

БІОЛОГІЧНІ ТА МЕДИЧНІ ПРИЛАДИ І СИСТЕМИ

УДК 681.5:613

С. В. Павлов¹, Й. Р. Салдан², С. М. Злепко¹, О. Д. Азаров¹,
Л. І. Тимченко³, Л. В. Абраменко³**МЕТОДИ ПОПЕРЕДНЬОЇ ОБРОБКИ ТОМОГРАФІЧНИХ
ЗОБРАЖЕНЬ ОЧНОГО ДНА**¹Вінницький національний технічний університет²Вінницький національний медичний університет ім. М.І. Пирогова³Державний університет інфраструктури та технологій

Анотація. В статті розглянутий ряд сучасних методів та алгоритмів обробки зображень, а саме: фільтра Кірша, нелінійних фільтрів Роберта та Собела, методів Уолесса та SUSAN, а також реалізовано програмне забезпечення для застосування цих методів на прикладі аналізу томографічних зображень макулярних змін сітківки ока. Найбільш інформативним для сегментації зображень є метод на основі фільтра Кірша та метод на основі нелінійного фільтра Собела. На відміну від так званих детермінованих спотворень, які часто описуються поелементними функціональними перетвореннями вхідного зображення, для опису випадкових впливів використовують моделі адитивного, імпульсного і мультиплікативного шумів. Для покращення якості оброблення біомедичних зображень застосовують, як правило, алгоритми Кірша, Робертса, Собела. Розглянуто особливості побудови високопродуктивних експертних систем та пристроїв оброблення та розпізнавання зображень шляхом створення нейроподібних методів та алгоритмів паралельно-ієрархічного перетворення, мережних методів попереднього оброблення зображень, розроблення та створення апаратних спеціалізованих образних комп'ютерів на їх базі, що у поєднанні із відомими перевагами око-процесорної обробки відеоданих (наприклад, використання ЛЧФ, Q-перетворення, метод паралельно-ієрархічного кодування зображень, просторово-часові матричні моделі організації обчислювальних процесів, паралельні алгоритми та структури, що їх відображають, для паралельної обробки зображень) є досить актуальним завданням. Таким чином, на відміну від так званих детермінованих спотворень, які часто описуються поелементними функціональними перетвореннями вхідного зображення, для опису випадкових впливів доцільно використовувати моделі адитивного, імпульсного і мультиплікативного шумів. Показано, що для покращення якості оброблення біомедичних зображень застосовують, як правило, алгоритми Кірша, Робертса, Собела.

Ключові слова: біомедичні зображення, макулярні зміни, нелінійна фільтрація

Аннотація. В статье рассмотрен ряд современных методов и алгоритмов обработки изображений, а именно: фильтра Кирша, нелинейных фильтров Роберта и Собела, методов Уолесса и SUSAN, а также реализовано программное обеспечение для применения этих методов на примере анализа томографических изображений макулярных изменений сетчатки глаза. Наиболее информативным для сегментации изображений является метод на основе фильтра Кирша и метод на основе нелинейного фильтра Собела. В отличие от детерминированных искажений, которые часто описываются поэлементно функциональными преобразованиями входного изображения, для описания случайных воздействий используют модели аддитивного, импульсного и мультипликативного шумов. Для улучшения качества обработки биомедицинских изображений применяют, как правило, алгоритмы Кирша, Робертса, Собела. Рассмотрены особенности построения высокопроизводительных экспертных систем и устройств обработки и распознавания изображений путем создания нейроподобных методов и алгоритмов параллельно-иерархического преобразования, сетевых методов предварительной обработки изображений, разработки и создания аппаратных специализированных образных компьютеров на их базе, в сочетании с известными преимуществами око-процессорной обработки видеоданных (например, использование ЛЧФ, Q-преобразования, метод параллельно-иерархического кодирования изображений, пространственно-временные матричные модели организации вычислительных процессов, параллельные алгоритмы и структуры для параллельной обработки изображений) является весьма актуальной задачей. Таким образом, в отличие от называемых детерминированных искажений, которые часто описываются поэлементно функциональными преобразованиями входного изображения, для описания случайных воздействий целесообразно использовать модели аддитивного, импульсного и мультипликативного шумов. Показано, что для улучшения качества обработки биомедицинских изображений применяют, как правило, алгоритмы Кирша, Робертса, Собела.

Ключевые слова: биомедицинские изображения, макулярные изменения, нелинейная фильтрация

Abstract. The paper deals with a number of modern methods and algorithms for image processing, namely: Kirsch filter, nonlinear filters Robert and Sobel, Wallace and SUSAN methods, as well as software for application of these methods on the example of analysis of tomographic images of macular changes of the retina. The most informative for image segmentation is the Kirsch filter based method and the Sobel nonlinear filter method. In contrast to the so-called deterministic distortions, which are often described by elemental functional transformations of the input image, models of additive, impulse and multiplicative noise are used to describe the random effects. As a rule, Kirsch, Roberts, Sobel algorithms are used to improve the quality of biomedical image processing. Features of construction of high-performance expert systems and devices of image processing and recognition by creation of neural-like methods and algorithms of parallel-hierarchical transformation, network methods of image pre-processing, development and creation of hardware specialized image computers based on them, which are combined with the eye, are considered. video processing (for example, the use of PCF, Q-conversion, parallel-hierarchical image encoding method, space-time ve matrix model of computing processes, parallel algorithms and structures that they reflect for parallel imaging) is a fairly urgent task. Thus, in contrast to the so-called deterministic distortions, which are often described by elemental functional transformations of the input image, it is advisable to use additive, pulse, and multiplicative noise models to describe the random effects. It is shown that, as a rule, the Kirsch, Roberts, Sobel algorithms are used to improve the quality of biomedical image processing.

Keywords: biomedical images, macular changes, nonlinear filtration

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-4-12>.

Вступ

Найважливішою операцією передобробки дискретизованого зображення є фільтрація амплітудного

і геометричного шумів, що є результатом зовнішніх і внутрішніх шумових впливів на процеси формування, реєстрації, перетворення і передачі дискретизованого зображення.

Розділення накладеного на зображення шуму на амплітудний і геометричний є вельми умовним і підкреслює лише ту обставину, що амплітудний шум викривлює значення відліків зображення об'єкту, тоді як геометричний шум змінює число і розташування відліків, що відносяться до цього зображення, тобто відліків, що перевищують, наприклад, загальний поріг виявлення сигналу [1,2,3]. З цього випливає, що методи фільтрації амплітудного і геометричного шумів різні і визначаються сутністю перетворень, що здійснюються при передобробці зображення. Так, в деяких прикладних задачах теорії розпізнавання образів, наприклад, при кореляційно – екстремальному визначенні координат зображення об'єкту, використовуються методи передобробки, що базуються на формуванні різнісного зображення з його наступною пороговою обробкою: дельта – модуляція [1], узагальнене контурне препарування [2], оконтурювання [4] і т. д., що призводять до посилення зашумленості перетвореного зображення.

В той же час сутність ряду евристичних методів передобробки зображення з метою фільтрації адитивного шуму полягає в сумуванні локальних груп відліків зображення, що перетворюється.

Актуальність

У зв'язку з цим актуальні проблеми:

- суміщення операцій нелінійної (порогової) різнісної передобробки зображення і усереднення локальних груп відліків для зменшення зашумленості перетвореного зображення, у всякому разі, до зашумленості його перед обробкою;
- визначення оптимальних розмірів $P_{\text{опт}} \cdot Q_{\text{опт}}$ локальної групи (вікна) відліків зображення, що усереднюються згідно алгоритмам згладжування евристичного типу [5, 6], або алгоритму передобробки по методу часткового Q – сумування [5], використання якого вельми ефективно в широкоформатних телевізійних кореляційно – екстремальних координаторах.

Амплітудна фільтрація дискретизованого зображення є перетворенням амплітуди центрального відліку локальної групи відліків по вибірковому середньому значенню амплітуди відліку цієї групи, а геометрична фільтрація цього зображення є перетворенням амплітуди центрального відліку локальної групи відліків по числу відліків в цій групі з вибірковим середнім значенням амплітуди відліку.

Мета

Метою статті є підвищення якості оброблення біомедичних зображень шляхом розробки алгоритмів фільтрації й сегментації біомедичних зображень на базі методів Кирша, Робертса, Собела, Уолесса та SUSAN.

Задачі

1. Аналіз сучасних методів та алгоритмів обробки зображень, а саме: фільтра Кирша, нелінійних фільтрів Роберта та Собела, методів Уолесса та SUSAN
2. Розробка програмно-алгоритмічного забезпечення для розділення фону і оброблення біомедичного зображення

Розв'язання задач

При створенні високопродуктивних експертних систем розвиток пристроїв оброблення та розпізнавання зображень буде розвиватися шляхом створення нейроподібних методів та алгоритмів паралельно-ієрархічного перетворення, мережних методів попереднього оброблення зображень, розроблення та створення апаратних спеціалізованих образних комп'ютерів на їх базі, що у поєднанні із відомими перевагами око-процесорної обробки відеоданих (наприклад, використання ЛЧФ, Q -перетворення, метод паралельно-ієрархічного кодування зображень, просторово-часові матричні моделі організації обчислювальних процесів, паралельні алгоритми та структури, що їх відображають, для паралельної обробки зображень) є досить актуальним завданням [1,2,3].

Для реалізації око-процесорного оброблення біомедичної інформації необхідно проводити сегментацію біомедичних зображень.

Виділення контурів використовується, в основному, для сегментації елементів зображення. Сегментація дозволяє виділити ділянки зображення, що здаються спостерігачеві однорідними.

Існує велика кількість різних підходів до виявлення границь на зображеннях і постійно розробляються нові.

До найбільш розповсюджених методів відносять [5,6,7]:

- градієнтні методи. Базуються на тому, що різним об'єктам на зображеннях відповідають області з порівняно однаковими значеннями яскравості. На границях же яскравість істотно змінюється;
- використання других похідних. Для підкреслення перепадів яскравості зображення можна використовувати другі похідні (оператор Лапласа або лапласіан);

- посилення диференціальними операторами. Всі диференціальні оператори являють собою лінійні фільтри високих частот. Характерною рисою таких фільтрів є посилення не тільки границь, але й дрібномасштабного шуму. Причому, чим вище порядок похідної, тим більше це посилення;
- детектор границь Canny. John Canny описав алгоритми виявлення границь, які з тих пір стали одними з найбільш широковикористовуваних. Вони стали класикою в області виявлення границь. Canny виходив із трьох критеріїв, яким повинен задовольняти детектор границь: достатнє виявлення (Canny трактував цю властивість як підвищення відношення сигнал/шум); нормальна локалізація (правильне визначення положення границі); єдиний відгук на одну границю. Із цих критеріїв потім будувалася цільова функція вартості помилок, мінімізацією якої знаходиться "оптимальний" лінійний оператор для згортки зображення;
- метод активних контурів. Використання для виявлення границь на зображенні кривих мінімальної енергії, названих снейками. Передбачається, що шукана границя на зображенні являє собою гладку лінію (лише для плоского зображення);
- алгоритм SUSAN. Smith & Brady звернули увагу на те, що сусідні точки однорідної області мають близьку до неї яскравість, а поблизу границі число сусідів з однаковою яскравістю зменшується. Відповідно до такого трактування границі й був розроблений алгоритм детектора границь SUSAN. Крім границь цей метод виявляє й інші особливості на зображенні (кути, тонкі лінії і т.д.).

В роботі були проаналізовані та розроблені алгоритми фільтрації й сегментації біомедичних зображень такими методами [7]. Для підвищення якості оброблення біомедичних зображень доцільно застосовувати методи Кирша, Робертса, Собела, Уолесса та SUSAN.

Фільтр Кирша працює із двовимірною апертурою 3×3 (частиною зображення, з яким фільтр працює безпосередньо в даний момент часу). Крім того, якщо мова йде про вікно, що являє собою ряд елементів зображення ($[X][X][X]$), то таке перетворення називається одновимірне; відповідно, існує й двовимірне перетворення.

Апертура має такий вигляд:

A_0	A_1	A_2
A_7	F	A_3
A_6	A_5	A_4

$$S_i = A_i + A_{i(+1)} + A_{i(+2)}$$

$$T_i = A_{i(+3)} + A_{i(+4)} + A_{i(+5)} + A_{i(+6)} + A_{i(+7)}.$$

Спочатку в циклі знаходяться всі значення змінних S_i й T_i , де i змінюється від 0 до 7, за наведеними вище формулами, у яких «(+))» означає додавання за модулем 8.

Далі знаходяться значення модулів різниці $|5 \cdot S_i - 3 \cdot T_i|$ для кожного i від 0 до 7 і значення максимуму серед цих модулів:

$$F' = \max_{i=0..7} (|5 \cdot S_i - 3 \cdot T_i|).$$

Остаточне значення F' заноситься в елемент F , після чого робоче вікно зсувається (рис. 1).

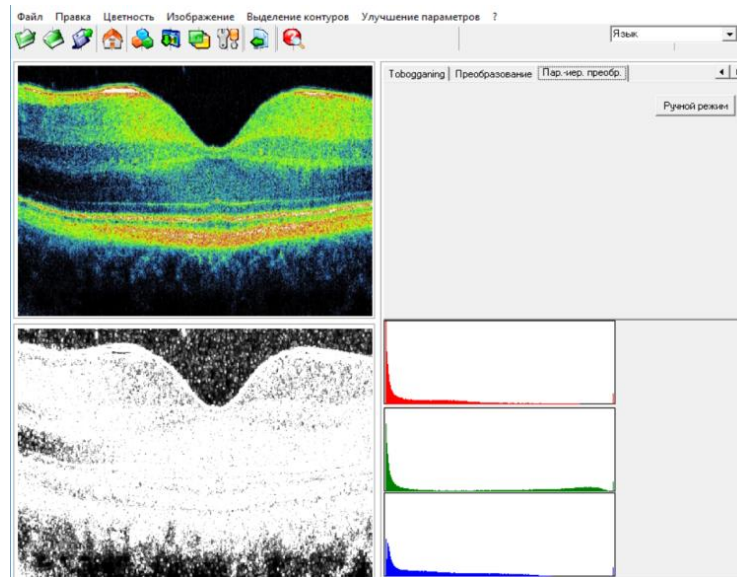


Рисунок 1 – Результат застосування фільтра Кирша при обробці томографічних зображень макулярних змін очного дна

Нелінійний фільтр Робертса – нелінійний метод контрастування, який використовує операції двовимірного дискретного диференціювання. Фільтр Робертса оперує квадратною апертурою розміром 2×2 , вагова функція якої приймається тотожно рівною одиниці:

$$A' = |A - D| + |B - C| = \sqrt{(A - D)^2 + (B - C)^2}$$

Друга форма запису (із квадратним коренем) працює повільніше, але точніше. Остаточне значення A' заноситься в елемент A , після чого робоче вікно зсувається.

Фільтр Робертса досить ефективно виділяє границі, але при цьому погано справляється з високочастотним шумом, хоча низькочастотний шум з невеликою (до 50%) амплітудою майже не впливає на сегментацію (рис. 2).

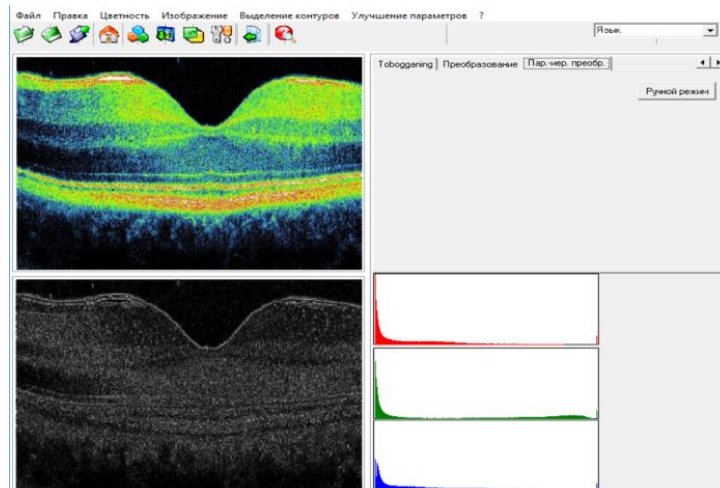


Рисунок 2 – Результат застосування нелінійного фільтра Робертса при обробці томографічних зображень макулярних змін очного дна

Алгоритм фільтрації за Робертсом є найпростішим (а значить і найшвидшим, що не менш важливо при обробці графіки) зі всіх розглянутих вище алгоритмів.

Нелінійний фільтр Собела – нелінійний оператор контрастування, що використовує апертуру 3×3 .

A_1	A_2	A_3
A_8	F	A_4
A_7	A_6	A_5

$$X = (A_3 + 2) \cdot (A_4 + A_5) - (A_1 + 2) \cdot (A_8 + A_7)$$

$$Y = (A_1 + 2) \cdot (A_2 + A_3) - (A_7 + 2) \cdot (A_6 + A_5)$$

Спочатку знаходяться значення змінних X й Y за наведеними вище формулами. Далі знаходиться нове значення центрального елемента:

$$F' = \sqrt{X^2 + Y^2}$$

Остаточне значення F' заміщує елемент F , після чого робоче вікно зміщується.

Фільтр Собела також досить ефективно виділяє границі, але при цьому погано справляється з високочастотним шумом (НЧ шум з невеликою (до 50%) амплітудою, як і фільтр Роберта, забирає майже повністю).

Як перевагу фільтрів Робертса (рис. 3) та Собела можна назвати поглинання низькочастотного шуму з невеликою (до 0.5) амплітудою. Але якщо на зображення накладено високочастотні імпульсні завади, перед застосуванням цих фільтрів зображення потрібно очистити за допомогою низькочастотних фільтрів (наприклад, медіанного двовимірного фільтра).

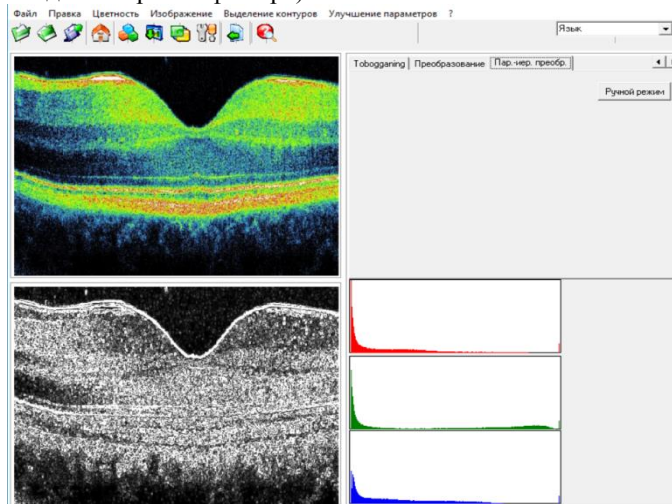


Рисунок 3 – Результат застосування нелінійного фільтра Собела при оброблянні томографічних зображень макулярних змін очного дна

Метод Уоллеса працює із двовимірною апертурою 3×3 :

0	1	2
7		3
6	5	4

$$F' = \frac{LN\left(\frac{F}{A_1} \cdot \frac{F}{A_3} \cdot \frac{F}{A_5} \cdot \frac{F}{A_7}\right)}{4}$$

Відразу знаходиться нове значення центрального елемента за наведеною вище формулою; при цьому, якщо знаменник (A_i з непарними значеннями i) дорівнює нулю, то до нього й до чисельника додається одиниця. Остаточне значення F' заноситься на місце F (рис. 4).

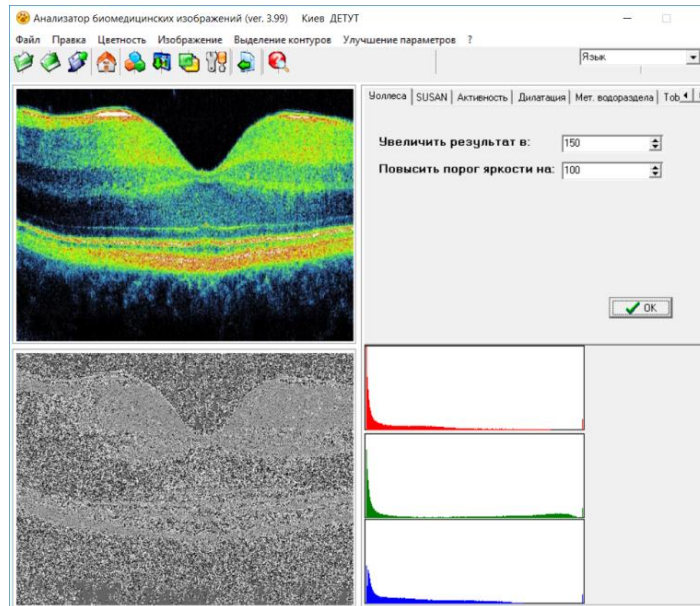


Рисунок 4 – Результат застосування методу Уоллеса при обробці томографічних зображень макулярних змін очного дна

Метод SUSAN. Основна ідея SUSAN – це те, що сусіди кожної точки в однорідній області мають близьку до неї яскравість, а поблизу границі число сусідів з однаковою яскравістю зменшується. Крім границь цей метод виявляє й інші особливості на зображенні (кути, тонкі лінії й т.п.).

Навколо кожного пікселя зображення будується маска, центральний піксель якої називається ядром (найчастіше використовується кругла маска з радіусом 3.4 пікселя, що включає 37 пікселів, або традиційна квадратна маска 3×3). Піксели в межах маски, що мають порівнянну з ядром яскравість, утворюють область SUSAN (Univaluse Segment Assimilating Nucleus - однорідний сегмент, асимільований ядром). Для виявлення двовимірних особливостей і границь використовуються розмір, центр ваги й інші ознаки SUSAN. Такий підхід виявлення особливостей відрізняється від відомих методів тим, що не використовує похідні зображення й, отже, не має необхідності в попередньому погашенні шуму.

Детектор границь SUSAN. Як було сказано вище, цей алгоритм використовує круглу маску з 37 пікселів або традиційну 3×3 . Маска міститься в кожену точку зображення, і яскравість кожної точки маски порівнюється з ядром (центральною точкою). Вихідне найпростіше рівняння такого порівняння має вигляд:

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| \leq \Delta; \\ 0 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| > \Delta. \end{cases} \quad (2)$$

Порівняння робиться для кожної точки маски й у підсумку підраховуємо n як суму c

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0). \quad (2)$$

Отже n і дає кількість пікселів в SUSAN, тобто дає площу SUSAN. Даний алгоритм максимально спрощений. Варто врахувати, що параметр Δ - різний для кожного зображення і у програмній реалізації повинен бути керованим.

Далі, для знаходження границі, n порівнюється з «геометричним порогом» g , що для більшості зображень можна прийняти на рівні 75% від n_{\max} . Реакція на границю має вигляд:

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{якщо } n(\vec{r}_0) < g; \\ 0 & \text{інакше.} \end{cases} \quad (3)$$

Описаний алгоритм достатньо непоганий, однак для більш стабільних й правильних результатів, особливо при обробці кольорових або високо-градуированих сірих зображень, доцільно використовувати замість функції (1) наведену нижче функцію:

$$c(\vec{r}, \vec{r}_0) = e^{-\frac{I(\vec{r}) - I(\vec{r}_0)}{\Delta}} \quad (4)$$

Для підвищення надійності алгоритму можна використати центр гравітації (ваги) SUSAN й основний напрямок симетрії.

Алгоритм SUSAN при досить високій якості зображення може ідеально правильно показати всі особливості форми об'єкта (або об'єктів) зображення (рис. 5).

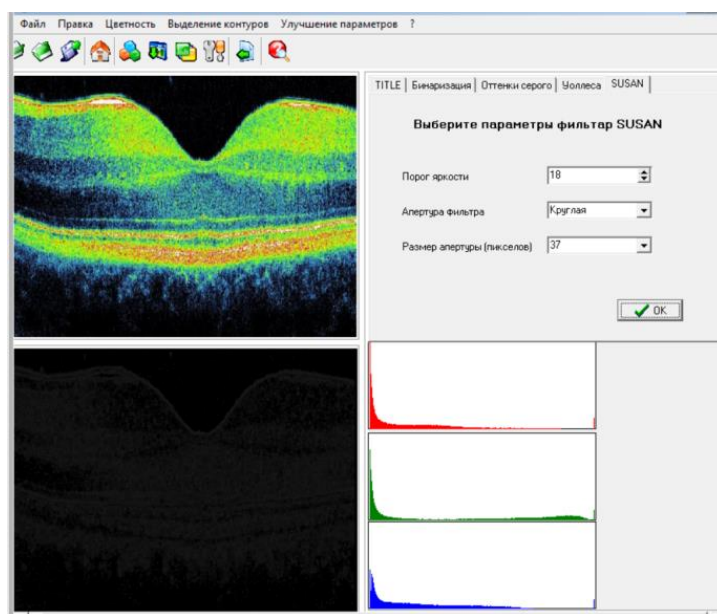


Рисунок 5 – Результат застосування методу SUSAN при обробці томографічних зображень макулярних змін очного дна

Висновки

Найважливішою операцією попередньої обробки дискретизованого зображення є фільтрація амплітудного і геометричного шумів, що є результатом зовнішніх і внутрішніх шумових впливів на процеси формування, ресстрації, перетворення і передачі дискретизованого зображення. Був таким чином розглянутий ряд сучасних методів та алгоритмів обробки зображень, а саме: фільтра Кірша, нелінійних фільтрів Роберта та Собела, методів Уоллеса та SUSAN, а також реалізовано програмне забезпечення для застосування цих методів на прикладі аналізу томографічних зображень макулярних змін сітківки ока. Найбільш інформативним для сегментації зображень є метод на основі фільтра Кірша та метод на основі нелінійного фільтра Собела.

Таким чином, на відміну від так званих детермінованих спотворень, які часто описуються поелементними функціональними перетвореннями вхідного зображення, для опису випадкових впливів доцільно використовувати моделі адитивного, імпульсного і мультиплікативного шумів.

Показано, що для покращення якості оброблення біомедичних зображень застосовують, як правило, алгоритми Кірша, Робертса, Собела.

Список літератури

- [1] Яне Б. Цифровая обработка изображений / Б. Яне. – М.: Техносфера, 2007. – 584 с.
- [2] Dougherty G. Digital Image Processing for Medical Applications / Geoff Dougherty – 2009. – 462 p.
- [3] Гонсалес Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.
- [4] Lim, Jae S. Two-Dimensional Signal and Image Processing / S. Jae Lim. – 1989. – 694p.
- [5] Павлов С. В. Оброблення біомедичних зображень із застосуванням швидкого перетворення Фур'є / С. В. Павлов, Д. В. Вовкотруб, Р. Ю. Довголюк // Оптико-електронні інформаційно-енергетичні технології. – 2011. – №2. – С. 96-101.
- [6] Yosyp R. Saldan, Sergii V. Pavlov, Dina V. Vovkotrub, Yulia Y. Saldan, Valentina B. Vassilenko, Nadia I. Mazur, Daria V. Nikolaichuk, Waldemar Wójcik, Ryszard Romaniuk, Batyrbek Suleimenov, Ulan Bainazarov, "Efficiency of optical-electronic systems: methods application for the analysis of structural changes in the process of eye grounds diagnosis", Proc. SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017, 104450S (7 August 2017); doi:

10.1117/12.2280977

- [7] Romanyuk O.N. Microfacet distribution function for physically based bidirectional reflectance distribution functions / O. N. Romanyuk, S. V. Pavlov, R. Yu. Dovhaliuk, N. P. Babyuk, M. D. Obidnyk, P. Kisala, B. Suleimenov Proceedings of SPIE - The International Society for Optical Engineering, 10.1117/12.2019338.
- [8] Pavlov S.V. Using of fuzzy expert method for diagnostic glaucoma / S.V. Pavlov, A.O. Rozhman, N.P. Babyuk, I.D. Ivasyuk // "Вимірювальна та обчислювальна техніка в технологічних процесах". Міжнародний науково-технічний журнал №2(43), 2013.– С.152-157.
- [9] Pavlov S.V. Fuzzy expert opto-electronic system for the analysis of biomedical images (for example diagnosing glaucoma) / S.V. Pavlov, O.D. Azarov, I.R. Saldan, A.O. Rozhman, N.P. Babyuk // "Інформаційні технології та комп'ютерна інженерія". Міжнародний науково-технічний журнал №1(26), 2013. — С.8-14.
- [10] Тимченко Л.І., Кокряцька Н.І., Герцій О.А., Петровський М.С., Степанюк Д.С., Паралельно-ієрархічні мережі для оброблення зображень. Теоретичні дослідження — Полтава : АСМІ, 2017. – 469 с.
- [11] Pavlov S. V. Information Technology in Medical Diagnostics //Waldemar Wójcik, Andrzej Smolarz, July 11, 2017 by CRC Press - 210 Pages.
- [12] Wójcik W., Pavlov S., Kalimoldayev M. Information Technology in Medical Diagnostics II. London: (2019). Taylor & Francis Group, CRC Press, Balkema book. – 336 Pages.
- [13] Yosyp R. Saldan, Sergii V. Pavlov, Dina V. Vovkotrub, Waldemar Wójcik, and etc. Efficiency of optical-electronic systems: methods application for the analysis of structural changes in the process of eye grounds diagnosis // Proc. SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017, 104450S; doi: 10.1117/12.2280977; <https://doi.org/10.1117/12.2280977>.

References

- [1] Yane B. Tsifrovaya obrabotka izobrazenyy // B. Yane. – М.: Technosfera, 2007. – 584 s.
- [2] Dougherty G. Digital Image Processing for Medical Applications / Geoff Dougherty – 2009. – 462 p.
- [3] Gonsales R. Tsifrovaya obrabotka v srede MATLAB / R. Gosales, R. Vuds, S. Eddins. – М.: Technosfera, 2006. – 616 s.
- [4] Lim, Jae S. Two-Dimensional Signal and Image Processing / S. Jae Lim. – 1989. – 694p.
- [5] Pavlov S.V. Obroblennya biomedichnih zobrazhen iz zastosuvannyam shvidkogo peretvorenniya Furie / S.V. Pavlov, D.V.Vovkotrub, R.Y.Dovgalyuk // Optiko-electronni informatsiyno-energetichni technologii. – 2011. – №2. – S. 96-101.
- [6] Yosyp R. Saldan, Sergii V. Pavlov, Dina V. Vovkotrub, Yulia Y. Saldan, Valentina B. Vassilenko, Nadia I. Mazur, Daria V. Nikolaichuk, Waldemar Wójcik, Ryszard Romaniuk, Batyrbek Suleimenov, Ulan Bainazarov, "Efficiency of optical-electronic systems: methods application for the analysis of structural changes in the process of eye grounds diagnosis", Proc. SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017, 104450S (7 August 2017); doi: 10.1117/12.2280977
- [7] Romanyuk O.N. Microfacet distribution function for physically based bidirectional reflectance distribution functions / O. N. Romanyuk, S. V. Pavlov, R. Yu. Dovhaliuk, N. P. Babyuk, M. D. Obidnyk, P. Kisala, B. Suleimenov Proceedings of SPIE - The International Society for Optical Engineering, 10.1117/12.2019338.
- [8] Pavlov S.V. Using of fuzzy expert method for diagnostic glaucoma / S.V. Pavlov, A.O. Rozhman, N.P. Babyuk, I.D. Ivasyuk // "Вимірювальна та обчислювальна техніка в технологічних процесах". Міжнародний науково-технічний журнал №2(43), 2013.– С.152-157.
- [9] Pavlov S.V. Fuzzy expert opto-electronic system for the analysis of biomedical images (for example diagnosing glaucoma) / S.V. Pavlov, O.D. Azarov, I.R. Saldan, A.O. Rozhman, N.P. Babyuk // "Інформаційні технології та комп'ютерна інженерія №1(26), 2013. — С.8-14.
- [10] Timchenko L. Kokryatska N. Etc. Paralelno-ierarchiseskie merezi dlya obroblennya zobrazhen— Poltava: АСМІ, 2017. – 469 S.
- [11] Pavlov S. V. Information Technology in Medical Diagnostics //Waldemar Wójcik, Andrzej Smolarz, July 11, 2017 by CRC Press - 210 Pages.
- [12] Wójcik W., Pavlov S., Kalimoldayev M. Information Technology in Medical Diagnostics II. London: (2019). Taylor & Francis Group, CRC Press, Balkema book. – 336 Pages.
- [13] Yosyp R. Saldan, Sergii V. Pavlov, Dina V. Vovkotrub, Waldemar Wójcik, and etc. Efficiency of optical-electronic systems: methods application for the analysis of structural changes in the process of eye grounds diagnosis // Proc. SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High

Energy Physics Experiments 2017, 104450S; doi:
10.1117/12.2280977; <https://doi.org/10.1117/12.2280977>.

Відомості про авторів

Павлов Сергій Володимирович, д.т.н., професор, проректор з наукової роботи Вінницького національного технічного університету.

Салдан Йосип Романович, д.м.н., професор кафедри очних хвороб та мікрохірургії ока Вінницького національного медичного університету ім. М. Пирогова.

Злепко Сергій Макарович, д.т.н., професор, зав. кафедри біомедичної інженерії Вінницького національного технічного університету.

Азаров Олексій Дмитрович – д.т.н., професор, декан факультету інформаційних технологій та комп'ютерної інженерії Вінницького національного технічного університету.

Тимченко Леонід Іванович - д.т.н., професор Державного університету інфраструктури та технологій, м. Київ.

Абраменко Людмила Володимирівна – аспірант Державного університету інфраструктури та технологій, м. Київ.

S. V. Pavlov¹, Y. R. Saldan², S. M. Zlepko¹, O. D. Azarov¹,
L. I. Tymchenko³, L. V. Abramenko³

METHOD OF PRE-PROCESSING TOMOGRAPHIC IMAGES OF THE FUNDUS

¹Vinnitsia National Technical University
²Vinnitsia National Pirogov Memorial Medical University
³State University of Infrastructure and Technologies

С. В. Павлов¹, И. Р. Салдан², С. М. Злепко¹, А. Д. Азаров¹,
Л. И. Тимченко³, Л. В. Абраменко³

МЕТОДЫ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ТОМОГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ ГЛАЗНОГО ДНА

¹Винницкий национальный технический университет
²Винницкий национальный медицинский университет им. Н.И. Пирогова
³Государственный университет инфраструктуры и технологий

УДК [681.7.01+66.088+615.478.6]

С. М. Злепко¹, А. М. Коробов², Вальдемар Вуйцік³, А. Ю. Клапоущак¹, В. С. Павлов¹, Т. І. Козловська¹, Ж. М. Хоменко⁴

ОПТИКО-ЕЛЕКТРОННІ ТЕХНОЛОГІЇ ОЦІНЮВАННЯ ЗМІНИ ПЕРИФЕРІЙНОГО КРОВООБІГУ ПРИ ФОТОННОМУ ВПЛИВІ ДЛЯ КОМПЛЕКСНОГО ЛІКУВАННЯ ХРОНІЧНИХ УСКЛАДНЕНЬ ЦУКРОВОГО ДІАБЕТУ

¹Вінницький національний технічний університет, Вінниця ²Харківський національний університет імені В.Н. Каразіна, Харків ³Люблінський технологічний університет «Любельська Політехніка», Люблін ⁴Державний університет «Житомирська політехніка», Житомир

Анотація. В статті розглянуто перспективність використання фотоплетизмографічного методу для оцінювання мікроциркуляторних змін периферійного кровообігу для вивчення процесів, які відбуваються при фотонному фізіотерапевтичному впливі для комплексного лікування хронічних ускладнень цукрового діабету. Показано, що фотонне випромінювання сприяє підвищенню еластичності стінок кровососних судин, еластичності еритроцитів, кисневотранспортній функції крові, активності клітинних мембран, прискоренню процесів регенерації тканин, зменшення окислення ліпідів, нормалізації реологічних показників крові, стимуляції утворення АТФ в мітохондріях, що підвищує біоенергетичний потенціал клітин. Після сеансів світлолікування у хворих зникав синдром «термоампутації», зменшувалися набряки ніг, а також значно зменшувалася біль в ногах, проходило оніміння, припинялися судороги ніг. Показано, що фотонне випромінювання апарату «Барва - СДС» призводить до протизапальної, десенсибілізуючої, знеболювальної, спазмолітичної, протинабрякової дій, що і підтверджено в даному дослідженні при лікуванні хворих з ускладненнями цукрового діабету. Для контролю рівня периферійного кровообігу використовувався фотоплетизмографічний метод, який порівняно з іншими відомими методами дослідження кровообігу має певні переваги, тому що дозволяє проводити дослідження як у прохідному, так і у відбитому світлі. Він дозволяє отримати корисну інформацію про судинну активність, судинному відповіді на специфічні речовини, а також проводити вивчення кровообігу в ізольованих судинах різного калібру (вени, артерії та ін.). Крім того, фотосенсор не викликає стиснення судин біологічної тканини, тобто не вносить порушення кровообігу. Таким чином, очевидно, що найбільш перспективним напрямком реєстрації фізіологічних параметрів є використання неінвазивних методів діагностування, серед яких широкого розвитку отримали оптичні методи реєстрації і перетворення біомедичної інформації і дозволяє проводити дослідження мікроциркуляції в тканинах людини без порушення кровообігу в досліджуваній ділянці, що є суттєвою перевагою для медичної практики.

Ключові слова: фотоплетизмограма, оптичні сенсори, фотонне випромінювання, фізіотерапевтичний вплив, цукровий діабет.

Анотація. В статье рассмотрены перспективность использования фотоплетизмографического метода для оценки микроциркуляторных изменений периферического кровообращения для изучения процессов, происходящих при фотонном физиотерапевтическом воздействии для комплексного лечения хронических осложнений сахарного диабета. Показано, что фотонное излучение способствует повышению эластичности стенок кровеносных сосудов, эластичности эритроцитов, кислородно-транспортной функции крови, активности клеточных мембран, ускорению процессов регенерации тканей, уменьшению окисления липидов, нормализации реологических показателей крови, стимуляции образования АТФ в митохондриях, что повышает биоэнергетический потенциал клеток. После сеансов светолечения у больных исчезал синдром «термоампутации», уменьшались отеки ног, а также значительно уменьшалась боль в ногах, проходило онемение, прекращались судороги ног. Показано, что фотонное излучение аппарата «Краска - СДС» приводит к противовоспалительному, десенсибилизирующему, обезболивающему, спазмолитическому, противотечному эффектам, что подтверждено в данном исследовании при лечении больных с осложнениями сахарного диабета. Для контроля уровня периферического кровообращения использовался фотоплетизмографический метод, который по сравнению с другими известными методами исследования кровообращения имеет определенные преимущества, так как позволяет проводить исследования как в проходящем, так и в отраженном свете. Он позволяет получить полезную информацию о сосудистой активности, сосудистом ответе на специфические вещества, а также проводить изучение кровообращения в сосудах различного калибра (вены, артерии и др.). Кроме того, фотосенсор не вызывает сжатия сосудов биологической ткани и не вносит нарушения при периферическом кровообращении. Таким образом, очевидно, что наиболее перспективным направлением регистрации физиологических параметров является использование неинвазивных методов диагностики, среди которых широкое развитие получили оптические методы регистрации и преобразования биомедицинской информации и позволяет проводить исследования микроциркуляции в тканях человека без нарушения кровообращения в исследуемой области, что является актуальным для медицинской практики.

Ключевые слова: фотоплетизмограма, оптические сенсоры, фотонное излучение, физиотерапевтический влияние, сахарный диабет.

Abstract. The paper deals with the prospect of using the photoplethysmographic method for the evaluation of microcirculatory changes in peripheral circulation to study the processes that occur during photonic physiotherapy exposure for the complex treatment of chronic complications of diabetes. It is shown that photonic radiation helps to increase the elasticity of the walls of blood vessels, the elasticity of erythrocytes, oxygen transport function of the blood, the activity of cell membranes, the acceleration of processes of tissue regeneration, the reduction of lipid oxidation, the normalization of rheologic indices. After procedures of phototherapy in patients, the syndrome of "thermoamputation" disappeared, swelling of the legs decreased, as well as significantly decreased pain in the legs, numbness took place, leg cramps stopped. It is shown that photonic radiation of the Barva apparatus leads to anti-inflammatory, desensitizing, analgesic, antispasmodic, and anti-edema actions, which was confirmed in this study in the treatment of patients with complications of diabetes. To control the level

С. М. Злепко, А. М. Коробов, Вальдемар Вуйцік, А. Ю. Клапоущак, В. С. Павлов, Т. І. Козловська,
Ж. М. Хоменко, 2019

of peripheral circulation, a photoplethysmographic method was used, which, compared to other known methods of blood circulation research, has certain advantages, because it allows to conduct studies in both transient and reflected light. It allows to obtain useful information about vascular activity, vascular response to specific substances, as well as to study blood circulation in isolated vessels of different caliber (veins, arteries, etc.). In addition, the photosensor does not cause compression of vessels of biological tissue, that is, does not cause impaired blood circulation. Thus, it is obvious that the most promising direction of registration of physiological parameters is the use of non-invasive methods of diagnosis, among which the wide development has received optical methods of registration and transformation of biomedical information and allows to study microcirculation in human tissues without impaired circulation in the study area, which is essential for medical practice.

Key words: Photoplethysmogram, optical sensors, photon radiation, physiotherapy effect, diabetes mellitus.

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-13-24>.

Вступ

Ускладнення цукрового діабету, такі як ангіопатії і полінейропатії, викликають серйозну занепокоєність ендокринологів у всьому світі у зв'язку з ростом кількості хворих з діабетичними нейропатіями в останні десятиліття. У лабораторії квантової біології та квантової медицини ХНУ імені В.Н. Каразіна під керівництвом А.М. Коробова та лабораторії біомедичної оптики ВНТУ під керівництвом С.В. Павлова проводяться дослідження дії низькоінтенсивного електромагнітного випромінювання оптичного діапазону спектра на розвиток цукрового діабету фотоплетизмографічними технологіями, а також розробляються методики і апарати серії «Барва» для лікування і профілактики ускладнень цукрового діабету (лабораторія квантової біології та квантової медицини ХНУ імені В.Н. Каразіна). В основу терапевтичної дії фотонних апаратів «Барва» покладено властивість світла видимого діапазону спектра посилювати мікроциркуляцію крові і лімфи, підвищувати еластичність стінок кровоносних судин, нормалізувати реологічні показники крові, а також нормалізувати роботу імунної, ендокринної, центральної і периферичної нервових систем.

Актуальність

Доведено, що фотоплетизмографічний метод є найкращим для дослідження стану периферичного кровообігу в тканинах людини, так як дозволяє проводити дослідження безконтактним способом як у прохідному так і у відбитому світлі, а також дозволяє проводити дослідження мікроциркуляції в тканинах людини без порушення кровообігу в досліджуваній ділянці, що є суттєвою перевагою для медичної практики.

Фотоплетизмографія заснована на просвічуванні досліджуваної ділянки тіла людини (монохроматичним) пучком світла, який після проходження або відбиття від нього реєструється оптико-електронними засобами, посилюється і записується. Встановлено, що інтенсивність променевої енергії, що пройшла через досліджуваний об'єкт або відбитий від нього, є функцією від вмісту в ній крові [29]. Оскільки коефіцієнт поглинання променевої енергії кров'ю значно вище, ніж тканинами, застосування фотоплетизмографічного методу дозволяє реєструвати лише зміни вмісту крові. При цьому розсіювання світла кров'ю відбувається в основному за рахунок відбиття від еритроцитів. Також, як і в реографії динаміка зміни оптичної щільності досліджуваних тканин $D(t)$ однозначно характеризує динаміку його кровонаповнення. Сигнал корисної інформації являє собою приріст оптичної щільності $\Delta D(t)$ та обумовлене перерозподілом крові з частотою серцевих скорочень щодо деякого постійного рівня D_0 .

Порівняно з іншими відомими методами дослідження кровообігу фотоплетизмографічний метод має певні переваги, тому що дозволяє проводити дослідження як у прохідному, так і у відбитому світлі. Він дозволяє отримати корисну інформацію про судинної активності, судинному відповіді на специфічні речовини, а також проводити вивчення кровообігу в ізольованих судинах різного калібру (вени, артерії та ін.). Крім того, фотосенсор не викликає стиснення судин біологічної тканини, тобто не вносить порушення кровообігу.

Таким чином, очевидно, що найбільш перспективним напрямком реєстрації фізіологічних параметрів є використання неінвазивних методів діагностування, серед яких широкого розвитку отримали оптичні методи реєстрації і перетворення біомедичної інформації.

Мета

Метою статті є підвищення ефективності вивчення мікроциркуляторних змін за допомогою фотоплетизмографічного методу при фотонному фізіотерапевтичному впливі на перебіг діабетичних ангіопатій.

Задачі

1. Провести аналіз біологічних ефектів при фотонному фізіотерапевтичному впливі.
2. Проаналізувати фізичні і математичні моделі для аналізу взаємодії оптичного випромінювання з біотканинами.
3. Реалізація оптико-електронного приладу діагностування стану периферичного кровообігу.
4. Проаналізувати результати терапевтичного впливу фотонного випромінювання та провести контроль ефективності фотоплетизмографічним методом.

Розв'язання задач

Аналіз біологічних ефектів при фотонному фізіотерапевтичному впливі

В результаті експериментальних досліджень з'ясовано, що відповідний відгук організму на фізіотерапевтичний вплив виникає на адаптаційному рівні і має періодичний характер, при цьому послідовно змінюються фази збудження і гальмування. Адаптаційна відповідь організму на терапевтичний вплив фотонним випромінюванням впливає на всі системи організму, як на клітинному, так і на органосистемному рівні.

Поглинання фотонного випромінювання викликає в біотканині біохімічні, біоелектричні та біоенергетичні ефекти. На рис. 1 показано біофізичні ефекти, які виникають після поглинання лазерного випромінювання малої і середньої потужності. Ці ефекти виникають при низьких густинах потужності та відносно тривалому часі опромінення.

Біохімічні ефекти викликають стимуляцію виділення гістамінів і серотонінів (це речовини, що відносяться до речовин-нейромедіаторів, а саме, фізіологічно активних речовин, що виконують роль хімічних посередників (медіаторів та нейрогормонів) у міжклітинних взаємодіях), можуть також стимулювати або гальмувати ензиматичні (ензими або ферменти – це речовини-каталізатори більшості хімічних реакцій, які відбуваються у живих організмах) реакції кислоти АТФ.

Біоелектричні ефекти приводять до нормалізації потенціалу біологічної тканини, а точніше її складових, зокрема і мембрани клітин.

Біоенергетичні ефекти стимулюють ріст і регенерацію клітин, а також міжклітинні процеси.

Вищеописані первинні ефекти, які безпосередньо виникають при опроміненні лазерним випромінюванням біотканин, спричиняють також вторинні ефекти і мають місце не тільки в опромінених, але одночасно і в сусідніх біологічних тканинах. Вторинні ефекти поділяються на три групи: протибольові, протизапальні та біостимуляційні.

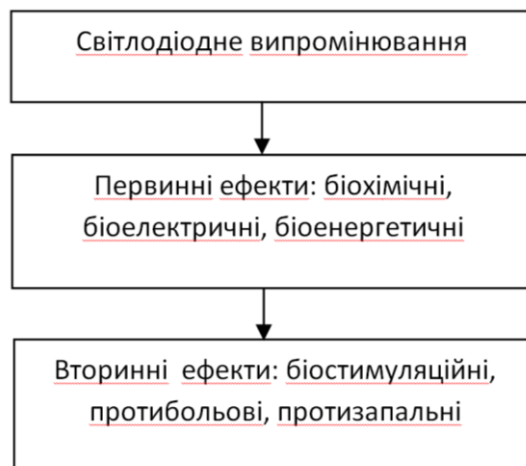


Рисунок 1 – Спрощена схема реакції біотканини на фотонне випромінювання малої або середньої потужності

Біоенергетичні ефекти стимулюють ріст і регенерацію клітин, а також міжклітинні процеси.

Вищеописані первинні ефекти, які безпосередньо виникають при опроміненні лазерним випромінюванням біотканин, спричиняють також вторинні ефекти і мають місце не тільки в опромінених, але одночасно і в сусідніх біологічних тканинах. Вторинні ефекти поділяються на три групи: протибольові, протизапальні та біостимуляційні.

Механізм дії ІЧ фотонного випромінювання (0,8–1,4 мкм) визначається його низькою енергією, нездатністю викликати виражений фотохімічний ефект. При поглинанні біотканинами ІЧ випромінювання перетворюється у теплову енергію вібрації молекул. Як наслідок різкого теплового розширення протоплазми клітин може виникнути гідродинамічний удар, який є вихідним імпульсом більш загальної дії ІЧ випромінювання. В результаті перебігу даних процесів різні біологічні сполуки переходять в активний стан або інактивуються. Фотонне випромінювання у ближній ІЧ області спектра слабо поглинається меланіном, водою та іншими компонентами біотканин, що обумовлює його високу проникність.

Таким чином, специфічність відповіді на дію фотонного світлодіодного випромінювання чітко проявляється у особливостях первинної відповіді (фотореакції) та на таких етапах, які запускаються в результаті локальних атомно-молекулярних змін. Після первинної резонансної абсорбції енергії кванта оптичного випромінювання або розподілу поглиненої енергії між збудженими станами окремих атомних

груп виникають перебудови макромолекул. Результатом цього може бути зміна активності ключових ферментів біотканинного метаболізму та мікроархітектоніки клітинних мембран.

Розглянемо принципи побудови математичних моделей для розрахунків взаємодії оптичного випромінювання з неоднорідними біотканинами. Як було відзначено раніше, неоднорідними називають середовища, в яких присутнє як поглинання так і розсіяння випромінювання. Одним із таких прикладів є шкіра людини. В той же час шкіра людини є живим багатошаровим середовищем, яке містить різноманітні включення, такі, як наприклад, кровоносні судини, в яких відбувається рух крові. Все це ускладнює розуміння процесів, які відбуваються при дії випромінювання на шкіру. Для опису цих процесів існує величезна кількість математичних і фізичних моделей, кожна з яких вирішує те чи інше конкретне завдання. Розглянемо основні принципи побудови математичних моделей, які описують взаємодію оптичного випромінювання з багатокомпонентними багатошаровими неоднорідними середовищами, таким як шкіра людини [1,6,7].

Практично всі моделі будуються за однаковою схемою (рис. 2).

Спочатку вибирають об'єкт дослідження, його геометрію. Потім визначаються оптичні і фізичні параметри всіх його компонентів. Далі відбувається розрахунок розподілу випромінювання в середовищі і для деяких моделей обчислення температурних полів.

Різниця між моделями стає помітною вже на етапі побудови геометрії. В більшості випадків шкіра являє собою послідовність плоских шарів з різними оптичними і теплофізичними характеристиками. Кількість шарів може змінюватися від одного до семи в залежності від поставлених завдань.

Побудуємо модель, яка дасть змогу оцінити інтенсивність випромінювання при відбиванні від судин.

Реальна біотканина являє собою неоднорідне середовище, в якому присутнє як поглинання так і розсіювання оптичного випромінювання, тому дослідити і описати процеси, які в ній відбуваються, практично неможливо. Розглянемо один із прикладів біотканини, а саме шкіру людини.

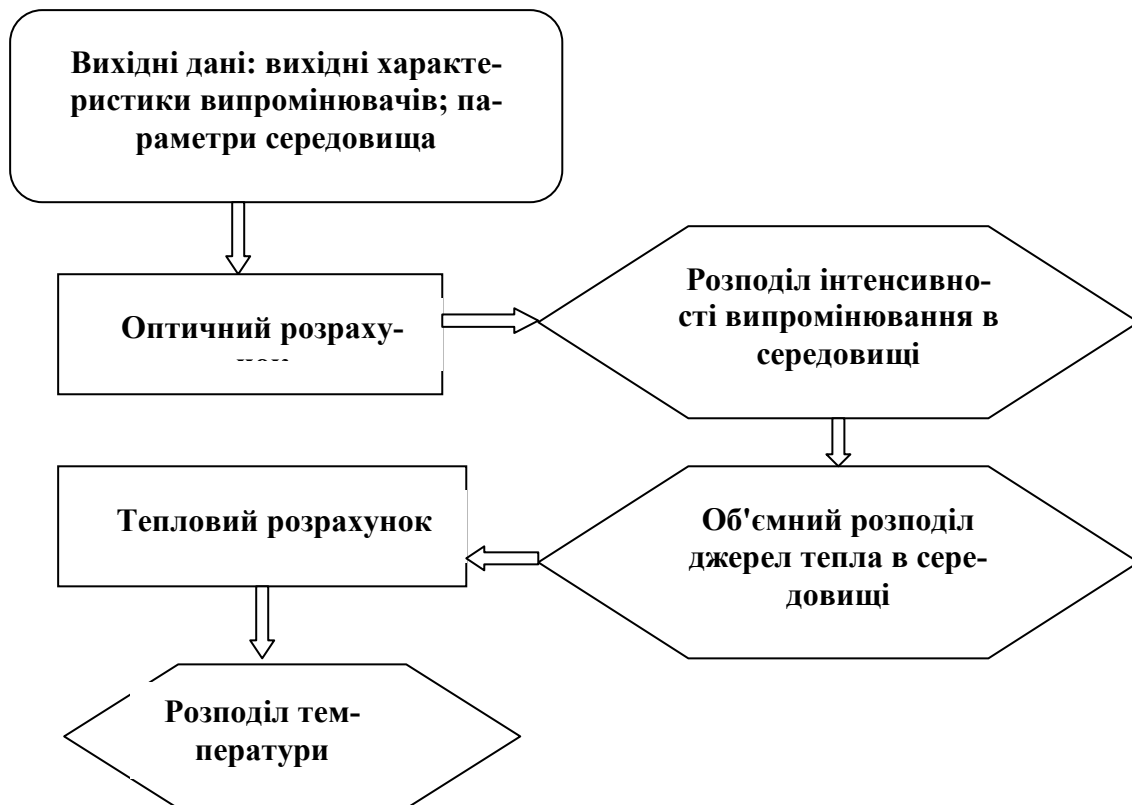


Рисунок 2 – Схема побудови моделей

Шкіра людини є живим багатошаровим середовищем, що містить різні включення, такі як, наприклад, кровоносні судини, в яких відбувається рух крові. Все це ускладнює розуміння процесів, що відбуваються, при дії випромінювання на шкіру. Для опису цих процесів на даний момент існує безліч різних математичних і фізичних моделей, кожна з яких покликана вирішити яку-небудь конкретну задачу,

описати окремий випадок. Спрощена модель шкіри, являє собою тришарову систему, причому кожен шар має різні біофізичні властивості, а тому і різні оптичні параметри (рис. 3).

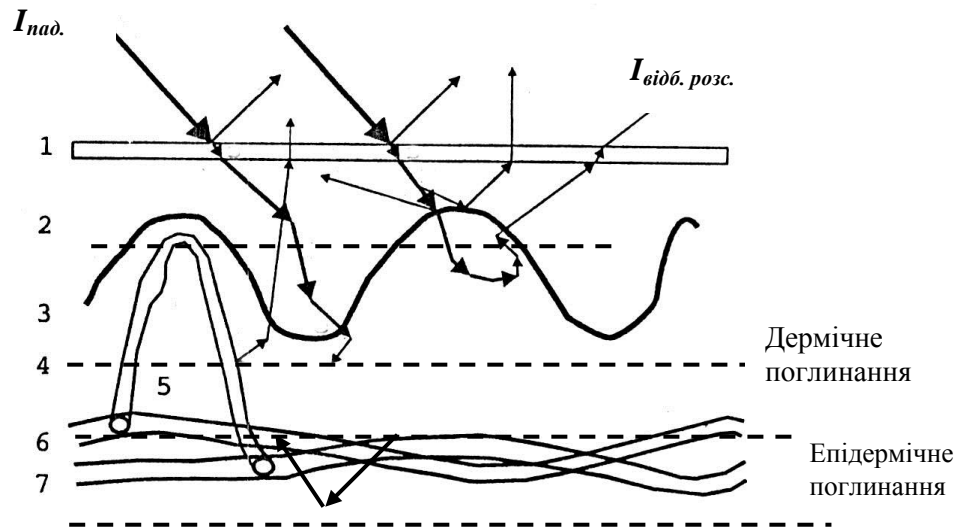


Рисунок 3 – Схема розповсюдження оптичного випромінювання в біооб'єкті:
1 – роговий шар; 2 – епідерміс; 3 – зона між епідермісом та дермою (базальна мембрана); 4 – дерма;
5 – артеріола; 6 – посткапілярна венула

Падаюче на шкіру випромінювання I_1 , частково відбивається від її поверхні I'_1 , проходить через епідерміс, де поглинається меланіном шкіри і частково розсіюється I_2 , а частина проходить через дерму і відбивається від судини I_3

$$\Delta I_{\text{відб}} = I_1 - (I'_1 + I_2 + I_3) \quad (1.1)$$

Позначимо $I_l = f(I_0, R, \theta)$ як функцію інтенсивності випромінювання, яке взаємодіє з біотканиною. $I_0 = f(\lambda)$, а також кута його падіння θ за умови, коли здійснюється закон Ламберта $I_\theta = I_0 \cos(\theta)$.

При відбиванні від поверхні шкіри випромінювання змінює інтенсивність в деякому тілесному куті

$$dI'_1 = r_1 I_0 \cos \Theta d\Omega, \quad (1.2)$$

де $d\Omega = \sin \Theta d\Theta d\varphi$ елементарний тілесний кут, r_1 – коефіцієнт відбивання шкіри.

При поглинанні у меланіні інтенсивність змінюється за законом Бугера – Ламберта

$$I_2(l) = I_0' e^{-\mu l c}, \quad (1.3)$$

де l – оптична товщина середовища, c – концентрація центрів поглинання, μ – коефіцієнт послаблення.

При відбиванні від судини інтенсивність зменшиться

$$dI_3 = I_0' r_2 2dl d\Omega. \quad (1.4)$$

Для обрахунку зміни інтенсивності запишемо отримане рівняння

$$dI(l) = I_0 \cos \Theta - (r_1 I_0' d\Omega + (1 - r_1) \exp(-\mu l c) + r_2 2dl d\Omega) \quad (1.5)$$

Отже, рівняння (5) дає можливість оцінити зміну інтенсивності випромінювання при відбиванні від судин залежно від їх кровонаповненості, що враховується зміною оптичної товщини середовища з врахуванням зміни інтенсивності при поглинанні і відбиванні у шарах шкіри [1,2].

Реалізація оптико-електронного приладу діагностування стану периферичного кровообігу

Конструктивно фотоплетизмографічні прилади складаються з чотирьох основних частин: вимірювального перетворювача, що включає в себе джерело оптичного випромінювання і фотоприймач, електронного блоку, призначеного для посилення напруги, що знімається з фотоприймача, і виділення корисного сигналу, реєстратора і блоку живлення, що служить для живлення електронних схем і джерела оптичного випромінювання.

В основу створення оптико-електронного приладу діагностування стану периферичного кровообігу, в якому за рахунок введення нових елементів та зв'язків підвищуються функціональні можливості та чутливість реєстрації пристрою. Оптико-електронний прилад діагностування стану периферичного кровообігу (рис. 4) містить джерело випромінювання 1 фотоприймач 2, що підключений до входу підсилювача 3. Вихід підсилювача 3 під'єднаний до входу мікроконтролера 4 обчислювача 5, вихід 6 мікроконтролера 4 з'єднаний зі входом блока спектрального аналізу 7, вихід 8 якого під'єднаний до входу мікроконтролера 4, вихід 9 мікроконтролера 4 з'єднаний зі входом блока еталонів 10, вихід 11 якого з'єднаний зі входом блока зберігання еталонів 12, перший вихід 13 якого з'єднаний зі входом блока еталонів 10, а другий його вихід 14 під'єднано до входу мікроконтролера 4.

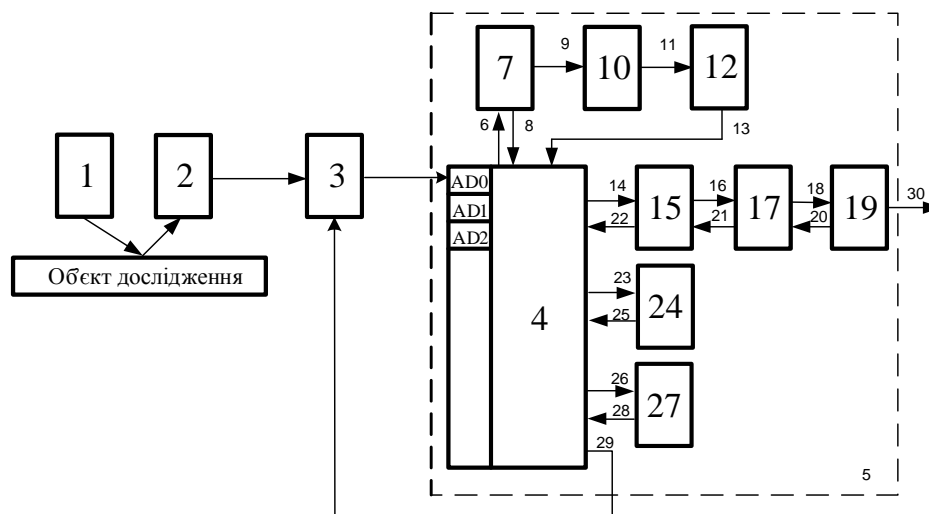


Рисунок 4– Структурна схема оптико-електронного приладу діагностування стану периферичного кровообігу

Оптико-електронний прилад діагностування стану периферичного кровообігу працює таким чином. Одразу після включення пристрою відбувається обнулення блоків обчислювача 5, а саме скид мікроконтролера 4 в нульовий стан та задання дозволу роботи пристрою. Після цього джерело випромінювання 1 випромінює світловий потік, що частково поглинаючись і частково розсіюючись біологічними тканинами досліджуваної ділянки тіла (об'єкту дослідження), подається на фотоприймач 2. Пульсації периферичних судин, що виникають за рахунок проходження пульсової хвилі, викликають коливання оптичної густини живої тканини, тому потік інфрачервоного випромінювання, що пройшов або відбився від тканин, модулюється по амплітуді і наводить у фотоприймачі 2 електричні сигнали, пропорційні цьому потоку.

Сигнал з фотоприймача 2 після фільтрації та посилення на підсилювачі 3 поступає на вхід мікроконтролера 4. Потім, сигнал надходить на вхід блока еталонів 10, де відбувається порівняння отриманого сигналу з записаним раніше в блоці зберігання еталонів 12, його еталонним значенням, в результаті чого визначається ступінь порушення периферичного кровообігу. У випадку, коли необхідно провести спектральний аналіз отриманого сигналу, вмикається блок спектрального аналізу 7, і, сигнал з мікроконтролера 4 надходить на його вхід, де проводиться виділення основних інформаційних ознак отриманого сигналу (рис.5.)

Оскільки мікроконтролер 4 має вбудований АЦП, то в ньому проводиться серія аналого-цифрових перетворень, після чого мікроконтролер 4 перетворює сигнал в цифровий код. З його виходу сигнал надходить на перший вхід 15 блока гальванічної розв'язки 16. З виходу 17 якого сигнал передається до USB-контролера 18. USB контролер 18 використовується для передачі даних в пам'ять персонального комп'ютера 20. Після того, як результати вимірювання потрапляють до персонального комп'ютера 20, на екрані висвітлюється оброблений фотоплетизмографічний сигнал (рис 6).

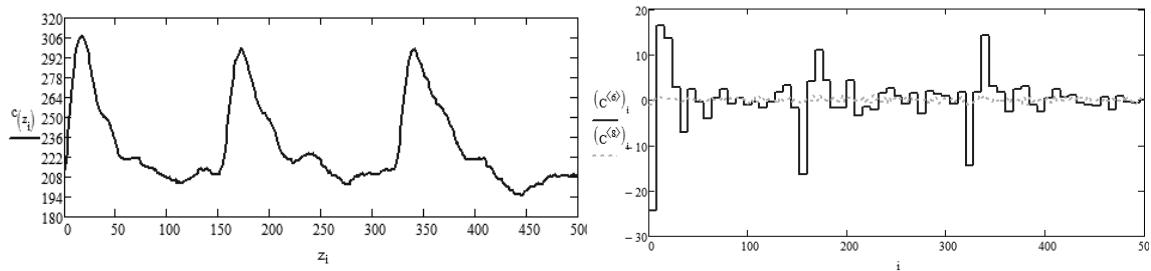


Рисунок 5 – Результат проведення спектрального аналізу фотоплетизмографічного сигналу, що реєструється за допомогою оптико-електронного приладу діагностування стану периферичного кровообігу

Коли сигнали потрапляють в персональний комп'ютер 20, він повідомляє мікроконтролер про те, що сигнал встановився. Це здійснюється через виходи: 21 персонального комп'ютера 20, 22 USB-контролера 18 та 23 блока гальванічної розв'язки 16.

Синхронізація роботи приладу, а саме підсилювача 3 здійснюється завдяки керуючому сигналу з виходу 30, який виробляє мікроконтролер 4.

Для забезпечення мобільності роботи даного приладу та розширення його функціональних можливостей в ньому використовується графічний рідкокристалічний індикатор 25, на який виводиться графічна інформація (фотоплетизмограма), через вихід 24 мікроконтролера 4, в якому запрограмовано модуль роботи з графічним рідкокристалічним індикатором. Це дозволяє проводити діагностування без використання персонального комп'ютера, що є важливим при обстеженні післяопераційних хворих. Крім того, прилад оснащений слотом 28 для SD-карти пам'яті, що дозволяє зберігати дані, та переносити їх в подальшому на персональний комп'ютер.

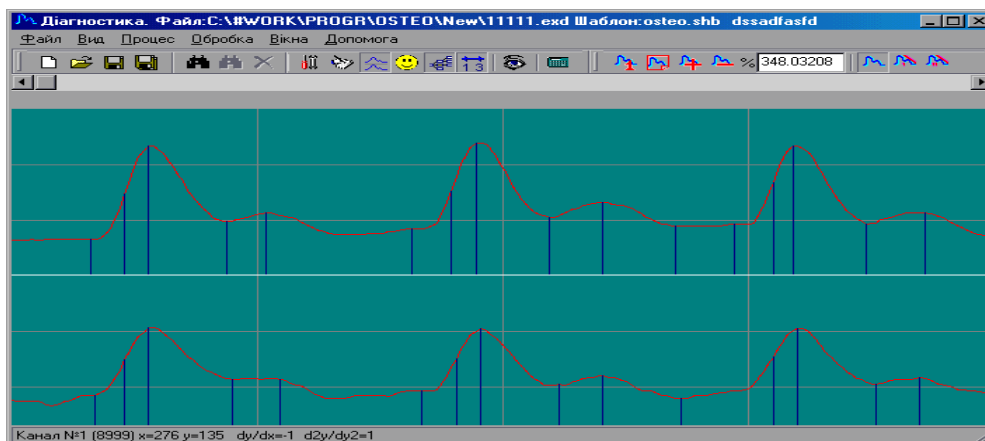


Рисунок 6 – Фотоплетизмографічний сигнал, що реєструється за допомогою оптико-електронного приладу діагностування стану периферичного кровообігу

Матеріал і методи досліджень

Фізіотерапевтичний вплив проводився за допомогою двосекційних фототерапевтичних апаратів Коробова А. - Коробова В. «Барва - СДС». В обстеженні брали участь 20 хворих з діабетичними ангіопатіями, що знаходилися на лікуванні в санаторії ендокринного профілю «Березовий гай» у м Миргороді. Вік обстежуваних осіб 40-72 року, з них 10 чоловіків і 10 жінок. Тривалість захворювання - 4-12 років. Всі учасники експерименту отримували комплексне лікування: светотерапія, дієтотерапія, корекція глікемії, гідропатичні процедури, прийом миргородської мінеральної води. Для контролю ефективності запропонованого методу фотонного впливу застосовували спосіб оцінювання біоелектричних властивостей клітинних ядер (визначали показник ЕНЯ% - відсоток електронегативних ядер в клітинах букального епітелію), який є інтегральним тестом, що відображає загальний функціональний стан організму обстежуваної особи. Для оцінки колатерального кровообігу в нижніх кінцівках хворих використовували методи лазерної фотоплетизмографія (ФПГ).

Результати експериментальних досліджень

Нижче наведені результати дослідження.

Приклад 1. Ефект впливу фотонного випромінювання за досліджуваними показниками у хворого Євгена Б., 1973 р.н.. (Зріст 184 см, вага 102 кг); діагноз: цукровий діабет 2 типу середньої тяжкості, ста-

дія субкомпенсації, мікроангіопатія. Фототерапія проводилася три дні випромінюванням зеленого діапазону спектра (525 нм), тривалість кожної процедури - 10 хвилин.

Вихідні значення показника ЕНЯ% - 57,7% по відношенню до вікової норми. Показник ЕНЯ% після першого сеансу світлолікування - 64,4% по відношенню до вікової норми. Показник ЕНЯ% після другого сеансу світлолікування - 71,1% по відношенню до вікової норми.

Показники ФПГ (рис. 7): ліва гомілка - посилення мікроциркуляції крові після трьох сеансів на 75%; права гомілка - посилення мікроциркуляції крові на 11%; ліва стопа - посилення мікроциркуляції крові на 2%; права стопа - посилення мікроциркуляції крові на 24%.

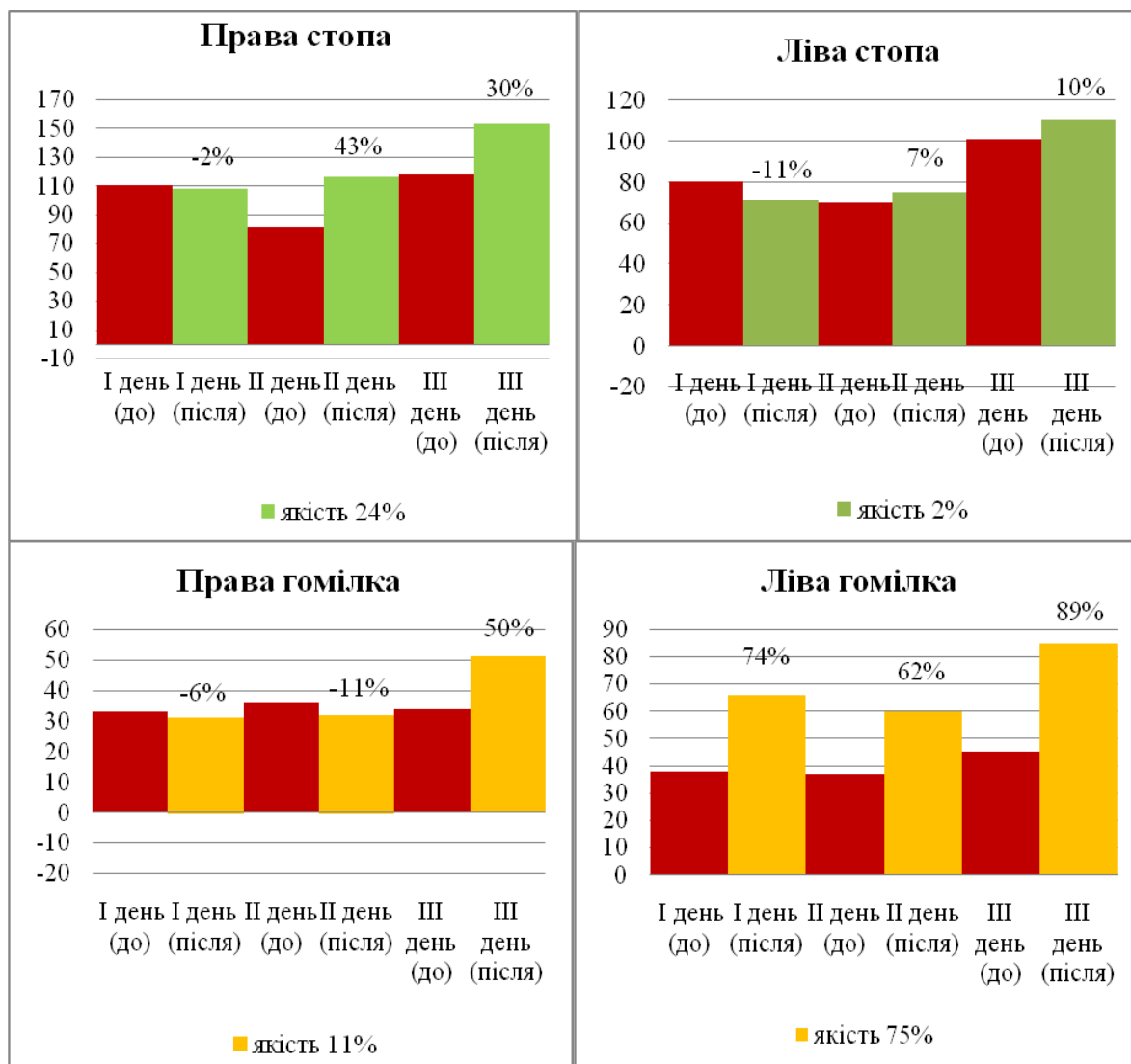


Рисунок 7 – Оцінювання рівня периферійного кровонаповнення при впливі фотонного випромінювання (приклад 1)

Приклад 2. Ефект впливу світлолікування за досліджуваними показниками у хворій Олені П., 1978 р.н.. (Зріст 168 см, вага 100 кг); діагноз: цукровий діабет 2 типу, комбінована ангіопатія. Фототерапія проводилася три дні випромінюванням синього діапазону спектра (470 нм), тривалість кожної процедури - 10 хвилин.

Вихідні значення показника ЕНЯ% - 32,4% по відношенню до вікової норми. Показник ЕНЯ% після першого сеансу світлолікування - 40,5% по відношенню до вікової норми. Показник ЕНЯ% після другого сеансу світлолікування - 45,9% по відношенню до вікової норми.

Показники ФПГ (рис. 8): ліва гомілка - посилення мікроциркуляції крові після трьох сеансів на 95%; права гомілка - посилення мікроциркуляції крові на 13%; ліва стопа - посилення мікроциркуляції крові на 41%; права стопа - посилення мікроциркуляції крові на 198%.

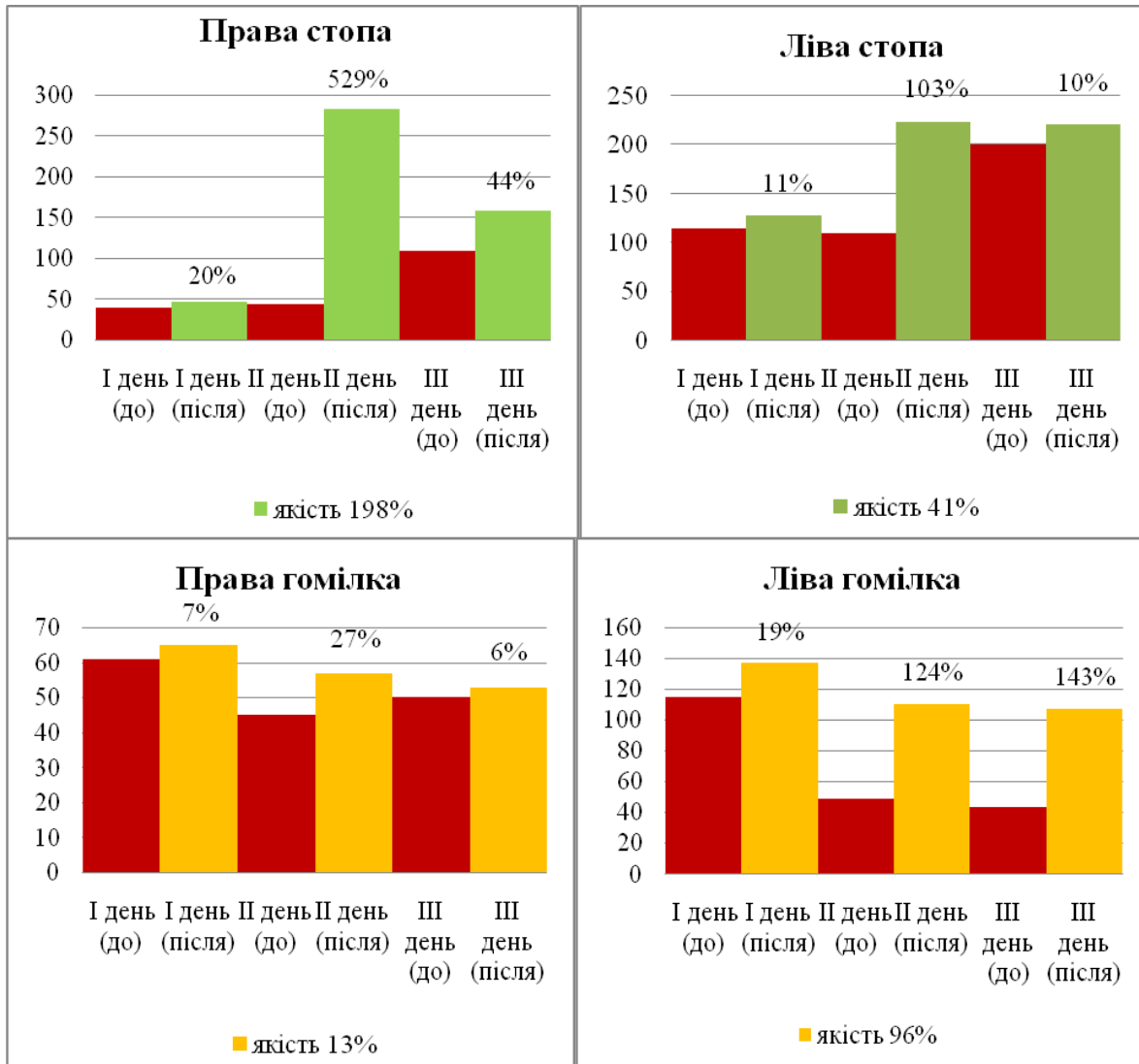


Рисунок 8 – Оцінювання рівня периферійного кровонаповнення при впливі фотонного випромінювання (приклад 2)

Висновки

1. В статті розглянуто перспективність використання фотоплетизмографічного методу для оцінювання мікроциркуляторних змін периферійного кровообігу для вивчення процесів, які відбуваються при фотонному фізіотерапевтичному впливі для комплексного лікування хронічних ускладнень цукрового діабету.

2. Показано, що фотонне випромінювання сприяє підвищенню еластичності стінок кровоносних судин, еластичності еритроцитів, кисневотранспортній функції крові, активності клітинних мембран, прискоренню процесів регенерації тканин, зменшення окислення ліпідів, нормалізації реологічних показників крові, стимуляції утворення АТФ в мітохондріях, що підвищує біоенергетичний потенціал клітин.

3. Після сеансів світлолікування у хворих зникнув синдром «термоампутації», зменшувалися набряки ніг, а також значно зменшувалася біль в ногах, проходило оніміння, припинялися судоми ніг.

4. Показано, що фотонне випромінювання апарату «Барва - СДС» призводить до протизапальної, десенсибілізуючої, знеболювальної, спазмолітичної, протинабрякової дій, що і підтверджено в даному дослідженні при лікуванні хворих з ускладненнями цукрового діабету.

Список літератури

1. Фізичні основи біомедичної оптики : монографія / [Павлов С. В., Кожем'яко В. П., Колісник П. Ф. та ін.] – Вінниця : ВНТУ, 2010. – 155 с.
2. Аналіз методів взаємодії оптичного випромінювання з біотканинами і шляхи їх удосконалення / [С.В.Павлов, В.П.Думенко, Т.І.Козловська та ін.] // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2008. – №2. – С. 129-135.

3. Лазерні медичні технології : навчальний посібник / [за редакцією Готри З. Ю. , Павлова С. В.] – Вінниця : ВНТУ, 2017. – 157 с.
4. ~~Timmis A, Townsend N, Gale C [et al.]. European Society of Cardiology: Cardiovascular Disease Statistics 2017. *European Heart Journal*. 39 (2018), 508–579. doi: 10.1093/eurheartj/ehx628.~~
5. The National Audit of Cardiac Rehabilitation (NACR). The National Audit of Cardiac Rehabilitation (NACR) annual statistical report. UK: University of York, 2016.
6. Röhrig B, Salzwedel A, Linck-Eleftheriadis S [et al.]. Outcome based center comparisons in inpatient cardiac rehabilitation – results from the EVA-Reha® Cardiology project. *Rehabilitation (Stuttg)*. 54(1) (2015), 45-52. doi: 10.1055/s-0034-1395556.
7. Soumagne D. Weber classification in cardiac rehabilitation. *Acta Cardiol*. 67(3) (2012), 285-290.
8. ~~Anderson L, Thompson DR, Oldridge N [et al.]. Exercise-based cardiac rehabilitation for coronary heart disease. *Cochrane Database Syst Rev*. 5 (1)(2016).CD001800. doi: 10.1002/14651858.CD001800.pub3.~~
9. ~~Widmer RJ, Allison TG, Lennon R [et al.]. Digital health intervention during cardiac rehabilitation: A randomized controlled trial. *Am Heart J*. 188 (2017), 65-72. doi: 10.1016/j.ahj.2017.02.016.~~
10. ~~Bhasipol A, Sanjaroensuttikul N, Pornsuriyasak P [et al.]. Efficiency of the home cardiac rehabilitation program for adults with complex congenital heart disease. *Congenit Heart Dis*. (2018) Sep 14. doi: 10.1111/chd.12659.~~
11. Olena V. Vysotska , Kostiantyn Nosov, Natalia B. Savina, and etc. "An approach to determination of the criteria of harmony of biological objects", *Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018*, 108083B (1 October 2018); doi: 10.1117/12.2501539.
12. Sergii V. Pavlov, Aleksandr T. Kozhukhar, et al. Electro-optical system for the automated selection of dental implants according to their colour matching // *PRZEGLĄD ELEKTROTECHNICZNY, ISSN 0033-2097, R. 93 NR 3/2017. — P. 121-124.* - doi:10.15199/48.2017.03.28
13. Vladimir V. Kholin, Oksana M. Chepurna, Sergii Pavlov et al. Methods and fiber optics spectrometry system for control of photosensitizer in tissue during photodynamic therapy, *Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*, 1003138 (September 28, 2016); doi:10.1117/12.2249259
14. Ronald H. Rovira; Stanislav Ye. Tuzhansky; Sergii V. Pavlov; Sergii N. Savenkov; Ivan S. Kolomiets, et al. Polarimetric characterisation of histological section of skin with pathological changes, *Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*, 100313E (September 28, 2016); doi:10.1117/12.2249373
15. S. V. Pavlov; V. B. Vassilenko; I. R. Saldan; D. V. Vovkotrub; A. A. Poplavskaya, et al. Methods of processing biomedical image of retinal macular region of the eye, *Proc. SPIE 9961, Reflection, Scattering, and Diffraction from Surfaces V*, 99610X (September 26, 2016); doi:10.1117/12.2237154
16. Ronald Rovira; Marcia M. Bayas; Sergey V. Pavlov; Tatiana I. Kozlovskaya; Piotr Kisała, et al. Application of a modified evolutionary algorithm for the optimization of data acquisition to improve the accuracy of a video-polarimetric system, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015*, 981619 (December 18, 2015); doi:10.1117/12.2229087
17. Natalia I. Zabolotna; Sergii V. Pavlov; Kostiantyn O. Radchenko; Vladyslav A. Stasenko; Waldemar Wójcik, et al. Diagnostic efficiency of Mueller-matrix polarization reconstruction system of the phase structure of liver tissue, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015*, 98161E (December 18, 2015); doi:10.1117/12.2229018
18. Oleg G. Avrunin; Maksym Y. Tymkovich; Sergii V. Pavlov; Sergii V. Timchik; Piotr Kisała, et al. Classification of CT-brain slices based on local histograms, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015*, 98161J (December 18, 2015); doi:10.1117/12.2229040;
19. Oksana Chepurna; Irina Shton; Vladimir Kholin; Valerii Voytsehovich; Viacheslav Popov, et al. Photodynamic therapy with laser scanning mode of tumor irradiation, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015*, 98161F (December 18, 2015); doi:10.1117/12.2229030.
20. Waldemar Wójcik, Andrzej Smolarz// *Information Technology in Medical Diagnostics* . London, July 11, 2017 by Taylor & Francis Group CRC Press Reference - 210 Pages.
21. Vassilenko, S Valtchev, JP Teixeira, S Pavlov. Energy harvesting: an interesting topic for education programs in engineering specialities / «Internet, Education, Science» (IES-2016) – 2016. – P. 149-156.
22. Roman Kvyetnyy, Olga Sofina, Pavel Orlyk, Andres J. Utreras, Waldemar Wójcik, and etc. "Improving the quality perception of digital images using modified method of the eye aberration correction", *Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*, 1003113 (28 September 2016).

References

1. Fizichni osnovy biomedichnoi optikt: monografiya/ [Pavlov S.V., Kojemiako V.P., Kolisnik P.F. ta insh.] – Vynnytsia: VNTU, 2010. – 155 s.
2. Analiz metodiv vzaenodii optichnogo viprominyuvannya z biotkaninoyu I shlyahi ih udoskonalennya / [S.V. Pavlov, V.P. Dumenko, T.I. Kozlovska ta insh] // Vimiryuvalna ta obchislyuvalna technika v technologichnih prosesah.– 2008. – №2. – S. 129-135.
3. Lazerni medichni tehnologii : navchalnii posibnik / [za redaksieyu Gotri Z.Yu., Pavlova S.V.] – Vynnytsia: VNTU, 2017. – 157 s.
4. Timmis A, Townsend N, Gale C [et al.]. European Society of Cardiology: Cardiovascular Disease Statistics 2017.*European Heart Journal*. 39 (2018), 508–579. doi: 10.1093/eurheartj/ehx628.
5. The National Audit of Cardiac Rehabilitation (NACR). The National Audit of Cardiac Rehabilitation (NACR) annual statistical report. UK: University of York, 2016.
6. Röhrig B, Salzwedel A, Linck-Eleftheriadis S [et al.]. Outcome based center comparisons in inpatient cardiac rehabilitation – results from the EVA-Reha® Cardiology project. *Rehabilitation (Stuttg)*. 54(1) (2015), 45-52. doi: [10.1055/s-0034-1395556](https://doi.org/10.1055/s-0034-1395556).
7. Soumagne D. Weber classification in cardiac rehabilitation. *Acta Cardiol*. 67(3) (2012), 285-290.
8. Anderson L, Thompson DR, Oldridge N [et al.]. Exercise-based cardiac rehabilitation for coronary heart disease.*Cochrane Database Syst Rev*. 5 (1)(2016).CD001800. doi: 10.1002/14651858.CD001800.pub3.
9. Widmer RJ, Allison TG, Lennon R [et al.]. Digital health intervention during cardiac rehabilitation: A randomized controlled trial. *Am Heart J*. 188 (2017), 65-72. doi: 10.1016/j.ahj.2017.02.016.
10. Bhasipol A, Sanjaroensuttikul N, Pornsuriyasak P [et al.]. Efficiency of the home cardiac rehabilitation program for adults with complex congenital heart disease. *Congenit Heart Dis*. (2018) Sep 14. doi: 10.1111/chd.12659.
11. Olena V. Vysotska, Kostiantyn Nosov, Natalia B. Savina, and etc. An approach to determination of the criteria of harmony of biological objects", *Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 108083B* (1 October 2018); doi: 10.1117/12.2501539.
12. Sergii V. Pavlov, Aleksandr T. Kozhukhar, et al. Electro-optical system for the automated selection of dental implants according to their colour matching // *PRZEGLĄD ELEKTROTECHNICZNY, ISSN 0033-2097, R. 93 NR 3/2017.* – P. 121-124. - doi:10.15199/48.2017.03.28
13. Vladimir V. Kholin, Oksana M. Chepurna, Sergii Pavlov et al. Methods and fiber optics spectrometry system for control of photosensitizer in tissue during photodynamic therapy, *Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016, 1003138* (September 28, 2016); doi:10.1117/12.2249259
14. Ronald H. Rovira; Stanislav Ye. Tuzhansky; Sergii V. Pavlov; Sergii N. Savenkov; Ivan S. Kolomiets, et al. Polarimetric characterisation of histological section of skin with pathological changes, *Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016, 100313E* (September 28, 2016); doi:10.1117/12.2249373
15. S. V. Pavlov; V. B. Vassilenko; I. R. Saldan; D. V. Vovkotrub; A. A. Poplavskaya, et al. Methods of processing biomedical image of retinal macular region of the eye, *Proc. SPIE 9961, Reflection, Scattering, and Diffraction from Surfaces V, 99610X* (September 26, 2016); doi:10.1117/12.2237154
16. Ronald Rovira; Marcia M. Bayas; Sergey V. Pavlov; Tatiana I. Kozlovskaya; Piotr Kisała, et al. Application of a modified evolutionary algorithm for the optimization of data acquisition to improve the accuracy of a video-polarimetric system, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 981619* (December 18, 2015); doi:10.1117/12.2229087
17. Natalia I. Zabolotna; Sergii V. Pavlov; Kostiantyn O. Radchenko; Vladyslav A. Stasenko; Waldemar Wójcik, et al. Diagnostic efficiency of Mueller-matrix polarization reconstruction system of the phase structure of liver tissue, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 98161E* (December 18, 2015); doi:10.1117/12.2229018
18. Oleg G. Avrunin; Maksym Y. Tymkovych; Sergii V. Pavlov; Sergii V. Timchik; Piotr Kisała, et al. Classification of CT-brain slices based on local histograms, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 98161J* (December 18, 2015); doi:10.1117/12.2229040;
19. Oksana Chepurna; Irina Shton; Vladimir Kholin; Valerii Voytsehovich; Viacheslav Popov, et al. Photodynamic therapy with laser scanning mode of tumor irradiation, *Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 98161F* (December 18, 2015); doi:10.1117/12.2229030.
20. Waldemar Wójcik, Andrzej Smolarz// *Information Technology in Medical Diagnostics* . London, July 11, 2017 by Taylor & Francis Group CRC Press Reference - 210 Pages.

21. Vassilenko, S Valtchev, JP Teixeira, S Pavlov. Energy harvesting: an interesting topic for education programs in engineering specialities / «Internet, Education, Science» (IES-2016) – 2016. – P. 149-156.
22. Roman Kvyetnyy, Olga Sofina, Pavel Orlyk, Andres J. Utreras, Waldemar Wójcik, and etc. "Improving the quality perception of digital images using modified method of the eye aberration correction", Proc. SPIE 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016, 1003113 (28 September 2016).

Відомості про авторів

Злепко Сергій Макарович, д.т.н., професор, зав. кафедри біомедичної інженерії Вінницького національного технічного університету.

Коробов Анатолій Михайлович – к.ф.-м.н, директор НВ медико-біологічної корпорації «Лазер та здоров'я», Харківський національний університет імені В.Н. Каразіна.

Вуйцик Вальдемар д.т.н., професор, директор інституту електроніки та інформаційних технологій Люблінського технологічного університету «Любельська Політехніка».

Клапоушак Андрій Юрійович, старший викладач кафедри біомедичної інженерії Вінницького національного технічного університету.

Павлов Володимир Сергійович, магістрант кафедри лазерної та оптико-електронної техніки Вінницького національного технічного університету.

Козловська Тетяна Іванівна, к.т.н., доцент кафедри фізики Вінницького національного технічного університету.

Хоменко Жанна Миколаївна, к.т.н., доцент Державного університету «Житомирська політехніка».

S. M. Zlepko¹, A. M. Korobov², Waldemar Wojcik³, A. Yu. Klauposhak¹,
V. S. Pavlov¹, T. I. Kozlovska¹, Z. M. Khomenko⁴

OPTICAL-ELECTRONIC TECHNOLOGIES OF EVALUATION OF CHANGE OF PERIPHERAL CIRCULATION AT PHOTON INFLUENCE FOR COMPLEX TREATMENT OF CHRONIC DIAGNOSIS

¹Vinnitsia National Technical University

²Kharkiv national University by V.N.Karazin

³Technological University "Politechnika Lubelska"

⁴State University "Zhytomyr Politechnika"

С. М. Злепко¹, А. М. Коробов², Вальдемар Вуйцик³, А. Ю. Клапоушак¹,
В. С. Павлов¹, Т. И. Козловська¹, Ж. Н. Хоменко⁴

ОПТИКО-ЭЛЕКТРОННЫЕ ТЕХНОЛОГИИ ОЦЕНКИ ИЗМЕНЕНИЯ ПЕРИФЕРИЧЕСКОГО КРОВООБРАЩЕНИЯ ПРИ ФОТОННОМ ВЛИЯНИИ ДЛЯ КОМПЛЕКСНОГО ЛЕЧЕНИЯ ХРОНИЧЕСКИХ ОСЛОЖНЕНИЙ САХАРНОГО ДИАБЕТА

¹Винницкий национальный технический университет

²Харьковский национальный университет имени В.Н. Каразина

³Люблинский технологический университет «Любельская Политехника»

⁴Государственный университет «Житомирская политехника»

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.032.26 + 519.72

Р. Н. Кветний, Ю. Ю. Іванов, В. В. Півошенко, А. С. Васюра

АЛГОРИТМ РОЗПІЗНАВАННЯ ТА КОРИГУВАННЯ ПОШКОДЖЕНИХ QR-КОДІВ

Вінницький національний технічний університет

Анотація. Інтенсивний розвиток інформаційних технологій призвів до створення систем обміну даними, які застосовують комбіноване стиснення, захист від пошкоджень та зберігання інформації. Подібні системи зазвичай використовують матричні коди, які дають можливість компактно зберігати великий обсяг інформації та швидко розпізнаються сканувальним обладнанням. У даній статті розв'язано задачу розпізнавання та коригування сильно пошкоджених матричних кодів, а саме QR-кодів, у яких присутній високий рівень шуму, відсутні ключові елементи або накладаються кольори. Такі зображення не розпізнаються декодуючим програмним забезпеченням, оскільки структура детекторів елементів пошкоджена, а коригуюча здатність вбудованих кодів Ріда-Соломона не дозволяє виправити необхідну частину помилок. Тому запропоновано алгоритм, який базується на використанні штучної нейронної мережі Хеммінга-Ліппмана з базою еталонів та обробленні зображень у режимі ковзного вікна, що спрощує процес навчання мережі без використання трудомістких обчислювальних операцій, великих обсягів пам'яті та витрат часу, навіть для зображень високої роздільної здатності та великих розмірів. Процес навчання мережі складається з двох частин: обробка зображень, розпізнавання та корекція зразка. Для того, щоб домогтися коректного розпізнавання, необхідно експериментально виявляти оптимальні параметри навчання, завдяки яким в матриці еталонних образів рядки будуть достатньо відрізнятися один від одного. Для цього авторами на мові C# розроблено програмне забезпечення, за допомогою якого і проведено необхідні експериментальні дослідження. Визначено умови коректної роботи нейронної мережі (оптимальні значення розміру ковзного вікна та порогу для різних розмірів зображень QR-кодів), а також випадки, коли можливі помилки розпізнавання та нестабільності її виходів. Результати досліджень показують, що розроблений алгоритм можна застосовувати як додаткову процедуру розпізнавання та виправлення QR-кодів у різноманітних системах обміну даними.

Ключові слова: нейронна мережа Хеммінга-Ліппмана, алгоритм навчання, режим ковзного вікна, розпізнавання зображень, корекція помилок, QR-коди, обчислювальна складність, оптимальні параметри.

Аннотация. Интенсивное развитие информационных технологий привело к созданию систем обмена данными, которые применяют комбинированное сжатие, защиту от поврежденной и хранение информации. Подобные системы обычно используют матричные коды, которые дают возможность компактно хранить большой объем информации и быстро распознаются сканирующим оборудованием. В данной статье решена задача распознавания и коррекции сильно поврежденных матричных кодов, а именно QR-кодов, в которых присутствует высокий уровень шума, отсутствуют ключевые элементы или накладываются цвета. Такие изображения не распознаются декодирующим программным обеспечением, поскольку структура детекторов элементов повреждена, а корректирующая способность встроенных кодов Рид-Соломона не позволяет исправить необходимую часть ошибок. Поэтому предложен алгоритм, основанный на использовании искусственной нейронной сети Хемминга-Липпмана с базой эталонных и обработке изображений в режиме скользящего окна, что упрощает процесс обучения сети без использования трудоемких вычислительных операций, больших объемов памяти и затрат времени, даже для изображений с высоким разрешением и большим размером. Процесс обучения сети состоит из двух частей: обработка изображений, распознавание и коррекция образца. Для того, чтобы добиться корректного распознавания, необходимо экспериментально выявлять оптимальные параметры обучения, благодаря которым в матрице эталонных образцов строки будут достаточно отличаться друг от друга. Для этого авторами на языке C# разработано программное обеспечение, с помощью которого и проведены необходимые экспериментальные исследования. Определены условия корректной работы нейронной сети (оптимальные значения размера скользящего окна и порога для различных размеров изображений QR-кодов), а также случаи, когда возможны ошибки распознавания и нестабильности ее выходов. Результаты исследований показывают, что разработанный алгоритм можно применять как дополнительную процедуру распознавания и исправления QR-кодов в различных системах обмена данными.

Ключевые слова: нейронная сеть Хемминга-Липпмана, алгоритм обучения, режим скользящего окна, распознавание изображений, коррекция ошибок, QR-коды, вычислительная сложность, оптимальные параметры.

Abstract. The intensive development of information technology has led to the creation of data exchange systems, that using combined compression, protection against damage and information storage. Such systems usually use matrix codes, that allow to store a large amount of information compactly and recognized it quickly by scanning equipment. In this article has been solved the task of recognition and correction of high damaged matrix codes, namely QR codes, where there is a high level of noise, there are no key elements or colors are overlaid. Such images are not recognized by decoding software as the structure of the detectors of elements is damaged and the correcting ability of the built-in Reed-Solomon codes doesn't allow to correct the necessary part of errors. That's why the algorithm based on the usage of artificial Hamming-Lippmann neural network with the base of samples and image processing in the sliding window mode is offered, which simplifies the learning process of the network without appliance of labor-intensive computational operations, large volumes of memory and high time consumption, even for images with high resolution and big size. The network learning process consists of two parts: image processing, sample recognition and correction. In order to achieve correct recognition, it is necessary to identify experimentally the optimal parameters of learning, thanks to which the matrix of samples the rows will differ sufficiently from each other. For this purpose, the authors developed software in C# language, with the helping of which the necessary experimental researches were performed. The conditions of correct work of the neural network (optimal values of the sliding window size and the threshold for different sizes of QR-code images) has been determined, as well as cases, when errors of recognition and instabilities of its outputs are possible. The results of the researches show, that the developed algorithm can be applied as an additional procedure of recognition and correction of QR codes in different data exchange systems.

Key words: Hamming-Lippmann neural network, learning algorithm, sliding window mode, images recognition, error corrections, QR codes, computational complexity, optimal parameters.

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-25-32>.

Вступ

Розвиток сучасних інформаційних технологій сприяє створенню та поширенню систем обміну даними, які застосовують комбіноване стиснення, захист від пошкоджень та зберігання інформації. Зазвичай подібні системи включають матричні коди або двовимірні штрих-коди, такі як *Data Matrix*, *Aztec* і QR-коди, які є найбільш поширеними. Їхньою основною перевагою є легке розпізнавання сканувальним обладнанням (наприклад, камерою смартфона) і великий обсяг компактно збереженої інформації (до 7089 цифр або 4296 букв), що дозволяє використовувати QR-коди в торгівлі, виробництві, логістиці тощо. За даними веб-порталу *comScore* більше 20 мільйонів громадян США використовують смартфони для сканування цих кодів; в Японії вони розміщуються на всіх буклетах, довідниках і практично всіх товарах [1].

Актуальність

Математична основа та принципи кодування і декодування QR-кодів представлені в стандарті ISO 18004 [2]. Для коригування помилок у даних кодах застосовують вбудовані недвійкові циклічні коди Ріда-Соломона з алгоритмом декодування Берлекемпа-Мессі (альтернатива – алгоритми Судана та Гао), пошуком положення помилок та оцінювання їх значень за Ченем та Форні [3, 4], що дозволяє відновити інформацію, якщо пошкоджено до $R \leq 30\%$ матриці даних QR-коду [5, 6]. Саме тому ряд досліджень направлено на розв'язання задачі відновлення матриці даних для сильно пошкоджених QR-кодів ($R > 30\%$), коли коригуюча здатність кодів Ріда-Соломона не дозволяє виправити необхідну частину помилок, тобто присутній високий рівень шуму або накладаються кольори. Для цього науковці застосовують модифіковані конструкції вбудованих кодів [7] та алгоритми виявлення ключових ознак [8], що вимагає використання досить трудомістких обчислювальних операцій у режимі реального часу. Це може бути недоцільним у ситуаціях, коли на матриці QR-коду відсутні ключові елементи-детектори або їхня структура значно пошкоджена, оскільки у цьому випадку декодує програму забезпечення не зможе навіть зчитати QR-код, а відповідно не буде намагатись розпізнати його. У даній роботі пропонується інший підхід до розв'язання задачі розпізнавання та коригування QR-кодів, який базується на додатковому використанні бази еталонів та штучної нейронної мережі Хеммінга-Ліппмана (НМХЛ) [9-11] з обробленням зображень у режимі ковзного вікна. Запропонований алгоритм можна застосовувати у системах обміну даними різного функціонального призначення у випадку, якщо матриця даних QR-коду пошкоджена настільки сильно, що немає можливості зчитати або розпізнати даний код звичайним чином.

Мета

Метою роботи є розробка алгоритму розпізнавання та коригування сильно пошкоджених QR-кодів, його експериментальне дослідження і пошук оптимальних параметрів роботи.

Задачі

Для досягнення мети необхідно виконати наступні задачі:

- 1) проаналізувати математичний апарат НМХЛ;
- 2) розробити програмне забезпечення, в якому реалізовано запропоновані алгоритми;
- 3) провести експериментальне дослідження розпізнавання та коригування сильно пошкоджених QR-кодів з використанням запропонованих розробок.

Алгоритми навчання нейронної мережі, розпізнавання та коригування QR-кодів

Нехай існує еталонний набір зображень (база даних), призначений для навчання мережі. Необхідно порівняти зображення із рівнем шуму, подане на вхід нейронної мережі, з усіма еталонними зразками і зробити висновок, що воно є одним із зразків, або його не можна розпізнати точно.

Використаємо НМХЛ (рис. 1), яка складається з m вхідних (S), n прихованих (Z, A), n вихідних (Y) нейронів і 4 прошарків. Значення n дорівнює кількості опорних зображень, збережених мережею і представлених у вигляді m -мірних біполярних векторів $S = (s_1, \dots, s_m)$. Також у цій мережі використовуються значення зміщення b . Матриця ваг W розмірністю $m \times n$ містить значення синаптичних зв'язків мережі прямого поширення, матриця E розмірності $n \times n$ зберігає значення синаптичних зв'язків нейронів мережі *MaxNet*. Функція активації нейронів є пороговою.

Алгоритм навчання НМХЛ включає в себе наступну послідовність кроків:

Крок 1.1. Проаналізувати матрицю P пікселів розміром $H \times B$, де H і B – висота і ширина зображення відповідно, рахуючи кількість темних пікселів. Щоб визначити, чи є аналізований піксель темним, необхідно зробити перетворення у відтінки сірого за формулою [12]:

$$Y = 0,298839 \cdot R + 0,586811 \cdot G + 0,11435 \cdot B, \quad (1)$$

де R, G, B – значення відповідних кольорних складових.

Крок 1.2. Вибрати певне граничне значення L .

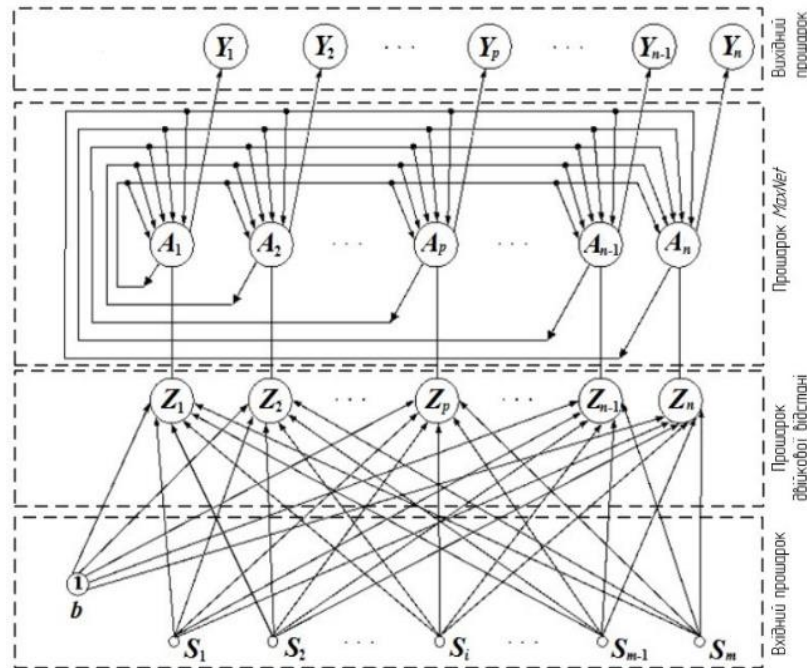


Рисунок 1 – Архітектура НМХЛ

Крок 1.3. Порівняти Y із L . Якщо $0 \leq Y \leq L$, тоді відповідний піксель вважається темним, інакше – білим.

Крок 1.4. Записати отриману двійкову матрицю зображення розміром $m = H \times B$. Таким чином, для всіх еталонних зображень n можна визначити відповідні бінарні вектори, які будуть використовуватися пізніше для навчання мережі і розпізнавання невідомих зразків.

Крок 1.5. Вибрати значення K (розмір ковзного вікна), яке дорівнює кількості рядків або стовпців, які будуть оброблятися (рис. 2). Змінюючи розмір вікна K , можна змінити точність аналізу зображення.

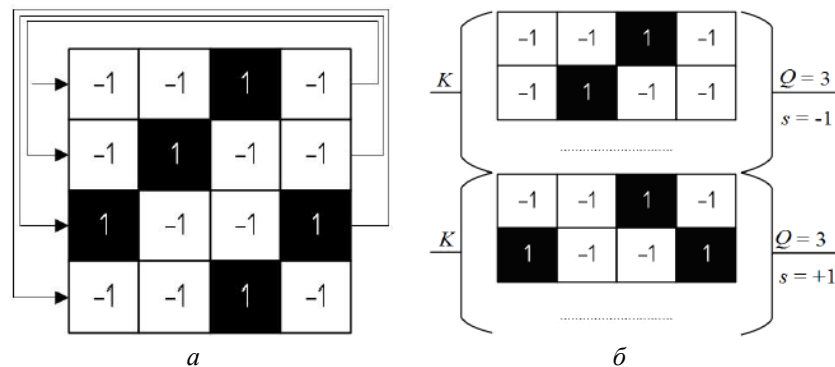


Рисунок 2 – Обробка зображення порядково (а) та у режимі ковзного вікна (б)

Крок 1.6. Вибрати порогову величину Q .

Крок 1.7. Обробити K рядків та обчислити N_i – кількість темних пікселів. Якщо $N_i \geq Q$, тоді i -ий елемент бінарного вектору S_i дорівнює «+1», інакше – «-1». Цей крок повторюється доки всі рядки не будуть оброблені. Тепер маємо вектор S із m елементами, для подачі на вхідні нейрони.

Крок 1.8. Повторити кроки 1.1-1.7 для всіх еталонних зображень із бази даних QR-кодів.

Крок 1.9. Базуючись на результатах, отриманих на кроці 1.8 (для всіх зображень), створити матрицю ваг W розмірністю:

$$M = m \times n = \lceil X / K \rceil \times n, \quad (2)$$

де X – висота H або ширина B даного зображення.

Модифікація, представлена в кроках 1.5-1.7 та 1.9, значно зменшить довжину вхідного вектору (особливо для зображень великих розмірів), що, в свою чергу, спростить обчислювальну складність навчання мережі.

Алгоритм розпізнавання та коригування QR-кодів можна сформулювати у такому вигляді:

Крок 2.1. Вибрати вхідне зображення для розпізнавання та виконати кроки 1.1-1.7.

Крок 2.2. Запустити процес розпізнавання [13, 14]. Вихідний сигнал для S нейронів має вигляд:

$$U_{output S_i} = f_s(U_{input S_i}) = s_i. \quad (3)$$

Крок 2.3. При поданні вхідного вектору кожен із Z нейронів від 1 до n обчислює свій вхідний сигнал відповідно до формули:

$$U_{input Z_k} = b + \sum_{i=1}^m w_{ik} \cdot U_{output S_i} = \frac{m}{2} + \frac{1}{2} \sum_{i=1}^m w_{ik} \cdot s_i, \quad (4)$$

де b – зміщення; w_{ik} – значення ваги від i вхідного нейрона до k нейрона Z (біполярне значення +1 або -1).

Крок 2.4. Обчислити значення функції активації для прихованих Z нейронів, використовуючи наступне правило:

$$U_{output Z_k} = f_z(U_{input Z_k}) = \begin{cases} 0, & \text{якщо } U_{input Z_k} \leq 0; \\ v \cdot U_{input Z_k}, & \text{якщо } 0 < U_{input Z_k} \leq U; \\ U, & \text{якщо } U_{input Z_k} > U, \end{cases} \quad (5)$$

де v – параметр нормалізації; $U = 1/v$ – константа.

Крок 2.5. Вихідний сигнал від Z нейронів є вхідним сигналом підмережі *MaxNet*:

$$U_{input A_k} = U_{output Z_k}. \quad (6)$$

Крок 2.6. Дана підмережа використовує ітераційний процес отримання максимального вхідного сигналу в наступному вигляді:

$$U_{output A_k}(0) = U_{output Z_k}, \quad (7)$$

$$U_{input Z_k}(t+1) = \sum_{i=1}^n w_{ik} \cdot U_{output A_i}(t) = U_{output A_k}(t) - \varepsilon \cdot \sum_{i=1, i \neq k}^n U_{output A_i}(t), \quad (8)$$

де t і $(t+1)$ – попередня та наступна ітерації; $0 < \varepsilon \leq 1$; $w_{ik} = 1$ для *MaxNet*, якщо $i = k$ (для мережі Хопфілда $w_{ik} = 0$) та $-\varepsilon$ – в іншому випадку.

Крок 2.7. Обчислити значення функції активації для прихованих A нейронів, використовуючи наступне правило:

$$U_{output A_k}(t) = f_A(U_{input A_k}(t)) = \begin{cases} U_{input A_k}(t), & \text{якщо } U_{input A_k} > 0; \\ 0, & \text{якщо } U_{input A_k} \leq 0. \end{cases} \quad (9)$$

Крок 2.8. Ітераційний процес (кроки 2.6 і 2.7) повторюється до тих пір, поки норма двох векторів не стане більшою, ніж значення eps_1 . Отже, умова стабілізації формулюється наступним чином [9]:

$$\|U_{output A_k}(t+1) - U_{output A_k}(t)\| = \sqrt{\sum_{k=1}^n (U_{output A_k}(t+1) - U_{output A_k}(t))^2} \leq eps_1. \quad (10)$$

Крок 2.9. Вихідні сигнали нейронів A надходять, як вхідні в нейрони Y , що можна представити у формі:

$$U_{output Y_k} = U_{output A_k}. \quad (11)$$

Крок 2.10. Обчислити значення функції активації для вихідних Y нейронів, використовуючи відхилення $eps_2 = 10^{-5} - 10^{-6}$ або нуль в ідеальному випадку. Це можна зобразити в такому вигляді:

$$U_{output Y_k} = f_Y(U_{input Y_k}) = \begin{cases} 1, & \text{якщо } U_{input Y_k} > eps_2; \\ 0, & \text{якщо } U_{input Y_k} \leq eps_2, \end{cases} \quad (12)$$

або, якщо потрібно виконати багато ітерацій, то необхідно вибрати максимальне значення $U_{input Y}$:

$$U_{output Y_k} = \max_{1,n} = (\{U_{input Y_k}\}). \quad (13)$$

Крок 2.11. Відкоригувати QR-код. Індекс позитивного елемента $k = \overline{1, n}$ вихідного сигналу нейронів $U_{\text{output } y_k}$ вказує номер еталонного зображення, яке найкраще підходить вхідному тестовому сигналу.

Виходом мережі в ідеальному випадку повинен бути вектор з одним позитивним елементом. Але іноді вхідна вибірка може бути дуже зашумленою, і на виході мережі може з'явитися кілька позитивних значень. У цьому випадку вважається, що мережа не може однозначно класифікувати вхідне зображення. Але вдало підібрані параметри алгоритму навчання можуть допомогти подати цей недолік. У наступному розділі статті продемонстрована працездатність запропонованих алгоритмів на практиці.

Результати експериментальних досліджень

Для вирішення поставленої задачі на мові програмування C# нами було розроблене універсальне програмне забезпечення. Щоб визначити оптимальні значення порогової величини Q і розміру ковзного вікна K для запропонованого алгоритму, були проведені експерименти, в яких використано зображення QR-кодів розміром 200×200 , 512×512 та 1024×1024 з високим рівнем шуму та пошкоджень, відсутністю деталей або злиттям кольорів.

Приклад. На рис. 3 і 4 представлено 5 навчальних і тестових зображень (QR-коди із пошкодженнями). Дані зображення не розпізнаються програмним забезпеченням для декодування, оскільки структура детекторів елементів пошкоджена, а коригуюча здатність вбудованих кодів Ріда-Соломона не дозволяє виправити необхідну частину помилок. Отже, застосуємо додаткову процедуру виправлення помилок, використовуючи НМХЛ, базу еталонних QR-кодів та запропонований алгоритм. Експериментальні результати розпізнавання зображень QR-кодів різних розмірів представлені в табл. 1-3. Використовувалась лінійна залежність $Q = \alpha \cdot K + \beta$ для порогової величини Q і розміру ковзного вікна K , для якої експериментально вибрані значення констант $[\alpha; \beta] = [(50; 0), (200; 0), (400; 100)]$.



Рисунок 3 – Навчальні зображення

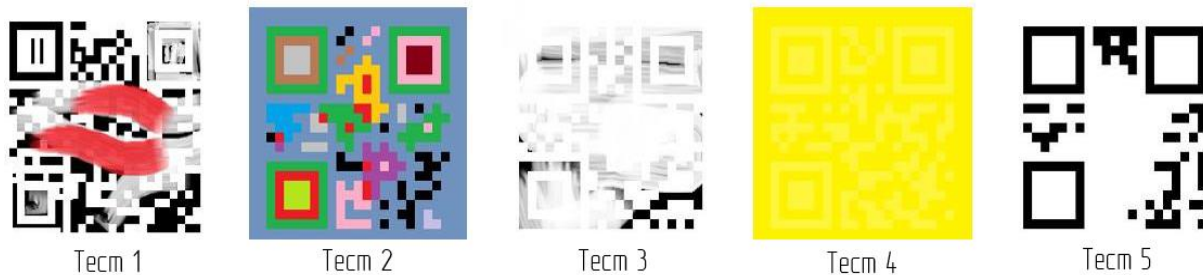


Рисунок 4 – Тестові зображення

Таблиця 1 – Результати експериментів (лінійна залежність $Q = 50 \cdot K$, зображення розміром 200×200)

K	Q	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Правильно розпізнані	Результат
1-7	50-350	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
8	400	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
9	450	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО
10	500	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО
11	550	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
12	600	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО
13	650	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
14	700	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
15	750	НЕВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
16	800	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО

Таблиця 2 – Результати експериментів (лінійна залежність $Q = 200 \cdot K$, зображення розміром 512×512)

K	Q	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат
9-15	1800-3000	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
16	3200	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
17	3400	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
18	3600	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
19	3800	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
20	4000	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
21	4200	ВІРНО	НЕВІРНО	НЕВІРНО	ВІРНО	ВІРНО	3	ПОГАНО
22	4400	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
23	4600	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
24	4800	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
25	5000	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	ВІРНО	3	ПОГАНО
26	5200	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
27	5400	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
28	5600	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО

Таблиця 3 – Результати експериментів (залежність $Q = 400 \cdot K + 100$, зображення розміром 1024×1024)

K	Q	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат
1-25	500-10100	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
26	10500	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
27	10900	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО
28	11300	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО

На основі даних з табл. 1-3 можна зробити висновок, що оптимальними значеннями параметрів K і Q для тестових зображень з розмірами H на $B \in [H \times B; K; Q] = [(200 \times 200; 8; 400), (512 \times 512; 16; 3200), (1024 \times 1024; 26; 10500)]$. Використовуючи ці значення, НМХЛ правильно розпізнає всі тестові QR-коди, але для досягнення меншої обчислювальної складності навчання і більшої точності розпізнавання може бути виконана процедура налаштування параметра K_i , коли $(n-1)$ тестових зображень були розпізнані, наприклад, $[H \times B; \{K_i\}] = [(200 \times 200; \{9; 10\}), (512 \times 512; \{17-20; 22-24; 26\}), (1024 \times 1024; \{27, 28\})]$.

У табл. 4 демонструється той факт, що, коли порогова величина Q менша за значення розміру ковзного вікна K або дорівнює йому, то НМХЛ не може правильно виконати розпізнавання зображень.

Таблиця 4 – Результати експериментів (зображення розміром 200×200 , $K \geq Q$)

K	Q	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат
1	1	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
4	2	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
4	4	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
9	4	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
9	9	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
16	8	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
16	16	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
25	12	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
25	25	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
36	18	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
36	36	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
49	24	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
64	32	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
81	40	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
100	50	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО

У табл. 5 показано приклад процедури налаштування, заснованої на інтервалах із табл. 1, для розміру ковзного вікна $K = 9$ (розмір зображення 200×200), яке більше за оптимальне значення, при якому всі QR-коди успішно розпізнаються.

Результати даних експериментів показали ефективність запропонованої процедури корекції, тобто її можна використовувати у системах обміну даними для виправлення сильно пошкоджених QR-кодів.

Таблиця 5 – Результати процедури налаштування в програмному забезпеченні

K_i	Q	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат	Оптимальне значення
9	400	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	3	ПОГАНО	$K = K_i - 1 = 8$
	455	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	460	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	465	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	470	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	475	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	480	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	485	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	490	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	495	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	500	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
510	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО		

Висновки

У статті запропоновано алгоритм розв'язання задачі розпізнавання та коригування сильно пошкоджених QR-кодів, який базується на використанні НМХЛ з базою еталонних зразків та обробці зображень у режимі ковзного вікна, що не потребує використання трудомістких обчислювальних операцій, великих обсягів пам'яті та значних витрат часу. У ході експериментальних досліджень з використанням розробленого програмного забезпечення визначено набір оптимальних параметрів алгоритму (розмір ковзного вікна K та порогового значення Q) для різних розмірів зображень QR-кодів. Розроблений алгоритм можна використовувати як додаткову процедуру виправлення QR-кодів у системах обміну даними тоді, коли скануючому обладнанню не вдається зчитати та розпізнати QR-код звичайним чином.

Список літератури

- [1] QR Code. [Electronic resource]. Available: <https://www.comscore.com/Insights/Infographics/20-Million-Americans-Scanned-a-QR-Code-in-October>. Accessed: 01-Oct-2019.
- [2] ISO/IEC 18004:2015. Estonian Centre for Standardization. [Electronic resource]. Available: <https://www.evs.ee/products/iso-iec-18004-2015>. Accessed: 01-Oct-2019.
- [3] T.K. Moon, *Error Correction Coding: Mathematical Methods & Algorithms*, John Wiley & Sons, 2005.
- [4] R.H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. Chichester: Wiley, 2006.
- [5] G. Barland, Error Correction and QR Codes. [Electronic resource]. Available: <https://www.stthomas.edu/media/collegeofartsandsciences/mathematics/pdf/2017AndersonGroupBarlandCAMReport.pdf>. Accessed: 01-Oct-2019.
- [6] QR Code Tutorial from Thonky. [Electronic resource]. Available: <https://www.thonky.com/qr-code-tutorial/>. Accessed: 01-Oct-2019.
- [7] I. Szentandrási, A. Herout, M. Dubska, "Fast Detection and Recognition of QR codes in High-Resolution Images", *Published in SCCG 2012*, pp. 129-136, 2012.
- [8] K. Suran, "QR Code Image Correction based on Corner Detection and Convex Hull Algorithm", *Journal of Multimedia*, vol. 8, no. 6, pp. 662-668, 2013.
- [9] R.P. Lippmann, B. Gold, M.L. Malpass, *A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification*, Ft. Belvoir: Defense Technical Information Center, pp. 1-35, 1987.
- [10] K. Koutroumbas, N. Kalouptsidis, "Generalized Hamming Networks and Applications", *Neural Networks*, vol. 18, pp. 896-913, 2005.
- [11] А.А. Ежов, А.С. Новиков, "Сеть Хэмминга и ее применение для решения задачи распознавания подписей", *Известия ТулГУ*, №11, С. 56-64, 2016.
- [12] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*. Prentice Hall, 2008.
- [13] Ю.Ю. Иванов, *Методи штучного інтелекту та наука про дані: лекції, лабораторні роботи, алгоритми та задачі*, Вінниця, Україна: ВНТУ, 2018.
- [14] L.V. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Delhi: Pearson Education, 2004.

References

- [1] QR Code. [Electronic resource]. Available: <https://www.comscore.com/Insights/Infographics/20-Million-Americans-Scanned-a-QR-Code-in-October>. Accessed: 01-Oct-2019.
- [2] ISO/IEC 18004:2015. Estonian Centre for Standardization. [Electronic resource]. Available: <https://www.evs.ee/products/iso-iec-18004-2015>. Accessed: 01-Oct-2019.
- [3] T.K. Moon, *Error Correction Coding: Mathematical Methods & Algorithms*, John Wiley & Sons, 2005.
- [4] R.H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. Chichester: Wiley, 2006.
- [5] G. Barland, Error Correction and QR Codes. [Electronic resource]. Available: <https://www.stthomas.edu/media/collegeofartsandsciences/mathematics/pdf/2017AndersonGroupBarlandCAMReport.pdf>. Accessed: 01-Oct-2019.
- [6] QR Code Tutorial from Thonky. [Electronic resource]. Available: <https://www.thonky.com/qr-code-tutorial/>. Accessed: 01-Oct-2019.
- [7] I. Szentandras, A. Herout, M. Dubska, "Fast Detection and Recognition of QR codes in High-Resolution Images", *Published in SCCG 2012*, pp. 129-136, 2012.
- [8] K. Suran, "QR Code Image Correction based on Corner Detection and Convex Hull Algorithm", *Journal of Multimedia*, vol. 8, no. 6, pp. 662-668, 2013.
- [9] R.P. Lippmann, B. Gold, M.L. Malpass, *A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification*, Ft. Belvoir: Defense Technical Information Center, pp. 1-35, 1987.
- [10] K. Koutroumbas, N. Kalouptsidis, "Generalized Hamming Networks and Applications", *Neural Networks*, vol. 18, pp. 896-913, 2005.
- [11] A.A. Ezhov, A.S. Novykov, "Set Khemmyuha y ee pryomenenye dlia resheniya zadachy raspoznavaniya podpysei", *Yzvestyia TulHU*, №11, S. 56-64, 2016.
- [12] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*. Prentice Hall, 2008.
- [13] І.І. Іванов, *Методи штучного інтелекту та наука про дані: лекції, лабораторні роботи, алгоритми та задачі*, Вінниця, Україна: VNTU, 2018.
- [14] L.V. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Delhi: Pearson Education, 2004.

Відомості про авторів

Кветний Роман Наумович – д. т. н., професор, завідувач кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ, член-кореспондент Національної академії педагогічних наук України, заслужений діяч науки і техніки, тел.: (0432) 59-82-43.

Іванов Юрій Юрійович – к. т. н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій.

Пивошенко Володимир Володимирович – студент кафедри автоматизації та інтелектуальних інформаційних технологій.

Васюра Анатолій Степанович – к. т. н., професор кафедри автоматизації та інтелектуальних інформаційних технологій, академік Української технологічної академії.

R. N. Kvetnyy, V. V. Pivoshenko, Yu. Yu. Ivanov, A. S. Vasyura

АЛГОРИТМ РАСПОЗНАВАНИЯ И КОРРЕКЦИИ ПОВРЕЖДЕННЫХ QR-КОДОВ

Винницкий национальный технический университет

R. N. Kvyetnyy, V. V. Pivoshenko, Yu. Yu. Ivanov, A. S. Vasyura

ALGORITHM FOR RECOGNITION HIGHLY CORRUPTED QR-CODES

Vinnitsa National Technical University

КОМП'ЮТЕРНІ СИСТЕМИ ТА КОМПОНЕНТИ

УДК 621.3

О. Д. Азаров, М. Р. Обертюх, С. А. Кирилашук

РЕЗИСТИВНІ МАТРИЧНІ ДІЛЬНИКИ СТРУМУ ДЛЯ БАГАТОРОЗРЯДНИХ ЦАП ІЗ ВАГОВОЮ НАДЛИШКОВІСТЮ

Вінницький національний технічний університет, Вінниця

Анотація. У статті розглянуті резистивні матричні дільники струму, які можуть застосовуватися в багаторозрядних цифроаналогових і аналогоцифрових перетворювачах з ваговою надлишковістю. Розглянуто запропоновані підходи щодо побудови секційних матричних дільників струму, в яких ваги розрядів у рамках секцій пропорційні вагам розрядів у двійковій системі числення, а надлишковість реалізується за допомогою резисторів зв'язку між секціями. Виведено аналітичні співвідношення для розрахунків значень резисторів зв'язку та кінцевих резисторів в даних дільниках для різних варіантів підбору зазначених типів резисторів. Також передбачена можливість завдання різних значень вагової надлишковості в вищезначених секційних матричних дільниках струму, в яких ваги розрядів у рамках секцій пропорційні вагам розрядів у двійковій системі числення. Проведено комп'ютерне моделювання розрахованих матричних дільників струму за допомогою програми для аналогового і цифрового моделювання електричних та електронних кіл MicroCap 11 та підтверджено правильність виведених співвідношень. Розглянуто запропоновані підходи щодо побудови секційних матричних дільників струму, в яких ваги розрядів секцій пропорційні вагам розрядів надлишкової системи числення на основі чисел Фібоначчі. Також виведено аналітичні співвідношення для розрахунків значень резисторів зв'язку та кінцевих резисторів в даних дільниках для різних варіантів підбору зазначених типів резисторів. Проведено комп'ютерне моделювання розрахованих матричних дільників струму за допомогою програми MicroCap 11 та підтверджено правильність співвідношень та розрахунків. Розраховані коефіцієнти надмірності отриманих резистивних матричних дільників струму і виконано порівняння їх характеристик. Дані рекомендації щодо їх застосування залежно від висунутих вимог.

Ключові слова: резистивні матричні дільники, ЦАП, вагова надлишковість.

Аннотация. В статье рассмотрены резистивные матричные делители тока, которые могут применяться в многоразрядных цифроаналоговых и аналого-цифровых преобразователях с весовой избыточностью. Рассмотрены подходы к построению секционных матричных делителей тока, в которых веса разрядов в рамках секций пропорциональны весам разрядов в двоичной системе счисления, а избыточность реализуется с помощью резисторов связи между секциями. Выведены аналитические соотношения для расчетов значений резисторов связи и оконечных резисторов в данных делителях для различных вариантов подбора указанных типов резисторов. Также предусмотрена возможность задания различных значений весовой избыточности в вышеуказанных секционных матричных делителях тока, в которых веса разрядов в рамках секций пропорциональны весам разрядов в двоичной системе счисления. Проведено компьютерное моделирование рассчитанных матричных делителей тока с помощью программы для аналогового и цифрового моделирования электрических и электронных цепей MicroCap 11 и подтверждена правильность выведенных соотношений. Рассмотрены предложенные подходы к построению секционных матричных делителей тока, в которых веса разрядов секций пропорциональны весам разрядов избыточной системы счисления на основе чисел Фибоначчи. Также выведены аналитические соотношения для расчетов значений резисторов связи и оконечных резисторов в данных делителях для различных вариантов подбора указанных типов резисторов. Проведено компьютерное моделирование рассчитанных матричных делителей тока с помощью программы MicroCap 11 и подтверждена правильность соотношений и расчетов. Рассчитаны коэффициенты избыточности полученных резистивных матричных делителей тока и выполнено сравнение их характеристик. Даны рекомендации по их применению в зависимости от предъявляемых требований.

Ключевые слова: резистивные матричные делители, ЦАП, весовая избыточность

Abstract. The article discusses resistive matrix current dividers that can be used in multi-digit digital-to-analog and analog-to-digital converters with weight redundancy. The proposed approaches to constructing sectional matrix current dividers in which the weights of the digits within the sections are proportional to the weights of the digits in the binary number system, and the redundancy is realized using connection resistors between the sections are given. Analytical relations are derived for calculating the values of connection resistors and terminal resistors in these dividers for various options for selecting these types of resistors. It is also possible to set different weight redundancy values in sectional matrix current dividers, in which the weight of the digits within the sections is proportional to the weight of the digits in the binary number system. Computer simulation of the calculated matrix current dividers was carried out using the MicroCap 11 program for analog and digital modeling of electrical and electronic circuits and the correctness of the derived relations was confirmed. Approaches to constructing sectional matrix current dividers in which the weights of the digits of the sections are proportional to the weights of the digits of the redundant number system based on the Fibonacci numbers are suggested. Analytical relations are also derived for calculating the values of connection resistors and terminal resistors in these dividers for various options for selecting these types of resistors. Computer simulations of the calculated matrix current dividers are also performed using the MicroCap 11 program and correctness of the relations and calculations are confirmed. The redundancy coefficients of the obtained resistive matrix current dividers are calculated and their characteristics are compared. Recommendations of their application, depending on the requirements are given.

Key words: resistive matrix dividers, DACs, weight redundancy.

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-33-39>.

Вступ

Резистивні матричні дільники струму широко застосовуються в цифрово-аналогових (ЦАП) та аналогово-цифрових (АЦП) перетворювачах [1, 7] для формування набору опорних струмів, які пропорційні вагам двійкової або іншої позиційної системи числення, яка застосовується. Але їх застосування обмежується характеристиками точності резисторів [5, 7]. У Вінницькому національному технічному універси

теті існує наукова школа [8, 9], що досліджує застосування надлишкових позиційних систем числення в АЦП та ЦАП. Це дає змогу досягти підвищення точності (або зниження вартості) та швидкодії [10, 11]. У статті розглядаються деякі аспекти побудови резистивних матричних дільників струму для цифрово-аналогових перетворювачів із ваговою надлишковістю.

Актуальність

Найбільш простим методом побудови резистивних матричних дільників струму є схема, яка наведена у [5]. Опорні струми будуть пропорційні вагам резисторів, і якщо прийняти $R_i = 2^{i-1} \cdot R$, то вони будуть пропорційні вагам розрядів двійкової системи числення. Якщо ж прийняти, що $R_i = \alpha^{i-1} \cdot R$, де $1 < \alpha < 2$, то струми будуть пропорційні вагам надлишкової позиційної системи числення з основою α . Недоліком такої схеми є великий розкид номіналів резисторів, особливо при збільшенні кількості розрядів n . Тому досягнення необхідної похибки при виготовленні резисторів ускладнюється.

Усунути такий недолік можливо шляхом застосування драбинкових дільників, відповідні схеми яких розглянуто в [5]. Щоб отримати опорні струми, що пропорційні вагам розрядів двійкової системи числення, потрібно прийняти $R_1 = R_0 = 2 \cdot R$ та $R_2 = R$, а для отримання опорних струмів, що пропорційні вагам розрядів надлишкової позиційної системи числення з основою α : $R_1 = \alpha \cdot R$, $R_i = \alpha \cdot R / (\alpha - 1)$, та $R_2 = (\alpha - 1) \cdot R$ [10]. Як бачимо, для заданих опорних струмів, які пропорційні вагам розрядів надлишкової позиційної системи числення, необхідно реалізувати резистори з ірраціональними номіналами, що викликає технологічні труднощі. Доцільніше мати дільники, в яких можна було б мати цілочислові ваги розрядів.

Отже, можна стверджувати, що створення простих у мікроелектронному виконанні резистивних решіток (із цілочисловими номіналами резисторів), які призначені для формування опорних струмів, що пропорційні вагам розрядів позиційної надлишкової системи числення, є актуальною задачею.

Мета

Запропонувати підхід проектування резистивних секційних матричних дільників струму з цілочисловими відношеннями ваг розрядів, що призначені для багаторозрядних ЦАП із ваговою надлишковістю.

Задачі досліджень

1. Аналіз запропонованих підходів побудови секційних матричних дільників струму, в яких ваги розрядів у рамках секцій пропорційні вагам розрядів у двійковій системі числення, а надлишковість реалізується за допомогою резисторів зв'язку між секціями.
2. Аналіз запропонованих підходів побудови секційних матричних дільників струму, в яких ваги розрядів секцій пропорційні вагам розрядів надлишкової системи числення на основі чисел Фібоначчі.
3. Порівняння характеристик розглянутих резистивних матриць та надання рекомендацій щодо їх використання залежно від висунутих вимог.

Розв'язання задач досліджень

Для розв'язання поставленої задачі пропонується використовувати секційний матричний дільник струму. Приклад такої схеми наведено на рис. 1

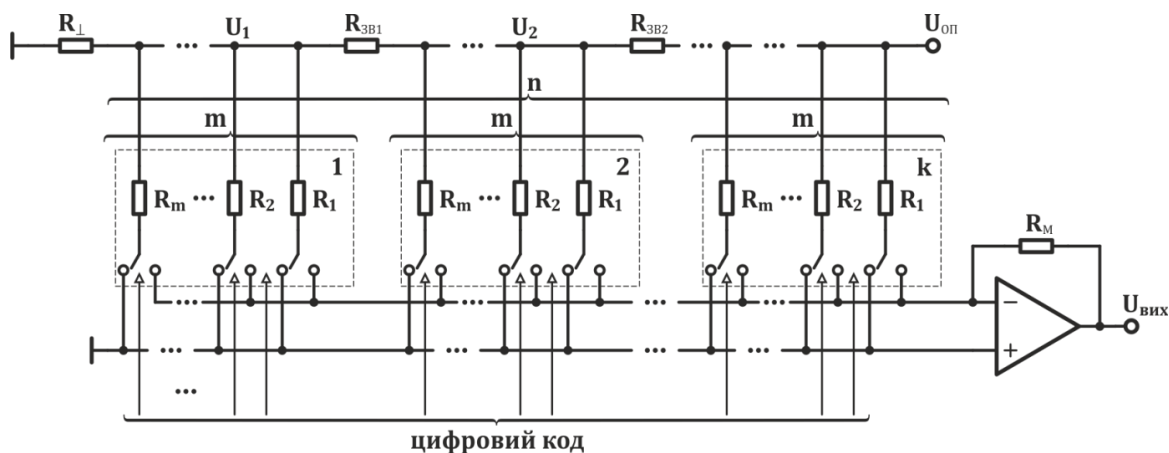


Рисунок 1 – ЦАП на основі секційного матричного дільника струму

Вона містить k секцій, кожна з яких складається з m паралельно з'єднаних резисторів (R_2, \dots, R_m), а між секціями розташовано резистори зв'язку (R_{3B1}, R_{3B2}, \dots), і кінцевий резистор R_{11} , що заземлюється. За допомогою ключів, якими керують цифрові коди, розрядні струми під'єднуються або на землю, або на

вхід операційного підсилювача, де за допомогою резистора R_m сумуються, масштабуються та перетворюються у вихідну напругу.

Виведемо основні відношення, що пов'язують значення струму та резисторів у схемі (рис. 1). Опори резисторів секції будуть дорівнювати R_1, R_2, \dots, R_m та опір всієї резистивної секції буде мати значення:

$$R_C = \frac{1}{\sum_{i=1}^m \frac{1}{R_i}}. \quad (1)$$

Сумарний струм, який протікає через R_C та $R_{\perp 1}$, дорівнює струму, який протікає через R_{3B1} :

$$\frac{U_1}{R_{\perp 1} \parallel R_C} = \frac{U_2}{R_{3B1} + R_{\perp 1} \parallel R_C} \Rightarrow \frac{U_2}{U_1} = \frac{R_{3B1}}{R_{\perp 1} \parallel R_C} + 1, \quad (2)$$

де \parallel - паралельне з'єднання резисторів.

Задамо коефіцієнт K_i як відношення струмів, що протікають через резистор R_m наступного каскаду та резистор R_1 попереднього каскаду, який можна використовувати для завдання надлишковості.

$$\frac{U_2}{R_m} = K_i \cdot \frac{U_1}{R_1} \Rightarrow \frac{U_2}{U_1} = K_i \cdot \frac{R_m}{R_1}. \quad (3)$$

Із умови (2) та (3) враховуючи (1) отримаємо:

$$R_{3B1} = (R_{\perp 1} \parallel R_C) \cdot \left(K_i \cdot \frac{R_m}{R_1} - 1 \right), \quad (4)$$

$$R_{\perp 2} = R_{3B1} + R_{\perp 1} \parallel R_C = K_i \cdot \frac{R_m}{R_1} \cdot (R_{\perp 1} \parallel R_C). \quad (5)$$

Для кожного розряду відповідно до [10], рівень надлишковості δQ , обчислюється за формулою:

$$\delta Q = \frac{\sum_{j=1}^{i-1} Q_j - Q_i}{\sum_{j=1}^i Q_j}. \quad (6)$$

Розглянемо випадок, коли струми, що протікають через резистори секції будуть пропорційні степеню двійки, тобто утворюють ряд $I_0, 2 \cdot I_0, 4 \cdot I_0, \dots, 2^n \cdot I_0$. У цьому випадку опори резисторів секції будуть дорівнювати $R_i = 2^{i-1} \cdot R$, де R – опір резистора старшого розряду. Відповідно до (1) опір секції визначається:

$$R_C = \frac{R}{\sum_{i=1}^m \frac{1}{2^{i-1}}} = \frac{2^{m-1}}{2^m - 1} \cdot R. \quad (7)$$

Будемо вважати наявністю надлишковості ситуацію, коли струми, які протікають через резистор R_m наступного каскаду та резистор R_1 попереднього каскаду рівні, тобто у формулі (3) коефіцієнт K_i дорівнює 1. (Можна задати інші умови надлишковості, наприклад, якщо струм через резистор R_m наступно-

го каскаду прийняти рівним сумі струмів через резистори R_1 та R_2 попереднього каскаду, то коефіцієнт K_i дорівнює $3/2$).

Розглянемо два підходи розрахунку резисторів $R_{\perp 1}$ та $R_{3B1}, R_{3B2}, \dots, R_{3Bn}$:

I спосіб. Резистор $R_{\perp 1}$ вибирають довільно. Тоді значення резистора R_{3B2} визначається за формулою (8), враховуючи значення обчисленні у (4), (5) та (7), причому замість $R_{\perp 1}$ приймають значення $R_{\perp 2}$, яке обчислюється за формулою (9). Інші значення R_{3B} , обчислюються рекурсивно.

$$R_{3B1} = \frac{R_{\perp 1} \cdot 2^{m-1} \cdot R}{R_{\perp 1} \cdot (2^m - 1) + 2^{m-1} \cdot R} \cdot (2^{m-1} - 1), \quad (8)$$

$$R_{\perp 2} = \frac{R_{\perp 1} \cdot 2^{2 \cdot (m-1)} \cdot R}{R_{\perp 1} \cdot (2^m - 1) + 2^{m-1} \cdot R}. \quad (9)$$

II спосіб. Значення $R_{3B1}, R_{3B2}, \dots, R_{3Bn}$ та відповідно, $R_{\perp 1}, R_{\perp 2}, \dots, R_{\perp n}$ рівні. З формул (4), (5) та (7) отримуємо:

$$R_{\perp} = \frac{2^{m-1} \cdot (2^m - 1)}{2^m - 1} \cdot R, \quad (10)$$

$$R_{3B} = \frac{(2^{m-1} - 1)^2}{2^m - 1} \cdot R. \quad (11)$$

Обчислимо значення R_{3B1} та R_{3B2} для способу 1 за умов: $p=1, m=3,4$ та приймемо $R=10\text{кОм}$, $R_{\perp 1}=2^{m-1} \cdot R$.

n	R_{3B1}	R_{3B2}
3	$\frac{3}{2} R = 15 \text{ кОм}$	$\frac{4}{3} R = 13.3 \text{ кОм}$
4	$\frac{7}{2} R = 35 \text{ кОм}$	$\frac{56}{17} R = 32.9 \text{ кОм}$

Результат моделювання схеми у MicroCap 11 для $m=4$ наведено на рис. 2а. Розрахуємо значення R_{\perp} та R_{3B} , для способу 2 за умов: $p=1, m=3,4$ та приймемо $R=10\text{кОм}$.

Коефіцієнт надлишковості для резистивної решітки з трьома секціями для старшого розряду та $m=4$ відповідно до (6) становитиме 0,06 (6%).

n	R_{\perp}	R_{3B}
3	$\frac{12}{7} R = 17.1 \text{ кОм}$	$\frac{9}{7} R = 12.9 \text{ кОм}$
4	$\frac{56}{15} R = 37.3 \text{ кОм}$	$\frac{49}{15} R = 32.7 \text{ кОм}$

Результат моделювання схеми у MicroCap 11 для $m=4$ наведено на рис. 2б.

Розглянемо випадок, коли струми через резистори секції пропорційні числам Фібоначчі, тобто утворюють ряд $I_0, 2 \cdot I_0, 3 \cdot I_0, 5 \cdot I_0, 8 \cdot I_0, 13 \cdot I_0, \dots$. Опори резисторів секції при цьому будуть дорівнювати $R, R/2, R/3, R/5, R/8, R/13, \dots$, де R – опір резистора молодшого розряду. Розглянемо секцію з п'яти резисторів. Тоді відповідно до (1):

$$R_C = \frac{1}{\frac{8}{R} + \frac{5}{R} + \frac{3}{R} + \frac{2}{R} + \frac{1}{R}} = \frac{R}{19}. \quad (12)$$

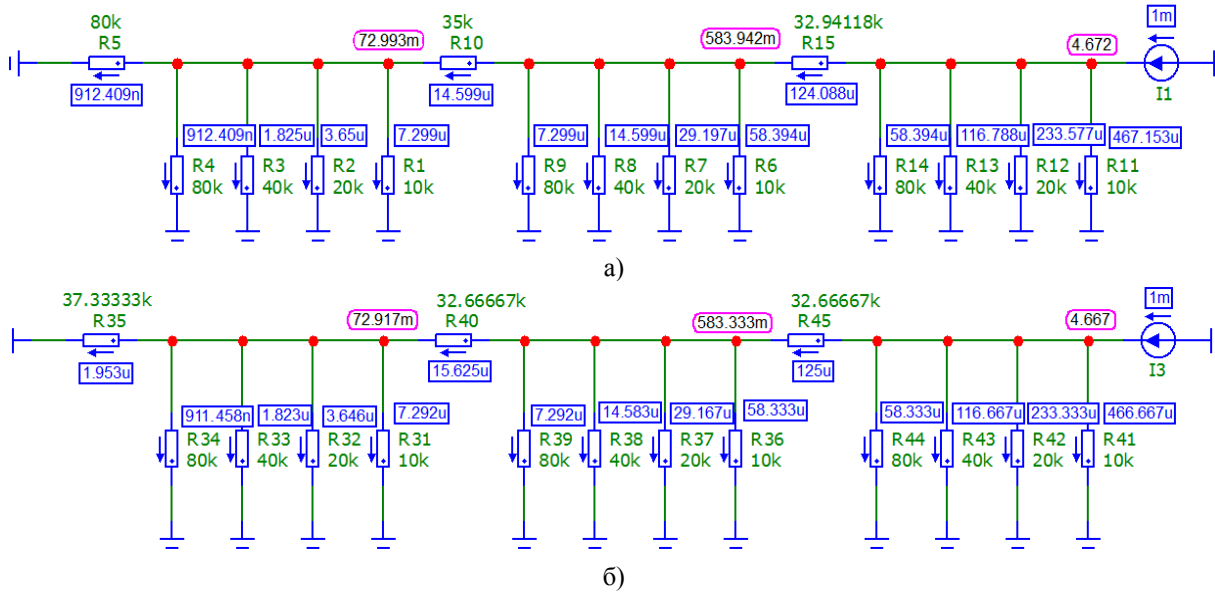


Рисунок 2 – Результат моделювання секційного матричного дільника струму із двійково-зваженими резисторами: а) I спосіб; б) II спосіб;

Завдання значень струмів, пропорційних числам ряду Фібоначчі є введенням надлишковості. Нехай відношення струмів, що протікають через резистор R_m наступного каскаду та резистор R_1 попереднього каскаду дорівнює $13/8$, відповідно ряду Фібоначчі ($k = 13/8$). Розглянемо два можливих способи розрахунку резисторів $R_{\perp 1}$ і R_{3B1} , R_{3B2} , ..., R_{3Bn} :

I спосіб. Якщо резистор $R_{\perp 1}$ вибираємо довільно. Тоді за формулами (4), (5) та (12) значення резистора R_{3B2} визначається за формулою (13), приймаючи замість $R_{\perp 1}$ значення $R_{\perp 2}$, що обчислюється за формулою (14). Інші значення R_{3B} обчислюються рекурсивно.

$$R_{3B1} = \frac{12 \cdot R_{\perp 1} \cdot R}{19 \cdot R_{\perp 1} + R}, \tag{13}$$

$$R_{\perp 2} = \frac{13 \cdot R_{\perp 1} \cdot R}{19 \cdot R_{\perp 1} + R}. \tag{14}$$

II спосіб. Значення R_{3B1} , R_{3B2} , ..., R_{3Bn} та відповідно, $R_{\perp 1}$, $R_{\perp 2}$, ..., $R_{\perp n}$ рівні. З формул (4), (5) та (12) отримуємо:

$$R_{\perp} = \frac{12}{19} \cdot R, \tag{15}$$

$$R_{3B} = \frac{144}{247} \cdot R. \tag{16}$$

Обчислимо значення R_{\perp} та R_{3B} , для способу 2, приймаючи $R=120\text{k}\Omega$.

n	R_{\perp}	R_{3B}
4	75.8 кОм	69.9 кОм

Результат моделювання схеми в MicroCap 11 наведено на рис. 3а.

Обчислимо значення R_{3B1} та R_{3B2} , для способу 1 приймаючи $R=120\text{k}\Omega$, $R_{\perp 1}=120\text{k}\Omega$.

n	R_{3B1}	R_{3B2}
4	$\frac{3}{5} R = 72 \text{ кОм}$	$\frac{52}{89} R = 70.1 \text{ кОм}$

Результат моделювання схеми в MicroCap 11 наведено на рис. 3б.

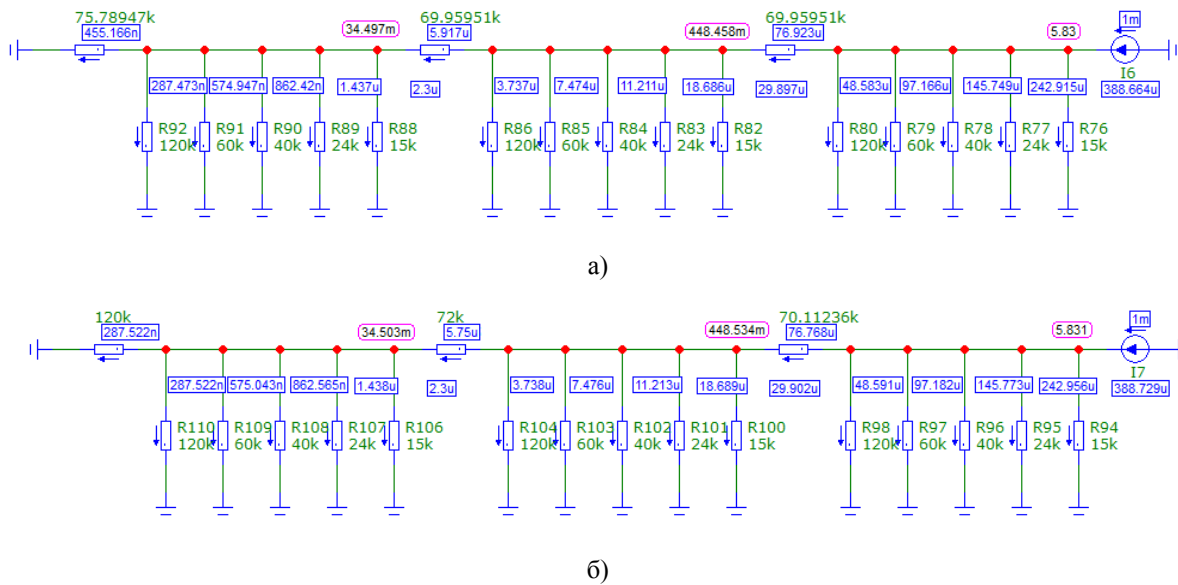


Рисунок 3 – Результат моделювання секційного матричного дільника струму із резисторами зваженими на основі ряду Фібоначчі: а) I спосіб; б) II спосіб.

Коефіцієнт надлишковості для резистивної решітки із трьома секціями для старшого розряду відповідно до (6) становитиме 0,22 (22%).

У розглянутих випадках побудови резистивних решіток значення R_L та R_{3B} отримуємо нецілими, але вимоги до їх точності можуть бути знижені шляхом використання самокалібровки. При використанні двійково-зважених резисторів не вимагається перетворення коду, що подається на ЦАП. Але коефіцієнт надлишковості сильно варіюється серед розрядів, що зменшує можливості калібровки окремих розрядів.

Висновки

1. Проаналізовані запропоновані підходи до побудови секційних матричних дільників струму, у яких ваги розрядів всередині секції пропорційні вагам розрядів у двійковій системі числення, а надлишковість задається вибором номіналів резисторів зв'язку між секціями.

2. Проаналізовані запропоновані підходи до побудови секційних матричних дільників струму, в яких ваги розрядів всередині секції пропорційні цілочисловим вагам розрядів надлишкової системи числення на основі чисел Фібоначчі.

3. Запропоновані рекомендації з проектування розглянутих секційних матричних дільників струму.

Література

- [1] Mosley J.D. Self-calibrating 16-bit A/D converter guarantees no missing codes to 50 kHz // EDN. – 1987. – vol. 32, № 2.
- [2] Hokazono, K., Kanemoto, D., Pokharel, R., Tomar, A., Kanaya, H., Yoshida, K.. “A low-glitch and small-logic-area Fibonacci Series DAC”. IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), 2011. Pages: 1-4.
- [3] Patent 6130632A US, Int. Cl. H03M 1/10, DIGITALLY SELF-CALIBRATING CURRENTMODE DIA CONVERTER, Ion E. Opris, Cupertino, Calif., Appl. No.: 09/062,157, Filed: Apr. 16, 1998, Date of Patent: Oct. 10, 2000, 11 p.
- [4] Patent 7605741B2 US, (51) Int. Cl. H03M 1/34 (2006.01), DIGITALLY CORRECTED SAR CONVERTER INCLUDING A CORRECTION DAC, Christopher Peter Hurrell, Cookham (GB), Appl. No.: 11/634,053, Filed: Dec. 5, 2006, Date of Patent: Oct. 20, 2009, 8 p.
- [5] Шляндін В. М. Цифровые измерительные преобразователи и приборы / Шляндін В. М. – М. : Высшая школа, 1973. – 280 с.
- [6] Шляндін В. М. Цифровые измерительные устройства : учебник для вузов / Шляндін В. М. – 2-е изд., перераб. и доп. – М. : Высшая школа, 1981. – 335 с.
- [7] Гнатек Ю.Р. Справочник по цифроаналоговым и аналогоцифровым преобразователям: Пер. с англ./ Под ред. Ю.А.Рюжина. - М.: Радио и связь, 1982. - 552 с., ил.
- [8] Стахов А. П. Алгоритмическая теория измерения / Стахов А. П. – М. : Знание, 1979. – 64 с.
- [9] Стахов А. П. Коды золотой пропорции / Стахов А. П. – М. : Радио и связь, 1984. – 152 с.

- [10] Азаров О.Д. Аналого-цифрове порозрядне перетворення на основі надлишкових систем числення з ваговою надлишковістю: монографія / Азаров О. Д.. – Вінниця : ВНТУ, 2010. – 231 с.
- [11] Азаров, О. Д. Багаторозрядні АЦП і ЦАП із ваговою надлишковістю, стійкі до параметричних відмов : монографія / О. Д. Азаров, О. В. Кадук. – Вінниця : ВНТУ, 2010. – 150 с.

References

- [1] Mosley J.D. Self-calibrating 16-bit A/D converter quarantees no missing codes to 50 kHz // EDN. – 1987. – vol. 32, № 2.
- [2] Hokazono, K., Kanemoto, D., Pokharel, R., Tomar, A., Kanaya, H., Yoshida, K.. "A low-glitch and small-logic-area Fibonacci Series DAC". IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), 2011. Pages: 1-4.
- [3] Patent 6130632A US, Int. Cl. H03M 1/10, DIGITALLY SELF-CALIBRATING CURRENTMODE DIA CONVERTER, Ion E. Opris, Cupertino, Calif., Appl. No.: 09/062,157, Filed: Apr. 16, 1998, Date of Patent: Oct. 10, 2000, 11 p.
- [4] Patent 7605741B2 US, (51) Int. Cl. H03M I/34 (2006.01), DIGITALLY CORRECTED SAR CONVERTER INCLUDING A CORRECTION DAC, Christopher Peter Hurrell, Cookham (GB), Appl. No.: 11/634,053, Filed: Dec. 5, 2006, Date of Patent: Oct. 20, 2009, 8 p.
- [5] SHlyandin V. M. Sifrovie izmeritelnie preobrazovateli i pribori / SHlyandin V. M. – M. : Visshaya shkola, 1973. – 280 s.
- [6] SHlyandin V. M. Sifrovie izmeritelnie ustroystva : uchebnik dlya vuzov / SHlyandin V. M. – 2-e izd., pererab. i dop. – M. : Visshaya shkola, 1981. – 335 s.
- [7] Gnatek YU.R. Spravochnik po cifroanalogovim i analogosifrovim preobrazovatelyam: Per. s angl./Pod red. YU.A.Ryuzina. - M.: Radio i svyaz, 1982. - 552 s., il.
- [8] Stahov A. P. Algoritmicheskaya teoriya izmereniya / Stahov A. P. – M. : Znanie, 1979. – 64 s.
- [9] Stahov A. P. Kodi zolotoy proporcii / Stahov A. P. – M. : Radio i svyaz, 1984. – 152 s.
- [10] Azarov O.D. Analogo-sifrove porozryadne peretvorenniya na osnovi nadlishkovih sistem chislennya z vagovoyu nadlishkovistyuu: monografiya / Azarov O. D.. – Vinnisya : VNTU, 2010. – 231 s.
- [11] Azarov, O. D. Bagatorozryadni ASP i SAP iz vagovoyu nadlishkovistyuu, stiyki do parametrichnih vidmov : monografiya / O. D. Azarov, O. V. Kaduk. – Vinnisya : VNTU, 2010. – 150 s.

Відомості про авторів

Азаров Олексій Дмитрович – д.т.н., професор, декан факультету інформаційних технологій та комп'ютерної інженерії Вінницького національного технічного університету, (0432) 59-83-64.

Обертюх Максим Романович – аспірант кафедри обчислювальної техніки Вінницького національного технічного університету

Кирилашук Світлана Анатоліївна – к.п.н, доцент, заступник декана факультета інформаційних технологій та комп'ютерної інженерії Вінницького національного технічного університету з навчально-методичної роботи, (0432) 59-84-85.

А. Д. Азаров, М. Р. Обертюх, С. А. Кирилашук

РЕЗИСТИВНЫЕ МАТРИЧНЫЕ ДЕЛИТЕЛИ ТОКА ДЛЯ МНОГОРАЗРЯДНЫХ ЦАП С ВЕСОВОЙ ИЗБЫТОЧНОСТЬЮ

Винницкий национальный технический университет, г. Винница

O. Azarov, M. Obertyukh, S. Kirilashchuk

RESISTIVE MATRIX DIVIDERS OF CURRENT FOR MULTIDIGIT DACS WITH WEIGHT REDUNDANCY

Vinnitsia National Technical University, Vinnitsa

УДК 004.9

Б. П. Русин¹, Л. В. Погрелюк¹, В. А. Висоцька², М. М. Осипов²,
Я. Ю. Варецький¹, О. В. Капшій¹

АРХІТЕКТУРА СИСТЕМИ ДЕДУБЛІКАЦІЇ ТА РОЗПОДІЛУ ДАНИХ У ХМАРНИХ СХОВИЩАХ ПІД ЧАС РЕЗЕРВНОГО КОПІЮВАННЯ

¹Фізико-математичний інститут імені Г.В. Карпенка НАН України, Львів
²НУ «Львівська політехніка», кафедра «Інформаційні системи та мережі», Львів

Анотація. Розроблена та детально описана концептуальна модель системи. Розроблена інтелектуальна система дедуплікації та розподілу даних у хмарному сховищі, описано опис програмного забезпечення, розглядаються етапи роботи користувача. Було проведено тестування роботи спроектованої системи. Описано кілька контрольних зразків, проаналізовано результати. Метою системи є дедублікації та розподілу даних у хмарних сховищах таким чином, щоб в кінцевому результаті резервного копіювання даних усунути повторюючі частини даних використовуючи потужності розподіленого обчислення та хмарних сховищ. Підбравши правильний підхід до розподілення завдань і даних під час дедублікації, можливо використати весь потенціал хмарних розподілених систем для збільшення швидкості резервного копіювання та його пропускну здатності. Проаналізовано (наведено недоліки та переваги використання різних підходів) та обрані ефективні методи вирішення задач: гібридну дедублікацію на рівні блоків, розбиття потоку даних на основі цифрового відбитку Рабіна, розподіл даних на основі хеш значень блоків дедублікації та використання розподіленого індексу. Дедублікація на рівні блоків передбачає два типи розбиття потоків даних на блоки, це розбиття з фіксованою довжиною та змінною на основі алгоритму. Розбиття з фіксованою довжиною досить тривіальне і швидко відносно складності алгоритму, проте недоліком є зміщення даних на початку потоку, оскільки блоки які будуть слідувати після змін будуть вважатися як нові. Проте у випадку з розбиттям блоків зі змінною довжиною, точку власне розбиття визначає алгоритм. Даний алгоритм повинен працювати з безкінечними потоками даних використовуючи кільцеву хеш функцію. Алгоритм поглинає кожен вхідний байт даних з потоку, і як тільки значення кільцевої хеш функції відповідає заданому раніше шаблону, це і слугує точкою розбиття потоку на блоки. Таким чином, при зміні або зміщенні даних на пару байтів, новим буде вважатись лише той блок даних який охоплює дані. Проте для того щоб відслідковувати зміни і правильно виставляти точки розбиття необхідно перевіряти вхідні дані на певний заданий цифровий шаблон – хеш значення. Поширеною практикою є обчислення хеш значення кожен раз на отримання вхідного байту в потоці даних. Точкою розбиття стане той момент, коли отримано хеш значення відповідатиме заданому шаблону. Щоб робити такі обчислення ефективно, було придумано алгоритм кільцевого хешу. Один із найпоширеніших алгоритмів кільцевого хешу є цифровий відбиток Рабіна. В ході аналізу засобів вирішення задач було обрано мову програмування Rust для написання клієнтської частини, мову програмування Scala для серверної частини, інструментарій Akka для менеджменту розподілених обчислень та Amazon S3 в якості хмарного сховища.

Ключові слова: дедублікація даних, розподіл даних, хмарне середовище, cloud computing, алгоритм Рабіна, хешування даних, гібридна дедублікація.

Аннотация. Разработана и подробно описана концептуальная модель системы. Разработана интеллектуальная система дедубликации и распределения данных в облачном хранилище, описано описание программного обеспечения, рассматриваются этапы работы пользователя. Было проведено тестирование работы спроектированной системы. Описаны несколько контрольных образцов, проанализированы результаты. Целью системы является дедубликации и распределения данных в облачных хранилищах таким образом, чтобы в конечном итоге резервного копирования данных усунуть повторяющиеся части данных, используя мощности распределенного вычисления и облачных хранилищ. Подобрал правильный подход к распределению задач и данных при дедубликации, возможно использовать весь потенциал облачных распределенных систем для увеличения скорости резервного копирования и его пропускной способности. Проанализированы (приведены недостатки и преимущества использования различных подходов) и выбранные методы решения задач: гибридную дедубликацию на уровне блоков, разбиение потока данных на основе цифрового отпечатка Рабина, распределение данных на основе хэш значений блоков дедубликации и использования распределенного индекса. Дедубликация на уровне блоков предусматривает два типа разбиения потоков данных на блоки, это разбиение с фиксированной длиной и переменной на основе алгоритма. Разбивка с фиксированной длиной достаточно тривиальное и быстрое относительно сложности алгоритма, однако недостатком является смещение данных в начале потока, поскольку блоки, которые будут следовать после изменений, будут считаться как новые. Однако в случае с разбивкой блоков с переменной длиной, точку собственно разбиения определяет алгоритм. Данный алгоритм должен работать с бесконечными потоками данных, используя кольцевую хэш-функцию. Алгоритм поглощает каждый входной байт данных из потока, и как только значение кольцевой хэш-функции соответствует заданному ранее шаблону, это и служит точкой разбиения потока на блоки. Таким образом, при изменении или смещении данных на пару байтов, новым будет считаться только тот блок данных, который охватывает данные. Однако для того чтобы отслеживать изменения и правильно выставлять точки разбиения необходимо проверять входные данные на определенный заданный цифровой шаблон - хэш значения. Распространенной практикой является вычисление хэш значения каждый раз на полу-ния входного байта в потоке данных. Точкой разбиения станет тот момент, когда полученное хэш значение будет соответствовать заданному шаблону. Для таких вычислений эффективно использовать алгоритм кольцевого хеша. Один из самых распространенных алгоритмов кольцевого хеша является цифровой отпечаток Рабина. В ходе анализа средств решения задач был выбран язык программирования Rust для написания клиентской части, язык программирования Scala для серверной части, инструментарий Akka для менеджмента распределенных вычислений и Amazon S3 в качестве облачного хранилища.

Ключевые слова: дедубликация данных, распределение данных, облачную среду, cloud computing, алгоритм Рабина, хеширование данных, гибридная дедубликация.

Abstract. The conceptual model of the system is developed and described in detail. An intelligent system of deduplication and distribution of data in the cloud storage is developed, the description of the software is described, the stages of the user's work are considered. Testing of the projected system was carried out. Several control samples are described and results are analyzed. The purpose of the system is to deduplicate and distribute data in cloud repositories in such a way that the end result of the backup is to eliminate duplicate pieces of data using

Б. П. Русин, Л. В. Погрелюк, В. А. Висоцька, М. М. Осипов, Я. Ю. Варецький, О. В. Капшій, 2019

distributed computing and cloud repositories. By picking the right approach to distribute tasks and data during deduplication, you can harness the full potential of cloud-based distributed systems to increase backup speed and bandwidth. Analyzes (disadvantages and advantages of using different approaches) are analyzed and effective methods of solution are selected: hybrid block-level deduplication, splitting of data flow on the basis of Rabin's digital imprint, distribution of data based on hash values of blocks of deduplication and use of distributed index. Block-level deduplication involves two types of data flow splitting into blocks, a fixed-length, algorithm-based split. Fixed-length partitioning is rather trivial and fast with respect to the complexity of the algorithm, but the downside is that data is displaced at the beginning of the stream, since the blocks that will follow after the changes will be considered new. However, in the case of partitioning of blocks of variable length, the point of proper partitioning is determined by the algorithm. This algorithm should work with infinite data flows using the ring hash function. The algorithm absorbs each input byte of data from the stream, and as soon as the value of the annular hash function corresponds to the previously specified template, it also serves as a point of splitting the stream into blocks. Thus, if the data is changed or displaced by a couple of bytes, only the data block that covers the data will be considered new. However, in order to track changes and correct set breakpoints, it is necessary to check the input data for a specific preset digital pattern - a hash value. It is a common practice to calculate a hash value every time an input byte is received in a data stream. The point of partition will be the moment when the resulting hash value matches the specified pattern. To do these calculations effectively, an algorithm has been devised for the ring hash. One of the most common ring hash algorithms is a digital Rabin imprint. During the analysis of the solutions, the Rust programming language for client-side writing, the Scala programming language for the server-side, the Akka distributed computing management tool, and Amazon S3 as the cloud repository were selected.

Key words: data deduplication, data sharing, cloud environment, cloud computing, Rabin algorithm, data hashing, hybrid deduplication.

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-40-63>.

Вступ

Для стабільної роботи компаній які працюють з інформацією є процес резервного копіювання даних. Не залежно від того, чи це банк, чи сайт по продажам господарських товарів, у разі несистемного збою та пошкодження основних серверів або баз даних – в найкоротший термін, система помилка повинна буде виправлена а резервні дані відновлені. Проте періодичне резервне копіювання викликає проблему з повторюваними даними, тим самим створюючи проблему для відділів фінансів в зв'язку з великими витратами на послуги постачальників хмарних сховищ або власних дата-центрів. Для економії дискового простору або коштів на послуги хмарних сховищ, багато систем резервного копіювання використовують дедублікацію даних [1-2]. Даний підхід дозволяє зменшити кінцевий розмір резервних копій, шляхом відсічення дублюючих даних або тих що уже відомі системі (були раніше збережені). Проте дані системи використовують хмарні технології лише в якості кінцевих точок сховищ. Підібравши правильний підхід до розподілення завдань і даних під час дедублікації, можливо використати весь потенціал хмарних розподілених систем для збільшення швидкості резервного копіювання та його пропускної здатності. З огляду на те, актуальною задачею є дослідження та подальша розробка інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах.

Актуальність

На основі того де відбувається розбиття потоку даних на блоки, обчислення їх хешів, побудова індексу та зберігання унікальних блоків, можна виділити такі типи дедублікації: вихідна, цільова, потокова та гібридна.

Вихідна дедублікація (англ. *source deduplication*) – це дедублікація яка виконується на пристрої, де знаходяться вихідні дані [3]. Будь-які дані, які помічені для резервного копіювання, будуть поділені на блоки, для яких порашовані хеші. На основі отриманих хешів, буде побудований індекс для подальшої дедублікації даних. Проте у даного підходу є два потенційні недоліки.

Перший потенційний недолік полягає в тому, що для процесу дедублікації використовуються ресурси девайсу, на якому знаходяться дані. Таким чином, користувачі повинні переконатися, що у них є достатньо вільної оперативної пам'яті та обчислювальної потужності процесора. А отже, резервне копіювання, наприклад поштового сервера або інтернет веб-сайту може призвести, щонайменше, до уповільнення їх роботи.

Другий недолік стосується розташування хеш-таблиці, тобто індексу дедублікації. У деяких випадках хеш-таблиця обмежується кожним окремим комп'ютером, який виконує дедублікацію. В такому випадку, індекс буде обмеженим лише локальними даними. Тобто переваги дедублікації будуть обмежені лише в межах одного комп'ютер. Якщо на інших серверах існують ідентичні фрагменти даних, поточний сервер не знатиме цього.

Цільова дедублікація (англ. *target deduplication*) – це дедублікація яка виконується на цільовому центральному сервері, де на кінцевому етапі будуть зберігатись дані [3]. Як тільки дані прибудуть до репозиторія, сервер отримає можливість побудувати індекс на їх основі та зробити перетин з існуючою, центральною хеш-таблицею, щоб зберегти лише унікальні дані. Перевагою даного підходу є великий об'єм даних. А оскільки чим більше даних, тим більший індекс, що в результаті дає більше шансів знайти ідентичні блоки, збільшуючи коефіцієнт дедублікації. Також вагомою перевагою є перенесення відповідальності за обчислювальні потужності від клієнта (пристрою з вихідними даними) на цільовий сервер, який розрахований лише на дедублікацію, унеможливаючи потенційну боротьбу за обчислювальні ресурси. Проте даний варіант не вирішує всі проблеми, є певні моменти які необхідно розглянути

і взяти до уваги. Цільова дедублікація вимагає щоб дані були збережені на диск перед самим процесом розбивки і хешування, щоб не перевантажувати клієнтів. А отже слабким місцем даного підходу є швидкість запису/читання з диску.

Другим потенційним недоліком є можливість колізій хеш-функцій, тобто можливість того, що два блоки даних які не є ідентичні, отримають однакове хеш значення. Хеш колізії можуть призвести до втрати консистентності даних, що у кінцевому випадку пошкодить оригінальні дані без можливості їх відновлення. Щоб уникнути колізії необхідно використовувати алгоритми, в яких шанс їх утримання менший, проте це призводить до більшого навантаження на обчислювальні ресурси. Проте, використання більш сильних та складних алгоритмів не є проблемою, оскільки сервери дедублікації використовують спеціалізоване апаратне обладнання під необхідні задачі.

Третім недоліком є те що повний об'єм даних для резервного копіювання повинен бути переданий через мережу від клієнтів до центрального сховища. Дана проблема є особливо актуальна якщо у клієнта наявне слабе або високо тарифіковане мережеве покриття. При незначних змінах після останнього резервного копіювання, надсилання абсолютно всіх даних є надлишковим. Також, проблемою може виникнути завантаженість мережі цільового сервера, що негативно вплине на кінцеву пропускну здатність.

Потокова дедублікація (англ. *inline deduplication*) використовує оперативну пам'ять в якості буферної зони для процесу розбивки даних і обрахування їх хешів [4]. А отже дані дедублюються перед записом на диск, це виключає витрати на швидкість запису/читання з диску. Потокову дедублікацію варто розглядати, як кращу версію цільової дедублікації. Перша, має всі переваги останньої, проте жодного недоліку пов'язаного з повільною роботою диску.

Гібридна дедублікація – це підхід при поєднанні вихідної та цільової (потокової) дедублікації[4]. Розбиття даних на блоки і обрахунок їх хешів відбувається на клієнті (пристроїв з вихідними даними). Отримані хеші надсилаються на сервер для їх ідентифікації, після чого, клієнт надсилає серверу лише ті блоки даних, які раніше йому не відомі.

Даний підхід нівелює недолік цільової та потокової дедублікації щодо використання інтернет мережі. Оскільки з клієнта буде надіслано лише ті блоки даних які раніше невідомі, а у випадку уже відомих, надсилаються лише їх хеші, які 2-3 тис. разів менші за оригінальні дані.

Хоча клієнт, у випадку з гібридною дедублікацією, все ще бере участь у розбивці даних на блоки та обчислення їх хешів, як і у випадку з вихідною дедублікацією. Проте робота з індексом лежить повністю на цільовому репозиторії, що значно розвантажує обчислювальні потужності клієнта. Якщо, даний момент все ще є недоліком для того чи іншого випадку, то розбивку даних на блоки та обчислення їх хешів можливо перенести на інший, транзитний пристрій який знаходиться всередині мережі клієнта або й повністю делегувати цю роботу до сервера. Дедублікація на рівні блоків передбачає два типи розбиття потоків даних на блоки, це розбиття з фіксованою довжиною та змінною на основі алгоритму [4]. Розбиття з фіксованою довжиною досить тривіальне і швидке відносно складності алгоритму, проте недоліком є зміщення даних на початку потоку, оскільки блоки які будуть слідувати після змін будуть вважатися як нові. Проте у випадку з розбиттям блоків зі змінною довжиною, точку власне розбиття визначає алгоритм. Даний алгоритм повинен працювати з безкінечними потоками даних використовуючи кільцеву хеш функцію. Алгоритм поглинає кожен вхідний байт даних з потоку, і як тільки значення кільцевої хеш функції відповідає заданому раніше шаблону, це і слугує точкою розбиття потоку на блоки. Таким чином, при зміні або зміщенні даних на пару байтів, новим буде вважатися лише той блок даних який охоплює дані. Проте для того щоб відслідковувати зміни і правильно виставляти точки розбиття необхідно перевіряти вхідні дані на певний заданий цифровий шаблон – хеш значення. Поширеною практикою є обчислення хеш значення кожен раз на отримання вхідного байту в потоці даних. Точкою розбиття стане той момент, коли отримане хеш значення відповідатиме заданому шаблону. Щоб робити такі обчислення ефективно, було придумано алгоритм кільцевого хешу [4-5]. Один із найпоширеніших алгоритмів кільцевого хешу є цифровий відбиток Рабіна [4].

Мета

Метою роботи є розробити загальну архітектуру системи дедублікації та розподілу даних у хмарних сховищах під час резервного копіювання.

Задачі

1. Здійснити аналіз сучасних технологій проектування та створення систем у сфері резервного копіювання та відновлення даних, здійснити порівняльний аналіз технічних реалізацій та алгоритмів, застосованих в реалізаціях.
2. Побудувати концептуальну модель інтелектуальної інформаційної системи дедублікації та розподілу даних у хмарних сховищах, задля автоматизації та ефективної роботи програмного забезпечення у сфері резервного копіювання та відновлення даних.

3. Провести аналіз методів та засобів реалізації інтелектуальної системи та зробити вибір для реалізації власної.

4. Спроекувати інтелектуальну систему дедублікації та розподілу даних у хмарних сховищах.

Розв'язання задач

Проектування інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах це складний процес, що потребує детального опису та ставить перед дослідниками множину питань. Оскільки об'єктом дослідження є процес дедублікації та розподілу даних у хмарних сховищах, важливим є на даному етапі провести системний аналіз обраної предметної області, для того, щоб розуміти повну картину роботи системи, процеси перебігу даних в цій системі, визначити головні можливості системи та основні бізнес процеси для того, щоб на етапі реалізації уникнути багатьох помилок в архітектурному плані та виборі моделей вирішення задачі дедублікації та розподілу даних у хмарних сховищах. Важливим є аналіз мети системи і розуміння шляхів її досягнення, застосувавши наявні ресурси - технічні засоби, певні методи та підходи, архітектури для побудови інформаційної системи. Системний аналіз дозволить розділити поставлену перед нами задачу на сукупність простих задач, враховуючи взаємозв'язки простіших задач між собою [6-10]. Таким чином складна система розбивається на складові та з чітко формуються бізнес процеси між частинами продукту. Конкретизація принципів системного підходу до аналізу інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах дає краще уявлення та бачення системи загалом, розуміння її функціонування, необхідних кроків для вдосконалення та певних особливостей системи. Метою даного дослідження є розробка програмного забезпечення, що матиме змогу проводити дедублікацію та розподілу даних у хмарних сховищах. Перш за все, спроектовану систему необхідно розглядати як множину більш простих елементів - її складових, які є окремими модулями між якими відбувається постійна комунікація, тобто елементи є взаємопов'язані. Потрібно також розуміти і те, що як і всередині системи відбуваються певні процеси, так і зовні на систему впливають фактори, які також вносять свою частину у функціонування системи - за допомогою цих факторів також відбувається перетворення потоків даних. Тож, можемо визначити основні принципи системного підходу при побудові інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах [11]:

- принципом кінцевої мети є дедублікації та розподілу даних у хмарних сховищах;
- принцип єдності полягає у тому, що система розглядається як сукупність певних складових, але з іншого боку - спроектоване програмне забезпечення є цілісною системою, оскільки функціональні процеси, якими вона володіє, логічно впливають з елементів, що формують систему, проте жоден елемент окремо не володіє цими функціями;
- принципом зв'язності підтверджується функціонуванням системи в певному середовищі - очевидно, що інтелектуальна система дедублікації та розподілу даних у хмарних сховищах буде корисною у використанні багатьма компаніями в задачі резервного копіювання даних, відповідно, необхідно передбачити механізм гнучкого налаштування для різних підходів;
- принцип модульної побудови визначається модульною структурою проекту, тобто повноцінна робоча система дедублікації та розподілу даних у хмарних сховищах поділяється на певну кількість модулів, між якими відбувається комунікація (модулі можуть бути підключеними або відключеними в залежності від потреб клієнтів);
- принцип функціональності визначає, що проектування структури системи можливо лише після визначення та повного розуміння роботи функцій системи, відповідно для початку маємо точно сформулювати та детально оглянути можливості системи дедублікації та розподілу даних у хмарних сховищах;
- принцип розвитку означає, що система має бути спроектована таким чином, щоб пізніше мати можливість розвиватись за рахунок розширення, підключення додаткових модулів, заміною імплементації інтерфейсів, накопичення даних в базах та можливістю масштабування;
- принцип децентралізації - це правильне співвідношення між централізацією та децентралізацією системи, що визначається метою та призначенням системи;
- невизначеності - ймовірність певних непередбачуваних факторів, що можуть вплинути на систему, щось, що може відбутись, а може і не відбутись.

Головним завданням системного аналізу є побудова моделі (системи), що дозволить розв'язати поставлену задачу. Кінцева система повинна складатись з взаємопов'язаними між собою елементами, що в сукупності приводять до вирішення конкретної проблеми. Базу системного аналізу складають система, мета, елемент, функція, оточення та інші. Під поняттям система розуміється сукупність окремих елементів між якими відбувається певна взаємодія, що веде до досягнення певної спільної мети, а також ця множина елементів утворюють цілісну систему, що взаємодіє з навколишнім середовищем як одне ціле [12-17].

Зовнішньою сутністю інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах, що функціонує у вигляді програми резервного копіювання даних є адміністратор, що проводить резервне копіювання даних підприємства, або користувач, що проводить резервне копіювання персональних даних. Елементами системи, що аналізується є: користувацький інтерфейс, вхідна точка для користувачів системи, надає змогу вибирати файли для резервного копіювання, переглядати існуючі копії, управляти їхнім життєвим циклом, розподілом та планувати завдання для періодичного резервного копіювання даних, а також відновлювати дані з резервних копій; програма-клієнт, яка отримує дані від користувацького інтерфейсу, проводить розподіл потоку даних на блоки та обчислення їх значень хеш-функції, тестування хеш значень блоків з індексом системи та надсилання нових блоків даних до хмарних сховищ; індекс - хеш таблиця яка містить інформацію про всі відомі унікальні блоки даних системи, відповідає за усунення дублікатів на момент збереження їх до хмарних сховищ; програма-сервер - виконує алгоритм дедублікації оперуючи блоками даних програми-клієнта та індексом, проводить розподіл завдань дедублікації між робітниками (окремими вузлами в кластерному середовищі) та комунікацію з хмарними середовищами для зберігання даних.

Метою системи є дедублікації та розподілу даних у хмарних сховищах таким чином, щоб в кінцевому результаті резервного копіювання даних усунути повторюючі частини даних використовуючи потужності розподіленого обчислення та хмарних сховищ. Важливим буде зрозуміти та визначити середовище функціонування системи, що означатиме вибір такої множини об'єктів, на яку мала би вплив розроблювана система і яка б змінювалась під дією цього впливу, а також такі, чия зміна впливала б на проєктовану систему [18-21]. Взаємодія інформаційної системи з зовнішнім середовищем відбувається завдяки наявним входам та виходам системи. Вплив зовнішнього середовища на проєктовану систему є, власне, входом, а результат цього впливу є виходом. Зазвичай прийнято розглядати лише найважливіші зв'язки системи з зовнішнім середовищем, в даному випадку це буде кінцевий користувач, що приймає участь в процесі резервного копіювання даних та певне підприємство, що використовуватиме розроблюваний програмний продукт та забезпечуватиме вхідні дані для проєктованої системи.

Відомо, що в методології системного аналізу системи діляться на два типи: складні та прості. Системи, що складаються з невеликої кількості елементів та, відповідно, зав'язків між ними, не мають розгалуженої структури називаються простими системами. Складними ж є ті, що побудовані з значно більшої кількості елементів, відповідно мають більшу кількість внутрішніх зав'язків, виконують складні функції та є неоднорідними [22-27]. Очевидно, що програмний продукт, що проєктується, є представником складних систем. Проєктована інтелектуальна система дедублікації та розподілу даних у хмарних сховищах складається з великої кількості елементів, матиме потужні канали комунікації між ними, ефективність роботи кожного з елементів окремо буде значно нижчою ніж ефективність використання їх сукупності у вигляді цілої системи.

Діаграма варіантів використання (англ. *use case diagram*) або діаграма прецедентів - це діаграма UML, на якій зображено відношення між акторами (зовнішні сутності) та прецедентами (варіанти використання) в системі [28-32]. Діаграма є графом, що складається з множини акторів, прецедентів обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. На діаграмі варіантів використання (рис. 1) також зображено наступні прецеденти (варіанти використання):

- *Надіслати дані* – користувач формує список файлів і надсилає їх до системи для резервного копіювання.
- *Розбити дані на блок* – система формує потік з вхідних даних і розбиває їх на маленькі блоки даних для комфортної дедублікації.
- *Обчислити хеші блоків* – система для кожного вхідного блоку обчислює хеш значення використовуючи задані хеш-функції.
- *Усунути дублікати блоків* – використовуючи *Індекс*, система перевіряє існуючість блоків за допомогою отриманих хеш значень.
- *Розподілити блоки між робітниками* – для збільшення пропускну здатності та швидкодії, система розподіляє роботу на блоками між робітниками в системі (наприклад вузли в кластері).
- *Зберегти унікальні блоки* – після усунення дублюючих блоків, система, використовуючи *Хмарні сховища*, зберігає унікальні блоки в сховищах.
- *Стиснути блоки* – для економії дискового простору, вхідні блоки можуть бути стиснуті один із алгоритмів компресії.
- *Зашифрувати блоки* – для додаткової захищеності даних, вхідні блоки можуть бути також зашифровані.
- *Сформувати резервну копію* – після закінчення обробки вхідних блоків, вихідний потік даних зберігається як послідовність посилань на хеш значення.

- *Отримати резервну копію* – після закінчення процесу дедублікації, користувач отримує посилання на резервну копію для майбутнього відновлення даних
- *Відновити дані* – використовуючи посилання на резервні копії, користувач може відновити оригінальні дані.

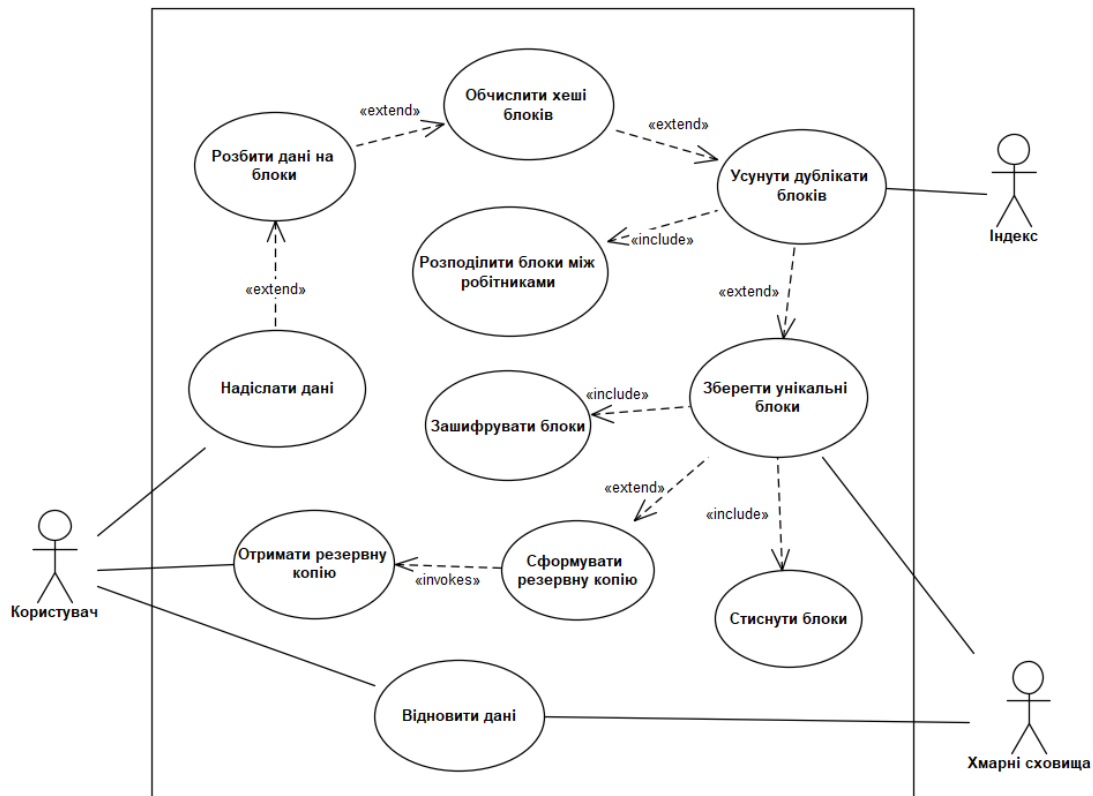


Рисунок 1 – Діаграма варіантів використання системи

Діаграма станів – це UML діаграма, що визначає зміну станів об'єкта у часі [15]. Діаграма діяльності описує як процеси виконання певних робіт, виконуваних або системою або зовнішніми сутностями координують між собою. Виконання певних робіт може бути залежним від деяких передумов, що мають бути виконані. Перехід до іншої дії відбувається після завершення виконання попередньої. Зазвичай дія на вході отримує певну інформацію - множину сигналів, що має бути опрацьована і перетворена на множину вихідних сигналів. На рис. 2 подано діаграму станів резервного копіювання даних користувачем інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах. Об'єктом діаграми станів є процес резервного копіювання, і далі в цьому підпункті ми будемо розуміти цей процес під назвою об'єкт. Стан *автентифікація користувача* є початковим для процесу резервного копіювання. Якщо дані автентифікації введено неправильно то об'єкт залишається в тому самому стані і очікує на нові команди. Якщо дані автентифікації, а саме логін та пароль, введено правильно то об'єкт переходить у стан *вибору файлів для резервного копіювання*. Наступний стан – *створення потоку даних*. З вибраних файлів для дедублікації система створює єдиний потік даних. Після успішного створення потоку даних, об'єкт переходить у стан *розбиття потоку даних на блоки*. Блок даних – це одиниця процесу дедублікації, саме дублікати блоків система старається уникати. Якщо в потоці даних є дані для розбиття на блоки, то об'єкт переходить у стан *обчислення хеш значення блоку*. Кожен блок даних повинен мати свій цифровий підпис, який набагато менший по обсягу, чим і являється хеш значення. Після того як отримано хеш значення, об'єкт переходить у стан *перевірки хеш значення на існуючість*. Для того щоб відкинути дублікати, треба перевірити чи даний блоку існує в системі, проте перевіряти весь блок цілком витратно, тому перевіряють на існуючість їх хеш значення. Якщо хеш значення уже відоме системі, то об'єкт переходить назад у стан *розбиття потоку даних на блоки*. А якщо хеш значення невідоме, то об'єкт переходить у стан *збереження блоку даних*. Новий блоку даних, раніше не відомий системі, зберігається у спеціально відведеному місці в хмарному сховищі, базуючись на його хеші (для спрощення його подальшого пошуку). Після збереження, об'єкт переходить у назад у стан *розбиття потоку даних на блоки*. Коли об'єкт знаходиться у цьому стані і більше не може отримати нових блоків з потоку даних, об'єкт переходить у

стан *побудови резервної копії*. Після процесу дедублікації, список з блоків даних представлений у вигляді списку з його хеш значень, що і являється резервною копією. Після її отримання, об'єкт переходить у стан *збереження резервної копії*. Даний стан є останнім перед входом об'єкт до кінцевого стану.

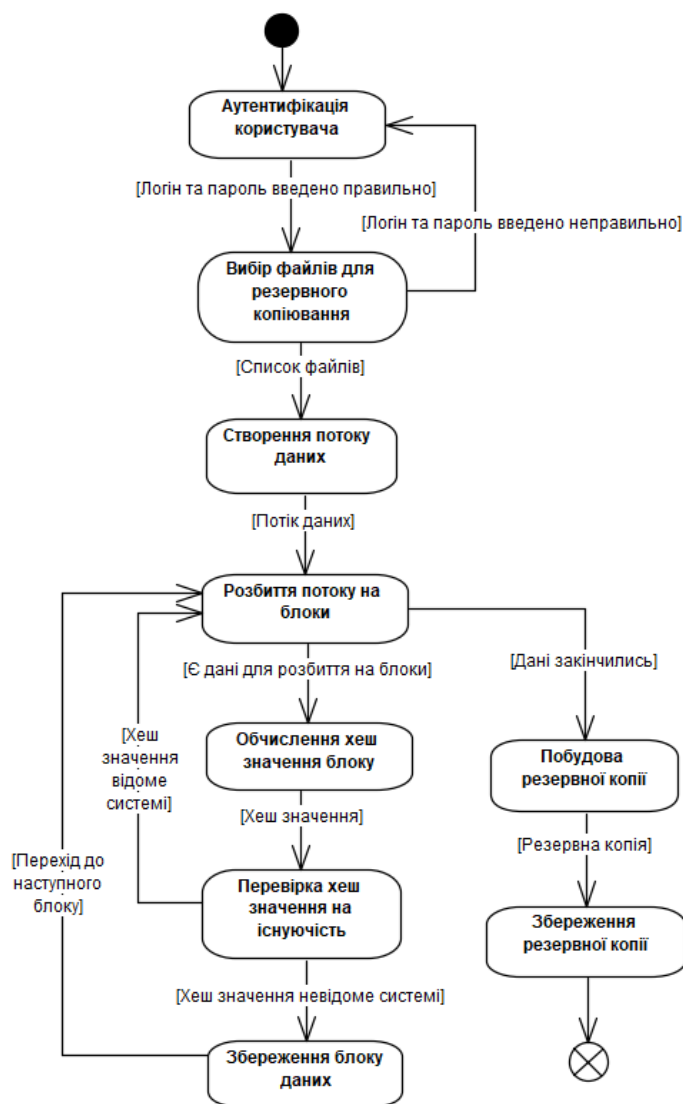


Рисунок 2 – Діаграма станів резервного копіювання

Діаграми послідовностей – це UML діаграма, яка відображає взаємодії об'єктів впорядкованих за часом [15]. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень. На діаграмі послідовностей показано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас. Надіслані повідомлення зображуються у вигляді горизонтальних ліній, в порядку відправлення. На рис. 3 подано діаграму послідовності резервного копіювання даних, використовуючи інтелектуальну систему дедублікації та розподілу даних у хмарних сховищах. На діаграмі зображено лінію життя актора *Користувач* та чотирьох об'єктів: *Програма-клієнт*, *Програма-сервер*, *Індекс* та *Хмарні сховища*.

- *Користувач* – єдиний актор в системі, обирає дані для резервного копіювання і отримує результат дедублікації.
- *Програма-клієнт* – програмне забезпечення на стороні користувача, яке відповідальне за групування користувацьких даних і формування потоку байтів на їх основі. Керує процесом дедублікації зі сторони користувача.
- *Програма-сервер* – програмне забезпечення на стороні сервера. Центральний координатор процесу дедублікації, виконує спілкування з програмою-клієнтом для отримання списку хешів та блоків; формує список необхідних, нових блоків на основі даних з індексу; вико-

нує запис нових блоків у хмарних сховищах; формує дерево посилання для відновлення вихідних даних.

- *Індекс* – хеш таблиця, містить усі відомі системі хеші та посилання на відповідний блок даних у хмарі чи іншому сховищі.
- *Хмарні сховища* – місце збереження блоків даних та резервних копій (дерева посилань).

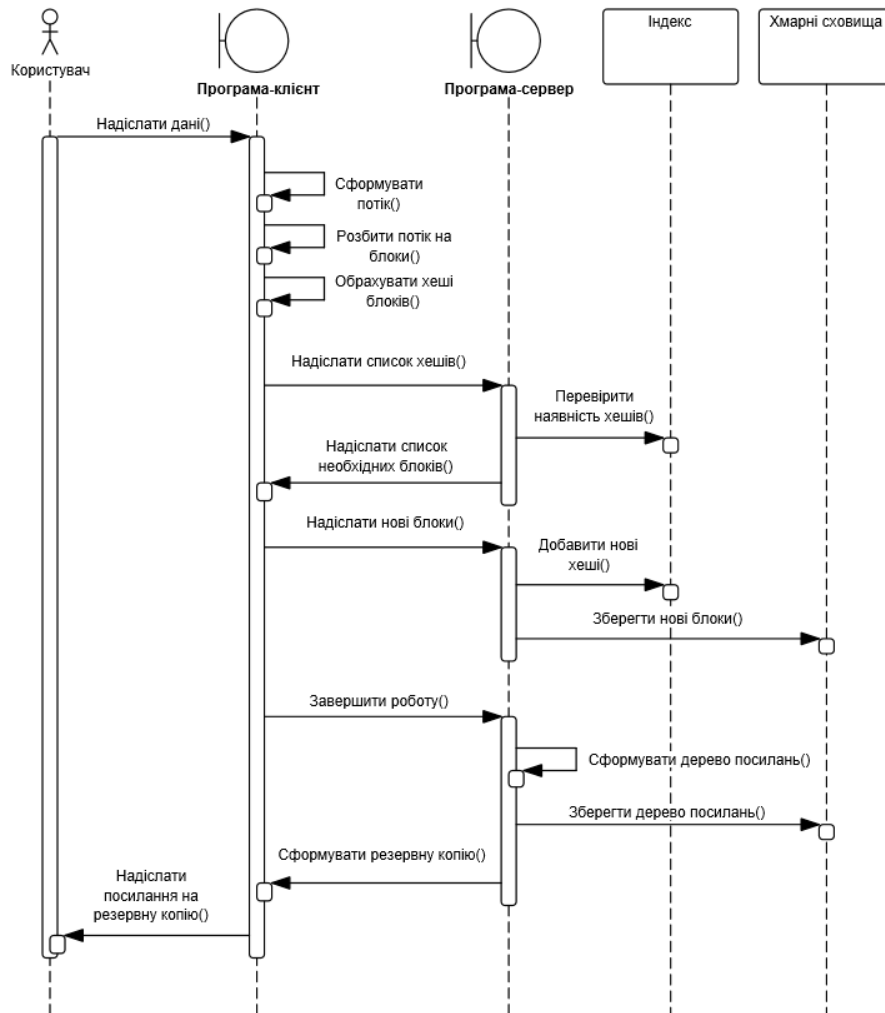


Рисунок 3 – Діаграма послідовності

Загалом робочий процес, згідно діаграми послідовності, виглядає наступним чином. *Користувач*, використовуючи користувацькій інтерфейс, у вигляді веб-сайту або настільного додатку, вибирає необхідні файли/дані для дедублікації і надсилає їх до *Програми-клієнта*. Який, в свою чергу, розбиває їх на блоки використовуючи заданий алгоритм. Для кожного блоку *Програма-клієнт* обчислює хеш значення та надсилає їх у вигляді списку до *Програми-сервера*. Останній перевіряє хеші з отриманого списку з даними з *Індексу* та відтинає ті хеші які вже відомі. В результаті назад до *Програми-клієнта* відсилаються лише ті хеші які раніше не відомі системі, а отже яких необхідно додати в систему. На основі цього списку, *Програма-клієнт* вибірково надсилає блоки до *Програми-сервера*. Кожний отримай блок зберігається в хмарних сховищах з унікальним ідентифікатором (посиланням на блок). Також *Програма-сервер* додає в хеш-таблицю *Індексу* нову пару ключ-значення: хеш блоку та посилання на нього. Даний цикл, перевірка на існуючість хеш значень та надсилання нових блоків виконується до тих пір, поки є дані в потоці. Коли потік завершиться, *Програма-клієнт* сигналізує про завершення процесу дедублікації. *Програма-сервер* на основі обробки блоків даних та їх хешів буде резервну копію у вигляді дерева посилань на хеші. Сформована копія повертається до *Користувача*.

Окрім поведінкових діаграм UML, необхідно зобразити систему використовуючи структурні діаграми [15]. Першою структурною діаграмою UML – це діаграма компонентів. На ній зображують компоненти та зв'язки між ними. Компоненти з'єднуються між собою за допомогою структурних

зв'язків (англ. *assembly connector*). Дані зв'язки встановлюються між інтерфейсами двох компонентів, при чому компоненти зазвичай мають як мінімум один інтерфейс. В такому зв'язку двох компонентів, один із них надає послуги необхідні іншому, за шаблоном клієнт-сервер. Також, компоненти можуть мати ієрархію, тобто один компонент може містити в собі інший.

На рис. 4 подано діаграму компонентів інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах, яка містить дві основні компоненти – *Client* та *Server*.

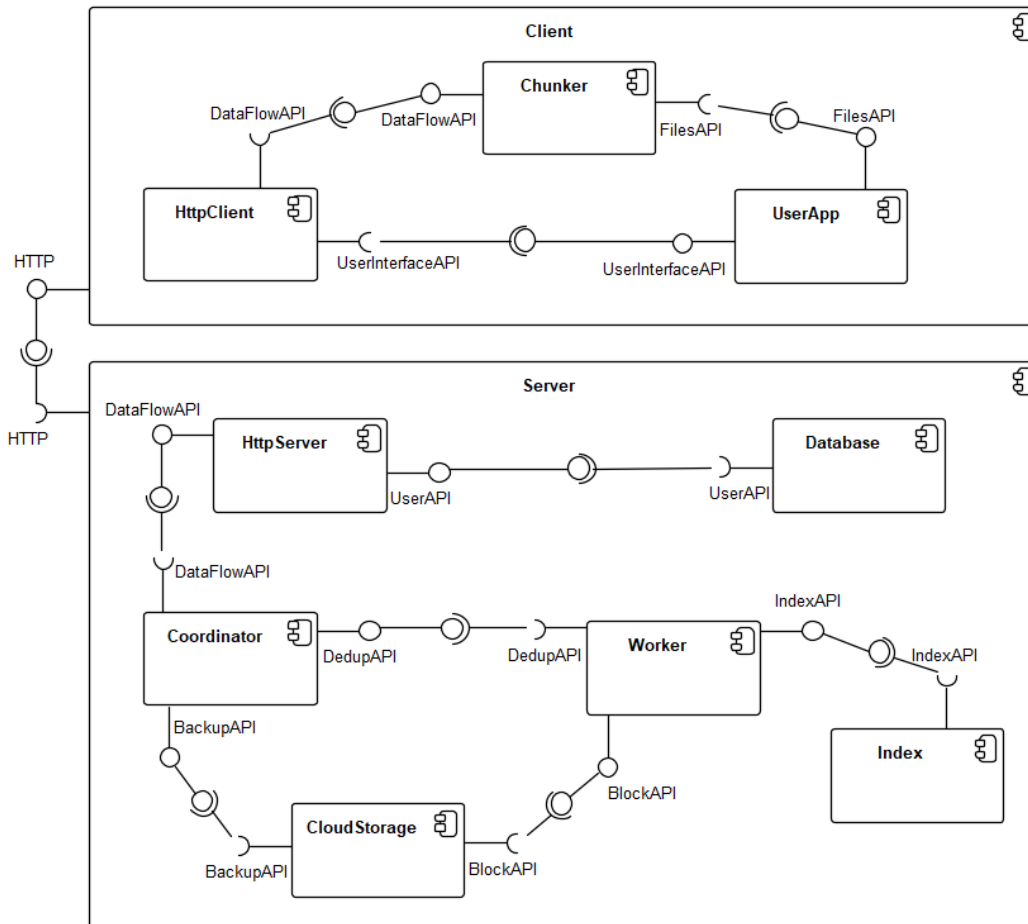


Рисунок 4 – Діаграма компонентів

Компонента *Client* містить програмні компоненти які знаходяться на стороні користувача:

- *UserApp* – компонент користувацької програми, яка надає змогу користувачу автентифікуватись в системі, вибрати файли для резервної копії, запустити сам процес резервного копіювання та відновити дані з резервної копії. Згруповані дані для резервної копії передає до компоненти *Chunker* через інтерфейс *FilesApi*, решта операцій зв'язаний зі взаємодією із сервером – через інтерфейс *UserInterfaceApi* із компонентною *HttpClient*.
- *Chunker* – компонент формування та розбиття потоку даних, який відповідає за перетворення вхідних користувацький файлів (отриманих від *UserApp* через інтерфейс *FilesApi*) у єдиний потік байтових даних, розбиття його на блоки даних та відправку цих блоків до *HttpClient* через інтерфейс *DataFlowApi*.
- *HttpClient* – компонент HTTP-клієнта, приймає дані та команди від користувацького інтерфейсу компоненти *UserApp* (через *UserInterfaceApi*), а також отримання блоків дедублікації від компоненти *Chunker* (через *DataFlowApi*). Усі отримані запити передаються на сервер через HTTP.

Компонента *Server* містить програмні компоненти які знаходяться на стороні сервера:

- *HttpServer* – компонент HTTP-сервера, який обробляє HTTP запити від компоненти *Client*. Вхідні запити пов'язані з процесом резервного копіювання передаються до компоненти *Coordinator* (через *DataFlowApi*), а ті що з процесами користувача (автентифікація користувача) до компоненти *Database* (через *UserApi*).

- *Database* – компонент бази даних, який відповідає за збереження та читання даних користувача та його резервних копій; а також відповідає за процес автентифікації, перевіряючи отримані логін та пароль зі записами в базі даних.
- *Coordinator* – компонент координатора системи, керує процесом резервного копіювання та дедублікації даних. Розподіляє вхідні блоки дедублікації між робітниками системи (компоненти *Worker*) та комунікує між ними через інтерфейс *DedupApi*. По завершенню процесу дедублікації, відповідає за формування резервних копій та збереження їх у хмарних сховищах (компонента *CloudStorage*), використовуючи *BackupApi*.
- *Worker* – компонент робітників системи, які відповідальні за процес дедублікації даних, а саме відсічення дублюючих блоків та збереження нових. Спілкується з компонентом *Index* (через інтерфейс *IndexApi*) для перевірки хеш значень блоків на існуєність у системі, а також збереження хеш значень нових блоків, які власне зберігаються у системі у хмарних сховищах, використовуючи компоненту *CloudStorage* через їх спільний інтерфейс *BlockApi*.
- *CloudStorage* – компонент хмарних сховищ, відповідає за збереження сформованих файлів з блоками даних та деревами посилань резервних копій у хмарних сховищах.
- *Index* – компонент, який відповідає за формування індексу дедублікації та перевірки хеш значень блоків на існуєність.

Діаграми класів – це структурна UML діаграма, яка описує структуру системи демонструючи класи системи, їхні атрибути, методи та зв'язок з іншими класами [15]. Діаграма класів це наріжний камінь моделювання об'єктно-орієнтованих програм. Використовується для загального концептуального моделювання структури програмного забезпечення. Класи в діаграмі класів представляють основні елементи програми, їхній взаємозв'язок в програмі та власне самі класи які повинні бути запрограмовані. У самих діаграмах класи зображені як прямокутники з трьома відсіками:

- ім'я класу, з великою букви, жирним шрифтом по середині відсіку;
- атрибути класу, з малої букви, вирівняні по лівому краю;
- методи класу, з малої букви, вирівняні по лівому краю.

Якщо класи зв'язані між собою то між ними проводять зв'язок асоціації. Якщо між двома класами визначена асоціація, то можна переміститися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарною. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів; вони називаються n-арними. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами. На рис. 5 подано діаграму класів досліджуваної інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах. Клас *User* представляє користувача системи. Містить інформацію про користувача та відповідає за створення резервних копій та відновлення даних. Атрибути та методи:

- *username*, атрибут типу *String* – унікальний ім'я користувача у інтелектуальній системі, використовується для його ідентифікації.
- *email*, атрибут типу *String* – електронна пошта користувача, використовується під час автентифікації.
- *password*, атрибут типу *String* – пароль користувача, використовується під час автентифікації.
- *auth*, метод з результуючим типом *boolean* – проводить автентифікацію користувача на основі його електронної пошти та паролю.
- *startBackup*, метод з вхідним параметром шляху до файлів для резервного копіювання – розпочинає процес дедублікації. Результатом є об'єкт типу *Stream*, який представляє потік даних для дедублікації.
- *getBackup*, метод з аналогічним вхідним параметром що і у *startBackup* – перевіряє чи дана резервна копія завершилась, і якщо так, то повертає об'єкт типу *Backup*.
- *recover*, метод зі вхідними параметрами об'єкту резервного копіювання типу *Backup* та шляху до відновлених файлів типу *String* – відновлює оригінальні дані з резервної копії у вказаних шлях на диску комп'ютера.

Клас *Backup* представляє резервну копію, яка містить інформацію про її час створення, оригінальний розмір, розмір після дедублікації та власника. Відповідає за відновлення оригінальних даних. Атрибути та методи:

- *id*, атрибут типу *String* – унікальний ідентифікатор резервної копії.
- *timestamp*, атрибут типу *DateTime* – час створення резервної копії.
- *originalSize*, атрибут типу *long* – оригінальний розмір даних у резервній копії.
- *dedupSize*, атрибут типу *long* – розмір даних резервної копії після дедублікації.

- *user*, атрибут типу *User* – власник резервної копії.
- *recover*, метод – відновлює файли з резервної копії. Результатом є список об'єктів типу *FileStream* – клас, що містить метадані про файл та потік байтів самого файлу.

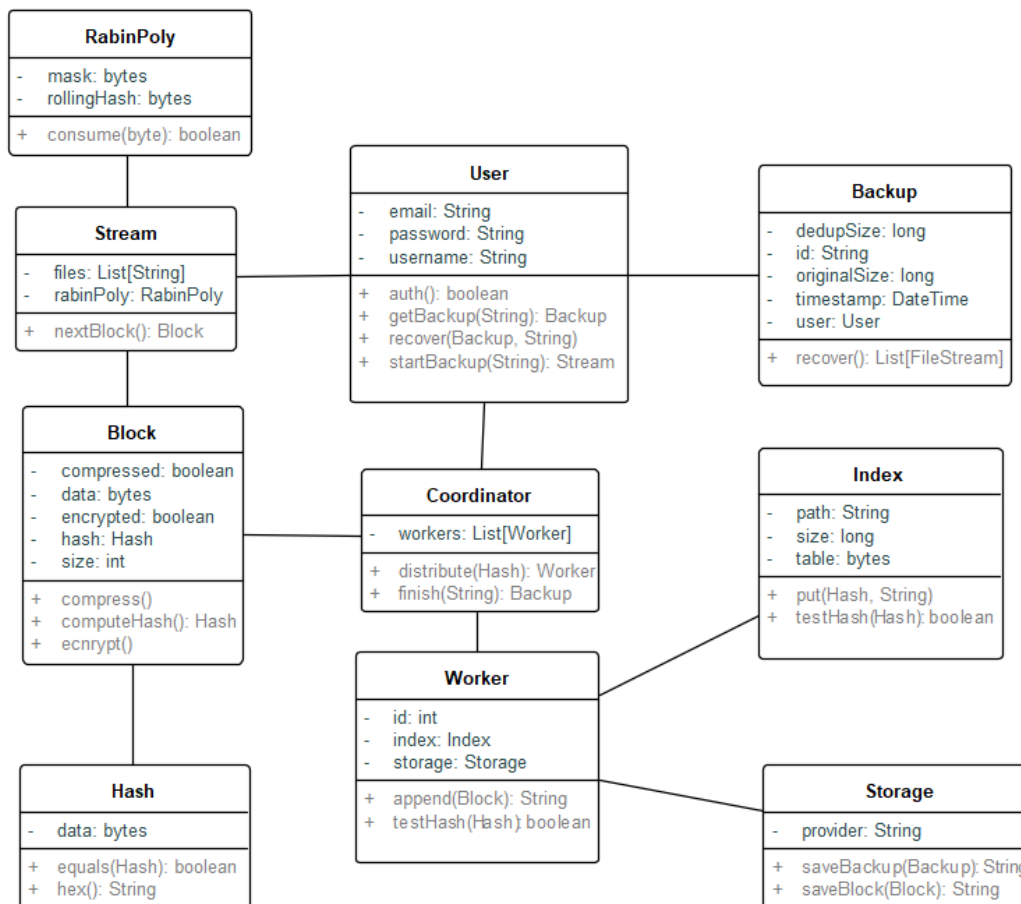


Рисунок 5 – Діаграма класів

Клас *RabinPoly* представляє компонент розбиття потоку даних на блоки використовуючи алгоритм Рабіна. Атрибути та методи:

- *mask*, атрибут типу *bytes* – встановлена маска яка використовується для логічного добутку з кільцевим хешем щоб отримати точку розбиття даних.
- *rollingHash*, атрибут типу *bytes* – кільцевий хеш, змінюється на основі кожного вхідного байту з потоку даних.
- *consume*, метод з вхідним параметром типу *byte*. Поглинає вхідні байти і обновлює кільцевий хеш на основі алгоритму Рабіна. Результат *true* або *false*, повідомляє чи потрібно розбивати потік на блок в даний момент чи продовжувати поглинання.

Клас *Stream* представляє потік даних, отриманих зі списку файлів наданих користувачем. Відповідає за утворення блоків даних для дедублікації. Атрибути та методи:

- *files*, атрибут типу *List[String]* – список файлів на основі яких будувати потік даних.
- *rabinPoly*, атрибут типу *RabinPoly* – містить об'єкт класу *RabinPoly*.
- *nextBlock*, метод з результатом типу *Block* – поступово вичитує дані з файлу викликає метод *consume* на атрибуті *rabinPoly* для кожного вхідного байту. Коли *consume* сповістить про момент розбиття, утворює об'єкт класу *Block* що і є результатом методу.

Клас *Block* представляє блок даних для дедублікації отриманих з розбиття вихідного потоку даних. Відповідає за стиснення, шифрування та обчислення хеш значення даних. Атрибути та методи:

- *data*, атрибут типу *bytes* – масив байтів даних.
- *size*, атрибут типу *int* – довжина масиву байтів оригінальних даних.
- *hash*, атрибут типу *Hash* – хеш значення блоку.

- *compressed*, атрибут типу *boolean* – вказує чи масив байтів даних стиснутий.
- *encrypted*, атрибут типу *boolean* – вказує чи масив байтів даних зашифрований.
- *computeHash*, метод з результатом типу *Hash* – обчислює хеш значення блоку даних використовуючи хеш-функції, результат зберігається в атрибуті *hash*.
- *compress*, метод – проводить стиснення даних, результат зберігається в атрибуті *data*, якщо змінено розмір блоку то новий записується в атрибуті *size*.
- *encrypt*, метод – проводить шифрування даних, результат зберігається в атрибуті *data*, якщо змінено розмір блоку то новий записується в атрибуті *size*.

Клас *Hash* представляє хеш значення отримане на основі блоку даних. Відповідає за представлення хешу у 16-ковій системі числення та порівняння з іншими хешами. Атрибути та методи:

- *data*, атрибут типу *bytes* – масив байтів який містить дані хеш значення.
- *equals*, метод вхідним параметром якого є інших хеш типу *Hash* – порівнює вхідне хеш значення із власним.
- *hex*, метод з результатом типу *String* – відображає хеш значення у 16-ковій системі числення.

Клас *Coordinator* призначений для координування та розподілення блоків між робітниками у кластері. Атрибути та методи:

- *workers*, атрибут типу *List[Worker]* – список відомих робітників у вигляді об'єктів класу *Worker*.
- *distribute*, метод вхідним параметром якого є об'єкт типу *Hash* – на основі 16-ого числа отриманого від вхідного об'єкту хешу (викликаючи метод *hex*) вирішує якому робітнику делегувати завдання. Результатом виклику методу є об'єкт типу *Worker*.
- *finish*, метод вхідним параметром якого є ідентифікатор резервного копіювання, завершує процес дедублікації та формує об'єкт резервної копії класу *Backup*.

Клас *Worker* представляє робітника системи який відповідальний за обробку вхідних блоків даних та перевірки хешів на існуєність у системі. Атрибути та методи:

- *id*, атрибут типу *int* – унікальний ідентифікатор робітника у кластері.
- *index*, атрибут типу *Index* – містить об'єкт класу *Index*.
- *storage*, атрибут типу *Storage* – містить об'єкт класу *Storage*.
- *testHash*, метод вхідним параметром якого є об'єкт класу *Hash* – перевіряє чи вхідний хеш існує в системі використовуючи атрибут *index* та викликаючи відповідний метод.
- *append*, метод вхідним параметром якого є об'єкт класу *Block* – зберігає блок даних в хмарних сховищах використовуючи атрибут *storage* та викликаючи на ньому відповідний метод.

Клас *Index* представляє індекс систему у вигляді хеш таблиці яка зберігає усі відомі хеші системі у вигляді ключа та посилання на блок даних у хмарі у вигляді значення. Атрибути та методи:

- *path*, атрибут типу *String* – містить шлях до індексу на диску.
- *size*, атрибут типу *long* – розмір індексу байтах.
- *table*, атрибут типу *bytes* – масив байтів що відповідає структурі хеш таблиці.
- *put*, метод вхідними параметрами якого є хеш типу *Hash* та шлях до блоку у хмарі типу *String* – додає в хеш-таблицю нову пару ключ-значення, де ключом є вхідний хеш, а значенням шлях до блоку даних у хмарі.
- *testHash*, метод вхідним параметром якого є хеш типу *Hash* – перевіряє чи вхідний хеш зберігається в хеш-таблиці.

Клас *Storage* представляє хмарні сховища. Відповідає за збереження блоків даних та резервних копій. Атрибути та методи:

- *provider*, атрибут типу *String* – містить інформацію про постачальника хмарних сховищ.
- *saveBlock*, метод вхідним параметром якого є блок даних типу *Block* – зберігає вхідний блок даних у хмарних сховищах.
- *saveBackup*, метод вхідним параметром якого є резервна копія типу *Backup* – зберігає резервну копію даних у хмарних сховищах.

З розглянутих видів дедублікації обрано дедублікацію на рівні блоків. Не зважаючи на те, що даний вид дедублікації є складнішим у реалізації та проектуванні, дедублікація на рівні блоків забезпечує зменшення вхідного об'єму даних до 95% [1]. В той час як дедублікація на рівні екземплярів напряму залежить від типу даних резервних копій, і найменші зміни у вихідних файлів вимагатимуть повторного запису їх загалом. Також, дедублікація на рівні блоків, забезпечує рівномірне розподілення задач та їх об'єму між робітниками у кластерному середовищі, що сприяє максимальному використанні наявних обчислювальних ресурсів, та рівномірному розподіленню даних між окремими вузлами хмарного сховища. Раніше розглянуто та детально описано чотири типи дедублікації: вихідна, цільова, потокова та гібридна. З поміж них необхідно вибрати той тип дедублікації який найкраще підходить для розподілених обчислень і збереження даних у хмарних сховищах, щоб задовільними цілі дослідження.

Вихідна дедублікація виконується на пристрої, де знаходяться вихідні дані. Оскільки вихідна дедублікація працює на комп'ютері користувачів, то єдиним способом збільшити її швидкість є вертикальна масштабованість, шляхом збільшення ресурсів комп'ютера користувача (покупка нового процесора, оперативної пам'яті та дискового простору). Проте, на комп'ютері користувача містяться і інші програми його повсякденного користування, а диск використовується для зберігання оригінальних даних. Тому, виділити додаткове місце для резервних копій буде фінансово витратно. Враховуючи ці фактори, вихідна дедублікація не підходить для цілей дослідження та завдань розробленої інтелектуальної системи. Цільова дедублікація виконується на цільовому центральному сервері, де на кінцевому етапі будуть зберігатись дані. Даний підхід краще підходить, ніж попередній тим, що підключає роботу центрального серверу, тим самим акумулюючи дані з різних джерел дедублікації. Також такий підхід підходить краще для масштабованості системи, оскільки обчислювальні ресурси, які встановлюють на виділені сервери зазвичай більшої продуктивності; а також, можна розраховувати на максимальне використання ресурсів, оскільки сервер дедублікації буде містити лише програмне забезпечення яке стосується лише дедублікації та збереження даних. Отже, даний тип дедублікації вигідніший з точки зору масштабованості. Проте, враховуючи той факт, що цільова дедублікація вимагає повного пересилання даних, збереження на диск і тільки тоді проводити відсічення дублікатів, оптимізація обмежується лише горизонтальною масштабованістю. Вертикальна масштабованість відпадає, оскільки не зважаючи на кількість процесорів, всі дані проходять через диск, який по своїй природі однопоточний і повільніший ніж оперативна пам'ять.

Потокова дедублікація по своїм природі створена для вирішення проблеми з диском у цільовій дедублікації. Процес розбиття потоку даних відбувається на момент отримання даних на сервері та відсічення дублікатів перед записом на диск. Такий підхід збільшує навантаження на оперативну пам'ять, тримаючи в пам'яті певне вікно даних, де відбувається розбиття і відсічення, що мінімізує навантаження на диск. Також це дозволяє масштабувати систему вертикально. Проте, все ще є єдиний недолік такого підходу, а саме навантаження на мережу та інтернет трафік користувача, оскільки всі дані користувача для резервного копіювання необхідно надіслати до серверу. Це стає великою проблемою коли виконується періодичне резервне копіювання і відсоток змін між копіями дуже малий.

Гібридна дедублікація – це суміш вихідного, цільового та потокового типу дедублікації. А саме розбиття даних на блоки та відсічення дублікатів відбувається на стороні користувача, а на сервер надсилаються лише нові блоки даних. Варто підкреслити, що комп'ютер користувача напряму не втягнений у процес усунення дублікатів напряму, програма-клієнт користувача надсилає список хеш значень блоків, а сервер вказує на ті хеш значення, які йому не відомі. Тобто робота з індексом лежить повністю на сервері. Таким чином можна уникнути додаткової навантаженості на комп'ютер користувача, на мережу та інтернет трафік. Також, враховуючи природну випадковість хеш значень, роботу з блоками легко розподілити між вузлами кластеру, тим самим масштабуючи систему щоб досягти максимальної продуктивності серверного обладнання. Враховуючи ці фактори, було обрано гібридний тип дедублікації для розробки інтелектуальної системи. Розбиття вхідного потоку даних на блоки для дедублікації на рівні блоків передбачає два алгоритми - з фіксованою довжиною та змінною. Згідно підрозділу 1.2 підхід з фіксованою довжиною передбачає розбиття потоку на рівні частини попередньо вибраної довжини, в той час як в підході зі змінною довжиною, точку розбиття вибирає алгоритм, на основі даних самого потоку. Алгоритм при розбитті зі змінною довжиною не гарантує того що блоки будуть мати однакову довжину, проте можливо встановити мінімальний і максимальний розмір блоків. Для прикладу візьмемо потік даних де одиницею є літера української абетки, а середня довжина блоку 5 літер. Вибравши літери від *A* до *Ф* отримаємо 5 блоків даних. Проте, якщо вихідний потік зазнає змін, наприклад цифру *1* буде виставлено між буквами *Г* та *Д*, алгоритми поведуть себе по різному. На рис. 6а подано заданий приклад, використовуючи підхід з розбиттям з фіксованою довжиною. Перший блок не зазнав змін, а отже вважається існуючим і повторне копіювання непотрібне. Проте усі наступні блоки будуть вважатись для системи новими, оскільки їхній контент буде відрізнитись від оригінальних.

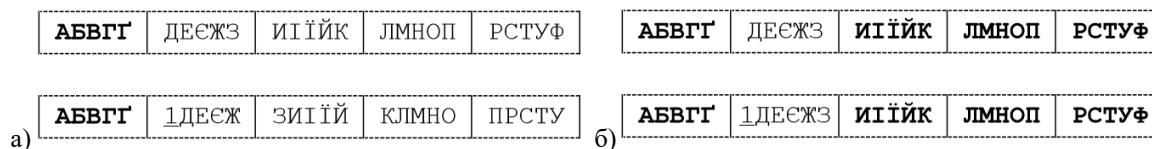


Рисунок 6 – Розбиття а) з фіксованою довжиною та б) зі змінною довжиною

На рис. 6б подано заданий приклад, використовуючи підхід з розбиттям зі змінною довжиною. На відміну від розбиття з фіксованою довжиною, алгоритм зі змінною довжиною, залишив літеру *З* у другому блоці, хоча це і збільшило вихідний розмір блоку, проте усі наступні блоки залишились ідентичними

до оригінальних. Таким чином, змінився лише той блок, в регіоні якого відбувся зсув. Алгоритм розбиття зі змінною довжиною працює не лише зі зсувом, а також із будь-якими змінами, використовуючи алгоритм кільцевого хешу. В якості алгоритму розбиття потоку даних на блоку було вибрано цифровий підпис Рабіна, який базується на основі обчислення кільцевого хешу. Найоптимальнішим розміром вікна є масив довжиною 16 біт, оскільки розміри більших значень не показують значного покращення роботи алгоритму [16]. А отже на кожен вхідний байт даних буде обчислено кільцеве хеш значення, для якого буде застосовано побітову операції I (AND) зі значенням середньої довжини блоку. Для даного дослідження було обрано середню довжину блоку 4096, мінімальну 2048, а максимальну 8192. Згідно досліджень вказаних вище, саме такі параметри дозволяють отримати максимальну точність розбиття даних при вікні довжиною 16 біт. Отже, виконуючи кожен раз для вхідного байту побітову операції AND над кільцевим хешем та середнім значенням довжини блоку – A, ми отримуємо число в межі від нуля до A. Якщо отриманий результат буде дорівнювати заданому шаблону, то поточний байт буде вважатись точкою розбиття потоку на наступний блок. Враховуючи межі довжини блоку, якщо точку знайдено раніше ніж мінімальна довжина, то вона ігнорується до моменту виходу за мінімальну довжину. Якщо точку розбиття не знайдено, а довжина блоку досягає максимальної довжини, то розбиття відбувається в момент досягнення максимальної довжини блоку. Такий випадок можливий при заповненні потоку даних повторюваними байтами, наприклад деякий регіон даних заповнений нулями, а отже кільцевий хеш, буде відмінний від заданого шаблону в будь-якій точці проблемного регіону.

Найважчим, з точки зору обчислень, в інтелектуальній системі є наповнення індексу та пошук в ньому ідентичних блоків даних на основі їхніх хеш значень. А отже дане завдання необхідно масштабувати горизонтально між вузлами кластеру системи та горизонтально між процесорами робочих машин. Для горизонтального масштабування необхідно вибрати ключ розподілу. Враховуючи випадкову природу хеш значень блоків, зобразивши хеш значення у вигляді чисел N-ної системи числення, можна майже задарма отримати зважений ключ розподілу. Для прикладу візьмемо 16-кову систему числення (оскільки даний тип найпоширеніший у представленні хеш значень). Взявши перше число хеш значення, ми отримуємо число від 1 до 16, зі ймовірністю появи . Таким чином ми отримуємо рівномірний розподіл блоків даних між 16-ма вузлами системи. Якщо в системі наявно менше число вузлів то вибір номеру вузла є остача від ділення першого числа хеш значення на кількість вузлів системи. Якщо вузлів є більше ніж 16, то береться друге число, тим самим отримуючи 256 комбінацій (16^2).

Визначившись з горизонтальним розподілом блоків даних між робітниками, можна досягти паралельного запису блоків у сховище кожного вузла. Але також, таким чином можна досягти і реалізації розподіленого індексу. Пошук дублікатів, використовуючи індекс, є лише процесорно-залежним завданням, оскільки індекс зберігається в оперативній пам'яті тим самим уникаючи роботи з диском. А процесорно-залежні завдання можливо масштабувати вертикально, розпаралелюючи роботу між процесорами комп'ютера, що у нашому випадку – пошук хеш значень у індексі. Досліджувана інтелектуальна система дедублікації та розподілу даних у хмарних сховищах передбачає роботу з блоками байтами даних та контролює їх розміщення на основі індексу (хеш-таблиці) в хмарних сховищах. А отже структуру традиційної реляційної бази даних необхідно будувати лише для мета інформацій, як от перелік користувачів та їх резервних копій. На рис. 7 подано схему реляційної бази даних, а саме дві таблиці: *Backup* та *User*, які з'єднані зв'язком багатов до одного. Загалом цих даних достатньо для функціонування досліджуваної системи, оскільки інформацію про інші дані система зберігає у хмарних сховища і вони не підходять для реляційних відношень. Таблиця *User* містить дані про користувачів системи, її колонки:

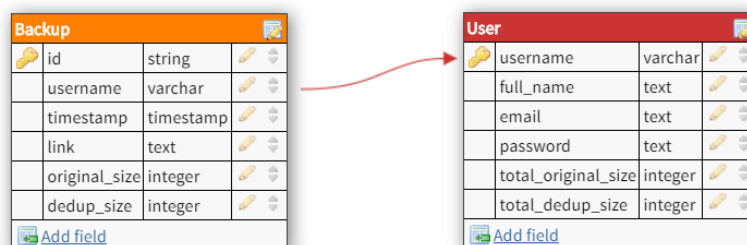


Рисунок 7 – Схема реляційної бази даних

- *username* – типу *varchar*, унікальний ключ таблиці, містить унікальне ім'я користувача і являється його ідентифікатором, може використовуватись для автентифікації користувача;

- *full_name* – типу *text*, містить повне ім'я користувача, використовується для формального звернення до користувача;
- *email* – типу *text*, електронна пошта користувача, може використовуватись для автентифікації користувача;
- *password* – типу *text*, хеш значення паролю, використовується для перевірки реального паролю під час автентифікації користувача;
- *total_original_size* – типу *integer*, загальний оригінальний розмір даних збережених користувачем, змінюється в залежності від додавання нових резервних копій або їх видалення;
- *total_dedup_size* – типу *integer*, загальний розмір даних збережених користувачем після дедублікації, змінюється в залежності від додавання нових резервних копій або їх видалення;

Маючи дані з колонок *total_original_size* та *total_dedup_size*, можна визначити коефіцієнт дедублікації: $1 - \text{total_dedup_size} / \text{total_original_size}$, та кількість місця на диску зекономлених завдяки дедублікації: $\text{total_original_size} - \text{total_dedup_size}$.

Таблиця *Backup* містить дані про резервні копії системи, її колонки:

- *id* – типу *text*, назва резервної копії, переважно складається з ідентифікатора користувача та з мітки часу створення резервної копії;
- *username* – типу *varchar*, унікальний ідентифікатор користувача, використовується для з'єднання з таблицею *User*, реалізуючи зв'язок багато до одного;
- *timestamp* – типу *timestamp*, мітка часу створення резервної копії, використовується для визначення віку резервної копії;
- *link* – типу *text*, посилання на резервну копію у хмарному сховищі;
- *original_size* – типу *integer*, оригінальний розмір резервної копії;
- *dedup_size* – типу *integer*, розмір резервної копії після дедублікації;

Для того щоб дані в колонках *total_original_size* та *total_dedup_size* таблиці *User* були завжди актуальними, необхідно створити тригер бази даних на таблиці *Backup* на діях додавання і видалення рядків. На рис. 8 подано sql тригер *user_total_change* зміни значень колонок *total_original_size* та *total_dedup_size*.

```

1 CREATE OR REPLACE TRIGGER user_total_change
2   AFTER INSERT OR DELETE ON backup FOR EACH ROW
3 BEGIN
4   IF TG_OP = 'INSERT' THEN
5     UPDATE user set
6       total_original_size = total_original_size + :new.original_size,
7       total_dedup_size = total_dedup_size + :new.dedup_size
8     WHERE username = :new.username;
9   ELSIF TG_OP = 'DELETE' THEN
10    UPDATE user set
11      total_original_size = total_original_size - :old.original_size,
12      total_dedup_size = total_dedup_size - :old.dedup_size;
13    WHERE username = :new.username;
14  END IF;
15 END;
```

Рисунок 8 – SQL тригер зміни колонок *total_original_size* та *total_dedup_size*

Даний тригер спрацює після успішного додавання або видалення записів з таблиці *Backup*, який працює наступним чином: якщо буде додано новий рядок, то буде виконано операцію *update*, яка збільшить значення в колонках *total_original_size* та *total_dedup_size* на *original_size* та *dedup_size* відповідно для зв'язаного користувача; якщо буде видалено старий рядок, то буде виконано операцію *update*, яка відповідно зменшить значення *total_original_size* та *total_dedup_size*.

Для зберігання даних на чи то на дисковий простір, чи у хмарні сховища, необхідно використати власний формат файлів, оскільки зберігання блоків даних у реляційних базах буде надлишковими витратами на читання, запис та їхній індекс. Оскільки реляційні база даних створені для зберігання різних типів даних які мають невизначену схему, що створюється під час виконання програм. Проте у випадку з дедублікацією, можна оминати багато таких обмежень, оскільки уся структура і схема даних відома наперед. Отже, для зберігання блоку даних на диску, необхідно мати наступні метадані – порядковий номер, фактичний розмір блоку, оригінальний розмір блоку, хеш значення, алгоритм стиснення, алгоритм шифрування. Варто додати, що фактичний розмір блоку може відрізнятися від оригінального, оскільки алгоритм стиснення та шифрування можуть змінити його. Порядковий номер блоку необхідний для його швидкого доступу під час відновлення резервних копій. На рис. 9 подано структуру файлу, який буде зберігати блоки даних.

Розмір (у байтах):	2	8	2	2	20	1	1	2048-8192
Контент:	MG	ID	CS	OS	HASH	C	E	DATA

Рисунок 9 – Структура файлу зі списком блоків даних

На стороні користувача необхідно мати операційну систему *Windows* з встановленим *Microsoft C++ Runtime Library 2015* і вище або операційну систему *Linux*. Також не залежно від операційної системи повинен бути встановлений браузер *Chrome* або його аналог *Chromium*. На стороні серверу, необхідне встановлення *JVM* (*Java Virtual Machine*). А також відкриті порти для доступу по *HTTP* протоколу. Також вимагається встановлення на основному вузлі кластеру одної із реляційних баз даних, таких як *Postgres* або *MySQL*. Для підключення хмарних сховищ необхідна оплачена та відкрита підписна на *Amazon S3*. В разі використання власного середовища для хмарних сховищ, необхідно мати сервер із операційною системою *Linux* та встановленим одним із безплатних *S3* середовищ – *LeoFS*, *RiakCS*, *Minio*, *Ceph*, *SwiftStack* та ін. Користувацький інтерфейс написаний на *JavaScript*, *Html* та *Css*. Програма-клієнт написаний на *Rust*. Програма-сервер на *Scala*. Інтелектуальна система дедублікації та розподілу даних у хмарних сховищах використовується в програмному забезпеченні для резервного копіювання даних. Дана система може використовуватись як окремий продукт, так і у вигляді модуля іншої системи.

Інтелектуальна система дедублікації та розподілу даних у хмарних сховищах є багатомодульною та складається з наступних модулів: користувацького інтерфейсу, програми-клієнта, індексу та програми-сервера. Вхідні дані користувача це набір файлів для резервного копіювання. Програма-клієнт формує на основі цих даних потік байтів даних і розбиває його на блоки використовуючи алгоритм розбиття на основі кільцевого хешу. Для кожного отриманого блоку даних обчислюється хеш значення і перевіряється на наявність цього хешу в системі. Якщо таких хеш уже відомий то використовується посилання на цей хеш, замість усього блоку. Якщо ні, то даний блок зберігається в системі а хеш записується в індекс з посиланням на блок даних. Даний процес повторюється для кожного блоку до кінця потоку. В результаті вхідний потік зберігається як список із хешів для майбутнього відновлення даних з резервної копії. Розподіл даних між робітниками в кластері відбувається на основі хеш значення. Хеш значення представлено у вигляді набору чисел з шістнадцяткової системи числення. Для вибору необхідного робітника використовуються перше число хешу, в залежності від кількості робітників (це повинно бути обов'язково число степені двійки). Наприклад, якщо число робітників 16, то значення числа хешу дорівнює номеру робітника, якщо менше ніж 16, то число масштабується до необхідної основи. Якщо кількість робітників більше 16 то використовується наступне число робітників і т.д. Варто також додати що індекс є розподіленим, тобто кожен робітник має свій власний індекс.

Модуль користувацького інтерфейсу написаний на мові програмування *JavaScript*. Для реалізації користувацького інтерфейсу як настільного додатку, використано бібліотеку *ElectronJS*, яка використовує потужності веб-браузера *Chrome* для відображення веб-сторінок поза межами браузера. Для побудови структури користувацького інтерфейсу було використано архітектурний шаблон модель-вигляд-контролер та його реалізацію *AngularJS*. Структурно модуль складається із:

- *package.json* – файл налаштувань, містить перелік підключених бібліотек та список задач побудови проекту.
- *main.js* – файл бізнес логіки головного вікна, містить конфігурації відкриття програми та перевірки локального кешу. В залежності чи користувач ідентифікований відкриває для нього основне вікно або вікно входу в систему.
- *main.css* – файл який містить таблиці стилів, описує зовнішній вигляд користувацького інтерфейсу.
- *main.html* – файл опису структури головного вікна, його макет.
- *login.html* – файл вигляду вікна входу.
- *dashboard.html* – файл вигляду загальної інформації про резервні копії користувача, дозволяє створювати нові та відновлювати дані з існуючих.
- *bridge.js* – файл бізнес логіки для зв'язку з програмою-клієнтом.

Модуль програма-клієнт написаний на мові програмування *Rust*. Програма реалізована як архітектурний шаблон інтерфейс командного рядка. Тобто дана програма може використовуватись без користувацького інтерфейсу шляхом виклику першої з передачею вхідних аргументів. Проте даний підхід використовується лише для відлагодження. Для обробки командного рядка використовується бібліотека *Clap-rs*. Використовує *Reqwest* в якості HTTP клієнта. Для розбиття потоку на блоки використовує бібліотеку *Rabinpoly*, а для обчислення їх хешів – *SHA-1*. Структурно модуль складається:

- *Cargo.toml* – файл налаштувань, містить перелік підключених бібліотек та список задач побудови проекту.

- `main.rs` – точка входу в програму, містить обробку вхідних аргументів та виклик необхідних функцій.
- `gabinpoly.rs` – розбиття потоку даних на блоки зі змінною довжиною використовуючи алгоритм Рабіна.
- `hash.rs` – обрахунок хеш значень блоків дедублікації.
- `http.rs` – інтерфейс для виклику HTTP запитів.
- `test_hashes.rs` – перевірка хешів на існуючість.
- `send_data.rs` – надсилання нових блоків до сервера.
- `backup.rs` – композиція наявних функцій та інтерфейсів для процесу резервного копіювання використовуючи дедублікації.
- `recover.rs` – відновлення даних з резервних копій.

Модуль програма-сервер написаний на мові програмування *Scala* та реалізує модель акторів за допомогою бібліотеки *Akka*. В якості HTTP серверу використовує *Akka-http*. Для роботи з хмарними сховищами використовує бібліотеку *Aws-sdk*, яка надає інтерфейс для спілкування з *Amazon S3*. Структурно модуль складається із:

- `build.sbt` – файл налаштувань, містить перелік підключених бібліотек та список задач побудови проекту.
- `MasterMain.scala` – точка входу в систему головного вузла кластеру, координатора.
- `WorkerMain.scala` – точка входу в систему робітника, вузла кластеру.
- `HttpServer.scala` – файл налаштувань HTTP сервера та маршрутизації веб запитів.
- `MasterActor.scala` – актор-координатор, координує вхідні хеші і на основі їх значень вибирає робітника для делегування задач.
- `WorkerActor.scala` – актор-робітник, спілкується з індексом та виконує надсилання блоків дедублікації до хмарних сховищ.

Програма-сервер та її робітники використовують хмарні сховища для зберігання блоків даних та дерева об'єктів у вигляді резервних копій. В якості хмарного сховища необхідно використовувати сервіси сумісні з S3 інтерфейсом.

Клієнтську частину інтелектуальної системи призначено для використання на персональних комп'ютерах, які працюють на Linux, MacOS та Windows. Системні вимоги:

- Оперативна пам'ять 1 ГБ.
- Місце на диску - не менше ніж 256 МБ.
- Відео пам'ять – не менше ніж 256 МБ.

Серверну частину інтелектуальної системи призначено для використання у розподіленому кластерно середовищі, з щонайменше одним робочим вузлом який працює на Linux, MacOS та Windows. Системні вимоги для кожного вузла окремо:

- Оперативна пам'ять 4-16 ГБ.
- Місце на диску 1 ТБ.

Користувацький інтерфейс та програму-клієнт необхідно спочатку встановити на комп'ютері користувача. Користувацький інтерфейс викликається шляхом запуску новоствореної іконки на робочому столі або запуском за допомогою скрипта розташованого в директорії встановленої системи. Програму-клієнт не потрібно додатково викликати, це відбувається автоматично в залежності від маніпуляцій з користувацьким інтерфейсом.

Для успішного запуску програми-сервера та індексу необхідно спочатку налаштувати робоче розподілене середовище у вигляді кластеру. Необхідно знати усі мережі IP-адреси кожного вузла кластеру і вказати їх під час запуску програми-сервера. Індекс будується автоматично при старті кожного робітника програми-клієнта.

Точкою входу зі сторони користувача є користувацький інтерфейс у вигляді настільного додатку. Точкою входу зі сторони сервера є скрипт запуску кластерного середовища.

Вхідними даними для користувацького інтерфейсу є маніпуляції користувача, список файлів для дедублікації та посилання на резервну копію для відновлення даних. Вхідними даними для програми-клієнта є файли для дедублікації, список блоків дедублікації для надсилання. Вхідними даними для програми-сервера є назва резервної копії, список хешів вхідних блоків дедублікації та унікальні блоки дедублікації. Вхідними даними для індексу є хеші та посилання на блоки у сховищі.

Вихідними даними для користувацького інтерфейсу є статус процесу резервного копіювання, посилання на резервну копію, файли резервної копії. Вихідними даними для програми-клієнта є усі блоки дедублікації, список хешів блоків дедублікації. Вихідними даними для програми-серверу є резервна копія. Вихідними даними для індексу є відповідь на наявність хешу блоку в системі.

Після розгортання серверу, користувач має змогу підключитись до нього за допомогою програми-клієнта. Проте спочатку необхідно автентифікуватись у системі. В разі неуспішної автентифікації (неправильно введена поштова скринька або пароль), програма попросить користувача, перевірити дані та ввести їх знову. Після введення правильного паролю, та натиску кнопки *Увійти*, система успішно автентифікує користувача та головне вікно користувацького інтерфейсу (рис. 10). Головне вікно програми складається з двох частин – бічної панелі та головного поля. На бічній панелі є можливість вибрати зміст активного поля – резервні копії або налаштування. За замовчуванням, при вході користувача в систему, йому відкривається поле із резервними копіями, на якому зображений список із раніше створених резервних копій. Інтерфейс дозволяє проводити сортування та пошук резервних копій за змістом рядків. Також присутня кнопка *Добавити*, яка дозволяє створити нову резервну копію, при натисканні на яку, відкривається вікно вибору файлів для резервного копіювання.

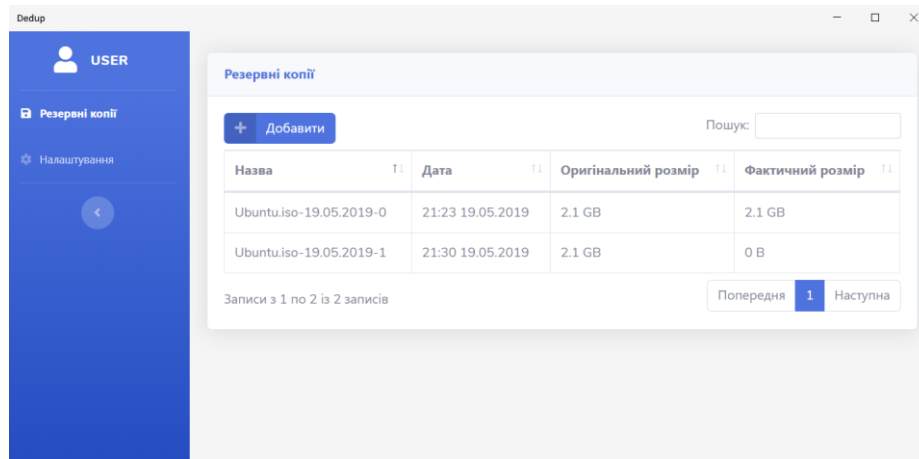


Рисунок 10 – Головне вікно програми

Для тестового запуску, вибрано документ *Диплом.docx*, розміром 5 мб. Після вибору файлу та натискання кнопки *Вибір* розпочнеться процес дедублікації, а саме: створення потоку даних на основі цього файлу та розбиття його на блоки; обчислення хеш значення кожного блоку та перевірка на існуючі його в системі; надсилання нових блоків даних на сервер; збереження резервної копії на хмарному сховищі та відображення її у програмі користувача (рис. 11-12). Для тестування процесу дедублікації під час резервного копіювання, необхідно додати файл *Документ* ще раз і подивитись на зміни поля *Фактичний розмір*. На рис. 13 подано вікно з резервними копіями де було вдруге додано файл *Документ*. Як видно з рисунку, назва резервної копії є *Документ.docx-19.05.2019-1* а її фактичний розмір – 0 байтів, тобто на сервері не було збережено жодних нових блоків. На рис. 14 подано останні повідомлення кластеру для резервної копії *Документ.docx-19.05.2019-1*. Оскільки усі дані уже відомі системі, в журналі немає повідомлень щодо нових блоків, а отже є лише два сповіщення, про початок та кінець. Останнє повідомлення, також вказує що система отримала 0 нових байтів та 0 нових блоків.

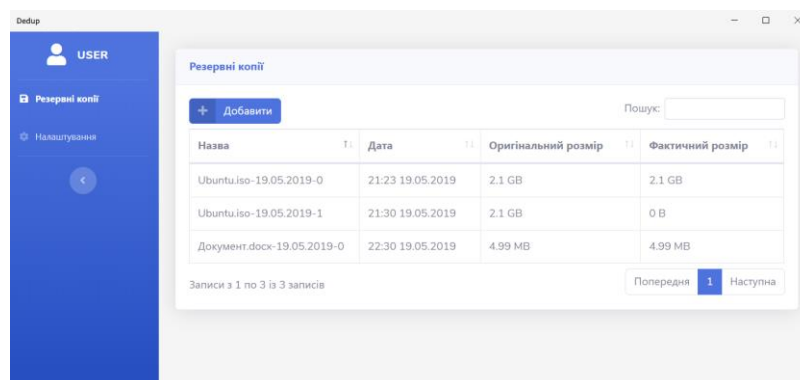


Рисунок 11 – Вікно резервних копій із доданим документом

```

22:29:570 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 0cacd9c9c3158f2e13c43d881d323289c2d61e94 : 4348
22:29:577 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 496a96d3bd7a0f7ffccff98bb095d7d65617741 : 7379
22:29:580 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 62c63821f4ad60cd9f39d2f902160183e651f7b : 5256
22:29:581 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 273094160ab75ab9e6641269a2b632297e606cc3 : 4269
22:29:589 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : c815ee1685fb0e18bbaa67573046a6029a251a24 : 7275
22:29:597 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : db7c7cbfb805990074866e19d49d969565ee8f5 : 3679
22:29:598 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 6abb04c8a488900fefe4e0290b676e482aa319ce : 5352
22:29:607 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 726b41f26b99659422d191ac17239c50465a42e3 : 6479
22:29:615 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 6550cf7aabc4efeb0211d558b6bcb379a89598f7a : 3359
22:29:627 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 6b6bc85ccadb6512801a29d4dfe27d35cabcc3dc : 2680
22:29:633 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : fe2507a3c018a6222597eb9bb382360af9dbee9 : 3742
22:29:639 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : ad3bae80b2634facd42c415022ed86995e687dc : 4516
22:29:643 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 0d167488d55a8352e2e3cb5f657190d2d1ba2e56 : 7184
22:29:639 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : dfb7c7920d70fdd120f7e911d504812212f53cc : 2451
22:29:646 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 8142e05b10d6ae3e6a3800de4c0e44d4eb508e596 : 3095
22:29:646 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 40d9cd5f2f181e2c5730bbc6e0798b6bad254d3e : 5281
22:29:654 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 0650cf5d3e9011dfa0f85045b8215cee4bd0d1f7 : 2078
22:29:659 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 15f30fba6da4780a91b976f6b6d8695ea2634d9 : 3223
22:29:667 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : eb135e66faf6c9b13334b5cb3a65d77aff5538f2 : 6251
22:29:670 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 3b08598a0aa641844b4f0dd640ef8c79cfc5dc9e : 2154
22:29:671 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 98638d137342f1b12299c272f06a76bf608e73b3 : 3965
22:29:680 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 8d6aace9d8dc0a2ae56c45b89ef4113463111c6 : 7649
22:29:685 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 276cbfe9b2a705e80039c5a4099da0442ed6781f : 7759
22:29:690 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 1a76827c2321b5774429dc4d447b50713aa03b0f0 : 6946
22:29:694 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 04d4c76e47733da3b0aa9e9ecb2da299987733c1f : 4532
22:29:696 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 587983b6b8f6e98f79a4341fd960dfa99eea7c34 : 3422
22:29:697 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : a8e4ac9aa00c8d615748e5543b1cdf1a9e72455 : 7856
22:29:702 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : dc88178f2db20baab4ed6c48609a1a33255f4e4 : 7641
22:29:708 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : bb1c612876f77288afb55af6297315d54d1f2b5a : 2120
22:29:710 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 4168d41cb2435eef3ad09bdf220d771873599e3 : 7627
22:29:719 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 4b63f065137cf98601cd79089c99e5983abf636 : 5349
22:29:723 [worker-1] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 45ab2cdd0f5b69741c9fd591da207b157930ac8d : 5016
22:29:733 [worker-2] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : dcb24f1d6880050e7981b68db7a72f00926fecfb : 5122
22:29:738 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 3b093efda2be455a94a0cc282e293844f3a044b1 : 6418
22:29:745 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 72078081259bd09985f3e13760f7a3cd56286bca : 3721
22:29:746 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : 6f62f16678eb3b8a6f613de66493df08fb6dd329 : 2704
22:29:755 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-0 : f1b49172485754ff80780cc5da8cf58104f0907e : 2622
22:30:012 [coordinator] INFO - FINISHED : Документ.docx-19.05.2019-0 : ALL_BYTES=52232394 : NEW_BYTES=52232394 : BLOCKS=10435(new=10435)
    
```

Рисунок 12 – Журнал повідомлень для першої резервної копії

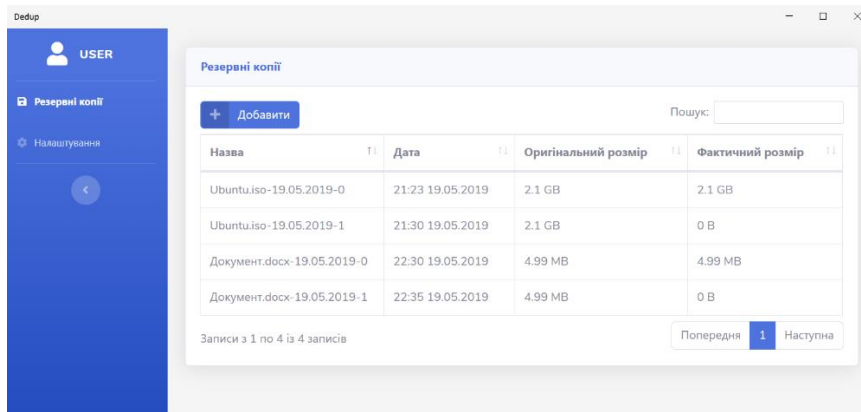


Рисунок 13 – Вікно резервних копій із вдруге доданим документом

```

22:33:703 [coordinator] INFO - STARTED : Документ.docx-19.05.2019-1 : USER=user
22:35:214 [coordinator] INFO - FINISHED : Документ.docx-19.05.2019-1 : ALL_BYTES=52232394 : NEW_BYTES=0 : BLOCKS=10435(new=0)
    
```

Рисунок 14 – Журнал повідомлень для другої резервної копії

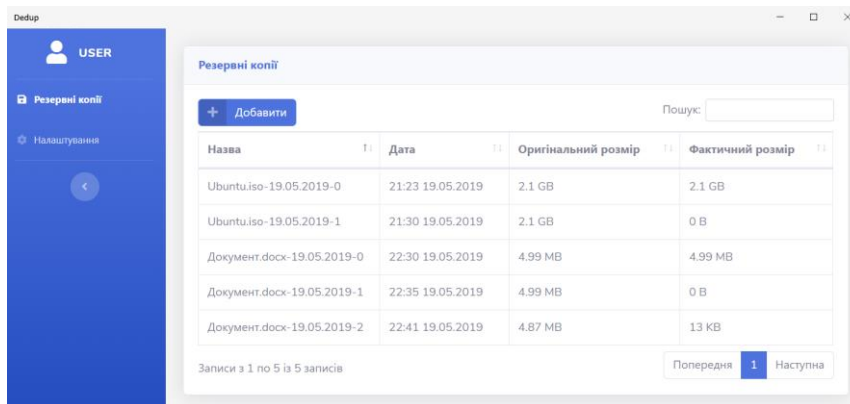


Рисунок 15 – Вікно резервних копій із втретє доданим документом

Для того щоб протестувати чи алгоритм розбиття потоку даних на блоки працює правильно, та будь-які не значні зміни в документі не потягнуть за собою заміну всіх блоків після точки змін. Для цьо-

го відкриємо оригінальний файл *Документ* та видалимо звідки трохи інформації і повторимо кроки для того щоб зробити нову резервну копію. На рис. 15 подано список резервних копій, з новоствореною *Документ.docx-19.05.2019-2*, яка містить файл *Документ* з дещо зміненими даними в середині файлу. Як видно з отриманих результатів, оригінальний розмір 4.87 мегабайт (оскільки було видалено частину даних), а фактичний – 13 кілобайт. Що означає, що алгоритм розбиття блоків виконав своє завдання, і змінив лише ті блоки, де відбувалися зміни. На рис. 16 подано останні повідомлення кластеру для резервної копії *Документ.docx-19.05.2019-2*. Згідно цих повідомлень, було добавлено лише 2 нові блоки (разом 13 кілобайт), які охоплюють те місце де було видалено дані з оригінального документа.

```
22:39:201 [coordinator] INFO - STARTED : Документ.docx-19.05.2019-2 : USER=user
22:40:365 [worker-3] INFO - NEW BLOCK : Документ.docx-19.05.2019-2 : 6019aef800db07e4c1f7b7a2a8236b235c0ff344 : 7651
22:40:387 [worker-0] INFO - NEW BLOCK : Документ.docx-19.05.2019-2 : f5eb3712d8ead6a4637dc82989acde77ffd64abf : 5246
22:41:856 [coordinator] INFO - FINISHED : Документ.docx-19.05.2019-2 : ALL_BYTES=5106565 : NEW_BYTES=12897 : BLOCKS=10415(new=2)
```

Рисунок 16 – Журнал повідомлень для другої резервної копії

Також була зменшена загальна кількість байтів та блоків. Отже, під час тестового запуску, файл *Документ* було тричі надіслано для резервного копіювання, що завдяки дедублікації, займає на сховищі даних лише 4.99 МБ + 131 КБ = 5.1 МБ. Тоді, якщо би копіювання відбувалось без дедублікації, то фактичний розмір був би 4.99 МБ + 4.99 МБ + 4.87 МБ = 14.85 МБ.

Висновки

1. Здійснено аналіз сучасних технологій проектування та створення систем у сфері резервного копіювання та відновлення даних, здійснено порівняльний аналіз технічних реалізацій та алгоритмів, застосованих в реалізаціях Загальною метою розроблення є створення інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах, що дозволить використовувати її в програмному забезпеченні резервного копіювання даних для усунення дублікатів та підвищення пропускної здатності вхідного потоку даних та загального швидкості процесу. Досліджувана інтелектуальна система дедублікації та розподілу даних у хмарних сховищах призначена довершити існуючі процеси дедублікації та їх інтеграцію з хмарними технологіями. Місцем застосування розроблюваної системи є підприємства або персональні користувачі метою яких є резервне копіювання даних. Було розглянуто відомі системи резервного копіювання, використовуючи дедублікацію даних у хмарних сховищах, а також їхні плюси та мінуси. Згідно їх опису, дані системи показують високі коефіцієнти дедублікації, а також можливість інтеграції в існуючі програми резервного копіювання та підключення хмарних сховищ різних постачальників і навіть приватних сховищ власного виробництва (за умови використання інтерфейсу S3). Проте оглянуті системи використовують хмарні сховища лише як кінцеву частину збереження даних, в той час як досліджувана інтелектуальна система націлена на повне використання хмарних технологій шляхом розподілу даних та завдань дедублікації. Таким чином, в той час як оглянуті аналогічні системи працюють на односерверних монолітних машинах, досліджувана система дозволяє підключати комп'ютерну мережу у вигляді обчислювального кластеру для виконання завдань дедублікації. Ефектом від впровадження інтелектуальної системи дедублікації та розподілу даних у хмарних сховищах є:

- зменшення витрати на збереження резервних копій – процес дедублікації передбачає усунення дублюючих даних, що є критично важливим для періодичних резервних копій, де відсоток зміни даних між ними досить малий;
- збільшення швидкодії, використовуючи алгоритми розподілення даних між робочими вузлами кластеру системи;
- можливість роботи системи на віддалених хмарних серверах та збереження даних у хмарних сховищах;
- легка інтеграції у існуючі рішення резервного копіювання даних.

2. Побудовано засобами UML концептуальну модель інтелектуальної інформаційної системи дедублікації та розподілу даних у хмарних сховищах, задля автоматизації та ефективної роботи програмного забезпечення у сфері резервного копіювання та відновлення даних

3. Проаналізовано (наведено недоліки та переваги використання різних підходів) та обрані ефективні методи вирішення задач: гібридну дедублікацію на рівні блоків, розбиття потоку даних на основі цифрового відбитку Рабіна, розподіл даних на основі хеш значень блоків дедублікації та використання розподіленого індексу. В ході аналізу засобів вирішення задач було обрано мову програмування Rust для написання клієнтської частини, мову програмування Scala для серверної частини, інструментарій Akka для менеджменту розподілених обчислень та Amazon S3 в якості хмарного сховища. Розроблено інтелектуальну систему дедублікації та розподілу даних у хмарних сховищах, наведено опис програмного забезпечення, розглянуті кроки для роботи користувача.

4. Розроблено інтелектуальну систему дедублікації та розподілу даних у хмарних сховищах. Проведено тестування роботи проєктованої системи та наведено декілька контрольних прикладів, проаналізо-

вано отримані результати. Підсумувавши вищесказане, можна зробити висновок, що мета розробки досягнута. Побудовано інтелектуальну систему дедублікації та розподілу даних у хмарних сховищах, яка при доробці систем перевірки цілісності резервних копій та оптимізації дискового простору після видалення старих резервних копій може використовуватись на практиці.

Список літератури

- [1] Understanding Data Deduplication [Електронний ресурс]. – Режим доступу: <https://www.druva.com/understanding-data-deduplication>
- [2] Explaining deduplication rates and single-instance storage to clients [Електронний ресурс]. – Режим доступу: <https://searchchannel.techtarget.com/tip/Explaining-deduplication-rates-and-single-instance-storage-to-clients><http://zakon.rada.gov.ua/laws/show/2469-19>
- [3] Inline vs. post-processing deduplication appliances [Електронний ресурс]. – Режим доступу: <https://searchdatabackup.techtarget.com/tip/Inline-vs-post-processing-deduplication-appliances>
- [4] Introduction to Data Deduplication [Електронний ресурс]. – Режим доступу: <https://www.petri.com/data-deduplication-introduction>
- [5] Rabin M. O. Fingerprinting by random polynomials / M. O. Rabin // Center for Research in Computing Technology Harvard University Report – Harvard, 1981.
- [6] Tanenbaum A.S. Distributed Systems / A.S. Tanenbaum, M. van Steen. – Upper Saddle River : Pearson Prentice Hall, 2017. – 15 с.
- [7] Amdahl G. The validity of the single processor approach to achieving large-scale computing capabilities. / G. Amdahl. – Atlantic City : Proceedings of AFIPS, 1967.
- [8] Using CloudReduce for cloud-based data deduplication [Електронний ресурс]. – Режим доступу: <https://cloud.google.com/solutions/partners/storereduce-cloud-deduplication>
- [9] OpenDedup Overview [Електронний ресурс]. – Режим доступу: <https://opendedup.org/odd/overview/>
- [10] 10. Rumbaugh J. The unified modeling language reference manual / J. Rumbaugh, I. Jacobson, G. Booch // Addison Wesley Longman Inc. – 1999.
- [11] Rolling hash, Rabin Karp, palindromes, rsync and others [Електронний ресурс]. – Режим доступу: <https://www.infoarena.ro/blog/rolling-hash>
- [12] Vysotska V. Methods based on ontologies for information resources processing / V. Vysotska, L. Chyrun, V. Lytvyn. - LAP Lambert Academic Publishing, 2016.
- [13] Vysotska V. Information technologies of gamification for training and recruitment / V. Vysotska, N. Shakhovska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [14] Висоцька, В.А. Особливості проектування та впровадження систем електронної комерції. / В.А. Висоцька // Комп'ютерні науки та інформаційні технології, Вісник Національного університету "Львівська політехніка". – Львів 2008. – № 629. – С. 34-45.
- [15] Vysotska V. Web resources processing based on ontologies / V. Vysotska, V. Lytvyn. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [16] Vysotska V. Tekhnolohiyi elektronnoyi komertsiyi ta Internet-marketynhu / V. Vysotska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [17] Vysotska V. Internet systems design and development based on Web Mining and NLP / V. Vysotska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [18] Vysotska V. Computer linguistics for online marketing in information technology: Monograph / V. Vysotska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [19] Lytvyn V. A Method of Construction of Automated Basic Ontology / V. Lytvyn, V. Vysotska, W. Wojcik, D. Dosyn // 1st International Conference Computational Linguistics and Intelligent Systems, COLINS'2017. – 21 April 2017, Kharkiv. – P. 75-83.
- [20] Lytvyn V. Intelligent System Structure for Web Resources Processing and Analysis / V. Lytvyn, V. Vysotska, L. Chyrun, A. Smolarz, O. Naum // 1st International Conference Computational Linguistics and Intelligent Systems, COLINS'2017. – 21 April 2017, Kharkiv. – P. 56-74.
- [21] Berko A. Features of information resources processing in electronic content commerce / Andriy Berko, Victoria Vysotska, Lyubomyr Chyrun // Applied Computer Science. ACS journal. – Volume 10, Number 2. – Poland, 2014. – ISSN 2353-6977 (Online), ISSN 1895-3735 (Print). – PP. 5-19.
- [22] Берко, А.Ю. Методи та засоби оцінювання ризиків безпеки інформації в системах електронної комерції / А.Ю. Берко, В.А. Висоцька, І.В. Рішняк // Інформаційні системи та мережі. Вісник Національного університету "Львівська політехніка". – Львів 2008. – № 610. – С.20-33.
- [23] Vysotska Victoria. Web Content Processing Method for Electronic Business Systems / Victoria Vysotska, Lyubomyr Chyrun // International Journal of Computers & Technology. – Vol 12, No 2. – December 2013. – PP. 3211-3220.

- [24] Висоцька В.А. Моделювання етапів життєвого циклу комерційного web-контенту / В.А. Висоцька, Л.Б. Чирун, Л.В. Чирун // Інформаційні системи та мережі. Вісник Національного університету "Львівська політехніка". – Львів 2011. – № 715. – С. 69-87.
- [25] Берко, А.Ю. Алгоритми опрацювання інформаційних ресурсів в системах електронної комерції / А.Ю. Берко, В.А. Висоцька, Л.В. Чирун // Комп'ютерні науки та інформаційні технології. Вісник Національного університету "Львівська політехніка". – Львів 2008. – № 616. – Стор.128-136.
- [26] Vysotska Victoria. Commercial Web Content Lifecycle Model: conference proceedings, November 16-19, 2011 / Victoria Vysotska, Lubomyr Chyrun, Lilya Chyrun // The 6th International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT'2011) which will be held November 16-19, 2011 at Lviv Polytechnic National University (Lviv, Ukraine) – Lviv 2011. – P. 160-163.
- [27] Берко А.Ю. Проектування навігаційного графу Web-сторінок бази даних систем електронної комерції. / А.Ю. Берко, В.А. Висоцька // Комп'ютерні науки та інформаційні технології, Вісник Національного університету "Львівська політехніка". – Львів 2009. – № 638. – С. 3-14.
- [28] Берко А.Ю. Семантична інтеграція неповних та неточних даних / А.Ю. Берко, В.А. Висоцька, В.В. // Збірник наукових праць «Системи обробки інформації. Безпека та захист інформації в інформаційних системах», Випуск 7 (79). – Харків 2009. – С. 93-98.
- [29] Берко, А.Ю. Моделі та методи проектування інформаційних систем електронної комерції / А.Ю. Берко, В.А. Висоцька // Автоматизовані системи управління та прилади автоматики. Науково-технічний журнал. – Харків 2007. – № 138. – С.55-66.
- [30] Алексеева К.А. Управління Web-ресурсами за умов невизначеності / К.А. Алексеева, А.Ю. Берко, В.А. Висоцька // Журнал «Технологический аудит и резервы производства». – Vol 2, No 2(22) (2015). – Харків, 2015. – ISSN (print) 2226-3780, ISSN (on-line) 2312-8372. – С. 4-7.
- [31] Vysotska V. Designing features of architecture for e-commerce systems / Victoria Vysotska, Lyubomyr Chyrun // MEST Journal (Management Education Science & Society Technologie). – Vol.2 No.1. – P. 57-70.
- [32] Vysotska V. Set-theoretic models and unified methods of information resources processing in e-business systems / Victoria Vysotska, Lyubomyr Chyrun // Applied Computer Science. ACS journal. – Volume 10, Number 3. – Poland, 2014. – ISSN 2353-6977 (Online), ISSN 1895-3735 (Print). – P. 5-2.

References

- [1] Understanding Data Deduplication [Electronic resource]. – Available: <https://www.druva.com/understanding-data-deduplication>
- [2] Explaining deduplication rates and single-instance storage to clients [Electronic resource]. – Available: <https://searchhitchannel.techtarget.com/tip/Explaining-deduplication-rates-and-single-instance-storage-to-clients><http://zakon.rada.gov.ua/laws/show/2469-19>
- [3] Inline vs. post-processing deduplication appliances [Electronic resource]. – Available: <https://searchdatabackup.techtarget.com/tip/Inline-vs-post-processing-deduplication-appliances>
- [4] Introduction to Data Deduplication [Electronic resource]. – Available: <https://www.petri.com/data-deduplication-introduction>
- [5] Rabin M. O. Fingerprinting by random polynomials / M. O. Rabin // Center for Research in Computing Technology Harvard University Report – Harvard, 1981.
- [6] Tanenbaum A.S. Distributed Systems / A.S. Tanenbaum, M. van Steen. – Upper Saddle River : Pearson Prentice Hall, 2017. – 15 с.
- [7] Amdahl G. The validity of the single processor approach to achieving large-scale computing capabilities. / G. Amdahl. – Atlantic City : Proceedings of AFIPS, 1967.
- [8] Using StorReduce for cloud-based data deduplication [Electronic resource]. – Available: <https://cloud.google.com/solutions/partners/storreduce-cloud-deduplication>
- [9] OpenDedup Overview [Electronic resource]. – Available: <https://opendedup.org/odd/overview/>
- [10] Rumbaugh J. The unified modeling language reference manual / J. Rumbaugh, I. Jacobson, G. Booch // Addison Wesley Longman Inc. – 1999.
- [11] Rolling hash, Rabin Karp, palindromes, rsync and others [Electronic resource]. – Available: <https://www.infoarena.ro/blog/rolling-hash>
- [12] Vysotska V. Methods based on ontologies for information resources processing / V. Vysotska, L. Chyrun, V. Lytvyn. - LAP Lambert Academic Publishing, 2016.
- [13] Vysotska V. Information technologies of gamification for training and recruitment / V. Vysotska, N. Shakhovska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.

- [14] Vysotska, V.A. Osoblyvosti proektuvannya ta vprovadzhennya system elektronnoyi komertsiiyi. / V.A. Vysotska // *Komp'yuterni nauky ta informatsiyni tekhnolohiyi*, Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika". – Lviv 2008. – № 629. – S. 34-45.
- [15] Vysotska V. Web resources processing based on ontologies / V. Vysotska, V. Lytvyn. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [16] Vysotska V. Tekhnolohiyi elektronnoyi komertsiiyi ta Internet-marketynhu / V. Vysotska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [17] Vysotska V. Internet systems design and development based on Web Mining and NLP / V. Vysotska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [18] Vysotska V. Computer linguistics for online marketing in information technology: Monograph / V. Vysotska. - Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2018.
- [19] Lytvyn V. A Method of Construction of Automated Basic Ontology / V. Lytvyn, V. Vysotska, W. Wojcik, D. Dosyn // 1st International Conference Computational Linguistics and Intelligent Systems, COLINS'2017. – 21 April 2017, Kharkiv. – P. 75-83.
- [20] Lytvyn V. Intelligent System Structure for Web Resources Processing and Analysis / V. Lytvyn, V. Vysotska, L. Chyrun, A. Smolarz, O. Naum // 1st International Conference Computational Linguistics and Intelligent Systems, COLINS'2017. – 21 April 2017, Kharkiv. – P. 56-74.
- [21] Berko A. Features of information resources processing in electronic content commerce / Andriy Berko, Victoria Vysotska, Lyubomyr Chyrun // *Applied Computer Science. ACS journal*. – Volume 10, Number 2. – Poland, 2014. – ISSN 2353-6977 (Online), ISSN 1895-3735 (Print). – PP. 5-19.
- [22] Berko, A.YU. Metody ta zasoby otsynuyvannya ryzkyiv bezpeky informatsiiyi v systemakh elektronnoyi komertsiiyi / A.YU. Berko, V.A. Vysotska, I.V. Rishnyak // *Informatsiyni systemy ta merezhi*. Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika". – Lviv 2008. – № 610. – S.20-33.
- [23] Vysotska Victoria. Web Content Processing Method for Electronic Business Systems / Victoria Vysotska, Lyubomyr Chyrun // *International Journal of Computers & Technology*. – Vol 12, No 2. – December 2013. – PP. 3211-3220.
- [24] Vysotska V.A. Modelyuvannya etapiv zhyttyevoho tsykladu komertsiiynoho web-kontentu / V.A. Vysotska, L.B Chyrun, L.V. Chyrun // *Informatsiyni systemy ta merezhi*. Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika". – L'viv 2011. – № 715. – Stor. 69-87.
- [25] Berko, A.YU. Alhorytmy opratsyuvannya informatsiiynkh resursiv v systemakh elektronnoyi komertsiiyi / A.YU. Berko, V.A. Vysotska, L.V. Chyrun // *Komp'yuterni nauky ta informatsiyni tekhnolohiyi*. Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika". – Lviv 2008. – № 616. – Stor.128-136.
- [26] Vysotska Victoria. Commercial Web Content Lifecycle Model: conference proceedings, November 16-19, 2011 / Victoria Vysotska, Lyubomyr Chyrun, Lilya Chyrun // *The 6th International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT'2011) which will be held November 16-19, 2011 at Lviv Polytechnic National University (Lviv, Ukraine)* – Lviv 2011. – P. 160-163.
- [27] Berko A.YU. Proektuvannya navihatsiiynoho hrafu Web-storinok bazy danykh system elektronnoyi komertsiiyi. / A.YU. Berko, V.A. Vysotska // *Komp'yuterni nauky ta informatsiyni tekhnolohiyi*, Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika". – Lviv 2009. – № 638. – S. 3-14.
- [28] Berko A.YU. Semantychna intehratsiia nepovnykh ta netochnykh danykh / A.YU. Berko, V.A. Vysotska, V.V. // *Zbirnyk naukovykh prats' «Systemy obrobky informatsiiyi. Bezpeka ta zakhyst informatsiiyi v informatsiiynykh systemakh»*, Vypusk 7 (79). – Kharkiv 2009. – S. 93-98.
- [29] Berko, A.YU. Modeli ta metody proektuvannya informatsiiynykh system elektronnoyi komertsiiyi / A.YU. Berko, V.A. Vysotska // *Avtomatyzovani systemy upravlinnya ta pryklady avtomatyky*. Naukovotekhnichnyy zhurnal. – Kharkiv 2007. – № 138. – S.55-66.
- [30] Alyeksyeyeva K.A. Upravlinnya Web-resursamy za umov nevyznachenosti / K.A. Alyeksyeyeva, A.YU. Berko, V.A. Vysotska // *Zhurnal «Tekhnolohycheskyy audyt y rezervy proyzvodstva»*. – Vol 2, No 2(22) (2015). – Kharkiv, 2015. – ISSN (print) 2226-3780, ISSN (on-line) 2312-8372. – S. 4-7.
- [31] Vysotska V. Designing features of architecture for e-commerce systems / Victoria Vysotska, Lyubomyr Chyrun // *MEST Journal (Management Education Science & Society Technology)*. – Vol.2 No.1. – P. 57-70.
- [32] Vysotska V. Set-theoretic models and unified methods of information resources processing in e-business systems / Victoria Vysotska, Lyubomyr Chyrun // *Applied Computer Science. ACS journal*. – Volume 10, Number 3. – Poland, 2014. – ISSN 2353-6977 (Online), ISSN 1895-3735 (Print). – P. 5-2.

Відомості про авторів

Русин Богдан Павлович – д.т.н., професор, завідувач відділу методів та систем дистанційного зондування Фізико-механічного інституту НАН України, вул. Наукова, 5, м. Львів, 79601, Україна.

Погрелюк Любомир Володимирович – аспірант відділу методів та систем дистанційного зондування Фізико-механічного інституту НАН України вул.Наукова,5,м.Львів,79601,Україна.

Висоцька Вікторія Анатоліївна – к.т.н., доцент, доцент кафедри «Інформаційні системи та мережі» Національного університету «Львівська політехніка», вул. Степана Бандери 12, м. Львів, 79013, Україна.

Осипов Михайло Михайлович – магістр кафедри «Інформаційні системи та мережі» Національного університету «Львівська політехніка», вул. Степана Бандери 12, м. Львів, 79013, Україна.

Варецький Ярема Юрійович – к.т.н.,ст.н.с. відділу методів та систем дистанційного зондування Фізико-механічного інституту НАН України,вул. Наукова,5,м. Львів, 79601, Україна.

Капший Олег Вірославович – к.т.н.,н.с. відділу методів та систем дистанційного зондування Фізико-механічного інституту НАН України, вул. Наукова,5,м. Львів, 79601, Україна.

B.P. Rusyn¹, L.V. Pohreliuk¹, V.A. Vysotska², M.M. Osypov²,
J.Y. Varetsky¹, OV Kapshiy¹

SYSTEM ARCHITECTURE OF DATA DEDUBLATION AND DISTRIBUTION IN CLOUD STORES DURING BACKUP

¹GV Physical and Mathematical Institute Karpenko NAS of Ukraine, Lviv

²NU Lviv Polytechnic, Department of Information Systems and Networks, Lviv

Б.П. Русын¹, Л.В. Погрелюк¹, В.А. Высоцька², М.М. Осыпов⁴,
Я.Ю. Варецький¹, А.В. Капший¹

АРХИТЕКТУРА СИСТЕМЫ ДЕДУБЛИКАЦИИ И РАСПРЕДЕЛЕНИЯ ДАННЫХ В ОБЛАЧНОЕ ХРАНИЛИЩЕ ПРИ РЕЗЕРВНОМ КОПИРОВАНИИ

¹Физико-математический институт имени Г.В. Карпенко НАН Украины, Львов

²НУ «Львовская политехника», кафедра «Информационные системы и сети», Львов

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОБЧИСЛЮВАЛЬНІ МЕТОДИ УДК 004.94

Т. Б. Мартинюк, О. І. Черняк, Б. І. Круківський, Мохамед Салем Нассер
Мохамед

ОБЧИСЛЮВАЛЬНА СКЛАДНІСТЬ МЕРЕЖЕВОЇ МОДЕЛІ СОРТУВАННЯ ЛІНІЙНОГО МАСИВУ ЧИСЕЛ

Вінницький національний технічний університет, м. Вінниця

Анотація. При розробці розвинутого програмного та апаратного забезпечення для сучасних обчислювальних засобів інтерес представляють удосконалені методи асоціативної обробки інформації, а саме процедури сортування і вибору. Це забезпечує реалізацію ефективного пошуку потрібної інформації в масивах даних. Необхідність паралельної необчислювальної обробки великих масивів інформації потребує відповідну організацію асоціативної пам'яті, а також розробку і використання відповідних перспективних технічних засобів. Сортування вважається важливою процедурою в таких прикладних областях, як рішення економічних задач, управління базами даних (СУБД), сортування IP адрес в комп'ютерних мережах, обробка сигналів і зображень (наприклад, при нелінійній медіанній фільтрації зображень). Аналіз відомих методів сортування показав, що найбільш ефективним методом паралельного сортування з урахуванням його апаратної реалізації сортуючою мережею є метод парного обміну. При цьому, ступінь паралелізму будь-якого методу сортування за його апаратної реалізації безпосередньо залежить від кількості схем порівняння, які спрацьовують паралельно при кожному перегляді. Для методу парного обміну ступінь паралелізму визначається величиною $\lfloor n/2 \rfloor$, де n - кількість вхідних числових величин або розмірність вхідного лінійного масиву чисел. У статті проаналізовано способи реалізації алгоритму сортування методом парного обміну з топологією зв'язків між елементами масиву чисел у вигляді «стрічки» і «кільця». Для прикладу описано паралельний алгоритм сортування методом парного обміну. Моделювання алгоритму виконано на мові високого рівня C++. Проаналізовано отримані статистичні та графічні результати моделювання. Аналіз графічних результатів моделювання свідчить про залежність виду $O(n)$ між кількістю циклів сортування і розмірністю n вхідного масиву. Це підтверджує ефективність апаратної реалізації сортування парним обміном на сортуючій мережі за рахунок регулярності структури і зв'язків в процесі сортування. Можливість статистично визначити не тільки кількість циклів сортування при заданій розмірності масиву чисел, але й відповідну кількість порівнянь і переміщень значно розширює можливості вдосконалення відомих і створення нових способів синхронного сортування елементів лінійного масиву апаратно у вигляді сортуючої мережі.

Ключові слова: обчислювальна складність, сортування, лінійний масив чисел.

Аннотация. При разработке развитого программного и аппаратного обеспечения для современных вычислительных средств интерес представляют усовершенствованные методы ассоциативной обработки информации, а именно процедуры сортировки и выбора. Это обеспечивает реализацию эффективного поиска нужной информации в массивах данных. Необходимость параллельной невычислительной обработки больших массивов информации требует соответствующую организацию ассоциативной памяти, а также разработку и использование соответствующих перспективных технических средств. Сортировка считается важной процедурой в таких прикладных областях, как решение экономических задач, управления базами данных (СУБД), сортировка IP адресов в компьютерных сетях, обработка сигналов и изображений (например, при нелинейной медианной фильтрации изображений). Анализ известных методов сортировки показал, что наиболее эффективным методом параллельной сортировки с учетом его аппаратной реализации сортирующей сетью является метод парного обмена. При этом, степень параллелизма любого метода сортировки при его аппаратной реализации напрямую зависит от количества схем сравнения, которые срабатывают параллельно при каждом просмотре. Для метода парного обмена степень параллелизма определяется величиной $\lfloor n/2 \rfloor$, где n - количество входных числовых величин или размерность входного линейного массива чисел. В статье проанализированы способы реализации алгоритма сортировки методом парного обмена с топологией связей между элементами массива чисел в виде «ленты» и «кольца». Для примера описан параллельный алгоритм сортировки методом парного обмена. Моделирование алгоритма выполнено на языке высокого уровня C++. Проанализированы полученные статистические и графические результаты моделирования. Анализ графических результатов моделирования свидетельствует о зависимости вида $O(n)$ между количеством циклов сортировки и размерностью n входного массива. Это подтверждает эффективность аппаратной реализации сортировки парным обменом на сортирующей сети за счет регулярности структуры и связей в процессе сортировки. Возможность статистически определить не только количество циклов сортировки при заданной размерности массива чисел, но и соответствующее количество сравнений и перестановок значительно расширяет возможности усовершенствования известных и создания новых способов синхронной сортировки элементов линейного массива апаратно в виде сортирующей сети.

Ключевые слова: вычислительная сложность, сортировка, линейный массив чисел

Abstract. In the development of advanced software and hardware for modern computing, the interest is the improvement of methods of associative processing of information such as procedures of sorting and selection. That ensures the realization of effective search for the required information in the data arrays. The need for parallel non-processing of large amounts of information entails the appropriate organization of associative memory, as well as the development and using of perspective technical devices. The sorting is important procedure in such application areas as solving economic problems, managing databases, sorting of IP addresses in computer networks, processing signals and images (for example, in nonlinear median image filtering). The analysis of known sorting methods have shown that the most effective method of parallel sorting, taking into account its hardware implementation by the sorting network, is the pairwise exchange. At the same time, the degree of parallelism of any sorting method for its hardware implementation directly depends on the number of comparison schemes that work in parallel in each view. For a pairwise exchange method, the degree of parallelism is determined by the value $\lfloor n/2 \rfloor$, where n is the number of input numerical values or the dimension of the input linear number array. In this article methods of implementing of the sorting algorithm by the method of pairwise exchange with the link topology between elements of the number array in the form of "tape" and "ring" are analyzed. For example, the parallel sorting algorithm using the pairwise exchange method is described. The simulation at a high-level C++ language is done. The obtained statistical and graphic results of modeling are analyzed. The analysis of graphical modeling results shows the dependence of the form $O(n)$ between the number of sort cycles and the dimensionality n of the input array. That confirms the effectiveness of the hardware implementation of sorting by pairwise exchange on the sorting network due to the regularity of the structure and connections in the sorting process. The ability to statistically determine not only the number of sorting cycles with a given dimension of

Т. Б. Мартинюк, О. І. Черняк, Б. І. Круківський, Мохамед Салем Нассер Мохамед, 2019

the number array, but also the corresponding number of comparisons and transposition greatly extends the possibilities of improving the known and creating new ways of synchronous sorting of elements of a linear array by hardware in the form of a sorting network.

Keywords: computational complexity, sorting, linear number array.

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-64-71>.

Вступ

При розробці розвиненого програмного та апаратного забезпечення для сучасних обчислювальних засобів інтерес представляє удосконалення методів асоціативної обробки інформації, основу якої складають такі процедури, як сортування і вибір, що забезпечує реалізацію швидкого пошуку потрібної інформації в значних масивах даних [1,2]. Це пов'язано з необхідністю паралельної необчислювальної обробки великих масивів інформації, що тягне за собою відповідну організацію асоціативної пам'яті, а також розробку і використання відповідних перспективних технічних засобів [3-5].

Актуальність тематики

Відомо, що сортування вважається важливою процедурою в таких прикладних областях, як рішення економічних задач, управління базами даних (СУБД), сортування IP адрес в комп'ютерних мережах, обробка сигналів і зображень (з використанням нелінійної медіанної фільтрації зображень) [6-9].

Аналіз відомих методів сортування показав, що найбільш ефективним методом паралельного сортування з урахуванням його апаратної реалізації на сортуючій мережі є метод попарного обміну [1,10]. При цьому, ступінь паралелізму будь-якого методу сортування при його апаратній реалізації безпосередньо залежить від кількості схем порівняння, які спрацьовують паралельно при кожному перегляді [1,4,10]. Для методу попарного обміну ступінь паралелізму визначається величиною $\lfloor n/2 \rfloor$, де n – кількість вхідних числових величин або розмірність вхідного лінійного масиву чисел [1,10].

Мета

Метою даної роботи є аналіз обчислювальної складності методу сортування попарним обміном в процесі комп'ютерного моделювання відповідного алгоритму сортування одновимірному масиву чисел.

Постановка задачі

Для оцінювання алгоритмів сортування найважливішим параметром вважається час сортування або обчислювальна складність, яка характеризує швидкість алгоритму [1,5,8]. Для процедури впорядкування масиву чисел (сортування) використовують найгіршу, середню і кращу поведінку (швидкодію) алгоритму в термінах розміру n списку. Відомо, що для типового алгоритму хороша поведінка – це залежність виду $O(n \log n)$ для його обчислювальної складності, погана – це $O(n^2)$, а ідеальна поведінка – це $O(n)$ [1,5,8,10]. Теоретична обчислювальна складність методу попарного обміну при його реалізації на сортуючій мережі дорівнює $O(n)$ [1,10].

В основі класичного методу попарного обміну лежить послідовність попарних переглядів, в яких елемент порівнюється зі своїм найближчим сусідом, а можливими переміщеннями (транспозиціями) є переміщення більших за значенням елементів на одну позицію вниз [10]. Сортування закінчується, коли при двох сусідніх переглядах не буде переміщень ні в одній парі елементів [10].

Для методу попарного обміну варіанти оптимізації можна отримати, по-перше, за рахунок створення нових зв'язків між крайніми елементами масиву (так зване замикання масиву в «кільце») в парних або непарних циклах. В результаті формується додаткова пара елементів, яка складається з елементів молодшої і старшої позиції в масиві. Так виконується перехід від сортуючої мережі вигляду «стрічка» до вигляду «кільце» [11]. По-друге, за рахунок введення додаткового елемента в масив з непарною кількістю чисел. Це дозволяє збільшити не тільки рівень паралелізму сортування, а й регулярність структури і зв'язків в пристроях сортування [11,12]. По-третє, за рахунок використання лише одного контрольного циклу замість двох на відміну від відомого алгоритму сортування цим методом [10], що дозволяє скоротити час сортування на цикл.

У статті [13] досліджено ефективність кільцевої сортуючої мережі в процесі комп'ютерного моделювання. Результати моделювання підтвердили, що створення зв'язку між крайніми вільними елементами масиву (формування «кільця») в непарних циклах сортування дозволяє істотно знизити кількість циклів у більшості випадків. При цьому мережева модель сортування попарним обміном дозволяє організувати формування пар елементів масиву в кожному циклі одночасно, що значно прискорює процес сортування. В результаті з урахуванням мінімізації кількості циклів обрано спосіб сортування з введенням мінімального (нульового) елемента в додаткову (старшу $(n+1)$ -у позицію) з утворенням «кільця» в непарних циклах сортування.

Апаратно організований паралельний метод сортування попарним обміном може бути реалізований як у вигляді сортуючої пам'яті [14], так і у вигляді сортуючої мережі [15]. Крім того, паралельний алгоритм сортування попарним обміном у вигляді синхронного алгоритму мультиобробки лінійного числового масиву описано в базисі систем алгоритмічних алгебр (САА) Глушкова [16,17], що є підставою для модифікованих варіантів його апаратної реалізації.

Особливості алгоритму сортування попарним обміном

У статті [16] приведено блок-схему алгоритму сортування за зростанням з паралельним способом формування пар елементів [15], в якому на відміну від алгоритму з послідовним формуванням пар елементів виконується порівняння всіх пар чисел одночасно.

Перед початком процесу сортування виконується визначення розмірності лінійного масиву чисел (парна чи непарна кількість елементів). Якщо кількість елементів n масиву чисел непарна, то додається мінімальне нульове число (\min) в максимальну $(n+1)$ -у позицію (\max). Потім обнуляються лічильники ЛчТ, ЛчЦ, де ЛчТ – лічильник транспозицій (переміщень), ЛчЦ – лічильник циклів.

Далі виконується формування всіх K пар чисел $a(2k-1)$, $a(2k)$ для першого (непарного) перегляду (циклу). Після цього всі елементи непарної $(2k-1)$ -ї позиції порівнюються зі своїм сусідом парної $2k$ -ї позиції і більший з них займає парну $2k$ -у позицію, а менший – непарну $(2k-1)$ -у позицію, причому $k = 1, 2, \dots, K$, $K = \lfloor n/2 \rfloor$, де K – кількість пар в масиві (ціла частина), n – розмірність масиву.

Облік переміщень (транспозицій) елементів в масиві відбувається таким чином: якщо елементи знаходяться в невідповідних позиціях у g парах, то вміст лічильника ЛчТ збільшується на величину g , де $g = 1, 2, 3, \dots, K$, і ці елементи міняються місцями в парах. Після цього перевіряється рівність нулю вмісту лічильника ЛчТ. Це перша умова, тобто умова відсутності переміщень (транспозицій) у всіх K парах елементів масиву. Якщо вона виконується, то перевіряється рівність вмісту лічильника ЛчЦ одиниці – це друга умова. Якщо друга умова виконується, то виконання алгоритму триває. Якщо не виконується перша умова, то збільшується вміст лічильника ЛчЦ на одиницю, обнуляється лічильник ЛчТ і перевіряється вміст лічильника ЛчЦ на парність. Це третя умова.

Якщо виконується друга і третя умови, то формуються всі K пари чисел $a(2k)$, $a(2k+1)$ і виконується їх порівняння, тобто починається другий (парний) перегляд (цикл). Транспозиція в кожній парі елементів виконується тільки для того, щоб більший з пари елементів зайняв непарну $(2k+1)$ -у позицію, а менший – парну $2k$ -у позицію. Знову перевіряється на рівність нулю вміст лічильника ЛчТ.

Таким чином, елементи з більшим значенням переміщуються в нижню частину списку. Якщо вміст лічильника ЛчЦ парний, то виконується перехід до непарного перегляду (циклу). Якщо перша умова виконується, а друга – не виконується, то алгоритм завершено. Інакше процес триває до тих пір, поки масив не буде відсортований.

Аналіз результатів комп'ютерного моделювання

В даній роботі виконано моделювання алгоритму сортування [16] на мові C++ [18-19] і отримано результати, що представлені на рис.1-5.

На рис.1 показано вікно вибору режиму сортування методом попарного обміну, на рис.2 показано вікно завантаження елементів масиву з файлу або генератора, а на рис.3 показано вікно вибору способу введення додаткового елемента в масив (підготовка масиву). При завантаженні елементів масиву непарної розмірності це вікно (рис.3) активне, а якщо масив парної розмірності, то це вікно буде неактивним.

Результати моделювання сортування попарним обміном на мові C++ у вигляді статистичних і графічних даних наведено на рис.4,5 відповідно. На рис.4 показано відсортований масив і виділено три параметри сортування (кількість порівнянь, переміщень (транспозицій), циклів) для відповідної розмірності лінійного масиву чисел. Аналіз отриманих графіків, які наведено на рис.5, показав, що кількість циклів сортування прямо пропорційна розмірності масиву чисел, а саме $O(n)$, що, в свою чергу, підтверджує той факт, що метод сортування попарним обміном є найбільш ефективним за швидкодією і відповідним для його реалізації апаратно, зокрема на сортуючій мережі [15].

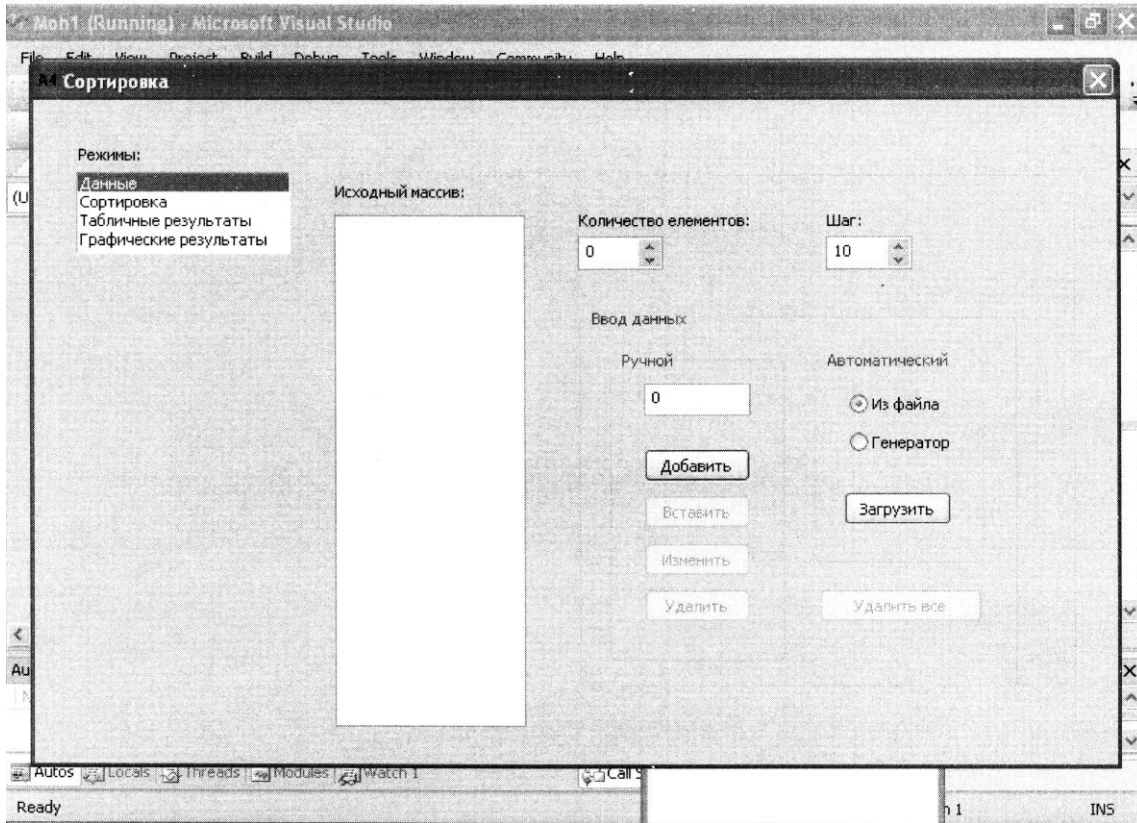


Рисунок 1 – Вікно вибору режиму для сортування методом попарного обміну

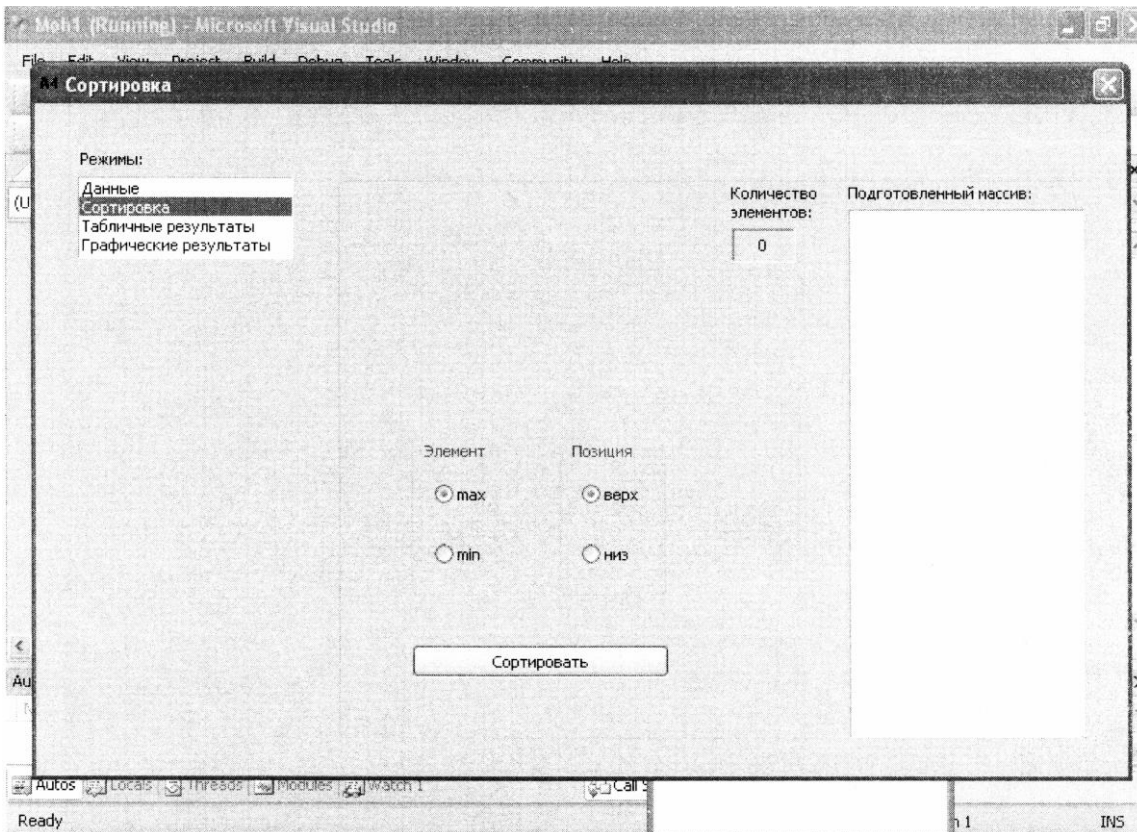


Рисунок 2 – Вікно для вибору додаткового елемента в масиві

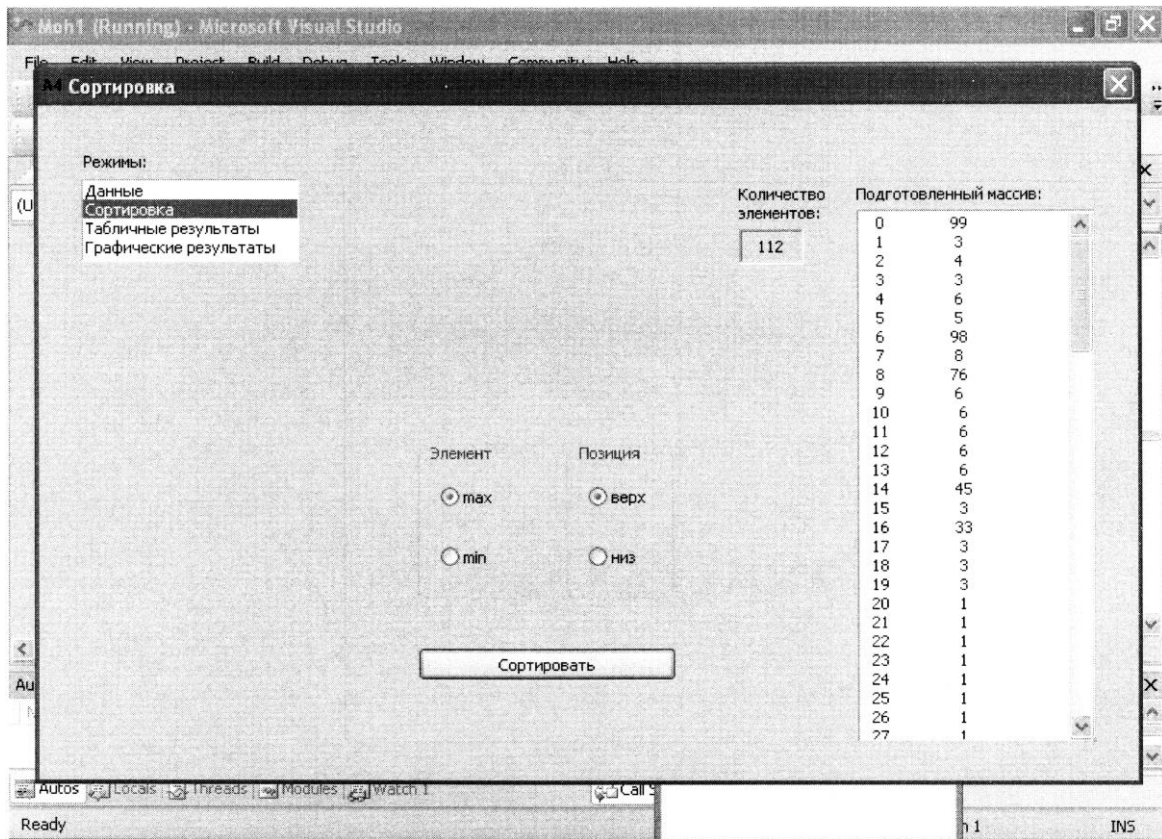


Рисунок 3 – Вікно вибору способу введення додаткового елемента в масиві (підготовка масиву)

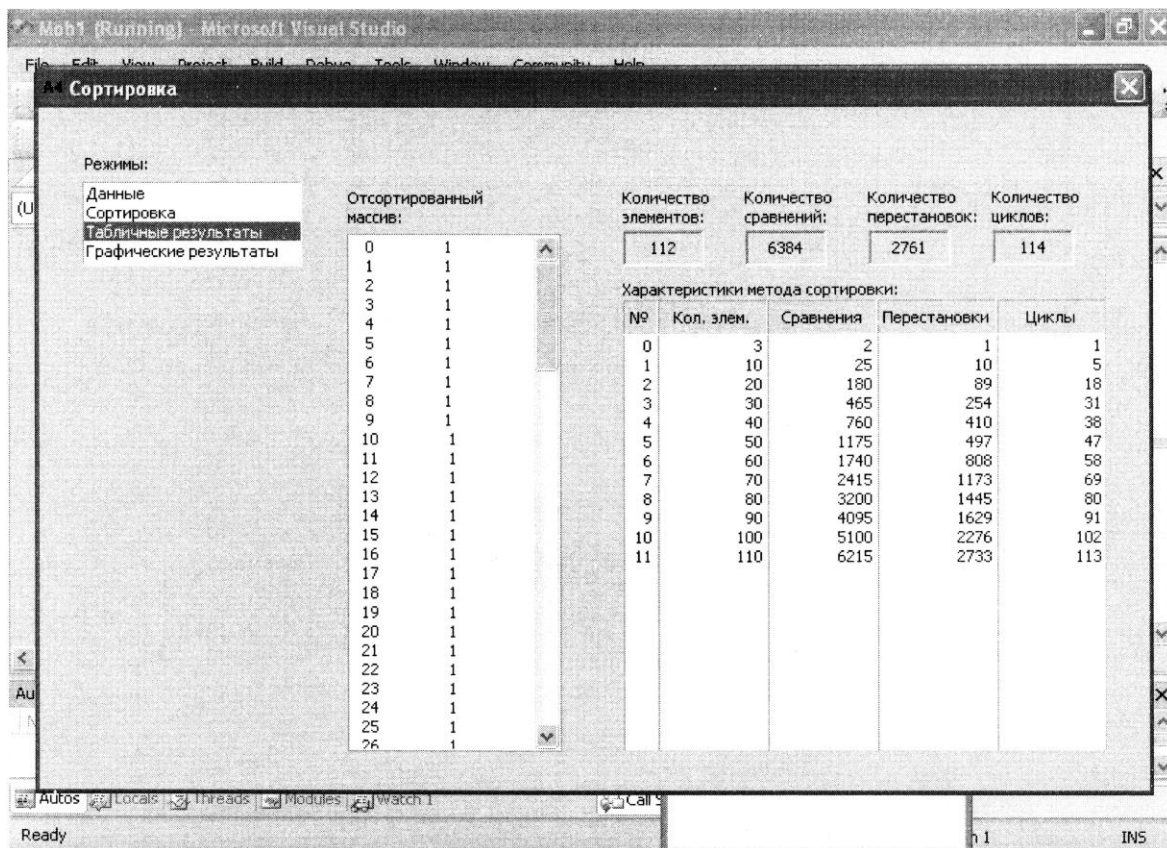


Рисунок 4 – Вікно з виводом табличних результатів

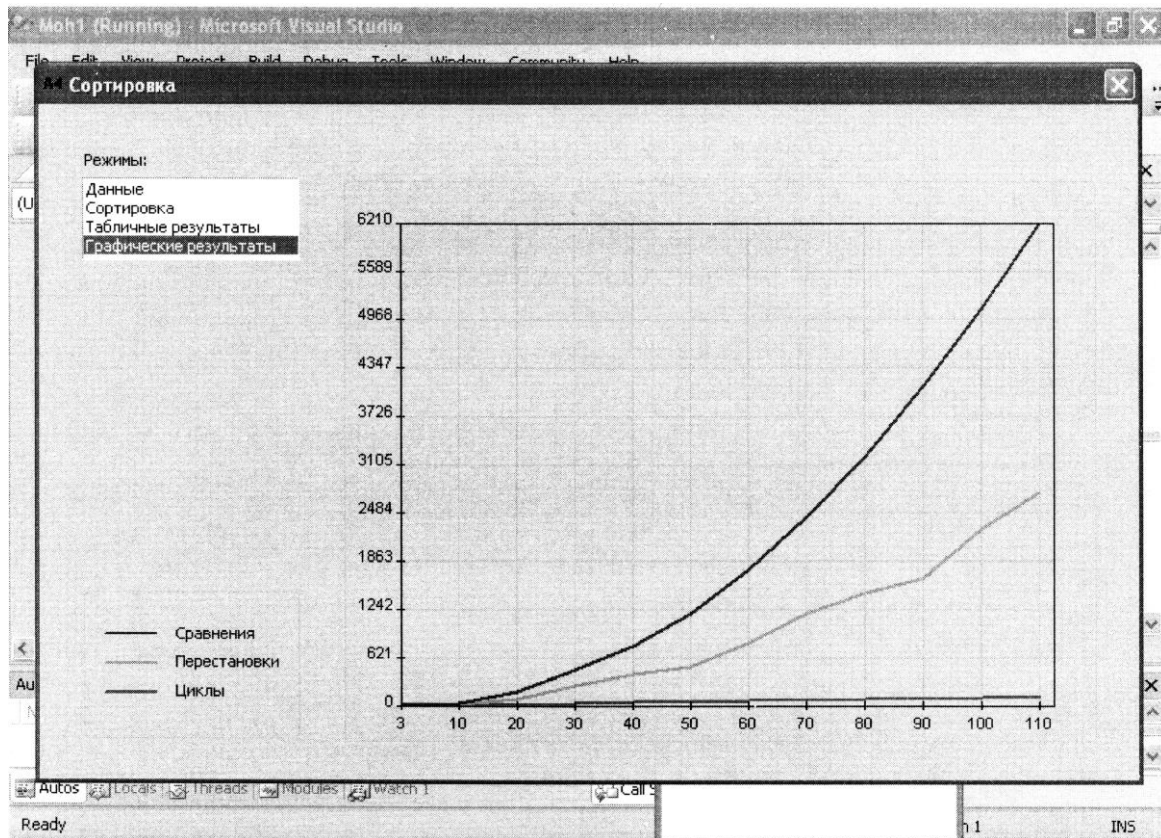


Рисунок 5 – Вікно з виводом графічних результатів

Висновки

1. Розроблені програми моделювання алгоритму сортування на мові високого рівня C++ дозволяють проаналізувати кількість циклів для методу попарного обміну в класичному (лінійному) варіанті і при додаванні min/max елемента в min/max позицію (кільцевому варіанті), а також представити результати у вигляді графіків. Це дозволяє в подальшому розрахувати швидкодню мережевої моделі сортування на конкретній елементній базі, наприклад, на ПЛІС.

2. Аналіз графічних результатів моделювання свідчить про залежність виду $O(n)$ між кількістю циклів сортування і розмірністю n вхідного масиву, а також, підтверджує ефективність апаратної реалізації сортування попарним обміном на сортуючій мережі за рахунок регулярності структури і зв'язків в процесі сортування.

3. Можливість статистично визначити не тільки кількість циклів сортування при заданій розмірності масиву чисел, але і відповідну кількість порівнянь і переміщень значно розширює можливості поліпшення відомих і створення нових способів синхронного сортування елементів лінійного масиву апаратно у вигляді сортуючої мережі.

Список літератури

- [1] Д.Э.Кнут, Искусство программирования. Т.3. Сортировка и поиск. М., Россия: Издательский дом «Вильямс», 2003.
- [2] Е.А.Яценко, «Регулярные схемы алгоритмов адресной сортировки и поиска», Управляющие системы и машины, №5, с.61-66. 2004.
- [3] Однокристалльный ассоциативный процессор САМ 2000, [Электронный ресурс]. Режим доступа: http://data.mf.grsu.by/citforum/htdocs/hardware/vich_sist/index.shtml Дата звернення: Травень. 15, 2019.
- [4] К.Дж.Тербер, Архитектура высокопроизводительных вычислительных систем. М., Россия: Гл.ред. физ-мат. Лит-ры, 1985.
- [5] В.П.Гергель, и Р.Г.Стронгин, Основы параллельных вычислений для многопроцессорных вычислительных систем. Нижний Новгород, Россия: Изд-во ННГУ им. Н.Лобачевского, 2003.

- [6] Е.Ф.Очин, Вычислительные системы обработки изображений. Л., Россия: Энергоатомиздат. Ленингр.отд-ние, 1989.
- [7] Хранение и сортировка адресов IP, [Электронный ресурс]. Режим доступа: <http://www.compodoc.ru/sort/index.html> Дата звернення: Травень. 17, 2019.
- [8] Алгоритмы. Сортировка и операции с упорядоченными последовательностями, [Электронный ресурс]. Режим доступа: <http://alglib.chat.ru/sort/index.html> Дата звернення: Травень. 10, 2019.
- [9] В.Р.Григорьев, «Нейросетевая организация алгоритмов сортировки на трехмерном оптическом нейрочипе», Автотметрия, №3, с.28-37. 1993.
- [10] Г.Лорин, Сортировка и системы сортировки. М.,Россия: Мир, 1983.
- [11] В.Мдовгань, В.А.Титенко, С.В.Выдрина, и Б.В.Клюйков, «Устройство для сортировки чисел», Патент России G06F7/08. №2246750 МКИ (2005), 20.02.2005.
- [12] Т.Б.Мартинюк, і В.В.Хом'юк, «Оцінювання ефективності алгоритмів мультиобробки масивів даних», Вісник Вінницького політехнічного інституту, №5, с.76-82. 2005.
- [13] Т.Б.Мартинюк, А.В.Кожем'яко, А.І.Колівошко, і О.В.Карась, «Дослідження ефективності кільцевої сортувальної мережі», Інформаційні технології та комп'ютерна інженерія, №1, с.68-71. 2015.
- [14] V.P.Kozhemyako, T.B.Martynyuk, R.A.Rasenko, and I.L.Pekhan, "Structure of Optoelectronic Sorting Memory", Оптико-електронні інформаційно-енергетичні технології, №1(3), с. 26-29. 2002.
- [15] Т.Б.Мартинюк, Мохамед Салем Нассер Мохамед, і В.В.Власійчук, «Модель сортувальної мережі», Інформаційні технології та комп'ютерна інженерія, №3, с.217-220. 2005.
- [16] Т.Б.Мартинюк, Л.В.Огороднійчук і Мохамед Салем Нассер Мохамед, «Оптимізований опис алгоритмів мультиобробки у базисі систем алгоритмічних алгебр Глушкова», Вісник Вінницького політехнічного інституту, №6, с.162-165. 2006.
- [17] В.П.Кожемяко, Т.Б.Мартынюк, и В.В.Хомюк, «Особенности структурного программирования синхронных алгоритмов», Кибернетика и системный анализ, №5, с.122-133. 2006.
- [18] Джесс. Либерти, С++ Энциклопедия. М.,Россия: Издательство ДиаСофт, 2000.
- [19] Р.Сэдживк, Фундаментальные алгоритмы на С++. Анализ/Структуры данных/Сортировка/Поиск. СПб.,Россия: ООО «ДиаСофтЮП», 2002.

References

- [1] D.E.Knut, Iskusstvo programmirovaniya. T.3. Sortirovka i poisk. M., Rossiya: Izdatelskiy dom «Vilyams», 2003.
- [2] Е.А.Yatsenko, «Regulyarnyie shemyi algoritmov adresnoy sortirovki i poiska», Upravlyayuschie sistemy i mashinyi, #5, s.61-66. 2004.
- [3] Odnokristalnyiy assotsiativnyiy protsessor SAM 2000, [Elektronniy resurs]. Rezhim dostupu: http://data.mf.grsu.by/citforum/htdocs/hardware/vich_sist/index.shtml Data zvernennya: Traven. 15, 2019.
- [4] K.Dzh.Terber, Arhitektura vyisokoproizvoditelnyih vyichislitelnyih sistem. M., Rossiya: Gl.red. fiz-mat. Lit-ryi, 1985.
- [5] V.P.Gergel, i R.G.Strongin, Osnovy parallelnyih vyichisleniy dlya mnogoprotsessornyih vyichislitelnyih sistem. Nizhniy Novgorod, Rossiya: Izd-vo NNGU im. N.Lobachevskogo, 2003.
- [6] Е.Ф.Очин, Vyichislitelnyie sistemy obrabotki izobrazheniy. L., Rossiya: Energoatomizdat. Le-ningr.otd-nie, 1989.
- [7] Hranenie i sortirovka adresov IP, [Elektronniy resurs]. Rezhim dostupu: <http://www.compodoc.ru/sort/index.html> Data zvernennya: Traven. 17, 2019.
- [8] Algoritmyi. Sortirovka i operatsii s uporyadochennyimi posledovatelnostyami, [Elektronniy re-surs]. Rezhim dostupu: <http://alglib.chat.ru/sort/index.html> Data zvernennya: Traven. 10, 2019.
- [9] V.R.Grigorev, «Neyrosetevaya organizatsiya algoritmov sortirovki na trehmernom opticheskom neyrochipe», Avtometriya, #3, s.28-37. 1993.
- [10] G.Lorin, Sortirovka i sistemyi sortirovki. M.,Rossiya: Mir, 1983.
- [11] V.Mdovgan, V.A.Titenko, S.V.Vyidrina, i B.V.Klyuykov, «Ustroystvo dlya sortirovki chisel», Patent Rossii G06F7/08. #2246750 MКИ (2005), 20.02.2005.
- [12] T.B.Martynyuk, I V.V.Hom'yuk, «OtsInyuvannya effektivnostI algoritmIv multiobrobki masivIv danih», VIsnik VInnitskogo polItehnIchogo Institutu, #5, s.76-82. 2005.
- [13] T.B.Martynyuk, A.V.Kozhem'yako, A.I.Kollvoshko, I O.V.Karas, «DosIldzhennya effektivnostI kIlItsevoYi sortuvalnoYi merezhI», InformatsIynI tehnologIYiI ta komp'yuterna InzhenerIya, #1, s.68-71. 2015.
- [14] V.P.Kozhemyako, T.B.Martynyuk, R.A.Rasenko, and I.L.Pekhan, "Structure of Optoelectronic Sorting Memory", Optiko-elektronnl InformatsIyno-energetichnl tehnologIYi, #1(3), s. 26-29. 2002.

- [15] Т.В.Мартинюк, Mohamed Salem Nasser Mohamed, I V.V.Vlaslychuk, «Model sortuvalnoYi merezhl», InformatsIynI tehnologIYiYi ta komp'yuterna InzhenerIya, #3, s.217-220. 2005.
- [16] Т.В.Мартинюк, L.V.Ogorodnychuk I Mohamed Salem Nasser Mohamed, «OptimIzovaniy opis algoritmIv multiobrobki u bazisI sistem algoritmIchnih algebr Glushkova», VIsnik VInnitskogo po-ItehnIchnogo Institutu, #6, s.162-165. 2006.
- [17] V.P.Kozhemyako, T.B.Martyinyuk, i V.V.Homyuk, «Osobennosti strukturnogo programmirovaniya si-nhronnyih algoritmov», Kibernetika i sistemnyi analiz, #5, s.122-133. 2006.
- [18] Dzhess. Liberti, S Entsiklopediya. M.,Rossiya: Izdatelstvo DiaSoft, 2000.
- [19] R.Sedzhvik, Fundamentalnyie algoritmyi na S . Analiz/Strukturyi dannyih/Sortirovka/Poisk. SPb.,Rossiya: OOO «DiaSoftYuP», 2002.

Мартинюк Тетяна Борисівна - д.т.н., професор кафедри обчислювальної техніки, Вінницький національний технічний університет;

Черняк Олександр Іванович - к.т.н., доцент кафедри обчислювальної техніки, Вінницький національний технічний університет;

Круківський Богдан Ігорович - магістр факультету комп'ютерних систем і автоматики, Вінницький національний технічний університет.

Мохамед Салем Нассер Мохамед - к.т.н., викладач Аденського університету, Ємен.

T. B. Martyniuk, O. I. Chernyak, B. I. Krukivskyi, Mohamed Salem
Nasser Mohamed

COMPUTATIONAL COMPLEXITY OF THE NETWORK MODEL OF SORTING OF A LINEAR NUMBER ARRAY

Vinnitsia National Technical University, Vinnitsa

T. B. Мартынюк, А. И. Черняк, Б. И. Круковский, Мохамед Салем
Нассер Мохамед

ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ СЕТЕВОЙ МОДЕЛИ СОРТИРОВКИ ЛИНЕЙНОГО МАССИВА ЧИСЕЛ

Винницкий национальный технический университет, г.Винница

ДО ВІДОМА АВТОРІВ

Найновіші правила оформлення і подання статей знаходяться на сайті журналу

<http://itce.vntu.edu.ua/>