

Міністерство освіти і науки України
Вінницький національний технічний університет

В. А. Лужецький, Л. А. Савицька

**МЕТОДИ ТА ЗАСОБИ
АДАПТИВНОГО УЩІЛЬНЕННЯ ДАНИХ
НА ОСНОВІ ЛІНІЙНОЇ ФОРМИ ФІБОНАЧЧІ**

Монографія

Вінниця
ВНТУ
2016

УДК 004.627:004.31

ББК 32.97

C13

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 9 від 25.02.2016 р.)

Рецензенти:

С. Г. Удовенко, доктор технічних наук, професор ХНУРЕ

О. А. Борисенко, доктор технічних наук, професор СумДУ

Лужецький, В. А.

C13 Методи та засоби адаптивного ущільнення даних на основі лінійної форми Фібоначчі : монографія / В. А. Лужецький, Л. А. Савицька, за заг. ред. В. А. Лужецького – Вінниця : ВНТУ, 2016. – 100 с.

ISBN 978-966-641-668-4

Розглянуто теоретичні та практичні питання створення принципово нових програмних і апаратних засобів адаптивного ущільнення даних на основі лінійної форми Фібоначчі. Наведено узагальнену модель адаптивного ущільнення даних, числові моделі джерел даних, методи кодування та декодування на основі лінійної форми Фібоначчі, а також результати експериментальних досліджень адаптивних методів ущільнення. Розрахована на наукових та інженерно-технічних працівників, які займаються питаннями створення компресорів і архіваторів даних, а також може бути корисною студентам і аспірантам.

УДК 004.627:004.31

ББК 32.97

ISBN 978-966-641-668-4

© В. Лужецький, Л. Савицька, 2016

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 МОДЕЛІ АДАПТИВНОГО УЩІЛЬНЕННЯ ДАНИХ	7
1.1 Підходи до ущільнення інформації	7
1.2 Узагальнена модель адаптивного ущільнення даних.....	9
1.3 Розробка правил моделювання джерела даних.....	10
1.4 Представлення цілих чисел у вигляді лінійної форми Фібоначчі	15
1.5 Розробка правил кодування та декодування блоків даних.....	17
РОЗДІЛ 2 МЕТОДИ АДАПТИВНОГО УЩІЛЬНЕННЯ ДАНИХ НА ОСНОВІ ЛІНІЙНОЇ ФОРМИ ФІБОНАЧЧІ	27
2.1 Метод ущільнення даних одним проходом з рівномірним розбиттям на блоки	27
2.2 Метод ущільнення даних фіксованою кількістю проходів з рівномірним розбиттям на блоки.....	33
2.3 Метод ущільнення даних змінною кількістю проходів з рівномірним розбиттям на блоки.....	36
2.4 Метод ущільнення даних одним проходом з нерівномірним розбиттям на блоки	41
2.5 Оцінки складності програмної реалізації методів ущільнення	44
РОЗДІЛ 3 СТРУКТУРА СПЕЦІАЛІЗОВАНОГО ПРОЦЕСОРА ДЛЯ УЩІЛЬНЕННЯ ДАНИХ	50
3.1 Узагальнена структура спеціалізованого процесора для ущільнення даних.....	50
3.2 Структура модуля моделювання джерела даних	55
3.3 Структура модуля кодування	60
3.4 Структура модуля декодування	62
3.5 Структура модуля оптимізації.....	68
3.6 Модуль формування структури послідовності	70
РОЗДІЛ 4 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ МЕТОДІВ УЩІЛЬНЕННЯ ДАНИХ НА ОСНОВІ ЛІНІЙНОЇ ФОРМИ ФІБОНАЧЧІ	76
4.1 Структури програмних засобів для адаптивного ущільнення даних на основі лінійної форми Фібоначчі.....	76
4.2 Результати дослідження впливу довжини блоків даних на коефіцієнт ущільнення	78
4.3 Результати дослідження методів адаптивного ущільнення ...	83
4.4 Порівняльний аналіз методів ущільнення даних.....	87
ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- RLE – Run Length Encoding (кодування серій послідовностей)
LZ77 – алгоритм ущільнення Лемпела–Зіва (1977 рік)
LZ78 – алгоритм ущільнення Лемпела–Зіва (1978 рік)
LZW – алгоритм ущільнення Лемпела–Зіва–Велча
PPM – Prediction by Partial Matching (прогнозування за частковим збігом)
DMC – Dynamic Markov Compression (динамічне ущільнення Маркова)
MFT – Move to Front (рухай вперед)
BWT – Burrows – Wheeler Transform (перетворення Барроуза–Вілера)
ЛФФ – лінійна форма Фібоначчі
СЛФФ – скорочена лінійна форма Фібоначчі
ОПРР – один прохід з рівномірним розбиттям
ФКПРР – фіксована кількість проходів з рівномірним розбиттям
ЗКПРР – змінна кількість проходів з рівномірним розбиттям
ОПНР – один прохід з нерівномірним розбиттям

ВСТУП

Обсяги інформації, яка зберігається і передається в сучасних комп'ютерних системах, з кожним роком значно зростають, що вимагає збільшення обсягів пам'яті і наявності високошвидкісних каналів передавання. Це призводить до постійного збільшення вартості таких систем. Тому економічно більш вигідним є зменшення обсягів інформації шляхом використання методів ущільнення.

У комп'ютерах і комп'ютерних системах ущільнення використовується для зменшення розмірів файлів при підготовці їх до передавання каналами зв'язку або до транспортування на зовнішніх носіях малої ємності, як засіб зменшення обсягу даних, що підлягають резервному копіюванню, для підвищення ефективності використання дискової пам'яті.

На цей час відома велика кількість методів ущільнення інформації. Одні з них мають тільки теоретичне значення, інші знайшли практичне застосування у вигляді компресорів й архіваторів.

Основи теорії ущільнення інформації були створені такими вченими: С. Е. Shannon [1], Р. М. Fano [2], Д. А. Huffman [3], J. Ziv, А. Lempel [4, 5], J. J. Rissanen [6, 7], Т. А. Welch [8], J. A. Storer, Т. G. Szymanski [9], S. W. Golomb [10], Р. Elias [11, 12], R. N. Horspool, G. V. Cormack [13, 14], I. Witten, R. Neal, J. Cleary [15], M. Burrows, D. J. Wheeler [16], Б. Я. Рябко [17].

Питання розвитку теорії та практики ущільнення інформації розглядаються в роботах Т. С. Bell [18–20], J. L. Bentley [21, 22], R. P. Brent [23], M. Jakobsson [24], G. G. Langdon [25–27], D. A. Lelewer [28], А. М. Moffat [29–31], О. А. Борисенка [32–34], А. В. Кадача [35], В. П. Майданюка [36], У. В. Поліщук [37], Г. А. Полякова [38], О. А. Ратушняка [39], О. І. Ситняковської [40], С. Г. Удовенка [41, 42], Д. А. Шкарина [43] та ін.

Існує два підходи до ущільнення інформації [44–48]. Перший підхід застосовується до інформації, що повинна зберігатися і передаватися в незмінному вигляді (тексти, складені природними мовами, мовами програмування та ін.). Другий підхід застосовується до інформації, що сприймається людиною (звук і зображення), і спрямований на ущільнення з втратами деякої частини інформації.

Теоретичні дослідження і практика застосування архіваторів показали, що не існує універсального методу ущільнення, який забезпечував би однаковий степінь ущільнення для різних типів даних. Врахо-

вуючи це, останнім часом прагнуть до створення адаптивних алгоритмів ущільнення певних типів даних. Причому степінь ущільнення збільшується значно, коли використовується комбінація декількох різних алгоритмів.

В основі існуючих методів ущільнення лежать два досить схожі підходи до опису властивостей джерел інформації: ймовірнісний і комбінаторний. Хоча вони і мають багато спільного, але існують незалежно один від одного. Перевага такого стану полягає в можливості вибору підходу з урахуванням властивостей конкретного джерела інформації. Однак на прикладі зростаючої послідовності натуральних чисел «1, 2, 3,...» легко переконатися в неспроможності здійснити ущільнення з використанням як ймовірнісного, так і комбінаторного підходів. В основі цих підходів орієнтація на повторюваність подій, а не на функціональну залежність, яка характерна для даного випадку. З цього прикладу випливає, що для моделювання подібних ситуацій потрібен альтернативний підхід до опису властивостей оброблюваної інформації. В цьому підході повинні враховуватися не тільки ймовірнісні й комбінаторні закономірності, але й функціональні закономірності [47].

У роботах [49–51] запропоновано оригінальний спосіб представлення цілих чисел у вигляді, так званої, лінійної форми Фібоначчі, яка забезпечує скорочення розрядності представлення великих чисел (256 двійкових розрядів і більше). Така форма представлення є основою нового підходу щодо ущільнення даних, який ґрунтується на оптимізуючих властивостях чисел Фібоначчі [52]. При цьому для ущільнення інформації блок цифрових даних розглядається як надвелике ціле додатне число, що представляється набором з трьох невеликих чисел. Однак відсутні дослідження цього підходу стосовно ефективності ущільнення певних типів даних і можливостей адаптації до властивостей джерел інформації. Тому метою даної роботи є створення моделей і методів адаптивного ущільнення даних на основі лінійної форми Фібоначчі та засобів, що реалізують їх.

Розділи 1, 2, 4 написано спільно В. А. Лужецьким і Л. А. Савицькою, а розділ 3 – Савицькою Л. А. Основні результати, викладені у розділах 1, 2, отримані під час дисертаційного дослідження Л. А. Савицькою.

Автори висловлюють вдячність рецензентам д. т. н., професору С. Г. Удовенку, д. т. н., професору О. А. Борисенку за поради, які сприяли покращенню отриманих результатів та їх подання.

РОЗДІЛ 1

МОДЕЛІ АДАПТИВНОГО УЩІЛЬНЕННЯ ДАНИХ

У цьому розділі розглядається узагальнена модель адаптивного ущільнення даних, яка передбачає використання множини правил моделювання джерела даних, множини правил кодування та функції оптимізації, що забезпечує вибір з множини кодованих послідовностей єдиної послідовності, яка відповідає найбільшому коефіцієнту ущільнення. Розглянуто особливості формування числових моделей джерела даних та особливості кодування даних з використанням лінійної форми Фібоначчі.

1.1 Підходи до ущільнення інформації

Існує кілька різних підходів до вирішення задачі ущільнення інформації. Одні мають досить складну теоретичну математичну базу, інші засновані на властивостях інформаційного потоку та алгоритмічно досить прості. Будь-який підхід, метод чи алгоритм, який реалізує ущільнення або компресію даних, призначені для зменшення обсягу вихідного потоку інформації в бітах за допомогою її оборотного або необоротного перетворення. Тому, перш за все, за критерієм, пов'язаним з характером або форматом даних, усі процеси ущільнення розділені на дві категорії: оборотне і необоротне ущільнення.

Під необоротним ущільненням розуміють [44–47] таке перетворення вхідного потоку даних, при якому вихідний потік, заснований на певному форматі інформації, досить схожий за зовнішніми характеристиками на вхідний потік (об'єкт), однак відрізняється від нього обсягом. Степінь подібності вхідного і вихідного потоків визначається ступенем відповідності деяких властивостей об'єкта (тобто ущільненої і неущільненої інформації, відповідно до певного формату даних), який представлений цим потоком інформації. Такі підходи і алгоритми використовуються для ущільнення, наприклад, даних растрових графічних файлів з низьким ступенем повторюваності байтів у потоці.

Часто необоротне ущільнення називають ущільненням з втратами. Необоротне ущільнення неможливо застосовувати в галузях, в яких необхідно мати точну відповідність інформаційної структури вхідного та вихідного потоків.

Оборотне ущільнення завжди приводить до зменшення обсягу вихідного потоку інформації без зміни його інформативності, тобто без втрати інформаційної структури. Більш того, з вихідного потоку за допомогою відновлювального алгоритму можна отримати вхідний. Процес відновлення називається декомпресією або розпаковуванням. Тільки після процесу розпакування дані придатні для обробки відповідно до їхнього внутрішнього формату. Оборотно ущільнення часто називають ущільненням без втрат [44–47].

Відомо три базові стратегії ущільнення [39].

1. **Трансформація потоку.** Опис даних, що надходять через уже оброблені. Результатом трансформації може бути кілька сформованих потоків. Навіть якщо сумарний обсяг потоків збільшується, їхня структура поліпшується і наступне ущільнення можна здійснити простіше, швидше й якісніше.

2. **Статистична стратегія:**

а) адаптивна (потокова). Обчислення ймовірностей для даних, що надходять, на підставі статистики вже оброблених даних. Кодування з використанням цих обчислених ймовірностей;

б) блокова. Окремо кодується й додається до ущільненого блоку його статистика.

3. **Трансформація блоку.** Вхідні дані розбиваються на блоки, які потім трансформуються цілком. Результатом трансформації можуть бути кілька блоків, а не один. Навіть якщо сумарна довжина блоків не зменшується, їхня структура значно поліпшується, і наступне ущільнення може відбуватися простіше, швидше й якісніше.

Одним з найважливіших досягнень у теорії ущільнення є вперше висловлена в [6] ідея поділу процесу ущільнення на моделювання (modeling) та кодування (encoding).

Моделювання забезпечує прогнозування ймовірності настання можливих подій, кодування забезпечує подання події у вигляді $-\log_2 p$ біт, де p – ймовірність настання події. Під кодуванням розуміють [44–47] обробку потоку символів у деякому алфавіті. Метою кодування є перетворення цього потоку в потік біт мінімальної довжини, яка досягається зменшенням ентропії вхідного потоку шляхом врахування частот появи символів. Довжина коду, який представляє символи з алфавіту потоку повинна бути пропорційна обсягу інформації вхідного потоку, а

довжина символів потоку в бітах може бути змінною. Завдання моделювання, як правило, більш складне. Це обумовлено високою складністю сучасних моделей даних. Разом з тим, кодування не є серйозною проблемою. Існує велика кількість кодерів, що різняться ступенем ущільнення і швидкодією. Як правило, у системах ущільнення використований кодер при необхідності може бути легко замінений іншим.

1.2 Узагальнена модель адаптивного ущільнення даних

Враховуючи те, що процес ущільнення поділяється на моделювання і кодування і, базуючись на стратегії ущільнення, що передбачає трансформацію блоку, автори пропонують таку узагальнену модель процесу адаптивного ущільнення даних:

$$C_A = \{P, A, M, C, P_M, P_C, P^*, S, f\},$$

де P – вихідна послідовність символів алфавіту $A = \{0,1\}$; $M = \{M_i\}$ – множина правил моделювання джерела даних; $C = \{C_j\}$ – множина правил кодування даних; $P_M = \{P_{M_i}\}$ – множина послідовностей, що є результатом моделювання; $P_C = \{P_{C_{ij}}\}$ – множина послідовностей, що є результатом кодування; P^* – послідовність ущільнених даних; S – правило формування структури послідовності; f – функція оптимізації.

Вихідна послідовність символів P перетворюється в множину послідовностей P_M з використанням множини правил моделювання M . Це описується відображенням $M: P \rightarrow P_M$. Правилу моделювання M_i відповідає відображення $P \xrightarrow{M_i} P_{M_i}$.

Кожна з послідовностей множини P_M кодується з використанням множини правил кодування $C: P_M \rightarrow P_C$, в результаті чого формується множина кодованих послідовностей P_C . Правилу кодування C_j відповідає відображення $P_{M_i} \xrightarrow{C_j} P_{C_{ij}}$.

Вибір із множини кодованих послідовностей P_C єдиної послідовності P^* , яка відповідає найбільшому коефіцієнту ущільнення, здійснюється за правилом, що описується функцією оптимізації $f: P_C \rightarrow P^*$.

Формування структури послідовності \mathbf{P}^* з вказівкою додаткової інформації, необхідної для відновлення послідовності \mathbf{P} , здійснюється на підставі правила S .

Таким чином, процес адаптивного ущільнення даних описується композицією відображень

$$C_A = M \circ C \circ f \circ S.$$

Пропонується така узагальнена модель процесу відновлення ущільнених даних:

$$DC_A = \{\mathbf{A}, \mathbf{P}^*, \mathbf{P}, \mathbf{D}, S\},$$

де $\mathbf{A} = \{0, 1\}$ – алфавіт; \mathbf{P}^* – послідовність ущільнених даних; \mathbf{P} – послідовність відновлених (вихідних) даних; $\mathbf{D} = \{D_j\}$ – множина правил декодування даних; S – правило формування структури послідовності \mathbf{P}^* .

Згідно з цією моделлю послідовність ущільнених даних \mathbf{P}^* розбивається на блоки за правилом S і кожен блок перетворюється за відповідним йому правилом декодування даних.

Узагальнена модель адаптивного процесу ущільнення даних передбачає використання множини правил моделювання джерела даних, множини правил кодування даних і функції оптимізації (адаптації). Така модель забезпечує опис з єдиних позицій існуючих методів ущільнення, оскільки на практиці адаптація процесу ущільнення до даних, що підлягають ущільненню, здійснюється перебором моделей джерела даних серед відомих моделей і алгоритмів кодування серед відомих алгоритмів для одержання найбільшого коефіцієнта ущільнення.

Основна перевага запропонованої узагальненої моделі адаптивного ущільнення даних полягає в тому, що формалізується підхід щодо розробки нових методів ущільнення. Це показано на прикладі формулювання нових правил моделювання джерела даних, правил кодування даних на основі лінійної форми Фібоначчі та опису функцій оптимізації.

1.3 Розробка правил моделювання джерела даних

Дані, що підлягають ущільненню, будемо розглядати як послідовність символів 0 і 1. Для подальших досліджень пропонується використовувати числову модель джерела даних. Відповідно до цієї моделі послідовність символів розбивається на блоки, що містять деяку кіль-

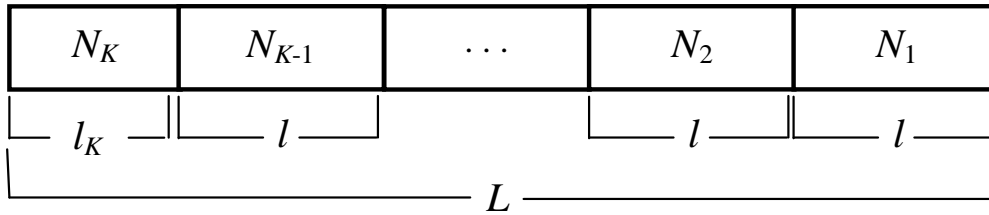


Рисунок 1.1 – Структурна модель вихідних даних з рівномірним розбиттям

Означення 1.2. Розбиття вихідної послідовності даних на блоки, що містять різну кількість символів, називається *нерівномірним розбиттям*.

У випадку нерівномірного розбиття маємо структурну модель даних, наведену на рис. 1.2.

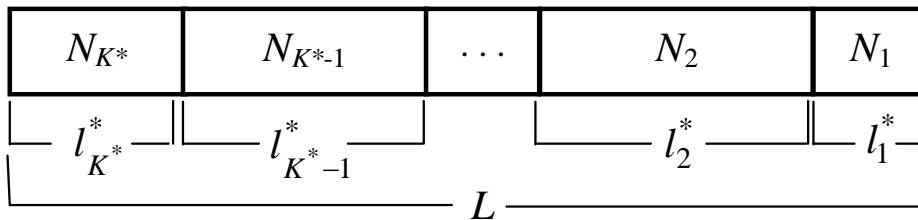


Рисунок 1.2 – Структурна модель вихідних даних з нерівномірним розбиттям

Довжина блоку змінюється від l_{\min} до l_{\max} і попередньо не задається. Для блоків, довжини яких мають значення

$$l, (l + 1 \cdot \Delta l), (l + 2 \cdot \Delta l), \dots, (l + g \cdot \Delta l),$$

l_{\min} l_{\max}

виконується ущільнення і вибирається та довжина, при якій забезпечується найбільший коефіцієнт ущільнення, тобто вона є результатом оптимального вибору серед різних значень довжин блоку.

Мінімальне значення l вибирається, виходячи з необхідного мінімального коефіцієнта ущільнення. Значення Δl повинне задовольняти умову $\Delta l \leq l$. Чим менше значення Δl , тим менша дискретність довжини і тим більший коефіцієнт ущільнення може бути отриманий. Але при цьому буде значно збільшуватися час ущільнення. Значення g вибирається з урахуванням умови, що довжина блоку $l_{\max} = (l + g \cdot \Delta l)$ задовольняє вимоги допустимих апаратних і часових витрат.

Для практичної реалізації зручним є представлення числа сукупністю байтів. При цьому кожен байт – це відповідний ASCII код, числовий еквівалент якого визначається за формулою:

$$s = \sum_{i=0}^7 a_i 2^i,$$

де a_i – цифра i -го розряду ASCII коду.

Виходячи з цього, блок даних довжини l байтів представляється як послідовність l чисел

$$s_0, s_1, s_2, \dots, s_{l-1}, \quad (1.1)$$

де s_j – числовий еквівалент ASCII коду j -го байту ($j = 0, 1, 2, \dots, l-1$).

Оскільки числові еквіваленти ASCII кодів належать діапазону чисел від 0 до 255, то число, що відповідає блоку даних, може бути обчислено за формулами

$$N_m = \sum_{j=0}^{l-1} s_j 256^j; \quad (1.2)$$

$$N_c = \sum_{j=0}^{l-1} s_{(l-j-1)} 256^{(l-j-1)}. \quad (1.3)$$

Відмінність цих формул полягає тільки в порядку використання елементів послідовності (1.1) відносно їх номерів: від молодшого до старшого та навпаки. Тут число 256 є основою системи числення і для кожного блоку даних вона незмінна.

Аналіз змісту блоків ущільнених даних стосовно числових еквівалентів ASCII кодів показав, що не в кожному блоці є байт з числовим еквівалентом 255. Тому пропонується визначати числовий еквівалент кожного блоку даних, виходячи з максимального значення $S_{\max} = \max(s_0, s_1, s_2, \dots, s_{l-1})$ числових еквівалентів ASCII кодів байтів, що

складають цей блок. При цьому числовий еквівалент визначається через представлення в системі числення з основою B за формулами

$$N_M = \sum_{j=0}^{l-1} s_j B^j; \quad (1.4)$$

$$N_C = \sum_{j=0}^{l-1} s_{(l-j-1)} B^{(l-j-1)}. \quad (1.5)$$

Число B для кожного блоку даних змінюється від $(S_{\max} + 1)$ до 256.

Покажемо це на прикладі. Нехай блок даних складається з байтів з такими числовими еквівалентами: 117; 45; 67; 136. Тут найбільшим є число 136. Отже, можна визначити числовий еквівалент блоку, використовуючи представлення в системах числення з основами від 137 до 256. Тобто для такого блоку можна отримати 120 різних числових еквівалентів.

Крім цього, є можливість ще збільшити кількість числових еквівалентів. Пропонується таке. Спочатку серед значень числових еквівалентів байтів визначається мінімальне значення $S_{\min} = \min(s_0, s_1, s_2, \dots, s_{l-1})$, а потім обчислюється числовий еквівалент блоку, за однією з формул

$$N_M = \sum_{j=0}^{l-1} d_j B^j \quad (1.6)$$

або

$$N_C = \sum_{j=0}^{l-1} d_{(l-j-1)} B^{(l-j-1)}, \quad (1.7)$$

де $d_j = s_j - S_{\min}$.

У цьому випадку число B для кожного блоку даних може змінюватися від $(S_{\max} - S_{\min} + 1)$ до 256.

Для наведеного вище прикладу кількість числових еквівалентів збільшується з 120 до 165.

Запропоновані моделі джерела даних характеризуються такими параметрами:

- довжина блоку (l_{const} – незмінна, l_{var} – змінна);
- основа системи числення (B_{const} – незмінна, B_{var} – змінна);
- значення числових еквівалентів байтів (S_0 – незмінні, S_{min} – зменшені);
- порядок використання елементів послідовності (1.1) (T_a – номери від молодшого до старшого, T_s – номери від старшого до молодшого).

З урахуванням наведених параметрів та можливості їхнього комбінування маємо таку множину правил моделювання джерела даних:

$$\begin{aligned}
 &M(l_{\text{const}}, B_{\text{const}}, S_0, T_a); \quad M(l_{\text{const}}, B_{\text{var}}, S_{\text{min}}, T_a); \quad M(l_{\text{var}}, B_{\text{const}}, S_{\text{min}}, T_s); \\
 &M(l_{\text{const}}, B_{\text{const}}, S_0, T_s); \quad M(l_{\text{const}}, B_{\text{var}}, S_{\text{min}}, T_s); \quad M(l_{\text{var}}, B_{\text{var}}, S_0, T_a); \\
 &M(l_{\text{const}}, B_{\text{const}}, S_{\text{min}}, T_a); \quad M(l_{\text{var}}, B_{\text{const}}, S_0, T_a); \quad M(l_{\text{var}}, B_{\text{var}}, S_0, T_s); \\
 &M(l_{\text{const}}, B_{\text{const}}, S_{\text{min}}, T_s); \quad M(l_{\text{var}}, B_{\text{const}}, S_0, T_s); \quad M(l_{\text{var}}, B_{\text{var}}, S_{\text{min}}, T_a); \\
 &M(l_{\text{const}}, B_{\text{var}}, S_0, T_a); \quad M(l_{\text{var}}, B_{\text{const}}, S_{\text{min}}, T_a); \quad M(l_{\text{var}}, B_{\text{var}}, S_{\text{min}}, T_s); \\
 &M(l_{\text{const}}, B_{\text{var}}, S_0, T_s).
 \end{aligned}$$

Таким чином, створенням множини числових моделей джерела даних забезпечується можливість адаптування процесу ущільнення до змісту даних.

1.4 Представлення цілих чисел у вигляді лінійної форми Фібоначчі

У роботах [49–51] запропоновано оригінальний спосіб представлення цілих чисел у вигляді, так званої, лінійної форми Фібоначчі, яка забезпечує скорочення розрядності представлення великих чисел (256 двійкових розрядів і більше).

Лінійна форма Фібоначчі рангу t – це представлення натурального числа із застосуванням чисел Фібоначчі, що має вигляд

$$n = xF_{t-1} + yF_t, \tag{1.8}$$

де x і y – цілі числа, $y \neq 0$; t – натуральне число.

Така форма представлення є основою нового підходу щодо ущільнення даних, який ґрунтується на оптимізуючих властивостях чисел Фібоначчі [52]. При цьому для ущільнення інформації блок цифрових даних розглядається як надвелике ціле додатне число, що представляється набором з трьох невеликих чисел.

Існує множина представлень виду (1.1), але для задачі ущільнення даних пропонується використовувати представлення максимального рангу, яке має найменші значення цілих додатних чисел x і y . Тільки в цьому випадку з'являється можливість максимально використовувати оптимізаційні властивості чисел Фібоначчі.

Задача ущільнення полягає у знаходженні для заданого числа n мінімального базису (a, b) такого, що $n = \Phi_k(a, b)$. Числа $\Phi_i(a, b)$ – це узагальнені числа Фібоначчі, що обчислюються за формулою

$$\Phi_i = \Phi_{i-1} + \Phi_{i-2}, \quad i = 2 \div k$$

для $\Phi_0 = a$; $\Phi_1 = b$.

Якщо $n = aF_{t-1} + bF_t$ є лінійною формою Фібоначчі максимального рангу, то аналогічні лінійні форми можна одержати і для коефіцієнтів a і b . Відповідні коефіцієнти лінійних представлень для них аналогічно розкладаються у чергові лінійні форми Фібоначчі. Рекурсивним повторенням цієї процедури одержується лінійне дерево Фібоначчі [50]. Глибина цього дерева для числа n обмежена значенням $O(\log \log n)$. Ця оцінка показує, що лінійні дерева Фібоначчі є зручним об'єктом для представлення великих і надвеликих чисел.

У роботі [49] описується алгоритм одержання лінійної форми Фібоначчі, який вимагає виконання операцій ділення. Ця операція у разі чисел великої розрядності виконується дуже повільно, що спричиняє великі витрати часу на реалізацію цього алгоритму. В [52] обґрунтовується можливість одержання лінійної форми Фібоначчі без використання операції ділення і пропонується прискорений алгоритм. Однак реалізація і цього алгоритму вимагає відносно великої кількості обчислень, що пов'язано з необхідністю множення надвеликих цілих чисел.

Найменшу кількість операцій для визначення мінімального базису (a, b) забезпечує алгоритм, запропонований в роботі [85]. Цей алгоритм передбачає виконання таких дій.

Обчислюється добуток заданого числа n на обернене значення золоті пропорції

$$q = n \frac{2}{1 + \sqrt{5}}.$$

Для забезпечення можливості одержання мінімального базису при обчисленні q необхідно використовувати число з розрядністю, що не менша за розрядність числа n .

Взявши числа $w_0 = n$ і $w_1 = q$ як початкові елементи, здійснюються обчислення за формулою

$$w_i = w_{i-2} - w_{i-1}, \quad i = 2, 3, \dots$$

до тих пір, поки $w_i > 0$.

Якщо обчислення були виконані k разів, то представленню $n = \Phi_k(a, b)$ буде відповідати такий мінімальний базис: $a = w_{k+1}$ і $b = w_k$.

1.5 Розробка правил кодування та декодування блоків даних

Основним правилом кодування є правило, за яким цілим додатним числом N , отриманим як результат моделювання джерела даних, ставиться у відповідність лінійну форму Фібоначчі

$$N = q_2 \varphi_1(j+1) + q_1 \varphi_1(j), \quad j = 1; 2; \dots \quad (1.9)$$

де q_1 і q_2 – цілі додатні числа (координати представлення); j – індекс представлення.

У розділі 1 відзначалося, що відомі алгоритми перетворення числа z у лінійну форму Фібоначчі (ЛФФ (*)), але вони є достатньо складними. Тому пропонується алгоритм пришвидшеного перетворення, що базується на ідеї, висловленій в роботі [85].

Обчислюється добуток числа N на обернене значення «золоті» пропорції

$$q = N \frac{2}{1 + \sqrt{5}}.$$

Взявши числа $w_0 = N$ і $w_1 = q$ як початкові елементи, здійснюються обчислення за формулою

$$w_i = w_{i-2} - w_{i-1}, \quad i = 2, 3, \dots,$$

поки $w_i > 0$.

Якщо обчислення були виконані j разів, то представленню (1.9) будуть відповідати такі мінімальні координати: $q_2 = w_{j+1}$ і $q_1 = w_j$.

Враховуючи, що

$$\lim_{l \rightarrow \infty} \frac{\varphi_1(l)}{\varphi_1(l+1)} = \frac{2}{1 + \sqrt{5}},$$

пропонується замінити множення числа z на обернене значення «золотої» пропорції зсувом управо на один розряд коду Фібоначчі числа $N = (a_n a_{n-1} \dots a_2 a_1)$. Тобто $q = \vec{N} = (0 a_n a_{n-1} \dots a_2)$.

Числа q і w_2 обчислюються, починаючи з a_n , за формулами:

$$q = \sum_{j=n}^1 a_j \varphi_1(j-1);$$

$$w_2 = \sum_{j=n}^2 a_j \varphi_1(j-2).$$

У байтовому представленні ЛФФ (*) має таку структуру:

$$\{Q_j // l_{Q_1} // l_{Q_2} // Q_1 // Q_2\}, \quad (1.10)$$

де Q_j – код числа j (2 байти); l_{Q_1} – довжина коду числа q_1 в байтах (2 байти); l_{Q_2} – довжина коду числа q_2 в байтах (2 байти); Q_1 – код числа q_1 (змінна кількість байтів); Q_2 – код числа q_2 (змінна кількість байтів).

Пропонується окреме правило кодування числа $N = 0$. ЛФФ (*) цього числа має координати $q_1 = 0$, $q_2 = 0$ та індекс $j = 0$. У даному випадку значення індексу достатньо щоб відновити число N . Тому в байтовому представленні складові структури (2.9) l_{Q_1} , l_{Q_2} , Q_1 і Q_2 можна не подавати у вигляді кодів, залишивши тільки складову Q_j . 3

урахуванням цього, числу $N = 0$ буде відповідати код довжини 2 байти зі структурою $\{Q_j\}$.

Дослідження, проведені авторами, показали, що для певних чисел ЛФФ (*) у байтовому представленні буде довшою за вихідне число. Тобто замість зменшення довжини блоку даних відбувається її збільшення. Для запобігання цьому, пропонується такий підхід. Число N , що перетворюється в ЛФФ (*), розбивається на дві складові: $N = N_1 + N_2$. Число N_1 вибирається таким, щоб йому відповідала компактна лінійна форма Фібоначчі, а число N_2 залишається неперетвореним. При цьому результат ущільнення має таку структуру в байтовому представленні:

$$\{Q_j // l_{Q_1} // l_{Q_2} // Q_1 // Q_2 // l_{Q_{N_2}} // Q_{N_2}\}, \quad (1.11)$$

де l_{Q_2} – довжина коду числа N_2 в байтах (2 байти); Q_{N_2} – код числа N_2 (змінна кількість байтів).

Оскільки на збільшення координат q_1 і q_2 суттєво впливають одиничні значення a_l для невеликих індексів, тому пропонується обчислювати N_1 і N_2 за формулами:

$$N_1 = \sum_{l=s}^n a_l \varphi_1(l), \quad (1.12)$$

$$N_2 = \sum_{l=2}^{s-1} a_l \varphi_1(l). \quad (1.13)$$

Лінійну форму Фібоначчі числа N_1 будемо називати скороченою і позначати СЛФФ (*).

Однак існують числа, для яких і ЛФФ (*) і СЛФФ (*) у байтовому представленні є довшими за байтові представлення цих чисел. У таких випадках пропонується залишати числа неперетвореними, вказуючи на це певним символом (ознакою).

Для зручності посилання на конкретне правило кодування введемо такі їх позначення:

- $C_{\text{лфф}}$ (правило кодування на основі ЛФФ (*));
- $C_{\text{слфф}}$ (правило кодування на основі СЛФФ (*));
- $C_{\text{нп}}$ (правило, що не передбачає перетворення коду блоку).

Таким чином, пропонуються три правила кодування, які породжують відповідні структури кодованих блоків (табл. 1.1).

Таблиця 1.1 – Правила кодування і структури блоків

Позначення правила кодування	Структура блоку	Позначення структури блоку
$C_{\text{лфф}}$	$\{Q_j // l_{Q_1} // l_{Q_2} // Q_1 // Q_2\}$	STR0
$C_{\text{слфф}}$	$\{Q_j // l_{Q_1} // l_{Q_2} // l_{Q_{N_2}} // Q_1 // Q_2 // Q_{N_2}\}$	STR1
$C_{\text{нп}}$	$s_0, s_1, s_2, \dots, s_{l-1}$	STR2

При ущільненні можуть утворюватися різні комбінації цих правил. Для вказівки використаного правила кодування до результату перетворення блоку даних дописуються ознаки.

Схеми алгоритмів, що реалізують правила кодування, наведено на рис. 1.3 і 1.4.

Розглянемо правила декодування даних для описаних вище кодових структур. Умовні позначення таких правил наведено в табл. 1.2.

Таблиця 1.2 – Позначення правил декодування

Позначення структури блоку	Позначення правила декодування
STR0	$D_{\text{лфф}}$
STR1	$D_{\text{слфф}}$
STR2	$D_{\text{нп}}$

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shannon C. E. A Mathematical Theory of Communication / C. E. Shannon // Bell System Technical Journal. – 1948. – V. 27. – P. 379–423.
2. Fano R. M. The transmission of information. Technical / R. M. Fano ; Research Laboratory of Electronics at MIT // Report № 65, 1949.
3. Huffman D. A. A Method for the Construction of Minimum – Redundancy Codes / D. A. Huffman // Proceedings of the Institute of Electrical and Radio Engineers. – 1952. – V. 40, № 9. – P. 1098–1101.
4. Ziv J. A universal algorithm for sequential data compression / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. – 1977. – V. 23, № 3. – P. 337–343.
5. Ziv J. Compression of individual sequences via variable-rate coding / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. – 1978. – V. IT–24, № 5. – P. 530–535.
6. Rissanen J. J. Universal modeling and coding / J. J. Rissanen, G. G. Langdon // IEEE Trans. Inf. Theory. – 1981. – V. IT–27, № 1. – P. 12–23.
7. Rissanen J. J. Arithmetic codings as number representations / J. J. Rissanen // Acta Polytechnic Scandinavica. – 1979. – V. Math. 31. – P. 44–51.
8. Welch T. A. A Technique for High Performance Data Compression / T. A. Welch // Computer. – 1984. – № 6. – P. 176–189.
9. Storer J. A. Data compression via textual substitution / J. A. Storer, T. G. Szymanski // Journal of the ACM. – 1982. – V. 29, № 4. – P. 928–951.
10. Golomb S. W. Run-length encodings / S. W. Golomb // IEEE Transactions on Information Theory. – 1966. – V. 12, № 3. – P. 399–401.
11. Elias P. Universal codeword sets and representations of the integers / P. Elias // IEEE Trans. Inf. Theory. – 1975. – V. IT–21, № 2. – P. 194–203.
12. Elias P. Interval and regency rank source coding: Two on-line adaptive variable-length schemes / P. Elias // IEEE Trans. Inf. Theory. – 1987. – V. IT–33, № 1. – P. 3–10.
13. Horspool R. N. Dynamic Markov modeling. A prediction technique / R. N. Horspool, G. V. Cormack // Proceedings of the International Conference on the System Sciences. – Honolulu : HI, 1986. – P. 700–707.
14. Cormack G. V. Data compression using dynamic Markov modeling / G. V. Cormack, R. N. Horspool // Comput. J. – 1987. – V. 30, № 6. – P. 541–550.

15. Witten I. Arithmetic Coding for Data Compression / I. Witten, R. Neal, J. Cleary // *Communications of the ACM*. – 1987. – V. 30, № 6. – P. 520–540.
16. Burrows M. A Block-sorting Lossless Data Compression Algorithm / M. Burrows, D. J. Wheeler // *SRC Research Report 124*. – Palo Alto : Digital Systems Research Center, 1994. – 18 p.
17. Рябко Б. Я. Сжатие данных с помощью стопки книг / Б. Я. Рябко // *Проблемы передачи информации*. – 1980. – Т. 16, вып. 2. – С. 16–21.
18. Bell T. Modeling for Text Compression / T. Bell, I. Witten, J. Cleary // *ACM Computing Surveys*. – 1989. – V. 21, № 4. – P. 557–591.
19. Bell T. C. A note on the DMC data compression scheme / T. C. Bell, A. M. Moffat // *Computer J*. – 1989. – V. 32, № 1. – P. 16–20.
20. Bell T. C. Better OPM/L test compression / T. C. Bell // *IEEE Trans. Commun. COM-34*. – 1986. – № 12. – P. 1176–1182.
21. A locally adaptive data compression scheme / J. L. Bentley, D. D. Sleator, R. E. Tarjan, V. K. Wei // *Commun.* – 1986. – V. 29, № 4. – P. 320–330.
22. Bentley J. Fast Algorithms for Sorting and Searching Strings / J. Bentley, R. Sedgwick // *Proc. ACM–SIAM Symp. on discr. Algorithms*. – New Orleans, USA, 1997. – P. 360–369.
23. Brent R. P. A linear algorithm for data compression / R. P. Brent // *Aust. Comput. J.* – 1987. – V. 19, № 2. – P. 64–68.
24. Jakobsson M. Compression of character string by an adaptive dictionary / M. Jakobsson // *BIT*. – 1985. – V. 25, № 4. – P. 593–603.
25. Langdon G. G. An introduction to arithmetic coding / G. G. Langdon // *IBM J. Res. Dev.* – 1984. – V. 28, № 2. – P. 135–149.
26. Langdon G. G. Compression of black – white images with arithmetic coding / G. G. Langdon, J. J. Rissanen // *IEEE Trans. Commun.* – 1981. – COM–29, № 6. – P. 858–867.
27. Langdon G. G. A simple general binary source code / G. G. Langdon, J. J. Rissanen // *IEEE Trans. Inf. Theory*. – 1982. – V. IT–28. – P. 800–803.
28. Lelewer D. A. Data compression / D. A. Lelewer, D. S. Hirschberg // *Comput. Surv.* – 1987. – V. 13, № 3. – P. 261–296.
29. Moffat A. Word-based text compression / A. Moffat // *Software – Practice & Experience*. – 1989. – V. 19. – P. 185–198.

30. Moffat A. A Data Structure for Arithmetic Coding on Large Alphabets / A. Moffat // Proceedings 11 Australian Computer Science Conference, Brisbane, Australia. – 1988. – P. 309–317.

31. Moffat A. Implementing the PPM Data Compression Scheme / A. Moffat // IEEE Transactions on Communications. – 1990. – V. 38, № 11. – P. 1917–1921.

32. Борисенко А. А. Сжатие информации методом локальных сдвигов / А. А. Борисенко, Ю. А. Зубань // Вісник Сумського державного університету. – 2000. – № 16. – С. 70–72.

33. Борисенко А. А. Комбинаторное сжатие информации / А. А. Борисенко // Вісник Сумського державного університету. – 1996. – № 1(5). – С. 79–83

34. Борисенко А. А. Метод сжатия на основе комбинаторного разложения передаваемых сообщений на классы эквивалентности / А. А. Борисенко, Ю. А. Зубань // Вісник Сумського державного університету. – 2003. – № 11. – С. 88–99.

35. Кадач А. В. Эффективные алгоритмы неискажающего сжатия текстовой информации : дис. ... к. ф-м. н. : 05.13.11 / А. В. Кадач ; Институт систем информатики им. А. П. Ершова. – Новосибирск, 1997. – 108 с.

36. Майданюк В. П. Ущільнення даних без втрат на основі перетворень / В. П. Майданюк, О. В. Кириченко // Оптико-електронні інформаційно-енергетичні технології. — 2008. — № 2 (16). — С. 71–76.

37. Поліщук, У. В. Ущільнення даних на основі нейроподібних структур автоасоціативного типу : автореф. дис.... канд. техн. наук. : 05.13.23 / У. В. Поліщук ; Національний університет «Львівська політехніка». – Львів, 2011. – 20 с.

38. Поляков Г. О. Методи ущільнення даних без втрат інформації з використанням конкуруючих моделей інформаційного джерела : автореф. дис. ... канд. техн. наук. : 05.13.06 / Г. О. Поляков ; Нац. авіац. ун-т. – К., 2003. – 18 с.

39. Ратушняк О. А. Методы сжатия данных без потерь с помощью сортировки параллельных блоков : дис. ... на к. ф-м. н. : 05.13.11 / О. А. Ратушняк ; Институт систем информатики им. А. П. Ершова. – Новосибирск, 2002. – 87 с.

40. Ситняковская Е. И. Построение эффективных побуквенных кодов для словарных методов сжатия данных / Е. И. Ситняковская // Проблемы передачи информации. – 1998. – Т. 34, вып. 2. – С. 47–55.

41. Модифицированный метод предиктивного кодирования для сжатия графической информации / С. Г. Удовенко, А. А. Шамраев., Е. О. Шамраева, С. Д. Лукьяненко // Системи обробки інформації. – 2011. – Вип. 5(95). – С. 115–119.
42. Удовенко С. Г. Lossless-метод сжатия томографических данных / С. Г. Удовенко, А. А. Шамраев., Е. О. Шамраева // Information Technologies & Knowledge. – Sofia : ITNEA. – 2014. – V. 8, № 1. – P. 29–32.
43. Шкарин Д. А. Повышение эффективности алгоритма RPM / Д. А. Шкарин // Проблемы передачи информации. – 2001. – Т. 37, вып. 3. – С. 44–54.
44. Кричевский Р. Е. Сжатие и поиск информации / Р. Е. Кричевский. – М. : Радио и Связь, 1989. – 176 с.
45. Фомин А. А. Основы сжатия информации / А. А. Фомин. – СПб., 1998. – 82 с.
46. Балашов К. Ю. Сжатие информации: анализ методов и подходов / К. Ю. Балашов. – Минск, 2000. – 42 с. (Препринт / Ин-т техн. Кибернетики НАН Беларуси; № 6)
47. Семенюк В. В. Экономное кодирование дискретной информации / В. В. Семенюк. – СПб. : СПбГИТМО (ТУ), 2001. – 115 с.
48. Методы сжатия данных : учебное пособие / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : Диалог-Мифи, 2002. – 384 с.
49. Анисимов А. В. Обратное преобразование Фибоначчи / А. В. Анисимов, Я. П. Рындин, С. Е. Редько // Кибернетика. – 1982. – № 3. – С. 9–11.
50. Анисимов А. В. Линейные формы Фибоначчи и параллельные алгоритмы большой размерности / А. В. Анисимов // Кибернетика и системный анализ. – 1995. – № 3. – С. 106–115.
51. Лужецький В. А. Спосіб зображення цілих чисел великого діапазону / В. А. Лужецький, Мохаммад Аль-Майта. // Вимірювальна та обчислювальна техніка в технологічних процесах. – 1998. – № 1. – С. 156–162.
52. Кшановський О. Д. Арифметичні методи ущільнення цифрової інформації / О. Д. Кшановський, С. В. Тітарчук, В. А. Лужецький // Вісник Вінницького політехнічного інституту. – 1999. – № 5. – С. 83–87.
53. Лужецький В. А. Узагальнена модель адаптивного ущільнення даних / В. А. Лужецький, Л. А. Савицька, Шахзада Ашрафул Хок // Інформаційні технології та комп'ютерна інженерія. – 2009. – № 1(14). – С. 56–63.

54. Лужецький В. А. Адаптивне ущільнення даних одним проходом з нерівномірним розбиттям на блоки / В. А. Лужецький, Л. А. Савицька // Вісник черкаського державного технологічного університету. – 2012. – № 4. – С. 3–7.

55. Лужецький В. А. Адаптивне ущільнення даних одним проходом з рівномірним розбиттям на блоки / В. А. Лужецький, Л. А. Савицька // Інформаційні технології та комп'ютерна інженерія. – 2012. – № 2 (24). – С. 23–30.

56. Лужецький В. А. Моделі і методи адаптивного ущільнення даних на основі лінійної форми Фібоначчі / В. А. Лужецький, Л. А. Савицька // Збірник наукових праць Харківського університету Повітряних сил. – 2015. – № 1 (42). – С. 53–57.

57. Лужецький В. А. Розробка та дослідження методів адаптивного ущільнення даних на основі лінійної форми Фібоначчі / В. А. Лужецький Л. А. Савицька // Східно-європейський журнал передових технологій. – 2015. – № 1/9 (73). – С. 16–22.

58. Лужецький В. А. Адаптивний метод ущільнення даних на основі лінійної форми Фібоначчі / В. А. Лужецький, Л. А. Савицька // Гармоничное развитие систем – третий путь человечества : коллективная монография / под. ред. Э. М. Сороко, Т. И. Егоровой-Гудковой – Одесса : Институт креативных технологий, 2011. – С. 101–108.

59. Лужецький В. А. Методи адаптивного ущільнення даних на основі лінійної форми Фібоначчі / В. А. Лужецький, Л. А. Савицька // Наукоемкие технологи в инфокоммуникациях: обработка и защита информации : коллективная монография / под. ред. В. М. Безрука, В. В. Баранника. – Х. : Компания СМІТ, 2013. – С. 241–261.

60. Савицька Л. А. Числові моделі джерела даних, що ущільнюються / Л. А. Савицька // Методи та засоби кодування, захисту й ущільнення інформації : тези доповідей II Міжнародної наук.-практичної конф.: (22–24 квітня 2009 р., Вінниця) / відп. ред. В. А. Лужецький. – Вінниця : УНІ-ВЕРСУМ–Вінниця, 2009. – С. 165–166.

61. Савицька Л. А. Спеціалізований процесор для ущільнення даних / Л. А. Савицька // Інформаційні технології та комп'ютерна інженерія : тези доповідей Міжнародної наук.-практичної конф.: (19–21 травня 2010 р., Вінниця) / відп. ред. В. А. Лужецький. – Вінниця : ВНТУ, 2010. – С. 391–392.

62. Савицька Л. А. Метод адаптивного ущільнення даних розбиттям на блоки різної довжини / Л. А. Савицька // Методи та засоби ко-

дування, захисту й ущільнення інформації : тези доповідей III Міжнародної наук.-практичної конф.: (20–22 квітня 2011 р., Вінниця) / відп. ред. В. А. Лужецький. – Вінниця : ВНТУ, 2011. – С. 182–183.

63. Савицька Л. А. Метод адаптивного ущільнення даних на основі лінійної форми Фібоначчі / Л. А. Савицька // Інформаційні технології та комп'ютерна інженерія : тези доповідей II Міжнародної наук.-практичної конф.: (29–31 травня 2012 р., Вінниця) / відп. ред. В. А. Лужецький. – Вінниця : ВНТУ, 2012. – С. 230–231.

64. Савицька Л. А. Метод адаптивного ущільнення даних на основі лінійної форми Фібоначчі / Л. А. Савицька // Інформаційні технології та комп'ютерна інженерія : тези доповідей III Міжнародної наук.-практичної конференції: (28–30 травня 2014 р., Вінниця) / відп. ред. В. А. Лужецький. – Вінниця : ВНТУ, 2014. – С. 227–230.

65. Terry A. A Technique for High-Performance Data Compression / A. Terry, T. Welch // *Computer*. – 1984. – V. 17, № 6. – P. 8–19.

66. Hirschberg D. Data compression / D. Hirschberg, D. Lelewer // *Computing Surveys*. – 1987. – V. 19, № 3. – P. 261–297.

67. Knuth D. E. Dynamic Huffman Coding / D. E. Knuth // *J. Algorithms*. – 1985. – V. 6, № 2. – P. 163–180.

68. Vitter J. S. Design and Analysis of Dynamic Huffman Codes / J. S. Vitter // *J. ACM*. – 1987. – V. 34, № 4. – P. 825–845.

69. Мастрюков Д. Алгоритмы сжатия информации : Ч. 1. Сжатие по Хаффмену / Д. Мастрюков // *Монитор*. – 1993. – № 7–8. – С. 14–20.

70. Nelson M. R. The Data Compression Book / M. R. Nelson. – Redwood City, CA : M&T Books, – 1991. – 527 p.

71. Rissanen J. J. Arithmetic coding / J. J. Rissanen, G. G. Langdon // *IBM J. Res. Dev.* – 1979. – V. 23, № 2. – P. 149–162.

72. Langdon G. G. An introduction to arithmetic coding / G. G. Langdon // *IBM J. Res. Dev.* – 1984. – V. 28, № 2. – P. 135–149.

73. Langdon G. G. Compression of black – white images with arithmetic coding / G. G. Langdon, J. J. Rissanen // *IEEE Trans. Commun.* – 1981. – COM–29, № 6. – P. 858–867.

74. Langdon G. G. A simple general binary source code / G. G. Langdon, J. J. Rissanen // *IEEE Trans. Inf. Theory*. – 1982. – V. IT–28. – P. 800–803.

75. Rubin F. Arithmetic stream coding using fixed precision registers / F. Rubin // *IEEE Trans. Inf. Theory*. – 1979. – V. IT – 25, № 6. – P. 672–675.

76. Рябко Б. Я. Эффективный метод адаптивного арифметического кодирования для источников с большими алфавитами / Б. Я. Рябко, А. Н. Фионова // Проблемы передачи информации. – 1999. – Т. 35, вып. 4. – С. 34–39.

77. Howard P. G. Design and Analysis of Fast Text Compression Based on Quasi-Arithmetic Coding / P. G. Howard, J. S. Vitter // 1993 DCC, Snowbird, Utah, Mar – 1993. – P. 98–107.

78. Martin G. N. Range encoding: an algorithm for removing redundancy from a digitized message / G. N. Martin // Proceedings of the Video & Data Recording Conference, Southampton. – 1979. – P. 18–21.

79. Long P. M. Text Compression via Alphabet ReRepresentation / P. M. Long, A. I. Natsev, J. S. Vitter // Proc. IEEE Data Compression Conf. – Snowbird, Utah, USA, Mar. 25–27. – 1997. – P. 161–170.

80. Schmidhuber J. Sequential Neural Text Compression / S. Heil, J. Schmidhuber // IEEE Trans. Neural Networks. – 1996. – V. 7. – P. 142–146.

81. Левенштейн В. И. Избыточность и задержка восстановительного кодирования натуральных чисел / В. И. Левенштейн // Проблемы кибернетики. – 1968. – № 20. – С. 173–179.

82. Rice R. F. Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data / R. F. Rice, J. R. Plaunt // IEEE Trans. on Commun. – 1971. – V. 16, № 9. – P. 889–897.

83. Even S. Economical encoding of commas between strings / S. Even, M. Rodeh // Communications of the ACM. – 1978. – V. 21, № 4. – P. 315–317.

84. Стахов А. П. Коды золотой пропорции / А. П. Стахов. – М. : Радио и связь, 1984. – 150 с.

85. Лужецький В. А. Високонадійні математичні Фібоначчі-процесори : монографія / В. А. Лужецький. – Вінниця : УНІВЕРСУМ–Вінниця, 2000. – 248 с.

86. Klein S. T. On the usefulness of Fibonacci compression codes / S. T. Klein, Kopel Ben-Nissan // Comput. J. – 2010. – V.53. – P. 701–716.

87. Klein S. T. Fast Decoding of Fibonacci Encoded Texts / S. T. Klein // Proceedings of the International Data Compression Conference, DCC'07. – Washington, 2007. – P. 388.

88. Klein S. T. Using Fibonacci Compression Codes as Alternatives to Dense Codes / S. T. Klein, M. K. Ben-Nissan // Proceedings of the International Data Compression Conference, DCC'07. – Washington, 2008. – P. 472–481.

89. Калошин Д. Б. Сравнение кодов переменной длины: коды Элиаса и Фибоначчи применительно к вопросам сжатия данных / Д. Б. Калошин, В. Ю. Башкирев, Е. В. Бурмин // Сейсмические приборы. – 2008. – Т. 44, № 3. – С. 64–69.
90. Somasundaram K. Compression of Image using Fibonacci Code (FC) in JPEG2000 / K. Somasundaram, P. Sumitra // International Journal of Engineering Science and Technology. – 2010. – V. 2, № 12. – P. 7311–7319.
91. Bastys R. Fibonacci Coding Within the Burrows-Wheeler Compression Scheme / R. Bastys // Electronics and Electrical Engineering. – 2010. – № 1(97). – P. 28–32.
92. Nelson M. R. The Data Compression Book / M. R. Nelson. – Redwood City, CA: M&T Books. – 1991. – 527 p.
93. Rodeh M. A Linear Algorithm for Data Compression via String Matching / M. Rodeh, V. R. Pratt, S. Even // Journal ACM. – 1981. – V. 28, № 1. – P. 16–24.
94. Bell T. C. A unifying theory and improvements for existing approaches to text compression : Ph. D. dissertation / T. C. Bell ; Dept. of Computer Science, Univ. of Canterbury. – New Zealand, 1987.
95. Fiala E. R. Data compression with finite windows / E. R. Fiala, D. H. Greene // Commun. ACM. – 1989. – V. 32, № 4. – P. 490–505.
96. Willems F. M. J. The context-tree weighting method: Basic properties / F. M. J. Willems, Y. M. Shtarkov, T. J. Tjalkens // IEEE Transactions on Information Theory. – 1995. – V. 41, № 3. – P. 653–664.
97. LZRW1. – Режим доступа: <http://www.ross.net/compression/lzrw1.html>.
98. LZRW2. – Режим доступа: <http://www.ross.net/compression/lzrw2.html>.
99. LZRW3. – Режим доступа: <http://www.ross.net/compression/lzrw3.html>.
100. LZRW4. – Режим доступа: <http://www.ross.net/compression/lzrw4.html>.
101. LZRW5. – Режим доступа: <http://www.ross.net/compression/lzrw5.html>.
102. LZW: a new data compression algorithm by Charles Bloom. – Режим доступа: <http://www.cbloom.com/papers/lzw.pdf>.
103. Salomon D. Data Compression. The Complete Reference / D. Salomon. – 4th Edition Book. – Berlin : Springer, 2006. – 1118 p.

104. Мастрюков Д. Алгоритмы сжатия информации : Ч. 3. Алгоритмы группы LZ / Д. Мастрюков // Монитор. – 1994. – № 2. – С. 10–13.
105. Мастрюков Д. Алгоритмы сжатия информации. Часть 4. Алгоритм LZW / Д. Мастрюков // Монитор. – 1994. – № 3. – С. 8–11.
106. Compress (Version 4.0) program and documentation / S. W. Thomas, J. McKie, S. Davies, [та ін.]. – 1985. Available from joe@petsd. UUCP.
107. Miller V. S. Variations on a theme by Ziv and Lempel / V. S. Miller, M. N. Wegman // Combinatorial Algorithms on Words / Eds. A. Apostolico, Z. Galil ; NATO ASI Series. – Berlin : Springer-Verlag, 1984. – V. F12. – P. 131–140.
108. Tischer P. A modified Lempel-Ziv-Welch data compression scheme / P. Tischer // Aust. Comp. Sci. Commun. – 1987. – V.9, № 1. – P. 262–272.
109. Cleary J. G. Data Compression Using Adaptive Coding and Partial String Matching / J. G. Cleary, I. H. Witten // IEEE Trans. Commun. – 1984. – V. 32, № 4. – P. 396–402
110. Jones D. W. Application of splay trees to data compression / D. W. Jones // Commun. ACM. – 1988. – V. 31, № 8. – P. 996–1007.
111. Schürmann T. Entropy estimation of symbol sequences / T. Schürmann, P. Grassberger // Chaos. – 1996. – V. 6. – P. 414–427.
112. Solving the Problems of Context Modeling by Charles Bloom – Режим доступа: <http://www.cbloom.com/papers/ppmz.pdf>
113. Langdon G. G. A Double-Adaptive File Compression Algorithm / G. G. Langdon, J. J. Rissanen // IEEE Trans. Comm. – 1983. – V. 31, № 11. – P. 1253–1255.
114. Williams R. N. Dynamic-History Predictive Compression / R. N. Williams // Information Systems. – 1988. – V. 13, № 1. – P. 129–140.
115. Mahoney M. Adaptive Weighing of Context Models for Lossless Data Compression / M. Mahoney // Florida Tech. Technical Report CS – 2005–16, 2005.
116. Буяновский Г. Ассоциативное кодирование / Г. Буяновский // Монитор. – 1994. – № 8. – С. 10–22.
117. Fenwick P. Block Sorting Text Compression / P. Fenwick // Proceedings of the 19 Australasian Computer Science Conference, Melbourne, Australia, 1996.
118. Nelson M. Data Compression with the Burrows-Wheeler Transform / M. Nelson // Dr. Dobb's Journal, September, 1996.
119. Bzip2. Режим доступа: <http://www.bzip.org/docs.html/>.

120. Борисенко А. А. Оптимальное неравномерное кодирование в методе локальных сдвигов / А. А. Борисенко, Ю. А. Зубань // Вісник Сумського державного університету. – 2001. – № 18. – С. 34–36.
121. Чередниченко В. Б. Метод сжатия двоичных кодов на основе биномиальных чисел / В. Б. Чередниченко // Вісник Сумського державного університету. – 2006. – № 4(88). – С. 61–68.
122. Борисенко А. А. Чередниченко В. Б. Нумерация равновесных кодов на основе биномиальных чисел / А. А. Борисенко, В. Б. Чередниченко // Право і Безпека. – 2004. – № 3–4. – С. 194–197.
123. Борисенко А. А. Биномиальное кодирование : монографія / А. А. Борисенко, И. А. Кулик. – Сумы : СумГУ, 2010. – 206 с.
124. Salomon D. Handbook of Data Compression / D. Salomon, D. Bryant, G. Motta. – London: Springer, 2010. – 1361 p.
125. Архиваторы. – Режим доступа: <http://compress.ru/article.aspx?id=9776>.
126. Краткое описание 20 популярных архиваторов. – Режим доступа: <http://compress.ru/article.aspx?id=10581&part=pa31ext11>.
127. Архиваторы. – Режим доступа: <http://www.win-rar.ru/support/knowledge/section.php?ID=201>.
128. Процкевич А. А. Форма хранения и передачи кодированных знаний / А. А. Процкевич, В. С. Яковишин // Комплексная защита информации. – 1999. – Вып. 2. – С. 95–103.
129. Nevill-Manning C. G. Compression and explanation using hierarchical grammars / C. G. Nevill-Manning, I. H. Witten // Computer Journal. – 1997. – V. 40, № 2/3. – P. 103–116.
130. Самофалов К. Г. Электронные цифровые вычислительные машины / К. Г. Самофалов, В. И. Корнейчук, В. П. Тарасенко. – К. : Вища школа, 1976. – 479 с.
131. Глушков В. М. Логическое проектирование дискретных устройств / В. М. Глушков, Ю. В. Капитонова, А. Т. Мищенко. – К. : Наукова думка, 1987. – 264 с.

Навчальне видання

**Савицька Людмила Анатоліївна
Лужецький Володимир Андрійович**

**МЕТОДИ ТА ЗАСОБИ АДАПТИВНОГО УЩІЛЬНЕННЯ
ДАНИХ НА ОСНОВІ ЛІНІЙНОЇ ФОРМИ ФІБОНАЧЧІ**

Монографія

Редактор С Малішевська

За заг. ред. В. А. Лужецького

Оригінал-макет підготовлено Л. Савицькою

Підписано до виготовлення 20.07.2016 р.

Системні вимоги:

процесор Pentium; 512 Mb RAM;

Windows XP,7,8; Acrobat Reader 6.0.

Один електронний оптичний диск (CD-ROM); Обсяг даних 1,35 Мб.

Наклад 100 (1-й запуск 1–30) прим. Зам. № E2016-04

Видавець та виготовлювач – Вінницький національний технічний університет,
Комп'ютерний інформаційно-видавничий центр Хмельницьке шосе, 95, ВНТУ,
ГНК, к. 114, м. Вінниця, 21021, тел.: (0432) 59-85-32, 59-81-59.
publish.vntu.edu.ua; *email*: kivc.vntu@gmail.com.

Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.