ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

УДК 004.451.53 : 004.67

В. А. Лужецький[1]
Т. М. Чеборака[1]

АДАПТИВНИЙ МЕТОД УЩІЛЬНЕННЯ ДАНИХ НА ОСНОВІ ВІДКИДАННЯ ПОСЛІДОВНОСТЕЙ НУЛІВ ТА ОДИНИЦЬ

[1]Вінницький національний технічний університет

Запропоновано адаптивний метод ущільнення даних без втрат, що базується на використанні методів відкидання послідовностей однакових символів, суть якого полягає в аналізі кожного блоку вихідних даних та відкиданні послідовності однакових символів у тих розрядах (молодших, старших, внутрішніх або молодших і старших), що забезпечує найбільший коефіцієнт ущільнення.

Ключові слова: адаптивний метод, ущільнення даних, послідовності однакових символів.

Вступ

Ущільнення даних є загальною вимогою для більшості прикладних програм, а також важливим і активним напрямом досліджень в галузі комп'ютерної науки [1]. Без методів ущільнення, жодна з новітніх технологій Інтернету, цифрового телебачення, мобільного зв'язку або покращених методів відео-зв'язку не набули б такого рівня розвитку, який вони мають зараз.

Процесом ущільнення даних є алгоритмічне перетворення, яке виконується з метою зменшення надлишковості інформації [2], що міститься у вхідних даних. В основі будь-якого методу ущільнення [3, 4] є модель джерела даних, або, точніше, модель надлишковості. Іншими словами, для ущільнення інформації використовуються відомості про формат та тип вхідних даних. Не володіючи такими відомостями про джерело даних, неможливо зробити ніяких припущень про тип перетворення, який б дозволив зменшити обсяг повідомлення. Модель надлишковості [5] може бути статичною, незмінною для всього повідомлення, або формуватися на етапі ущільнення (і відновлення). Адаптивні методи [6] дозволяють на основі властивостей вхідних даних змінювати модель надлишковості інформації, що забезпечує підвищення ефективності роботи методів ущільнення інформації.

Запропоновані в роботі [7] методи ущільнення даних на основі відкидання послідовностей нулів та одиниць є неадаптивними і тому не можуть забезпечити однаково високий коефіцієнт ущільнення для різних типів даних. Тому, актуальним є розробка адаптивних методів ущільнення інформації, які б ущільнювали дані з різними властивостями та характеристиками.

Метою роботи є підвищення значення коефіцієнта ущільнення даних без втрат, на основі відкидання послідовностей нулів та одиниць, шляхом створення адаптивного методу ущільнення.

Постановка задач

Для досягнення поставленої мети необхідно розв'язати такі задачі:

— розробити метод адаптивного ущільнення даних на основі відкидання послідовностей нулів та одиниць;

— провести експериментальне дослідження ефективності ущільнення запропонованого методу на тестових файлах різного формату та обсягу.

Правила ущільнення блоків даних

Пропонується набір з 5 правил ущільнення блоків даних, який забезпечує можливість адаптації до змісту вхідних даних, з метою забезпечення найбільшого коефіцієнта ущільнення.

Правило ущільнення C_1 полягає у відкиданні q однакових символів у старших розрядах блоку, при виконанні умови $q > \log_2 n$, де n — розрядність блоку ущільнюваних даних.

Результатом реалізації цього правила є структура перетвореного блоку STR1 (рис. 1), яка має такі складові:

— *type* (код правила 001);

— c (символи, що відкидаються, $c = 0$ або $c = 1$);

— q (двійковий код кількості однакових символів послідовності, що відкидаються);

— поле X...X (код, що залишається без змін).

X	…	X	q	c	*type*

Рис. 1. Структура перетвореного блоку STR1

Правило ущільнення C_2 реалізується аналогічно правилу C_1. Відмінність полягає лише у тому, що відбувається відкидання однакових символів в молодших розрядах блоку. При цьому поле *type* приймає значення 010.

Правило ущільнення C_3 полягає у відкиданні q однакових символів у внутрішніх розрядах блоку, при виконанні умови $q > 2\log_2 n - 1$.

Результатом реалізації цього правила є структура перетвореного блоку STR3 (рис. 2), яка має такі складові:

— *type* (код правила 011);

— l (двійковий код кількості символів, що залишається без змін у молодших розрядах, розрядність коду $\log_2 n$);

— поле $X^{(l)} \dots X^{(l)}$ (код, що залишається без змін у молодших розрядах, розрядність коду l);

— поле $X^{(h)} \dots X^{(h)}$ (код, що залишається без змін у старших розрядах, розрядність коду $n - q - l - 2$).

$X^{(h)}$	…	$X^{(h)}$	$X^{(l)}$	…	$X^{(l)}$	l	q	c	*type*

Рис. 2. Структура перетвореного блоку STR3

Правило ущільнення C_4 полягає у відкиданні q однакових символів у старших і молодших розрядах блоку, за виконання умов:

$$\begin{cases} q > \log_2 n + 2, \text{ якщо } q_l = q_h; \\ q > 2\log_2 n + 2, \text{ якщо } q_l \neq q_h, \end{cases}$$

де $q = q_l + q_h$, q_h (двійковий код числа q, що відкидається у старших розрядах); q_l (двійковий код числа q, що відкидається в молодших розрядах).

Результатом реалізації цього правила є структура перетвореного блоку STR4 (рис. 3), яка має такі складові:

— *type* (код правила 100);

— $q_{\min}$ (двійковий код розрядністю $\log_2 n$, який відповідає значенню $\min(q_l, q_h)$);

— c_h (символи, що відкидаються у старших розрядах);

— c_l (символи, що відкидаються у молодших розрядах);

— $t_e = \begin{cases} 01, \text{ якщо } q_l > q_h; \\ 10, \text{ якщо } q_l < q_h; \\ 11, \text{ якщо } q_l = q_h; \end{cases}$

— Δq (різниця значень $|q_l - q_h|$, розрядність $\log_2 n$);

— поле X...X (код, що залишається без змін, розрядність коду $n - q - 2$).

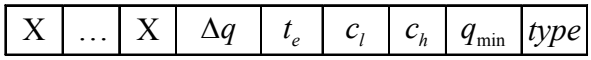

Рис. 3. Структура перетвореного блоку STR4

Коли жодне з правил ущільнення на основі відкидання послідовності однакових символів не може бути реалізоване для блоку даних, то такий блок залишається незмінним, але до нього дописується код 000 в полі *type,* що відповідає правилу ущільнення C_5 (рис. 4).

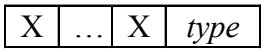

Рис. 4. Структура перетвореного блоку STR5

З урахуванням вищенаведених структур перетворених блоків B_i^* маємо структуру ущільнених даних, що наведена на рис. 5.

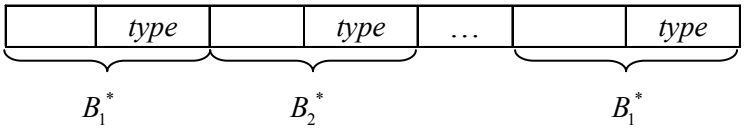

Рис. 5. Структура ущільнених даних

Правила відновлення блоків даних

Правило відновлення блоків даних визначається кодом поля *type.* Тому, спочатку зчитується три біти поля *type,* а потім певна кількість бітів, що відповідає цьому типу ущільнення.

Схему відновлення блоку даних за правилом D_1 (*type* = 001) наведено на рис. 6.

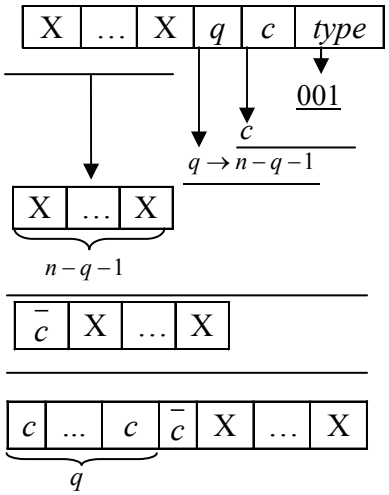

Рис. 6. Схема відновлення блоку даних за правилом D_1

Для реалізації правила D_1 виконується така послідовність дій. Зчитується 1 біт *c*, який визначає відкинуті символи. Зчитується $\log_2 n$ біт, що є кодом числа q. Визначається число $n-q-1$, яке є кількістю бітів, що залишилися без змін. Зчитується така кількість біт з ущільнених даних і записується у відновлений блок. Далі до цих бітів з боку старших розрядів дописується q символів, що визначаються *c*. Також дописується один символ $\bar{c}$ протилежний символу *c*.

Правило відновлення D_2 (*type* = 010) реалізується аналогічно правилу відновлення D_1. Відмінність полягає лише у тому, що q однакових символів дописується у молодші розряди блоку (рис. 7).

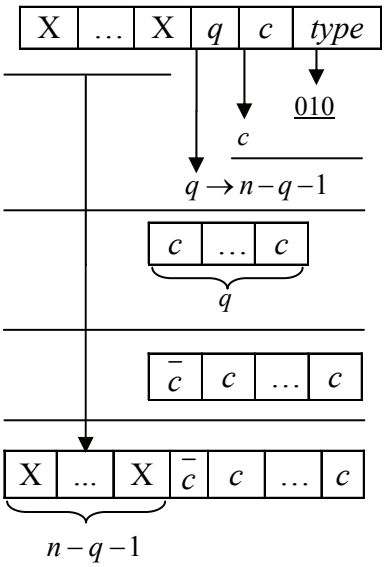

Рис. 7. Схема відновлення блоку даних за правилом D_2

Схему відновлення блоку даних за правилом D_3 (*type* = 011) наведено на рис. 8.

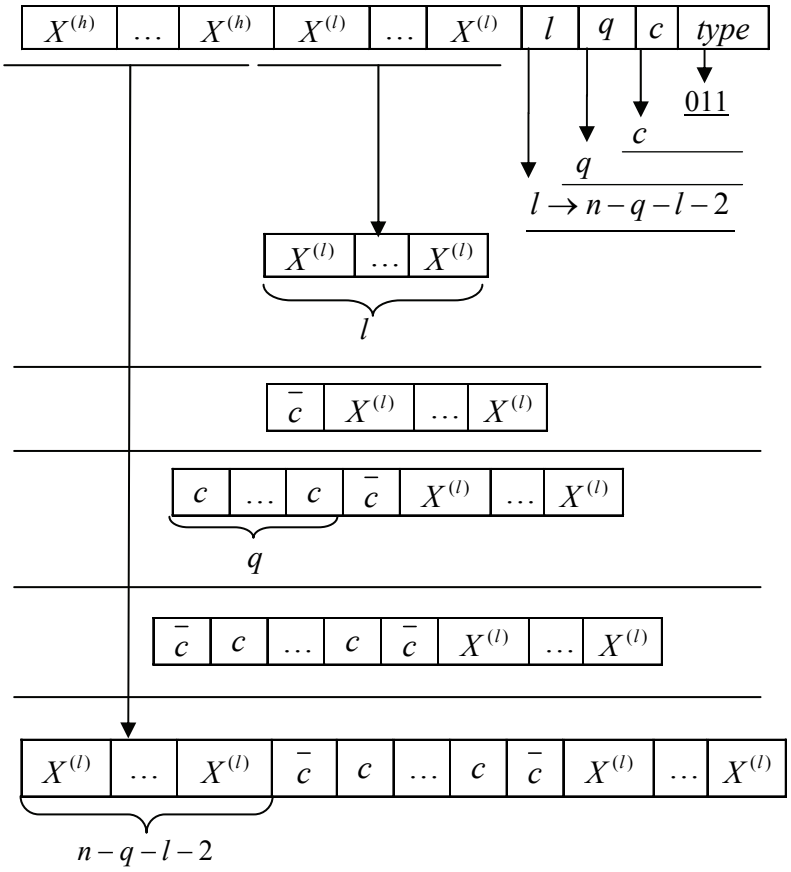

Рис. 8. Схема відновлення блоку даних за правилом D_3

Для реалізації правила D_3 виконується така послідовність дій. Зчитується 1 біт *c*. Зчитується $\log_2 n$ біт q. Зчитується $\log_2 n$ біт l — кількість символів, що залишилися без змін у молодших розрядах. Зчитується ця кількість біт з ущільнених даних і записується у відновлений блок. Внутрішні розряди відновленого блоку заповнюються q символами, що визначаються *c*, а також двома

протилежними символами $\overline{c}$ по краях послідовності однакових символів. Потім визначається число $n-q-l-2$, яке є кількістю бітів у старших розрядах, що залишилися без змін.

Схему відновлення блоку даних за правилом D_4 (*type* = 100) показано на рис. 9.

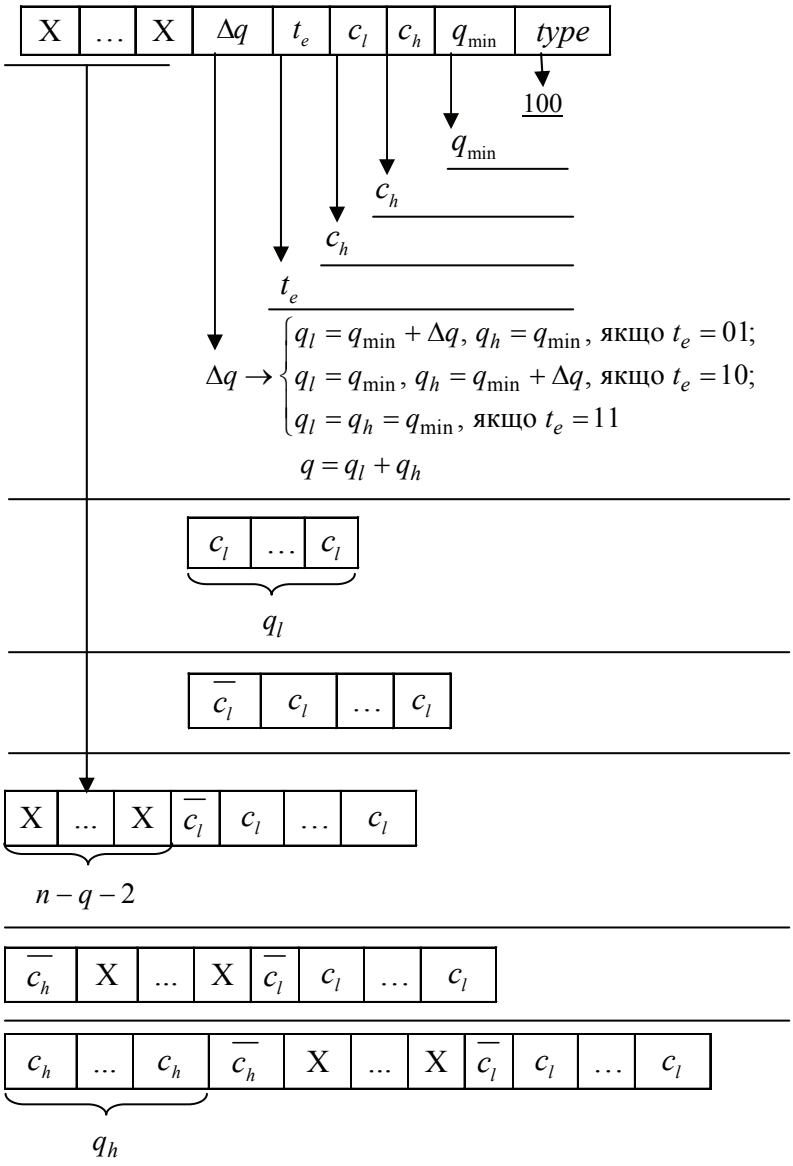

Рис. 9. Схема відновлення блоку даних за правилом D_4

Для реалізації правила D_4 виконується така послідовність дій. Зчитується $\log_2 n$ біт числа $q_{\min}$. Зчитується 1 біт c_h, який визначає тип відкинутого символу у старших розрядах. Зчитується 1 біт c_l, який визначає тип відкинутого символу у молодших розрядах. Зчитуються 2 біти t_e типу нерівності q_l і q_h. Якщо $t_e \neq 11$, то також зчитується $\log_2 n$ біт Δq. На основі $q_{\min}$, t_e і Δq визначається q_l і q_h таким чином:

— якщо $t_e = 01$, то $q_l = q_{\min} + \Delta q$, $q_h = q_{\min}$;

— якщо $t_e = 10$, то $q_l = q_{\min}$, $q_h = q_{\min} + \Delta q$;

— якщо $t_e = 11$, то $q_l = q_h = q_{\min}$.

Після цього молодші розряди заповнюються q_l, а старші q_h символами, що визначаються c. До молодших та старших розрядів також дописується один протилежний символ $\overline{c}$, якщо $q \neq n$ і $q_l \leq n-1$ та $q \neq n$ і $0 < q_h \leq n-1$, відповідно. Визначається число $n-q-2$, яке є кількіс-

тю бітів, що залишилися без змін. Зчитується це число біт з ущільнених даних і записується у відновлений блок.

Відновлення блоку даних за правилом D_5 ($type$ = 000) полягає лише у зчитуванні n біт з ущільнених даних.

Результати дослідження

Для автоматизації експериментального дослідження запропонованого адаптивного методу ущільнення на основі відкидання послідовностей нулів та одиниць розроблено програмний засіб.

Під час проведення дослідження розрядність початкового блоку даних може набувати значень в діапазоні: 16, 32, …, 2048. Дослідження відбувалося за такими показниками: коефіцієнт ущільнення k, тривалість ущільнення t_c, тривалість відновлення t_{dc}.

Результати дослідження запропонованих методів ущільнення за показниками k, t_c, t_{dc} виводяться у табличному та графічному вигляді (рис. 10).

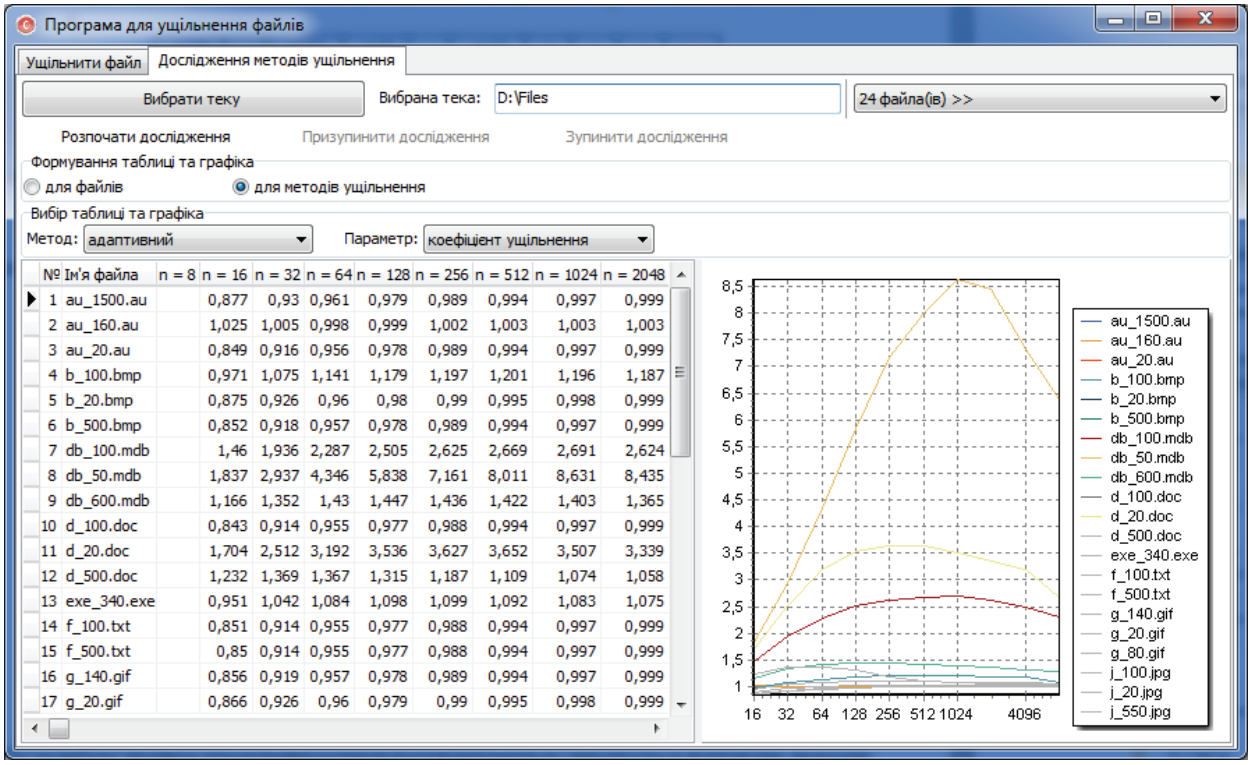

Рис. 10. Головне вікно програмного засобу

Відновлення вихідного файлу до початкового вигляду виконується без втрат, оскільки процес ущільнення формує усю необхідну інформацію для відновлення початкового файлу.

Для проведення експериментальних досліджень адаптивного методу ущільнення сформовано тестову вибірку із файлів найпоширеніших форматів. Вибірка містить 24 файли таких форматів та розмірів:

— *.doc — 20 кБ, 100 кБ, 500 кБ;
— *.txt — 100 кБ, 500 кБ;
— *.bmp — 20 кБ, 100 кБ, 500 кБ;
— *.gif — 20 кБ, 80 кБ, 140 кБ;
— *.jpg — 20 кБ, 100 кБ, 550 кБ;
— *.au — 20 кБ, 160 кБ, 1500 кБ;
— *.mp3 — 40 кБ, 330 кБ, 500 кБ;
— *.exe — 340 кБ;
— *.mdb — 50 кБ, 100 кБ, 600 кБ.

Результати досліджень коефіцієнта ущільнення подано у таблиці та показано як графік на рис. 11.

Результати дослідження коефіцієнта ущільнення

Формат та обсяг файлу	Розрядність							
	16	32	64	128	256	512	1024	2048
au, 160 кБ	1,025	1,005	0,998	0,999	1,002	1,003	1,003	1,003
bmp, 100 кБ	0,971	1,075	1,141	1,179	1,197	1,201	1,196	1,187
mdb, 100 кБ	1,46	1,936	2,287	2,505	2,625	2,669	2,691	2,624
mdb, 50 кБ	1,837	2,937	4,346	5,838	7,161	8,011	8,631	8,435
mdb, 600 кБ	1,166	1,352	1,43	1,447	1,436	1,422	1,403	1,365
doc, 20 кБ	1,704	2,512	3,192	3,536	3,627	3,652	3,507	3,339
doc, 500 кБ	1,232	1,369	1,367	1,315	1,187	1,109	1,074	1,058
exe, 340 кБ	0,951	1,042	1,084	1,098	1,099	1,092	1,083	1,075
mp3, 40 кБ	0,892	0,962	1	1,021	1,032	1,036	1,038	1,038

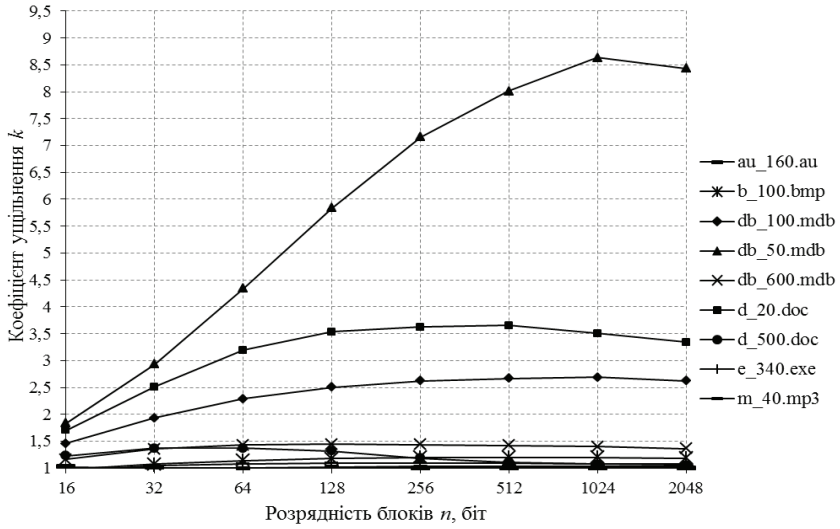

Рис. 11. Графіки результатів дослідження коефіцієнта ущільнення

З табл. та рис. 11 видно, що більшість файлів піддаються ущільненню запропонованим адаптивним методом, за винятком файлів таких форматів: mp3, у разі розбиття вхідної послідовності на блоки по 16 і 32 розряди; au, у разі розбиття вхідної послідовності на блоки по 64 і 128 розрядів; bmp при розбитті на блоки по 16 розрядів та exe, у разі розбиття вхідної послідовності на блоки по 16 розрядів. В результаті досліджень найбільшого коефіцієнта ущільнення досягнув файл бази даних (mdb, 50кБ) у разі розбиття на блоки розрядністю 1024, його вдалось ущільнити більше, ніж у 8 разів.

Щодо тривалості ущільнення та відновлення файлів запропонованими методами, то експериментальне дослідження на комп'ютері типу IBM/PC з процесором Intel Core i3 M370 (2.4GHz) і ОЗП розміром у 3 ГБ показало, що для найбільшого файлу (au, 1500 кБ) процедура ущільнення найдовше тривала за найменшої розрядності блоків: від 0,01 с до 0,73 с, а процедура відновлення — від 0,01 с до 0,44 с.

Висновки

Встановлено, що запропонований адаптивний метод ущільнення даних на основі відкидання послідовностей однакових символів дозволяє ущільнювати файли різних форматів та розмірів, оскільки обробляє вихідну послідовність даних на бітовому рівні, що дозволяє зменшити її залежність від типу даних. Найбільший коефіцієнт ущільнення забезпечує адаптивний метод при ущільненні файлу бази даних (mdb, 50кБ). У разі розбиття вихідної послідовності даних на блоки розрядністю 1024 біта файл вдалось ущільнити більш ніж у 8 разів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Salomon D. Handbook of Data Compression / D. Salomon, G. Motta. — London: Springer, 2010. — 1361 p.
2. Методы сжатия данных / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин — М. : ДИАЛОГ-МИФИ, 2002. — 384 с.
3. Storer J. A. Data Compression: Methods and Theory / Storer J. A. // Computers Science Press. — 1988. — V. 47, 1. — P. 23—29.
4. Lopez D. Important Concepts in Signal Processing, Image Processing and Data Compression. — University of Delhi, 2012. — 73 p.
5. Кричевский Р. Е. Сжатие и поиск информации / Р. Е. Кричевский. — М. : Радио и Связь, 1989. — 176 с.
6. Рябко Б. Я. Эффективный метод адаптивного арифметического кодирования для источников с большими алфавитами / Б. Я. Рябко, А. Н. Фионова // Проблемы передачи информации. — 1999. — С. 34—39.
7. Лужецький В. А. Методи ущільнення даних на основі відкидання послідовностей нулів та одиниць / В. А. Лужецький, Т. М. Чеборака // Інформаційні технології та комп'ютерна інженерія. — 2014. — № 1. — С. 18—26.

Рекомендована кафедрою захисту інформації ВНТУ

Стаття надійшла до редакції 19.03.2015

Лужецький Володимир Андрійович — д-р техн. наук, професор, завідувач кафедри захисту інформації, e-mail: lva_zi@mail.ru;

Чеборака Тетяна Михайлівна — аспірант кафедри захисту інформації, e-mail: altamira_90@mail.ru.

Кафедра захисту інформації, Вінницький національний технічний університет, Вінниця

V. A. Luzhetskyi[1]
T. M. Cheboraka[1]

The Adaptive Method of Data Compression Based on Truncation of Zeros and Ones Sequences

[1]Vinnytsia National Technical University

The adaptive method of lossless data compression based on using methods of truncation sequences of the same characters is proposed. The subject matter of adaptive method is to analyze blocks of input data and truncate sequences of the same characters in positions (low, high, internal or low and high order positions), leading to the highest increasing of data compression ratio.

Keywords: adaptive method, data compression, sequence of same characters.

Luzhetskyi Volodymyr A. — Dr. Sc. (Eng.), Professor, Head of the Chair of Information Protection, e-mail: lva_zi@mail.ru;

Cheboraka Tetiana M. — Post-Graduate Student of the Chair of Information Protection, e-mail: altamira_90@mail.ru

В. А. Лужецкий[1]
Т. М. Чеборака[1]

Адаптивный метод сжатия данных на основе отбрасывания последовательности нулей и единиц

[1]Винницкий национальный технический университет

Предложен адаптивный метод сжатия данных без потерь, основанный на использовании методов отбрасывания последовательности одинаковых символов, суть которого заключается в анализе каждого блока исходных данных и отбрасывании последовательности одинаковых символов в тех разрядах (младших, старших, внутренних или младших и старших), что приводит к наибольшему повышению коэффициента сжатия данных.

Ключевые слова: адаптивный метод, сжатие данных, последовательности одинаковых символов.

Лужецкий Владимир Андреевич — д-р техн. наук, профессор, заведующий кафедрой защиты информации, e-mail: lva_zi@mail.ru;

Чеборака Татьяна Михайловна — аспирант кафедры защиты информации, e-mail: altamira_90@mail.ru