УДК 658.012:681.32:621.38

Л. І. Тимченко, д.т.н., професор
(завідувач кафедри «Телекомунікаційні технології та автоматика», Державний університет інфраструктури та технологій)
Н. І. Кокряцька, к.т.н., доцент
(доцент кафедри «Телекомунікаційні технології та автоматика», Державний університет інфраструктури та технологій)
І. Д. Івасюк, к.т.н., доцент
Ю. В. Майстренко
(аспірант, Державний університет інфраструктури та технологій)

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ УЩІЛЬНЕННЯ ДАНИХ НА ОСНОВІ ІЄРАРХІЧНОГО КОДУВАННЯ

У даній статті проводиться аналіз методів ущільненя даних, що використовується на даний момент в комп'ютерній техніці та розглядаються шляхи розробки технології ущільнення даних на основі ієрархічного кодування, зокрема ітераційного та побітового методів.

Ключові слова: *ущільнення, стиск, кодування, інформація, дані, кодер, декодер, модель, алгоритм, біт, адаптивне кодування, ієрархічне кодування, побітове кодування.*

Постановка проблеми. Останні 30 – 40 років обсяги цифрової інформації постійно збільшуються, поступово вся інформація оцифровується, а отже, відповідно до цього потрібно збільшувати в потужності накопичувачі інформації та канали її передачі. Досягти цього можна лише двома шляхами: збільшуючи саму вмістимість накопичувачів та потужність каналів передачі даних або зменшити обсяг інформаціїї, не втрачаючи її. На сьогодення розвиток техніки рухається паралельно в обох напрямах, адже неможливо досягти значних результатів, вибравши тільки один напрям. Вірніше можливо, тільки шляхом збільшення потужностей, але через свою собівартість не буде достигнуто поставленої мети в глобальному маштабі. Шлях зменшення обсягів даних значно зменшує витрати на побудову таких систем. А отже в процесі розвитку комп'ютерної техніки, техніка ущільнення даних стосовно до інформаційних технологій удосконалювалась і підійшла до тієї межі, коли потрібно шукати нові методології ущільнення даних, оскільки існуючі через деякий час перестануть задовільняти потреби. Отже, візьмемо для нас більш приорітетним методологію ущільнення даних.

Метою роботи є проведення стислого класифікаційного аналізу методів ущільнення, що використовуються в комп'ютерних технологіях сьогодення та пошук рішень поставленої проблеми.

Виклад основного матеріалу. В даний час у комп'ютерній техніці ущільнення використовуються статистичні і словникові методи.

З статистичних методів найпоширенішим є кодування Шеннона-Фано і кодування Хаффмена (статичне, динамічне і з блокуванням), а зі словникових − сімейство алгоритмів, заснованих на широко відомому методі кодування Лемпеля-Зіва-Велча. Можна відзначити деякі інші методи ущільнення, що використовуються в комп'ютерних технологіях. Це арифметичне кодування і кодування на основі стопки книг. Проаналізуємо деякі з них.

Кодування Хаффмена має мінімальну надмірність за умови, що кожний символ кодується окремим ланцюжком в алфавіті {0,1}. Його недоліком є залежність ступеня ущільнення від близькості ймовірності символів до негативного ступеня 2, що пов'язано з кодуванням кожного символу цілим числом біт.

Арифметичне кодування є методом, що дозволяє упаковувати символи вхідного алфавіту без утрат за умови, що відомий розподіл частот цих символів. Концепція методу була викладена в 60-х роках у роботах Еліаса. При реалізації цього методу виникають дві проблеми: по-перше, необхідна арифметика дійсних чисел, узагалі говорячи, необмеженої точності, а по-друге, результат кодування стає відомий лише при закінченні вхідного потоку.

В основу словникових методів покладене кодування ланцюжків символів (LZ77 − compression). Суть його в такому: пакувальник постійно береже деяку кількість останніх оброблених символів у деякому буфері − ковзному словнику. В міру обробки вхідного потоку символи, що знову надійшли, потрапляють у кінець буфера, зрушуючи попередні символи і витісняючи найстаріші. Час ущільнення при такій реалізації пропорційний добутку довжини вхідного потоку на розмір буфера, що є непридатним для практичного використання. Для поліпшення процедури швидкого пошуку в словнику використовується двійкове дерево, що дозволило дещо підвищити швидкість роботи.

Далі пропонується група так званих методів ієрархічного кодування, що дозволяють розділити процес ущільнення даних на процедури моделювання і безпосередньо кодування.

Основна ідея цієї групи методів заснована на представленні вхідного потоку у вигляді багаторівневого (ієрархічного) потоку даних і їхньому кодуванні на кожному рівні ієрархії, взаємодії між якими визначає однозначний процес декодування.

Класифікаційний аналіз найвідоміших методів ущільнення з урахуванням запропонованих, приведений на рис. 1.

Коротко зупинимося на тих методах, що не згадувалися вище. Сімейство алгоритмів LZ78 (LZW, MW, AP, Y) відрізняє висока швидкість роботи як при упаковуванні, так і при розпаковці, достатньо скромні вимоги до пам'яті і прямої апаратної реалізації. Недолік − менший ступінь ущільнення в порівнянні зі схемою двоступінчатого кодування LZ77.

Алгоритм LZW має важливу властивість префіксності, а алгоритм MW не є префіксним і для нього реалізація словника важка.

Алгоритм AP нарощує словник набагато швидше, ніж MW і ущільнення є кращим, ніж MW.

Y -кодування, відкрите Д. Бернстейном, додає в словник один рядок на кожний читаний символ і подібно алгоритму LZW має властивість префіксності. Найцікавіші алгоритми LZFG (Fiala and Greene, 1989), що являють собою комбінацію LZ77 і LZ78

з оригінальною структурою словника, а також моделювання методом пророкувань часткових збігів PPM (Prediction by Partial Matching), що досягає найбільш високих ступенів ущільнення в сполученні з арифметичним кодером.

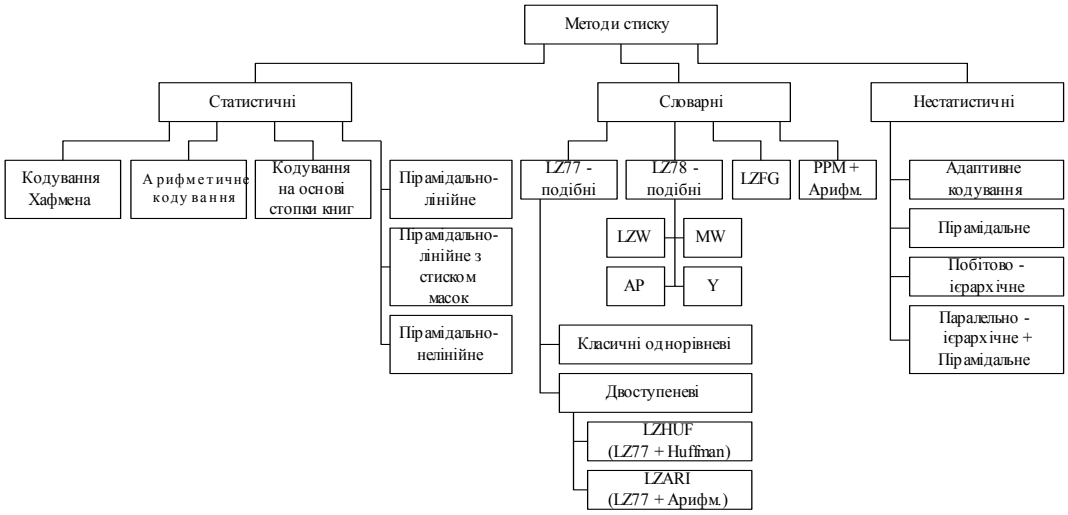


Рис. 1. **Класифікаційний аналіз методів ущільнення**

По суті, всі розглянуті вище алгоритми зводяться до двох основних ідей: «скорочувати часте» (статистичні методи), «повторювати старе» (словникові методи).

У даному розділі досліджуються нові напрями в техніці ущільнення, що зводяться до таких ідей: аналізу щільності розподілу однакових символів (статистичний метод) і аналізу багаторівневого представлення даних на кожному рівні ієрархії (група нестатистичних методів).

Адаптивний метод ущільнення. Розглянемо декілька реалізацій нестатичного ущільнення на прикладі так званого адаптивного кодера (adaptive coder). Загальна ідея організації такого кодера полягає в тому, щоб ввести в загальну таблицю кодування функції кодування, що змінюються залежно від початкових даних. Алгоритми, засновані на даній ідеї, однопрохідні і дають великий ступінь ущільнення, оскільки аналізуються локальні структури даних; обумовлені на основі функцій адаптивного кодування. Розглянемо дві реалізації функцій кодування. Перша з них пов'язана з безпосереднім завданням функцій кодування для визначеної (або визначених) кодової (кодових) комбінацій. Друга реалізація заснована на завданні функцій кодування у вигляді таблиці. Розглянемо першу реалізацію, тобто функцію адаптивного кодування Y_k − вираз (1), у якому $P_1, P_2 \{0,1\}$ – керуючі біти, P_1 − керуючий біт для диференціації комбінацій − $x_{i-1} = x_i = x_{i+1} = 0 \cup 1$ і комбінацій, для яких $x_{i-1} \oplus x_i = 0, x_{i+1} = \bar{x}_i$, P_2 − керуючий біт для диференціації комбінацій − $x_{i-1} = x_i = x_{i+1} = 0$ и $x_{i-1} = x_i = x_{i+1} = 1, x_y$ − керуюча бітова послідовність. У нижченаведених табл. 1, табл. 2, табл. 3 вхідний потік даних перетвориться у вихідний потік першого $\left(x^{(1)}\right)$ і другого $\left(x^{(2)}\right)$ ієрархічних рівнів.

Інакше, вхідний потік даних представляється у вигляді вихідної інформації $\{x_j^{(1)}\}$ і $\{x_j^{(2)}\}$, закодованої на двох ієрархічних рівнях, взаємодія між якими однозначно визначає процес її декодування.

Таблиця 1. **Кодування вхідних даних у вікні**

Вхідні дані (x) x_{i-1}, x_i, x_{i+1}	Вхідні дані на першому рівні $(x^{(1)}), x_{j-1}^{(1)}, x_j^{(1)}$	Вхідні дані на другому рівні $(x^{(2)}), x_{j-1}^{(2)}, x_j^{(2)}$
000	Y_k	Y_k
001	01	1
010	0	00
011	0	01
100	1	00
101	1	01
110	10	1
111	Y_k	Y_k

Таблиця 2.**Кодування вхідних даних між вікнами**

Вхідні дані (x) x_{i-1}, x_i, x_{i+1}	Вхідні дані на першому рівні $(x^{(1)}), x_{j-1}^{(1)}, x_j^{(1)}, x_{j+1}^{(1)}$	Вхідні дані на другому рівні $(x^{(2)}), x_{j-1}^{(2)}, x_j^{(2)}, x_{j+1}^{(2)}$
000	00	1
001	01	1
010	0	00
011	0	01
100	1	00
101	1	01
110	10	1
111	11	1

Таблиця 3. **Кодування вхідних даних у вікні**

Вхідні дані (x) x_{i-1}, x_i, x_{i+1}	Вхідні дані на першому рівні $(x^{(1)}), x_{j-1}^{(1)}, x_j^{(1)}$	Вхідні дані на другому рівні $(x^{(2)}), x_{j-1}^{(2)}, x_j^{(2)}$
000	-	-
001	01	1
010	0	00
011	0	01
100	1	00
101	1	01
110	10	1
111	-	-

Розглянемо іншу модифікацію нестатичного ущільнення на прикладі адаптивного кодера, функція кодування в якому задана таблично. Ідея реалізації такого адаптивного

кодера полягає в тому, щоб ввести в основну таблицю кодування допоміжну, дія якої залежить від структури початкових даних.

$$Y_k = \begin{cases} x_{j-1}^{(2)} = x_j^{(2)} = 1, \;\; x_{j-1}^{(1)} \oplus x_j^{(1)} = 0 \cup x_{j+1}^{(1)} \oplus x_{j+2}^{(1)} = 0, x_{i-1} = x_i = x_{i+1} = 0. \\ x_{j-1}^{(2)} = x_j^{(2)} = x_{j+1}^{(2)} = 1, \;\; x_{i-1} = x_i = x_{i+1} = 1, x_{j-1}^{(1)} \oplus x_j^{(1)} = 0 \cup x_{j+1}^{(1)} \oplus x_{j+2}^{(1)} = 0 \cup x_{j+3}^{(1)} \oplus x_{j+4}^{(1)} = 0. \\ x_{j-1}^{(1)} = x_j^{(1)} = 0, \;\; x_{i-1} = x_i = x_{i+1} = 0, x_{j-1}^{(2)} = 1, x_j^{(2)} = 0. \\ x_{j-1}^{(1)} = x_j^{(1)} = 1, \;\; x_{i-1} = x_i = x_{i+1} = 1, x_{j-1}^{(2)} = 1, x_j^{(2)} = 0. \\ x_{j-1}^{(2)} = x_j^{(2)} = 1, \;\; x_y = P_1 \cup P_1, P_2, \end{cases} \quad (1)$$

Такий алгоритм також однопрохідний і дає великий ступінь ущільнення оскільки аналізуються локальні структури даних на основі допоміжної таблиці кодування. Основна і допоміжна таблиці кодування подані відповідно в табл. 2. і табл. 3.

У загальному вигляді алгоритм адаптивного кодування полягає в такому. Відповідно до функції кодування (2) визначаються ділянки (або вікна) потоку даних, для якого виконується співвідношення:

$$n_н + n_y < n_s, \quad (2)$$

де $n_н$ – число біт, що витрачаються на кодування координати початку вікна, n_y – число біт у керуючій бітовій послідовності x_y, n_s – число заборонених комбінацій у вікні, що дають стиск вхідних даних.

Визначення 1. Комбінація вважається забороненою у вікні, якщо при її кодуванні виконуються такі співвідношення: $x_{j-1}^{(1)} \oplus x_j^{(1)} = 0$, а $x_{j-1}^{(2)} = 1$.

Визначення 2. Комбінація вважається забороненою у вікні й у той же час, що дає стиск, якщо при її кодуванні виконуються такі логічні співвідношення:
$x_{j-1}^{(2)} = x_j^{(2)} = 1, \;\; x_{j-1}^{(1)} \oplus x_j^{(1)} = 0 \cup x_{j+1}^{(1)} \oplus x_{j+2}^{(1)} = 0, x_{i-1} = x_i = x_{i+1} = 0;$ а також
$x_{j-1}^{(1)} = x_j^{(1)} = 0, \;\; x_{i-1} = x_i = x_{i+1} = 0, x_{j-1}^{(2)} = 1, x_j^{(2)} = 0,$ і
$x_{j-1}^{(1)} = x_j^{(1)} = 1, x_{i-1} = x_i = x_{i+1} = 1, x_{j-1}^{(2)} = 1, x_j^{(2)} = 0.$

Для полегшення кодування координати кінця вікна його довжина вибирається фіксованою і визначається довжиною двійкової розрядної сітки числового діапазону помилкових комбінацій координат початку вікна.

Умовою формування вікна, у межах якого діє допоміжна табл. 3., є така.

Якщо після комбінацій $x_{i-1} = x_i = x_{i+1} = 0 \cup 1$, слідують комбінації, для яких справедливо $x_{i-1} = x_i, x_{i+1} = x_i,$ то дані комбінації є дозволеними. У противному випадку ці комбінації є забороненими. Якщо у вікні є заборонені комбінації, то кодування вхідних даних проводиться відповідно до табл. 2. Розглянемо більш докладно процес формування координати початку вікна.

Початком вікна може бути послідовність двох комбінацій, для яких дотримуються такі логічні умови:

$$x_{j-1}^{(1)} \oplus x_j^{(1)} = 0 \;\cap\; x_{j+1}^{(1)} \oplus x_{j+2}^{(1)} = 0, x_{j-1}^{(2)} \oplus x_j^{(2)} = 0. \quad (3)$$

Координати початку і кінця вікна будемо визначати відповідно до таких міркувань. Координату початку вікна найзручніше визначати, підраховуючи число комбінацій, для яких виконуються логічні умови (3).

Тоді скорочення числа біт у кожному вікні визначається співвідношення

$$N = k + 2, \tag{4}$$

де k − число заборонених комбінацій у вікні.

Координату кінця найдоцільніше визначати по заданому числу заборонених комбінацій у вікні (k).

Очевидно, що вікно, розмірністю N, яке задовольняє умовам (3) кодується відповідно до табл. (2). лише в тому разі, якщо справедливе таке співвідношення:

$$N < n_{H}, \tag{5}$$

Кодування вхідних даних початку вікна у вікнах і між ними проводиться, відповідно, на основі табл. 2. і табл. 3.

Можна ввести менш жорсткі обмеження на умови (5) формування вікна, для чого необхідно використовувати керуючу кодову послідовність. Дана послідовність диференціює ті вхідні комбінації, для яких на першому ієрархічному рівні виконується логічна умова:

$$X_{j-1}^{(1)} \oplus X_{j}^{(1)} = 0, \tag{6}$$

У разі виконання умови (6) на другому ієрархічному рівні повинно бути:

$$X_{j-1}^{(2)} = 0. \tag{7}$$

Тоді умова (5) формування вікна трансформується в таке співвідношення:

$$N < n_{H} + n_{Y}, \tag{8}$$

де n_{Y} − число біт, що затрачаються на диференціацію тих вхідних кодових комбінацій, для яких справедливе виконання умов (6) і (7).

Ітераційний метод ущільнення на основі ієрархічного кодування. Ідея даного засобу полягає в розбивці початкової бітової послідовності на бітові тріади – трибітове слово і відрізняється проведенням попередньої обробки відповідно до табл. 2.

Основна мета такої попередньої обробки − в обумовленому вікні звузити динамічний діапазон представлення трибітових слів.

Після зазначеної попередньої обробки, що фактично знижує середню довжину бітових серій, мабуть найдоцільніше використовувати як початкову інформацію для подальшого кодування вихідні дані першого ієрархічного рівня (табл. 2.). Надалі ці дані першого рівня також представляють у вигляді послідовності трибітових слів. Потім із трибітових слів формуються початкові множини для наступного пірамідального кодування.

Суть ітераційного ієрархічного кодування полягає у виборі в кожній множині мінімального елемента $(a_{\min})$ і формуванні різниць $a_i - a_{\min}$. Це перетворення записується в такий спосіб:

$$\{a_i\}_{i=1}^{k} \Rightarrow \left(a_{\min}, \bigcup_{i=1}^{k} (a_i - a_{\min}) \right), \tag{9}$$

де k − число елементів у множині.

Перетворення (9) проводиться в тому разі, якщо функція керування для j-ї множини $Y_y^{(j)} = 1,$ тобто:

$$Y_{y_1}^{(j)} = \begin{cases} 1, & \text{я кщо } n_1 - n_2 \geq 1 \cup a_i - a_{\min} > n-1 \\ 0, & \text{в іншому випадку,} \end{cases} \tag{10}$$

де n_1 і n_2 − число біт, що затрачаються на кодування відповідно до і після попередньої обробки і наступного кодування, n − розрядність таблично перетворених кодових трибітових ланцюжків.

Можливість організації адаптивного кодування пов'язана з запам'ятовуванням: $\sum_{j=1}^{m} \left(Y_{y1}^{(j)} = 0 \right)$ і введенням на основі цієї інформації керуючої байтової послідовності Y_{y_2}

Запишемо умову ущільнення для перетворення вигляду (9), для чого необхідно виконання такої нерівності:

$$n_2 + n_{y1} + n_{y2} < n_1, \tag{11}$$

де n_{y1}, n_{y2} − число біт, що затрачаються в першій і другій керуючих кодових послідовностях.

Операція декодування проводиться в оберненій послідовності, тобто спочатку аналізуються керуючі послідовності Y_{y_2} , Y_{y1} , а потім реалізується операція вигляду:

$$a_{\min} + \bigcup_{i=1}^{k} (a_i - a_{\min}). \tag{12}$$

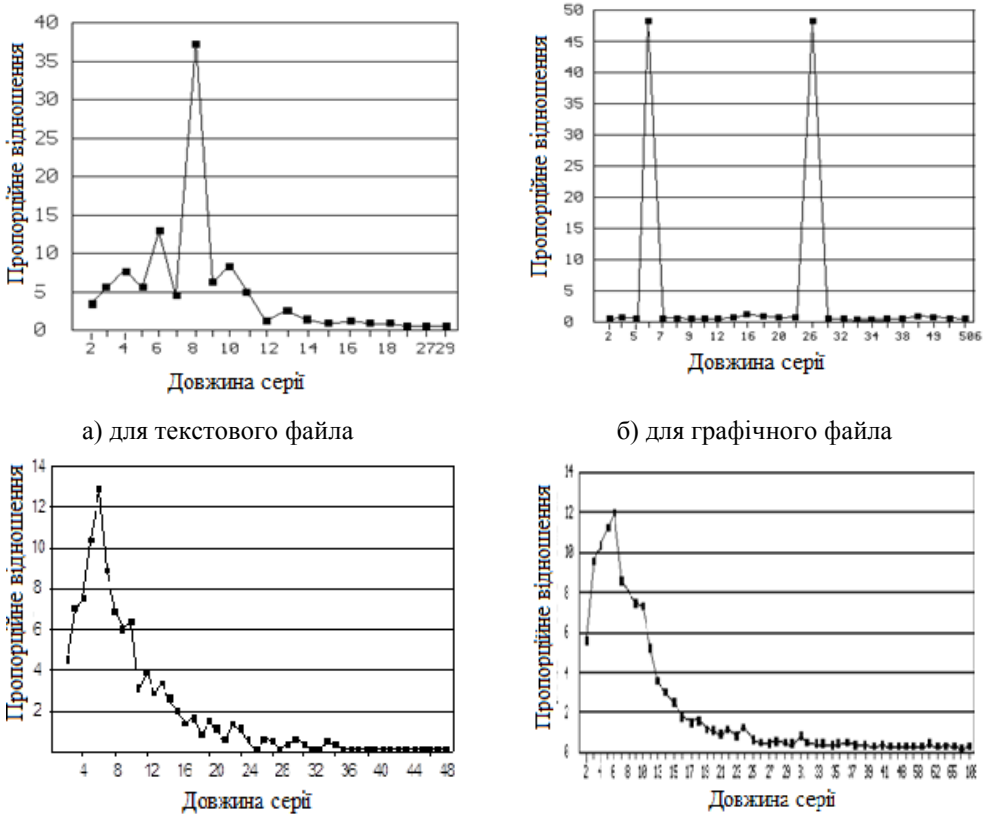
Використовуючи вихідні дані табл. 2 нескладно одержати початкову вхідну інформацію. Аналіз розглянутої групи методів свідчить, що найбільша результативність ущільнення (до 5-7%) досягається в тому разі, якщо як вхідні дані використовуються архівні дані.

Побітово-ієрархічний метод кодування. Початкові дані, наприклад файл, розбивають на бітові ланцюжки таким чином, що одноелементні серії (серія − послідовність однакових бітів) можуть знаходитися тільки на початку і кінці ланцюжка. Причому на початку не більш однієї одноелементної серії. Кожний ланцюжок закінчується однією однобітовою серією зі збільшенням ще на один біт, узятий із наступної серії (незалежно від того, яка серія наступна).

Після такої попередньої обробки файла, ланцюжки готові для кодування. Кодування полягає в ієрархічному процесі зменшення всіх серій на один біт. Причому всі одноелементні серії (крім випадку, якщо одноелементна серія знаходиться на початку ланцюжка), узагалі не розглядаються. Таким чином, початковий файл заміняється двома послідовностями. Одна складається з того, що залишилося від початкового файла, інша − із різницевих довжин ланцюжків, на які був розбитий файл.

На рис. 2, а – г і 3, а – г приведені відповідно частотний розподіл довжин серій і результати моделювання алгоритму побітово-ієрархічного кодування для різноманітних типів файлів.

Визначимо умову ущільнення для даного методу кодування. Позначимо через $\Phi_i^k(x,y)$ − функцію довжини ланцюжка, де k − кількість елементів ланцюжка, $k=\overline{3,n}$, а n − кількість біт у файлі, x − кількість початкових одноелементних серій у ланцюжку, причому $x=2,3$, y − кількість неодноелементних серій у ланцюжку, i − кількість ланцюжків. Мінімальна кількість ланцюжків у файлі може бути $i=1$, у разі, коли $k=n$, а максимальне $i=\frac{n}{3}$ при $k=3$, тобто $i=\overline{1,\frac{n}{3}}$. Позначимо $\sum_i x=X$, а $\sum_i y=Y$.



а) для текстового файла

б) для графічного файла

в) для виконуваного файла

г) для архівного файла

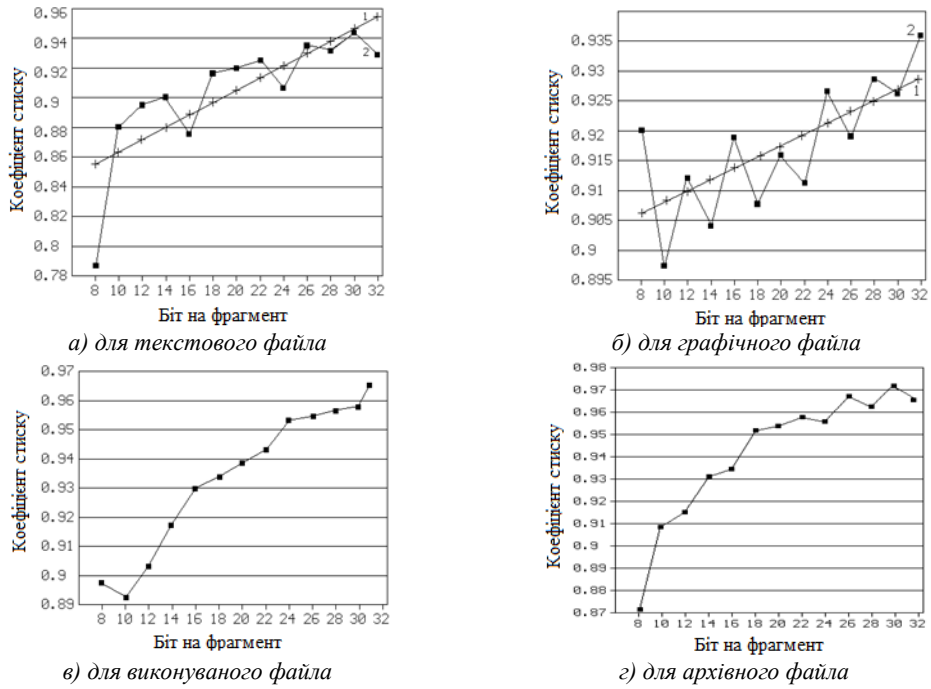
Рис. 2. **Частотний розподіл довжин серій для різноманітних типів файлів**

Функція приймає три значення для кожного i -го ланцюжка:

$$\Phi_i^k(x,y)=\begin{cases} k \text{ при } x=2,3 \\ k-(y+1) \text{ при } x=0, \\ k-(y+2) \text{ при } x=1 \end{cases} \tag{13}$$

Розглянемо приклад кодування бітової послідовності: 100101000110101100010. Даний фрагмент розбивається на три ланцюжка (1,2,3): 1001010; 00110101; 100010. Результатом кодування будуть послідовності нових ланцюжків: 10; 001; 100, а також службова інформація 7; 1; -1. Службова інформація формується на підставі середньої довжини початкових ланцюжків.

Алгоритм відновлення стисненної інформації заснований на аналізі останього елемента ланцюжка. При декодуванні наступним елементом ставиться той самий як і останій елемент в кодованому ланцюжку даних, а наступні елементи формуються почергово один за одним. За допомогою службової інформації можна контролювати кількість потрібних елементів в кожному ланцюжку даних відносно першого елемента в службовій інформації. Для вищезгаданного прикладу операція декодування буде записуватись так, як це подано на рис. 4.



а) для текстового файла *б) для графічного файла*

в) для виконуваного файла *г) для архівного файла*

Рис. 3. **Коефіцієнт ущільнення для різноманітних типів файлів**

Запишемо умову ущільнення, позначивши через p − відносний коефіцієнт ущільнення. Тому що функція $\Phi_i^k(x,y)$ приймає три значення (включаючи прийняте значення до перетворення), то кожний ланцюжок кодується по одному з двох напрямків відповідно до

Процес відновлення ланцюга данних			Службова інформація	Число елементів відновленного ланцюга
10 →	01010 →	1001010	(7)	(7)
Результат кодування	Добавлені елементи	Відновлен-ний код		
001 →	10101 →	00110101	(1)	(7+1)
100 →	0101 →	100010	(-1)	(7-1)

Рис. 4. **Процес відновлення ланцюга даних**

прийнятих функцією значеннями. Тому позначимо через $\sum_i \Phi_i^k(1)$ − суму значень функції до кодування, $\sum_i \Phi_i^k(2)$ − суму значень функції після кодування по першому напрямку, $\sum_i \Phi_i^k(3)$ – суму значень функції після кодування по другому напрямку.

$$\text{Тоді } p_1 = \frac{\sum_i \Phi_i^k(1)}{\sum_i \Phi_i^k(2)} \text{ і } p_2 = \frac{\sum_i \Phi_i^k(1)}{\sum_i \Phi_i^k(3)}.$$

$$p_1 = \frac{\sum_i \Phi_i^k(1)}{\sum_i \Phi_i^k(2)} = \frac{\sum_i k}{\sum_i (k-(y+1))} = \frac{n}{\sum_i k - \sum_i y - \sum_i 1} = \frac{n}{n\text{-}Y\text{-}i} > 1$$

$$p_2 = \frac{\sum_i \Phi_i^k(1)}{\sum_i \Phi_i^k(3)} = \frac{\sum_i k}{\sum_i (k-(y+2))} = \frac{n}{\sum_i k - \sum_i y - \sum_i 1} = \frac{n}{n\text{-}Y\text{-}2i} > 1.$$

Вирішуючи ці нерівності маємо:

$$\begin{aligned} &n > n\text{-}Y\text{-}i, && n > n\text{-}Y\text{-}2i, \\ &Y + i > 0, && Y + 2i > 0, \\ &p_1 = Y + i + 1, && p_2 = Y + 2i + 1. \end{aligned} \tag{14}$$

Отже, умовою ущільнення є величина суми кількості неодноелементних серій і кількості ланцюжків, а відносний коефіцієнт ущільнення показує, які можливості представляє результат кодування файла для необхідного обсягу службової інформації. Дана властивість методу свідчить про його ефективність для кодування переходів типу $0 \to 1$ і $1 \to 0$, тобто для ущільнення, наприклад, контурних зображень.

Алгоритм декодування ще більш простий, тому що збільшуємо кожну серію (крім першої) у всіх ланцюжках на один біт, доповнивши одиничними серіями (виходячи зі службової інформації) наприкінці кожного ланцюжка. Причому доповнювати одиничні серії починаємо з «1», якщо остання неодиночна серія ланцюжка складається з нулів і з «0», якщо – з одиниць. У результаті початковий файл відновлений. Виходячи з умови ущільнення (14) даного алгоритму кодування очевидно, що чим більше одиночних серій, тим алгоритм ефективніший, а з огляду на той факт, що середня довжина серій найменша в архівних файлах (≈2 біт) у

порівнянні з іншими типами файлів, то даний алгоритм кодування необхідно застосовувати в першу чергу до архівних файлів, додатково стискаючи їх ще до 10%.

Висновки. Моделювання побітово-ієрархічного методу ущільнення показує, що з метою зменшення середньої довжини бітових серій безпосередньо перед операцією кодування, до початкових вхідних даних доцільно застосовувати операцію їхнього перекодування відповідно до табл. 2. Причому подальший стиск відповідно до вищевикладеного алгоритму може проводитися над вихідними даними табл. 2 як першого, так і другого ієрархічних рівнів. Проведене моделювання доводить, що застосовуючи таку попередню обробку, можна підвищити ефективність ущільнення даних у середньому на 5 – 10%. Реалізація адаптивного кодування по даному методу припускає постійне коректування таблиці кодування (табл. 2) відповідно до структури вхідного потоку, що змінюється. Це не потребує значних витрат на перебалансування кодової таблиці відповідно до нових вхідних даних на кожному кроці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *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.

2. *Шевчук Б.М. та ін*.Адаптивне кодування даних, формування та передавання завадостійких пакетів інформації в мережах промислового моніторингу. /Б.М. Шевчук, В. Резаі, В.П. Зінченко // Керуючі системи та машини. – 2015. – № 4. – С. 40 – 51.

3. *Саломон Д*. Посібник зі стиснення даних / Д. Саломон, Г. Мотта. − Лондон, 2010. − 1361 с.

4. *Лопес Д*. Важливі поняття в обробці сигналів, обробці зображень та стиснення даних. – Делі: Університет Делі, 2012. − 73 с.

5. *Ватолин Д. та ін*. Методи стиснення даних / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкін. − М.: ДІАЛОГ-МІФІ, 2002. − 384 с.

6. *Рябко Б.Я*. Ефективний метод адаптивного арифметичного кодування для джерел з великими алфавітами / Б. Я. Рябко, А. Н. Фіонова // Проблеми передачі інформації. − 1999. − С. 34-39.

7. *Тимченко Л.І., Кокряцька Н.І. та ін*. Паралельно-ієрархічні мережі для оброблення зображень.Теоретичні дослідження : монографія / Л.І. Тимченко, Н.І. Кокряцька, О.А. Герцій, М.С. Петровський, Д.С. Степанюк. – Полтава: АСМІ, 2017. – 469 с.

8. *Тимченко Л.І.* та ін. Паралельно-ієрархічні мережі для оброблення біомедичних зображень та зображень плям лазерних пучків. Експериментальні дослідження : монографія / Л.І. Тимченко, Н.І. Кокряцька, О.А. Герцій, М.С. Петровський, Д.С. Степанюк. – Полтава: АСМІ, 2017. – 363 с.

REFERENCES

1. V. A. Luzhetsky. *Metodi uschilnenya danih na osnovi vidkidanya poslidovnostey nuliv ta odinic* [Methods of data compression based on discarding sequences of zeros and ones]. A. Luzhetsky, T. M. Cheboraca // *Informaciyni tehnologii ta computerna ingeneria* [Informational technologies and computer engineering]. – 2014. – №. 1. – S. 18 – 26.

2. *Adaptivne coduvanya danih, formuvanya ta peredavanya zavodostiykih paketiv informacii v merazhah promislovogo monitoringu* [Adaptive encoding of data, formation and transmission of error-correcting packets of information networks industrial monitoring] / B. M. Shevchuk, V. Ruta, V. P. Zinchenko // *Keruyuchi sistemi ta mashini* [Control systems and machines]. – 2015. – №. 4. – S. 40 – 51.

3. Salomon, D. Handbook of data compression / D. Salomon, G. Motta. − London, 2010. − 1361 S.

4. Lopez D. Important concepts in signal processing, image processing and data compression. − University Of Delhi, 2012. − 73 p.

5. *Metodi stisnenya danih* [Methods data compression] / D. Vatolin, A. Ratushniak, N. Smirnov, V. UCN − Moscow: DIALOG-MIFI, 2002. − 384 p.

6. Ryabko, B. Y. *Efectivniy metod adaptivnoga arifmetichnogo coduvanya dlya dzerel z velicimi alfavitami* [An effective method of adaptive arithmetic coding for sources with large alphabets] / B. Y. Ryabko, A. N. Fionova // Problems of information transmission. − 1999. − P. 34-39.

7. Timchenko L. I., Kokracka N. I., Hertz V. A., Petrovsky M. S., Stepanyuk, D. S. *Paralelno-irarhichni merezi dlya obrobleny zoobragen. Teoritichi doslidgenya: monographia* [Parallel-hierarchical network for image processing.Theoretical studies : monograph] / L. I. Timchenko, N. I. Kocracka, A. Hertciy, M. S. Petrovsky, D. S. Stepanyuk — Poltava: ASMI, 2017. – 469 p.

8. Timchenko L. I., Kokracka N. I., Hertz V. A., Petrovsky M. S., Stepanyuk, D. S. *Paralelno-irarhichni merezi dlya obrobleny biomedichnih zoobragen ta zoobragen plyam lazernih puchkiv. Experementalni doslidgenya: monographia* [Parallel-hierarchical network for processing biomedical images, and images of spots of laser beams. Experimental studies : monograp] / L. I. Timchenko, N. I. Kocracka, A. Hertciy, M. S. Petrovsky, D. S. Stepanyuk – Poltava: ASMI, 2017. – 363 p.

Л. И. Тимченко, д.т.н., профессор
(заведующий кафедры «Телекоммуникационные технологии и автоматика», Государственный университет инфраструктуры и технологий)
Н.І. Кокряцкая, к.т.н., доцент
(доцент кафедры «Телекоммуникационные технологии и автоматика», ГУИТ)
И. Д. Ивасюк, к.т.н., доцент
Ю. В. Майстренко (аспирант, ГУИТ)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ УПЛОТНЕНИЯ ДАННЫХ НА ОСНОВЕ ИЕРАРХИЧЕСКОЙ КОДИРОВКИ

В данной статье проводится анализ методов уплотнений данных используется на данный момент в компьютерной технике и рассматриваются пути разработки технологии сжатия данных на основе иерархического кодирования, в частности, итерационного и побитового методов.

Ключевые слова: *уплотнение, сжатие, кодирование, информация, данные, кодер, декодер, модель, алгоритм, бит, адаптивное кодирование, иерархическое кодирование, побитовое кодирования.*

Leonid I. Timchenko, Doctor of Science (Technical Sciences), Professor
(head of the Department of Telecommunications Technologies and Automatics», State University for Infrastructure and Technology)
Natalia I. Kokryatskaya, PhD (Technical Sciences), Associate Professor
(associate Professor of the Department of Telecommunications Technologies and Automatics», State University for Infrastructure and Technology)
I. D. Ivasuk (PhD (Technical Sciences), Associate Professor)
Y. V. Maystrenko (postgraduate of State University for Infrastructure and Technology)

INFORMATION TECHNOLOGY DATA COMPRESSION BASED ON A HIERARCHICAL CODING

This article analyzes the methods of compaction of data currently used in computer technology and examines the ways of developing data compression technology based on hierarchical coding, in particular iterative and bitwise methods.

Keywords: *compaction, compression, coding, information, data, encoder, decoder, model, algorithm, bits, adaptive coding, hierarchical coding, a bitwise encoding.*

Стаття надійшла до редакції 15.12.2018 р.