УДК 681.397.6

Л. Я. Глинчук

СТИСНЕННЯ ДВІЙКОВИХ ПОСЛІДОВНОСТЕЙ НА ОСНОВІ СИСТЕМИ ЧИСЛЕННЯ ШТЕРНА–БРОКО

The new algorithm of compression is examined without losses for binary sequences, that is based on the use of scale of notation of Shterna–Broko. For the estimation of efficiency of algorithm middle length of binary sequence is determined after a compression, mid-coefficient of compression.

Keywords: *binary sequence, compression, aspect ratio.*

Розглянуто новий алгоритм стиснення без втрат для двійкових послідовностей, що базується на використанні системи числення Штерна–Броко. Для оцінки ефективності алгоритму проаналізовано середню довжину двійкової послідовності після стиснення та середній коефіцієнт стиснення.

Ключові слова: *двійкова послідовність, стиснення, коефіцієнт стиснення.*

Як відомо, у комп'ютері вся інформація відображається у двійкових кодах, тому, зважаючи на зростання інформаційних потоків у системах зв'язку і комп'ютерних мережах, задача стиснення на рівні двійкових кодів є сьогодні актуальною.

Для стиснення розроблені та використовуються різні методи – від застосування методів Шеннона, Фано, Хаффмана до арифметичного кодування та інших. Однак важливу роль у методах перетворення кодів відіграє специфіка стискуваної інформації. Наприклад, задачі стиснення телевізійних зображень мають суттєві відмінності від задач обробки текстових повідомлень, таблиць чи графіків. Наявні теоретичні підходи та методи не завжди досить ефективно справляються з вищезазначеними та іншими специфічними задачами [1].

Ця стаття присвячена питанню стиснення двійкової інформації без втрат з використанням елементів системи числення Штерна–Броко[3]. Аналогічні алгоритми досліджено та описано в роботах [2], [4], [6]. Суть алгоритму полягає у використанні відповідності між дробом дерева та двійковим шляхом, за яким він знаходиться у дереві. Цей підхід відрізняється від [4], [6] досить простими алгоритмами, нескладною апаратною реалізацією, що підвищує надійність перетворення кодів за наявності високого коефіцієнта стиснення.

Коефіцієнт стиснення [5] розглядаємо як відношення кількості бітів нестиснутих кодів до стиснутих. Якщо в результаті стиснення коефіцієнт дорівнює одиниці, то стиснення немає, якщо більший від одиниці – стиснення відбулося.

Основна частина. Основою методу є система числення (дерево) Штерна–Броко, для якої задано спеціальну нумерацію елементів, тобто кожному елементу (елементом є нескоротний додатний дріб) ставиться у відповідність двійкова послідовність – шлях, за яким можна знайти відповідний дріб. Алгоритми, що дають змогу за двійковою (бінарною) послідовністю обчислити дріб і навпаки детально описано у [3].

Алгоритм стиснення (блок-схема зображена на рис. 1) на основі системи числення Штерна–Броко полягає у переході від бінарної послідовності до дробу, який їй відповідає. Алгоритм відновлення (блок-схема на рис. 2) – зворотний. Оскільки ця система числення будується в глибину, то яку б ми двійкову послідовність не вибрали б, їй у відповідність завжди можна поставити дріб.

Рис. 1. Блок-схема алгоритму стиснення.

Рис. 2. Блок-схема алгоритму відновлення.

Отже, для застосування такого алгоритму до будь-яких двійкових комбінацій на вхід потрібно передати чисельник та знаменник їх нескоротних дробів.

Щоб оцінити ефективність такого алгоритму, розглянемо його характеристики. Для цього проведемо деякі розрахунки. Нехай на вхід подається двійкова послідовність довжиною n біт. Тоді в результаті перетворення отримаємо два числа: c − чисельник дробу, z − знаменник. Кількість бітів для зберігання чисельника обчислюватимемо так: переводимо чисельник з десяткової системи числення у двійкову, рахуємо довжину послідовності d_c – кількість бітів для зберігання чисельника та довжину d_z − кількість бітів для зберігання знаменника. Сумарна довжина l стиснутої двійкової послідовності довжиною n бітів: $l = d_c + d_z$.

Наприклад, вхідна двійкова послідовність 0111111111 після перетворення 1 в алгоритмі відобразиться як дріб 10/11, на чисельник та знаменник потрібно по 4 біти, тоді l = 8 бітів, маємо зменшення довжини на 2 біти.

Оскільки довжина l_i стиснутої i-ї двійкової послідовності довжиною n біт залежить від кількості одиниць у послідовності та їх розміщення, то доцільно було б для подальших досліджень розрахувати середнє значення $l_{i_{cp}}$ за формулою $l_{i_{cp}} = \frac{\sum_{i=1}^{m} l_i}{m}$, де m – кількість можливих двійкових комбінацій, що обчислюється за формулою $m = C_n^k = \frac{n!}{k!(n-k)!}$, де n – довжина послідовності, у якій k одиниць.

Коефіцієнт стиснення i-ї двійкової послідовності довжиною n біт з кількістю одиниць k буде таким: $K_i = \frac{n}{l_{i_{cp}}}$.

На практиці кодові послідовності з’являються з різною ймовірністю p_i, тоді середня довжина дробів після стиснення двійкових кодових комбінацій

$$l_{cp} = \sum_{i=0}^{n} p_i \cdot l_{i_{cp}} .$$

Середній коефіцієнт стиснення Y_{cp} в результаті перетворення послідовності кодових комбінацій обчислимо за формулою $Y_{cp} = n / l_{cp}$. Чим більше Y_{cp}, тим ліпше стиснення.

Як видно з останньої формули, для обчислення середнього коефіцієнта стиснення $Y_{ср}$ необхідно обчислити $l_{ср}$. У табл. 1 наведено розрахунки середніх значень довжин двійкових послідовностей після стиснення для кодів довжиною $n = 8, 16, 32$ розрядів з різною кількістю одиниць в кодах. Було задано чотири варіанти розподілу ймовірностей появи кодових комбінацій − p_i (варіанти однакові для кожного n). Таблиця 2 містить результуючі дані коефіцієнта стиснення $Y_{ср}$.

Таблиця 1. Розрахунки середніх значень довжин $l_{ср}$ у випадку різних варіантів ймовірностей p_i

n		k Вар.	0	1	2	3	4
8	$l_{ср}$	$l_{i_{ср}}$	5,000	7,620	9,320	10,300	10,660
	5,262	p = варіант 1	0,900	0,100			
		$p_i \cdot l_{i_{ср}}$	4,500	0,762			
	5,694	p = варіант 2	0,800	0,100	0,100		
		$p_i \cdot l_{i_{ср}}$	4,000	0,762	0,932		
	6,486	p = варіант 3	0,600	0,200	0,100	0,100	
		$p_i \cdot l_{i_{ср}}$	3,000	1,524	0,932	1,030	
	7,314	p = варіант 4	0,400	0,300	0,100	0,100	0,100
		$p_i \cdot l_{i_{ср}}$	2,000	2,286	0,932	1,030	1,066
16	$l_{ср}$	$l_{i_{ср}}$	6,000	10,000	12,910	15,220	16,950
	6,400	p = варіант 1	0,900	0,100			
		$p_i \cdot l_{i_{ср}}$	5,400	1,000			
	7,091	p = варіант 2	0,800	0,100	0,100		
		$p_i \cdot l_{i_{ср}}$	4,800	1,000	1,291		
	8,413	p = варіант 3	0,600	0,200	0,100	0,100	
		$p_i \cdot l_{i_{ср}}$	3,600	2,000	1,291	1,522	
	9,908	p = варіант 4	0,400	0,300	0,100	0,100	0,100
		$p_i \cdot l_{i_{ср}}$	2,400	3,000	1,291	1,522	1,695
32	$l_{ср}$	$l_{i_{ср}}$	7,000	12,410	16,910	20,640	23,700
	7,541	p = варіант 1	0,900	0,100			
		$p_i \cdot l_{i_{ср}}$	6,300	1,241			
	8,532	p = варіант 2	0,800	0,100	0,100		
		$p_i \cdot l_{i_{ср}}$	5,600	1,241	1,691		
	10,437	p = варіант 3	0,600	0,200	0,100	0,100	
		$p_i \cdot l_{i_{ср}}$	4,200	2,482	1,691	2,064	
	12,648	p = варіант 4	0,400	0,300	0,100	0,100	0,100
		$p_i \cdot l_{i_{ср}}$	2,800	3,723	1,691	2,064	2,370

Таблиця 2. Середні коефіцієнти стиснення $Y_{ср}$, обчислені на підставі знайдених у табл. 1 середніх довжин $l_{ср}$

n	8	16	32
p = варіант 1	1,520	2,500	4,243
p = варіант 2	1,405	2,256	3,751
p = варіант 3	1,233	1,902	3,066
p = варіант 4	1,094	1,615	2,530

З табл. 1 та 2 можна зробити такі висновки:

1) чим більша ймовірність появи кодових комбінацій з малою кількістю одиниць чи нулів, як буде описано далі, тим ліпше стиснення, і навпаки;

2) відповідно до обраних комбінацій ймовірностей, стиснення стає ефективніше зі зростанням довжини кодової комбінації.

Аналіз формули середнього коефіцієнта стиснення $Y_{ср}$ дає змогу зробити висновок, що можливі такі параметри кодових комбінацій, коли результуюча двійкова послідовність буде довша від початкової, тобто Y<1. Для визначення інтервалів стиснення ми обчислювали коефіцієнти стиснення K_i та розраховували середнє значення $l_{i_{ср}}$ для кодів довжиною n = 8, 16, 32 розрядів з різною кількістю одиниць в кодах. Доведено, що значення середніх коефіцієнтів стиснення для кількостей одиниць i та n-i збігаються. Результати розрахунків наведені у табл. 3, 4, в яких інтервали стиснення виділені сірим кольором.

Таблиця 3. Коефіцієнти стиснення K_i та середні значення $l_{i_{ср}}$ для кодів довжиною n = 8, 16, 32

n	k	0	1	2	3	4	5	6	7	8
8	m	1	8	28	56	70				
	$l_{i_{ср}}$	5,00	7,62	9,32	10,30	10,66				
	K_i	1,60	1,05	0,86	0,78	0,75				
16	m	1	16	120	560	1820	4368	8008	11440	11700
	$l_{i_{ср}}$	6,00	10,00	12,91	15,22	16,95	18,20	19,08	19,57	21,71
	K_i	2,67	1,60	1,24	1,05	0,94	0,88	0,84	0,82	0,74
32	m	1	32	496	4960	35960	201376	906192	3365856	10518300
	$l_{i_{ср}}$	7,00	12,41	16,91	20,64	23,70	26,34	28,60	30,54	32,12
	K_i	4,57	2,58	1,89	1,55	1,35	1,22	1,12	1,05	0,99

	k	9	10	11	12	13	14	15	16
32	m	28048800	64512240	129024480	...	...	...	...	...
	$l_{i_{ср}}$	33,64	34,85	35,84	36,65	37,26	37,67	37,89	38,34
	K_i	0,95	0,92	0,89	0,87	0,86	0,85	0,84	0,83

Таблиця 4. Зведені результати розрахунків K_i для різних n

k / n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
8	,6(	,0[	0,86	0,78	0,75												
16	[,6[	,6(	,2[	,0[	0,94	0,88	0,84	0,82	0,74								
32	[,5[	[,5[	,8[	,5[	,3[	,2[	,1[	,0[	0,99	0,95	0,92	0,89	0,87	0,86	0,85	0,84	0,83

Оскільки коефіцієнти стиснення симетричні відносно $k = n/2$, то отримаємо такі графіки (рис. 3), які ілюструють вищезазначені обчислення:

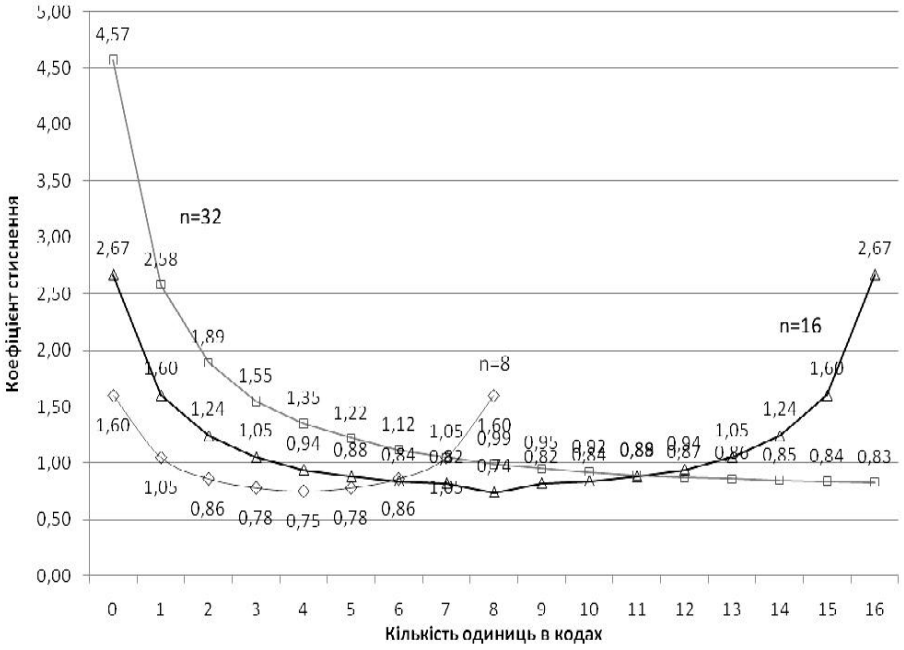

Рис. 3. Залежності коефіцієнтів стиснення від кількостей одиниць у кодах довжини n.

Запропонований алгоритм виконує відображення 2^n рівномірних кодів на нерівномірні. При цьому кодам з кількістю одиниць (нулів) з інтервалів [0; n/4–1] та [3/4n+1; n] ставляться у відповідність коротші відносно n коди, а іншим – довші. Тобто половина кодів відображаються в коротші нерівномірні коди, а інша половина – в довші.

Як видно з рис. 3, між інтервалами стиснення є внутрішній інтервал, в якому після перетворення відбувається не стиснення, а подовження кодових комбінацій. Посередині цього інтервалу довжина перетворюваного коду набуває максимального значення.

ВИСНОВКИ

Отже, розглянутий метод дає змогу перетворювати будь-які двійкові послідовності довжиною n та може використовуватися для зменшення кодової надлишковості, коли переважають ймовірності появи кодових комбінацій з малою кількістю одиниць або нулів. Крім того, використовуються досить прості алгоритми, що дають можливість реалізувати їх на апаратному рівні та поліпшити надійність роботи.

Розроблений алгоритм дає подібні результати із запропонованим та дослідженим алгоритмом [6]. Але, наприклад, під час застосування алгоритмів до

стиснення деякого бінарного зображення довжиною 256 бітів коефіцієнт стиснення виявився вищим від коефіцієнта стиснення алгоритму [6].

Для порівняння з іншими аналогічними програмами було використано деякий тестовий набір файлів CalgCC (з набору взято 8 файлів). У табл. 5 наведені результати застосування запропонованого алгоритму, коли довжина $n=8$ (названого NA) до набору CalgCC та результати інших архіваторів.

Таблиця 5. Коефіцієнти стиснення різних архіваторів та NA

	ARJ	PKZIP	ACE	RAR	7-Zip	**NA(8)**
Bib	3,08	3,16	3,38	3,39	3,62	**1,24**
Book1	2,41	2,46	2,78	2,80	2,94	**1,23**
Book2	2,90	2,95	3,36	3,39	3,59	**1,24**
News	2,56	2,61	3,00	3,00	3,16	**1,23**
Paper1	2,84	2,85	2,91	2,93	3,07	**1,22**
Paper2	2,74	2,77	2,86	2,88	3,01	**1,22**
Progp	4,32	4,37	4,55	4,57	4,73	**1,18**
Trans	4,65	4,79	5,19	5,23	5,56	**1,26**
Середнє значення	3,188	3,245	3,504	3,524	3,710	**1,228**

З табл. 5 можна зробити висновок, що не має значення, яка інформація у файлі, застосовуючи запропонований алгоритм, одержуємо дає коефіцієнти стиснення, які приблизно однакові для кожного файла з набору.

У перспективі планується:

1) провести додаткові дослідження можливостей попередньої обробки (препресингу), що ставить у відповідність кодам з близькими кількостями нулів та одиниць коди, які стискаються запропонованим алгоритмом;

2) дослідити залежність коефіцієнта стиску від довжини вхідної послідовності;

3) удосконалити програми запропонованого алгоритму для визначення часу кодування та декодування.

1. *Кодирование* информации (двоичные коды) / И. Т. Березюк и др. – Харьков: Вищ. шк., 1978. – 252 с.
2. *Борисенко А. А., Протасова Т. А.* О комбинаторном подходе к сжатию информации // Вісник СумДУ. – 2006. – № 4(88). – С. 56–61.
3. *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики. – М.: Мир, 1998. – 703 с.
4. *Кулик И. А., Харченко С. Н.* Динамическое адресно-векторное сжатие двоичных последовательностей // Вісник СумДУ. – 2006. – № 4(88). – С. 73–79.
5. *Сжатие* информации – Викизнание. – [Електронный ресурс]. – Режим доступу: *www.wikiznanie.ru/ru-wz/index.php/Компрессия_данных*.
6. *Чередниченко В. Б.* Метод сжатия двоичных кодов на основе биномиальных чисел // Вісник СумДУ. – 2006. – № 4(88). – С. 61–68.

Волинський національний університет ім. Лесі Українки

Одержано 22.03.2010