УДК 519.6: 621.391

В.А. Лужецький, Л.А. Савицька

Вінницький національний технічний університет, Вінниця

МОДЕЛІ І МЕТОДИ АДАПТИВНОГО УЩІЛЬНЕННЯ ДАНИХ НА ОСНОВІ ЛІНІЙНОЇ ФОРМИ ФІБОНАЧЧІ

Розглянуто особливості формування числових моделей джерела даних і правила кодування, що забезпечують підвищення коефіцієнта ущільнення за рахунок адаптації виконуваних перетворень до конкретного змісту ущільнюваних даних. Запропоновано алгоритм прискореного формування лінійної форми Фібоначчі. Запропоновано і досліджено два методи адаптивного ущільнення даних на основі лінійної форми Фібоначчі.

***Ключові слова**: адаптивне ущільнення, джерело даних, числова модель, правила кодування, лінійна форма Фібоначчі, коефіцієнт ущільнення.*

Вступ

Обсяги інформації, яка зберігається і передається, з кожним роком значно зростають, що вимагає збільшення об'ємів пам'яті і наявності високошвидкісних каналів передавання в сучасних комп'ютерних системах. Це призводить до постійного збільшення вартості таких систем. Тому економічно більш вигідним є зменшення обсягів інформації шляхом використання методів ущільнення. Ущільнення інформації скорочує обсяг пам'яті, що необхідна для її зберігання, і кількість часу, який потрібен для її передавання каналом з фіксованою пропускною здатності.

Існує кілька підходів до ущільнення інформації, що породжують цілу низку методів ущільнення, для реалізації яких, у свою чергу, використовується величезна кількість алгоритмів [1-4]. Одні з них мають науковий характер, а інші знайшли практичну реалізацію у вигляді архіваторів.

Теоретичні дослідження і практика застосування архіваторів показали, що не існує універсального методу ущільнення, який забезпечував би однаковий степінь ущільнення для різних типів даних. Тому продовжуються пошуки нових підходів, що були б ефективними для певних типів сигналів і даних з точок зору степеня ущільнення, програмної й апаратної реалізації. Однак дані навіть одного типу, з погляду ущільнення, мають різні властивості і характеристики. З огляду на це, останнім часом прагнуть до створення адаптивних алгоритмів ущільнення даних.

І хоча на практиці широко використовуються архіватори, створені на основі методів й алгоритмів ущільнення даних, що враховують статистику символів у повідомленні [1, 2] або базуються на побудові словника [3, 4], однак продовжуються пошуки нових підходів до ущільнення даних.

У роботах [5, 6] запропоновано оригінальний спосіб представлення цілих чисел у вигляді, так називаної, лінійної форми Фібоначчі, яка забезпечує скорочення розрядності представлення великих чисел. Така форма представлення є основою нового методу ущільнення даних, який ґрунтується на оптимізуючих властивостях чисел Фібоначчі.

Постановка задач

У роботі [7] показано, що існує принципова можливість ущільнення даних з використанням лінійної форми Фібоначчі. Однак відсутні дослідження цього методу стосовно ефективності ущільнення певних типів даних.

У роботі [5] описується алгоритм одержання лінійної форми Фібоначчі, який вимагає виконання операцій ділення. Ця операція для чисел великої розрядності виконується дуже повільно, що призводить до великих витрат часу на реалізацію даного алгоритму.

У [6] обґрунтовано можливість одержання лінійної форми Фібоначчі без використання операції ділення і запропоновано прискорений алгоритм. Однак реалізація і цього алгоритму вимагає відносно великої кількості обчислень, що пов'язано з необхідністю множення надвеликих чисел. Оскільки основна складність процедури ущільнення визначається саме перетворенням числа в лінійну форму Фібоначчі, то виникає потреба у прискоренні цього перетворення.

Одним з найважливіших положень теорії ущільнення інформації є висловлена в [8] ідея поділу процесу ущільнення на дві процедури: моделювання і кодування.

Моделювання визначає характеристики джерела даних, що ущільнюються. Залежно від того, якою моделлю джерела сигналу описується інформація, її поділяють на бінарну інформацію (виконувані файли, програмні бібліотеки і т.д.) і текстову. Окремим випадком текстової або бінарної інформації є числові дані (залежно від того, у якому вигляді вони представлені – ASCII коді або двійковому).

Метою кодування є перетворення потоку символів у потік біт мінімальної довжини.

У роботі [9] запропоновано узагальнену модель процесу адаптивного ущільнення, яка передбачає наявність множини правил моделювання джерела даних, множини правил кодування даних і функції оптимізації, що забезпечують можливість адаптації до конкретного змісту ущільнюваних даних.

Запропонований в роботі [7] метод ущільнення даних з використанням лінійної форми Фібоначчі передбачає тільки одну числову модель даних та одне правило кодування. Отже, для підвищення коефіцієнта ущільнення за рахунок адаптації потрібно сформувати набори моделей даних і правил кодування.

Метою дослідження є підвищення степеня ущільнення даних на основі лінійної форми Фібоначчі шляхом розробки методів адаптивного ущільнення.

Для досягнення поставленої мети розв'язувалися такі задачі:

- розробка набору числових моделей даних;
- розробка правил прискореного кодування даних;
- розробка і дослідження методів адаптивного ущільнення даних на основі лінійної форми Фібоначчі.

Числові моделі даних

Вихідні дані, що підлягають ущільненню, будемо розглядати як послідовність символів 0 і 1. Відповідно до числової моделі джерела даних, послідовність символів розбивається на блоки, що містять деяку кількість символів, і кожному символу в блоці відповідає своя певна вага. Якщо вага k-го символу блоку дорівнює 2^k (k=0,1..., n-1), то блок є двійковим кодом деякого числа.

Для практичної реалізації більш 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 довжиною l байтів представляється як послідовність l чисел:

$$s_0, s_1, s_2, ..., s_{l-1}, \qquad (1)$$

де s_j - числовий еквівалент ASCII коду j-го байту (j=0,1,2,…,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:

$$N_м = \sum_{j=0}^{l-1} s_j 256^j , \qquad (2)$$

$$N_c = \sum_{j=0}^{l-1} s_{(l-j-1)} 256^{(l-j-1)} . \qquad (3)$$

Відмінність цих формул полягає тільки в порядку використання елементів послідовності (1) відносно їх номерів: від молодшого до старшого та навпаки. Тут число 256 є основою системи числення і для кожного блоку даних вона незмінна (фіксована). Виходячи з цього, позначимо моделі джерела даних таким чином: $M_{фм}$ і $M_{фс}$, відповідно.

Аналіз змісту блоків ущільнюваних даних стосовно числових еквівалентів ASCII кодів показує, що не в кожному блоці є байт з числовим еквівалентом 255. Тому є можливість визначати числовий еквівалент блоку через представлення в системі числення з основою не тільки 256.

Покажемо це на прикладі. Нехай блок даних складається з байтів з такими числовими еквівалентами: 176; 48; 94; 36. Тут найбільшим є число 176. Отже, можна визначити числовий еквівалент блоку, використовуючи представлення в системах числення з основами від 177 до 256. Тобто для такого блоку можна отримати 80 різних числових значень.

Для кожного блоку даних визначається максимальне значення B_{max} серед значень числових еквівалентів байтів. Виходячи з цього, є можливість обчислити такі числові еквіваленти:

$$N_м = \sum_{j=0}^{l-1} s_j B^j , \qquad (4)$$

$$N_c = \sum_{j=0}^{l-1} s_{(l-j-1)} B^{(l-j-1)} , \qquad (5)$$

де $B = B_{max} + 1$.

Тут число B є основою системи числення і для кожного блоку даних воно змінюється від $(B_{max} + 1)$ до 256. Виходячи з цього, позначимо моделі джерела даних таким чином: $M_{зм}$ і $M_{зс}$, відповідно.

Для забезпечення більшої кількості таких моделей пропонується використовувати багатомодульне представлення чисел:

$$N = (\alpha_1, \alpha_2, ..., \alpha_k) , \qquad (6)$$

де $\alpha_i = N \bmod p_i$, $i = 1,2,...,k$.

Таке представлення забезпечує не тільки менше значення B_{max}, але і зміну статистики значень. Наприклад, для модулів $p_1 = 37$, $p_2 = 41$ і $p_3 = 47$, які забезпечують представлення двобайтних чисел, маємо таке: 1739=(6, 203)=(0, 17, 0); 1519=(5, 239)=(2, 2, 15); 6493=(25, 93)=(18, 15, 7).

Замість послідовності байтів з числовими еквівалентами (6, 203, 5, 239, 25, 93) маємо послідовність (0, 17, 0, 2, 2, 15, 18, 15, 7).

Правила кодування

Відомо [10], що будь-яке ціле додатне число z можна представити у вигляді:

$$z = \sum_{l=2}^{n} a_l \phi_l(l) ,$$

де $a_l = \{0;1\}$; $\phi_l(l)$ - l-е число Фібоначчі, що обчислюються за формулою:

$$\phi_l(l) = \phi_l(l-1) + \phi_l(l-2)$$

для $\phi_l(0) = 0$, $\phi_l(1) = 1$.

Набір $a_n a_{n-1} \ldots a_2 a_1$ називається кодом Фібоначчі числа z [10].

Для визначення значень a_l використовується такий алгоритм.

Алгоритм 1.

П.1. У послідовності чисел Фібоначчі знайти елемент, що задовольняє умові $z = \phi_l(l) + r$, де r - залишок, $0 \le r < \phi_l(l-1)$.

П.2. $a_l = 1$.

П.3. Залишок r прийняти за початкове число і перейти до п.1.

Пункти 1-3 виконувати доти, поки залишок r не буде дорівнювати нулю.

Усі a_l, значення яких не дорівнюють 1, мають значення, що дорівнюють нулю.

Для реалізації даного алгоритму пропонується формувати убутну послідовність чисел Фібоначчі, користуючись рекурентним співвідношенням:

$$\phi_l(l) = \phi_l(l+2) - \phi_l(l+1), \quad l = n, n-1, \ldots 2$$

для початкових значень $\phi_l(n+2), \phi_l(n+1)$.

У лінійній формі Фібоначчі будь-яке ціле додатне число *z* представляється у вигляді [10]:

$$z = q_2 \phi_l(j+1) + q_l \phi_l(j) , \quad j = 1; 2; \ldots \qquad (7)$$

де q_l і q_2 - цілі додатні числа (координати представлення); j - індекс представлення.

Цю форму будемо позначати ЛФФ(*).

Із (7) випливає, що *z* визначається трьома цілими додатними числами, тобто $z = f(q_1, q_2, j)$.

Існує множина представлень (7), серед яких є єдине мінімальне представлення (М-представлення), що має властивість $\{\min(q_l) ; \min(q_2)\}$.

У байтовому представленні ЛФФ(*) має таку структуру:

$$\{Q_j \parallel l_{Q_l} \parallel l_{Q_2} \parallel Q_l \parallel Q_2\}, \qquad (8)$$

де Q_j - код числа *j* (2 байти); l_{Q_l} - довжина коду числа q_l в байтах (2 байти); l_{Q_2} - довжина коду числа q_2 в байтах (2 байти); Q_l - код числа q_l (змінна кількість байтів); Q_2 - код числа q_2 (змінна кількість байтів).

Для спрощення процедури перетворення числа z в ЛФФ(*) пропонується алгоритм, що базується на ідеї, висловленій в роботі [10].

Обчислюється добуток числа z на обернене значення «золотої» пропорції:

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

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

$$w_i = w_{i-2} - w_{i-1} , \qquad i = 2, 3, \ldots .$$

поки $w_i > 0$.

Якщо обчислення були виконані j разів, то представленню (7) будуть відповідати такі мінімальні координати: $q_2 = w_{j+1}$ і $q_l = w_j$.

Враховуючи, що $\lim\limits_{l\to\infty} \dfrac{\phi_l(l)}{\phi_l(l+1)} = \dfrac{2}{1+\sqrt{5}}$, можна замінити множення числа z на обернене значення «золотої» пропорції зсувом управо на один розряд кода Фібоначчі числа $z = (a_n a_{n-1} \ldots a_2 a_1)$. Тобто $q = \vec{z} = (0 a_n a_{n-1} \ldots a_2)$. Одночасно з цим обчислюється $w_2 = w_0 - w_1 = \vec{q} = (00 a_n a_{n-1} \ldots a_3)$. Для визначення значень a_l використовується алгоритм 1, а числа q і w_2 обчислюються починаючи з a_n за формулами:

$$q = \sum_{j=n}^{1} a_j \phi_l(j-1) \text{ і } w_2 = \sum_{j=n}^{2} a_j \phi_l(j-2) .$$

Дослідження, проведені авторами статті, показали, що для певних чисел ЛФФ(*) в 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.

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

$$N_1 = \sum_{l=s}^{n} a_l \phi_l(l), \qquad (9)$$

$$N_2 = \sum_{l=2}^{s-1} a_l \phi_l(l). \qquad (10)$$

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

Таке правило кодування забезпечує не тільки підвищення коефіцієнта ущільнення, але і спрощення обчислень.

Методика та результати дослідження неадаптивного методу ущільнення

Дослідження здійснювалося з використанням програмного засобу, що реалізує неадаптивний метод ущільнення на основі лінійної форми Фібоначчі полягає в такому. Ущільнювані дані розбиваються на блоки довжини l байтів. У загальному випадку, останній (k-й) з послідовності блоків може мати довжину $l_k < l$. Для кожного блоку визначається числовий еквівалент N за формулою (2), що відповідає моделі джерела даних $M_{фм}$. Якщо N=0, то формується тільки ознака p=0, а якщо N>0, то формується ознака p=1 і здійснюється перетворення числа N в лінійну форму Фібоначчі.

Ущільнені дані мають таку структуру:

$$S = \left\{ l \parallel l_k \parallel \pi \parallel Бл^*1 \parallel Бл^*2 \parallel ... \parallel Бл^*k \right\}, \quad (11)$$

де l - значення довжини блоку (2 байти); l_k - значення довжини останнього блоку (2 байти); π - послідовність ознак p у байтовому представленні; Бл*i - структура складових лінійної форми Фібоначчі числового еквівалента i-го блоку ($i = 1,2,...,k$).

Довжина блоку l змінювалася в межах від 100 байт до 1000 байт з кроком 100 байт.

Оскільки числові еквіваленти блоків даних мають великі значення (від 2^{800} до 2^{8000}), то програмний засіб було створено мовою програмування Python.

Досліджувалося ущільнення файлів різних типів і різного обсягу. Аналіз отриманих результатів показав, що ущільнюються файли типів *.doc, *.dat, *.bmp, *.lib, *.mp3, *.mdb, *.dll, а файли типів *.txt, *.xml, *.kbd, *.fdb, *.sdb, *.cpp, *.gif , *.jpg і *.au збільшують обсяг.

Вплив довжини блоків даних l на коефіцієнт ущільнення є різним для різних типів файлів. Найбільший коефіцієнт ущільнення для різних типів файлів досягається для різної довжини блоку даних. Тому потрібно здійснювати адаптацію також шляхом підбору довжини блоку даних.

Методи адаптивного ущільнення даних

Пропонується такий метод адаптивного ущільнення даних.

Ущільнювані дані розбиваються на блоки довжини l байтів. Для кожного блоку визначається максимальне значення B_{max} серед значень числових еквівалентів байтів. Потім обчислюється два числові еквіваленти $N_м$ і N_c за формулами (4) і (5).

Якщо $N_м=N_c=0$, то формується тільки ознака p=0, а інакше, формується ознака p=1 і здійснюється перетворення чисел $N_м$ і N_c в лінійну форму Фібоначчі ЛФФ($N_м$) і ЛФФ(N_c). Якщо в байтовому представленні $ЛФФ(N_м) \le ЛФФ(N_c)$, то формується ознака c=0 і результатом перетворення блоку є ЛФФ($N_м$), а інакше, формується ознака c=1 і результатом перетворення блоку є ЛФФ(N_c).

Обчислення числових еквівалентів блоку $N_м$ та N_c і перетворення їх у лінійну форму Фібоначчі виконується для всіх значень B, що не перевищують 256. Вибір оптимального значення B_o здійснюється з використанням функції оптимізації:

$$f_{бл}^{оп} = \min\left\{ (\min\{l_{бл}^{(м)}, l_{бл}^{(c)}\})^{(i)} \right\},$$

де $l_{бл}^{(зм)}$ - довжина перетвореного блоку для i-ої моделі джерела даних $M_{зм}$; $l_{бл}^{(зс)}$ - довжина перетвореного блоку для i-ої моделі джерела даних $M_{зс}$.

Ущільнені дані мають таку структуру:

$$S = \left\{ l \parallel l_k \parallel \pi \parallel C \parallel Бл^*1 \parallel Бл^*2 \parallel ... \parallel Бл^*k \right\},$$

де C – послідовність ознак c у байтовому представленні.

Тут Бл*i має структуру виду (8), доповнену значенням B_o.

Аналіз результатів ущільнення за даним методом показав, що для різних типів файлів коефіцієнт ущільнення збільшується від 1,23 до 3,7 разів порівняно з неадаптивним методом.

Варто відзначити, що файли типів *.xml, *.txt, *.cpp і *.kbd, які не ущільнювалися з використанням неадаптиного методу, ущільнені за даним методом у 1,15 рази.

Пропонується другий метод адаптивного ущільнення даних, який передбачає використання числової моделі джерела даних виду (7) з модулями $p_1 = 37, p_2 = 41, p_3 = 47$, числових моделей виду (4) і (5) та правила кодування з розбиттям числового еквіваленту блоку даних на складові N_1 і N_2.

Ущільнювані дані розбиваються на блоки довжини l байтів. Кожен байт блоку представляється сукупністю трьох чисел згідно з моделлю (7). Потім визначається максимальне значення B_{max} серед чисел отриманої послідовності. Обчислюється два числові еквіваленти $N_м$ і N_c за формулами (4) і (5).

Якщо $N_м=N_c=0$, то формується тільки ознака p=0, а інакше, формується ознака p=1 та обчислюються ЛФФ($N_м$), ЛФФ(N_c), СЛФФ($N_м$) і СЛФФ(N_c). Вибирається та форма, яка в байтовому представленні має найменшу довжину, і формується віповідна їй ознака h.

Обчислення $N_м$ та N_c і перетворення їх у лінійні форми Фібоначчі виконується для всіх значень B, що не перевищують 256.

Ущільнені дані мають таку структуру:

$$S = \left\{ l \parallel l_k \parallel \pi \parallel C \parallel H \parallel Бл^*1 \parallel Бл^*2 \parallel ... \parallel Бл^*k \right\},$$

де H – послідовність ознак h у байтовому представленні; Бл*i має структуру виду (8), доповнену значенням B_o.

Аналіз результатів ущільнення за даним методом показав, що для різних типів файлів коефіцієнт ущільнення збільшується від 3,83 до 4,25 разів порівняно з неадаптивним методом. Найбільший коефіцієнт ущільнення 4,25 досягнуто для файлів типу *.mdb. Файли типів *.xml, *.txt, *.cpp і *.kbd ущільнені за даним методом у 1,15 рази. Порівняння результатів ущільнення показує, що другий адаптивний метод забезпечує більші коефіцієнти ущільнення, але він реалізується довше (до 12 разів).

Висновки

1. Запропоновано набір числових моделей, які враховують певні властивості блоків даних. Можливість зміни моделі джерела даних дозволяє вибирати таку модель, яка забезпечує найбільший коефіцієнт ущільнення для заданого правила кодування.

2. Показано, що можна одержати лінійну форму Фібоначчі для надвеликих цілих чисел, використовуючи тільки додавання і віднімання, які для надвеликих чисел виконуються швидше, ніж операції ділення та множення. Це забезпечує прискорення одержання лінійної форми Фібоначчі в 1,45-3,6 рази порівняно з відомими алгоритмами.

3. Запропоновано два методи ущільнення даних на основі лінійної форми Фібоначчі, які передбачають використання наборів моделей джерела даних і правил кодування даних та функції оптимізації (адаптації). Вибір на основі функції оптимізації моделі джерела даних і правила кодування даних сприяє адаптації виконуваних перетворень до конкретного змісту ущільнюваних даних. Адаптація забезпечує підвищення коефіцієнта ущільнення до 4,25 разів порівняно з неадаптивним методом ущільнення на основі лінійної форми Фібоначчі, який передбачає тільки одну модель джерела даних й одне правило кодування для кожного блоку вхідних даних.

Список літератури

1. Shannon C. E. A Mathematical Theory of Communication [Текст] / C. E. Shannon // Bell System Technical Journal. – 1948. – V. 27. – P. 379-423, 623-656.

2. Huffman D. A. A Method for the Construction of Minimum-Redundancy Codes [Текст] / D. A. Huffman // Proceedings of the Institute of Electrical and Radio Engineers. – 1952. – V. 40, №9. – P. 1098-1101.

3. Ziv J. Compression of individual sequences via variable-rate coding [Текст] / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. – 1978. – V. IT–24, №5. – P. 530-535.

4. Welch T. A. A Technique for High Performance Data Compression [Текст] / T. A. Welch // Computer. – 1984. – №6. – P. 176-189.

5. Анисимов, А.В. Линейные формы Фибоначчи и параллельные алгоритмы большой размерности [Текст] / А.В. Анисимов // Кибернетика и системный анализ. – 1995. – № 3. – С. 106-115.

6. Лужецький В.А. Спосіб зображення цілих чисел великого діапазону [Текст] / В.А. Лужецький, Мохаммад Аль-Майта // Вимірювальна та обчислювальна техніка в технологічних процесах. – 1998. – № 1. – С. 156-162.

7. Кшановський О.Д. Арифметичні методи ущільнення цифрової інформації [Текст] / О.Д. Кшановський, С.В. Тітарчук, В.А. Лужецький // Вісник ВПІ. – 1999. – №5. – С. 83–87.

8. Rissanen J. J. Universal modeling and coding [Текст] / J. J. Rissanen, G. G. Langdon // IEEE Trans. Inf. Theory. – 1981. – V. IT–27, № 1. – P. 12-23.

9. Лужецький В.А. Узагальнена модель адаптивного ущільнення даних [Текст] / В.А. Лужецький, Л.А. Савицька, Шахзада Ашрафул Хок // Інформаційні технології та комп'ютерна інженерія. – 2009. - № 1(14). – С. 56-63.

10. Лужецький В.А. Високонадійні математичні Фібоначчі-процесори: монографія [Текст] / В.А. Лужецький. - Вінниця: Універсум-Вінниця, 2000. – 248 с.

Надішла до редколегії 3.12.2014

Рецензент: д-р техн. наук, проф. О.А. Смірнов, Кіровоградський національний технічний університет, Кіровоград.

МОДЕЛИ И МЕТОДЫ АДАПТИВНОГО СЖАТИЯ ДАННЫХ НА ОСНОВЕ ЛИНЕЙНОЙ ФОРМЫ ФИБОНАЧЧИ

В. А. Лужецкий, Л. А. Савицкая

Рассмотрены особенности формирования числовых моделей источника данных и правила кодирования, обеспечивающие повышение коэффициента сжатия за счет адаптации выполняемых преобразований к конкретному содержанию сжимамых данных. Предложен алгоритм ускоренного формирования линейной формы Фибоначчи. Предложено и исследовано два метода адаптивного сжатия данных на основе линейной формы Фибоначчи.

***Ключевые слова**: адаптивное сжатие, источник данных, числовая модель, правила кодирования, линейная форма Фибоначчи, коэффициент сжатия.*

MODELS AND METHODS FOR ADAPTIVE DATA COMPRESSION BASED ON LINEAR FIBONACCI FORM

V.A. Luzhetsky, L.A. Savitskaya

Features of data source numerical models desinging and coding rules are considered, those provide compression ratio increasing by using transformations adaptation to the compressed data specific content. The algorithm of accelerated linear Fibonacci form formation is proposed. Two methods of adaptive data compression based on linear Fibonacci form are proposed and researched.

***Keywords**: adaptive compression, data source, numerical model, coding rules, linear Fibonacci form, compression ratio.*