УДК 519.684.4

РОЗПАРАЛЕЛЕНИЙ АЛГОРИТМ СИНТЕЗУ ЕМПІРИЧНИХ МОДЕЛЕЙ ОПТИМАЛЬНОЇ СКЛАДНОСТІ

М. І. Горбійчук*[*], *В. М. Медведчук, Б. В. Пашковський

Івано-Франківський національний технічний університет нафти і газу,
вул. Карпатська, 15, м. Івано-Франківськ, 76019, e-mail: gorb@nung.edu.ua

Запропонований метод, який значно розширює клас емпіричних моделей і дозволяє синтезувати моделі оптимальної складності, спираючись на зовнішній критерій відбору моделей. Показано, що із збільшенням розмірності задачі синтезу емпіричних моделей зростають затрати машинного часу на їх програмну реалізацію. Тому актуальною науковою задачею є зменшення затрат машинного часу, що дозволить синтезувати емпіричні моделі високої розмірності. Одним із шляхів розв'язання поставленої задачі – розпаралелення алгоритму синтезу моделей оптимальної складності. Проведений аналіз алгоритму побудови таких моделей, який показав, що найбільш затратними операціями є розв'язання системи лінійних алгебраїчних рівнянь. Ці операції виконуються багаторазово. Для виявлення ефективності паралельних алгоритмів синтезу моделей оптимальної складності обчислено кількість арифметичних операцій, які мають місце при реалізації паралельних алгоритмів і здійснено їх порівняння з відповідним послідовним алгоритмом.
Ключові слова: система, ген, хромосома, критерій пристосування, розмірність задачі, система рівнянь, арифметична операція, аналіз алгоритму.

Предложен метод, который значительно расширяет класс эмпирических моделей и позволяет синтезировать модели оптимальной сложности опираясь на внешний критерий отбора моделей. Показано, что с увеличением размерности задачи синтеза эмпирических моделей увеличиваются затраты машинного времени на их программную реализацию. Поэтому актуальной научной задачей является уменьшение затрат машинного времени, что позволит синтезировать эмпирические модели высокой размерности. Одним из путей решения поставленной задачи - распараллеливание алгоритма синтеза моделей оптимальной сложности. Проведенный анализ алгоритма построения таких моделей показал, что наиболее затратными операциями являются решения системы линейных алгебраических уравнений. Эти операции выполняются многократно. Для выявления эффективности параллельных алгоритмов синтеза моделей оптимальной сложности вычислено количество арифметических операций, которые имеют место при реализации параллельных алгоритмов и осуществлено их сравнение с соответствующим последовательным алгоритмом.
Ключевые слова: система, ген, хромосома, критерий приспособления, размерность задачи, система уравнений, арифметическая операция, анализ алгоритма.

The method is developed to extend the class of empirical models and it allows to synthesize models of optimal complexity based on external criterion of models selection. At the same time, with the increase of the dimension of the problem of synthesis of the empirical models increases the cost of the computer time on their software implementation. Therefore, the actual scientific problem is the reduction of the computing time, which allows synthesizing the empirical models of high dimension. The parallelization of the algorithm of models synthesis of optimal complexity is the way to solve this problem. According to the analysis of the algorithm for the constructing of the empirical models of optimal complexity, the most expensive operations is the solution of systems of linear equations and calculating the output of the system. These operations are performed repeatedly. The number of arithmetic operations is calculated for the detection of the efficiency of the parallel algorithms for models synthesis of optimal complexity. This arithmetic operations is used in the implementing of parallel algorithms. Their comparison with the corresponding sequential algorithms of the solving of system of linear equations and the computation of the output of the empirical model are given in this article.
Keywords: The system, the gene, the chromosome, the criterion of adaptation, the dimension of the problem, the system of linear equations, arithmetic operation, the analysis of algorithm.

Вступ

При побудові емпіричних моделей найчастіше використовують поліноміальну залежність виду [1]

$$Y = \sum_{i=0}^{M-1} a_i \prod_{j=1}^{n} x_j^{z_{ij}}, \qquad (1)$$

де a_i – параметри моделі (1); M - кількість параметрів моделі.

Степені z_{ij} аргументів x_j приймають значення 0, 1, 2, … і задовольняють обмеженню $\sum_{j=1}^{n} z_{ij} \le r$; де r – максимальна степінь полінома (1).

У роботі [2] запропоновано емпіричну модель (1) синтезувати з використанням генетичних алгоритмів, суть якого у наступному. Утворимо упорядковану структуру довжиною M, в якій на i-тому місці буде стояти одиниця або нуль в залежності від того чи параметр a_i, $i = \overline{0, M-1}$ моделі (1) відмінний від нуля, чи нульовий. У теорії генетичних алгоритмів така упорядкована послідовність носить назву хромосоми або особі, а атомарний елемент хромосоми (одиниця або нуль) – це ген. Набір хромосом утворює популяцію. Важливим поняттям у теорії генетичних алгоритмів є функція пристосування, яка визначає ступінь пристосування окремих осіб у популяції. Вона дає змогу із всієї популяції вибрати особі, які є найбільш пристосованими, тобто такі, які мають найбільше (найменше) значення функції пристосування. У задачі синтезу емпіричних моделей функцією пристосованості виступає певний критерій селекції. У роботі [2] таким критерієм є критерій регулярності або мінімального зміщення. Таким чином, задачу синтезу емпіричної моделі сформуємо наступним чином: із початкової популяції хромосом шляхом еволюційного відбору вибрати таку хромосому, яка забезпечує найкраще значення функції пристосування (мінімальне значення критерію селекції). Реалізація розробленого алгоритму вимагає багаторазового розв'язування системи лінійних алгебраїчних рівнянь.

Якщо n – число незалежних змінних моделі, то кількість M коефіцієнтів моделі визначається [2] за формулою

$$M = \frac{(n+r)!}{n!\,r!}. \qquad (2)$$

У роботі [3] синтезована емпірична модель, в якій $n = 7$ і $r = 4$. У такому випадку $M = 330$. Таке значне число параметрів приведе до того, що процес розв'язання рівняння (2) займе досить багато часу. Тому з метою зменшення машинних затрат часу на розв'язування системи рівнянь (2) було запропоновано розпаралелення методу Гаусса.

Аналіз сучасних публікацій

На даний час можливості підвищення ефективності обчислювальних процесів завдяки збільшенню тактової частоти процесорів практично вичерпані або є економічно невигідними.

Одним із способів підвищення ефективності обчислювального процесу є його розпаралелення. Для реалізації паралельних алгоритмів використовують обчислювальні засоби як універсального, так і спеціального призначення. Універсальні обчислювальні системи розвиваються за такими основними напрямками [4]: MPP (Massively Parallel Processng), SMP (Symmetric Multi-Processing), векторноконвеєрні та кластери. Типовим представником SMP-систем є сучасні багатоядерні процесори фірми Intel.

Розвиток паралельних обчислювальних систем привів до появи цілого ряду наукових робіт, де запропоновані паралельні алгоритми розв'язання задач, які вимагають значних обсягів обчислень. Як приклади таких задач можна назвати задачі цифрової фільтрації [5, 6], оцінка якості функціонування складних динамічних систем [7], розв'язання систем лінійних алгебраїчних рівнянь з розрідженими матрицями [8], де дана оцінка ефективності паралельних алгоритмів за умови, що процесори завантажені рівномірно. Така ситуація практично не має місця. Як правило розподілення блоків матриці між процесорами є нерівномірним. Тому актуальною науковою задачею є розроблення алгоритму і його дослідження для випадку, коли розподілення блоків матриці між процесорами нерівномірне.

Виклад основного матеріалу

Для зняття проблеми великої розмірності комбінаторного методу застосуємо генетичний підхід. Як емпіричну модель виберемо поліном (1) степені r.

Як і класичний генетичний алгоритм [9], алгоритм синтезу моделей оптимальної складності, що ґрунтується на засадах генетичних алгоритмів, складається із таких кроків.

К1. Формування початкової популяції

(ініціалізація). На цьому кроці роботи алгоритму випадковим чином формулюється популяція із I особин, кожна із яких є хромосомою довжиною M. Число генів у хромосомі визначається за формулою (2).

К2. Оцінка пристосованості хромосоми у популяції. У відповідності з вибраним критерієм селекції формується матриця F_A розміром $N_A \times M$, де F_A – матриця, рядки якої значення функцій при коефіцієнтах a_i полінома (1); N_A – кількість точок спостережень за вихідною величиною Y. Із матриці F_A вилучається i-тий стовпець, якщо на i-тій позиції хромосоми знаходиться нуль; якщо одиниця, то відповідний стовпець залишається без змін. У результаті отримаємо матрицю $\tilde{F}_A$, із якої вилучено c стовпців (за кількістю нулів у хромосомі). Розмір такої матриці буде $N_A \times (M-c)$. На множині точок N_A обчислюються ненульові коефіцієнти a_{Aj}, $j = \overline{1, M-c}$, моделі (1) шляхом розв'язання нормального рівняння Гауса

$$\tilde{M}_{F,A}\bar{a}_A = \tilde{F}_A^T \bar{Y}_A, \qquad (3)$$

де $\bar{a}_A = \left(a_{A0}, a_{A1}, \ldots, a_{A,M-c-1}\right)^T$ – вектор ненульових параметрів моделі, який асоційований з черговою хромосомою; $\tilde{M}_{F,A} = \tilde{F}_A^T \tilde{F}_A$; $\bar{Y}_A = \left(Y^{(1)}, Y^{(2)}, \ldots, Y^{(N_A)}\right)$ – вектор експериментальних значень на множині N_A.

За знайденими коефіцієнтами $\bar{a}_A$ поліноміальної моделі на множині точок N_A обчислюють значення функції пристосування для кожної j-тої хромосоми із початкової популяції потужністю I ($j = \overline{1, I}$).

К3. Перевірка умови зупинки алгоритму. Якщо мінімальне значення критерію селекції не перевищує деякого заданого значення E, то відбувається зупинка обчислень. Зупинка обчислень також може відбутись у випадку, коли у результаті виконання алгоритму немає суттєвого зменшення функції пристосування, або у тому випадку, коли виконано задане число ітерацій.

Після виконання однієї із трьох умов із чергової популяції вибирається та хромосома ch^*, для якої критерій селекції набуває мінімального значення. Ця хромосома задає структуру моделі оптимальної складності і формує матрицю F^* таким чином, що із початкової матриці вилучаються стовпці, які асоційовані з нульовими значеннями відповідних генів. Перерахунок параметрів моделі (1) здійснюється на всій множині точок N за допомогою методу найменших квадратів (МНК).

К4. Селекція хромосом. За розрахованими на другому кроці алгоритму значеннями функції пристосування здійснюється відбір тих хромосом, які будуть брати участь у створенні потомків для нової популяції. Такий відбір проводиться за принципом природного відбору, коли найбільші шанси у створенні нової популяції мають хромосоми з найкращими значеннями функції пристосування. Найпоширенішим методами селекції є метод рулетки і турнірний метод [9]. Метод рулетки можна застосовувати тоді, коли функція пристосованості додатна, що робить його придатним лише для задач максимізації. Турнірний метод можна використовувати як у задачах максимізації, так і у задачах мінімізації.

При турнірній селекції всі хромосоми розбиваються на підгрупи з наступним вибором із кожної підгрупи хромосоми з найкращою пристосованістю. Підгрупи можуть мати довільний розмір, але частіше за все популяцію ділять на підгрупи по 2 – 3 особини у кожній.

К5. Формування нової популяції потомків. Над відібраними особинами на четвертому кроці алгоритму здійснюють відозміну за допомогою двох основних операторів: схрещування і мутації. Слід зауважити, що оператор мутації застосовується рідше у порівнянні з оператором схрещування. Вірогідність схрещування досить висока ($0 \le P_c \le 1$). Тоді вірогідність мутації $P_м$ становить: $0 \le P_m \le 0{,}1$ [3].

К6. Після виконання оператора схрещування відбувається перехід до кроку К2.

Реалізація генетичного алгоритму синтезу емпіричних моделей оптимальної складності показала, що значна частина машинного часу витрачається на розв'язування системи лінійних алгебраїчних рівнянь (3).

Нормальне рівняння (3) запишемо у такій формі:

$$\tilde{A}\bar{a}_A = \tilde{b}, \qquad (4)$$

де $\tilde{A} = \tilde{M}_{F,a}$; $\tilde{b} = \tilde{F}_A^T \bar{Y}_A$.

Рівняння (4) будемо розв'язувати методом гаусового виключення, де система (4) приводиться до верхньої діагональної форми за такими формулами [8]:

$$\tilde{a}_{i\bullet}^{(i)} = \frac{\tilde{a}_{i\bullet}^{(i-1)}}{\tilde{a}_{ii}^{(i-1)}},\ i = \overline{1, M-c}; \qquad (5)$$

$$\tilde{a}_{k\bullet}^{(i)} = \tilde{a}_{k\bullet}^{(i-1)} - \tilde{a}_{i\bullet}^{(i)} \tilde{a}_{kj}^{(i-1)},\ k = \overline{i+1, M-c},$$
$$j = \overline{1, M-c+1}, \qquad (6)$$

де $\tilde{a}_{i\bullet}^{(i)}$, $\tilde{a}_{k\bullet}^{(i)}$ – i-тий та k-тий рядок розширеної матриці, яка утворена шляхом приєднання до матриці $\tilde{A}$ вектор-стовпця $\tilde{b}$; $\tilde{a}_{i\bullet}^{(0)} = \tilde{a}_{i\bullet}$, $i = \overline{1, M-c}$; $\tilde{a}_{k\bullet}^{(0)} = \tilde{a}_{k\bullet}, k = \overline{i+1, M-c}$.

В обчислювальному процесі цикл за індексом k є внутрішнім по відношенню до зовнішнього циклу за індексом i. Результатом застосування ітераційних процедур (5) і (6) є верхня діагональна матриця з одиницями на головній діагоналі

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} & \cdots & u_{1,M-c} & u_{1,M-c+1} \\ 0 & 1 & u_{23} & \cdots & u_{2,M-c} & u_{2,M-c+1} \\ 0 & 0 & 1 & \cdots & u_{3,M-c} & u_{3,M-c+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & u_{M-c,M-c+1} \end{bmatrix},$$

де $u_{ij} = a_{ij}^{(i)}$, $i = \overline{1, M-c}$, $j = \overline{i+1, M-c+1}$.

Таким чином, рівняння (4) перетвориться в еквівалентну систему лінійних алгебраїчних рівнянь

$$\tilde{U}\,\bar{a}_A = \bar{b}_A, \qquad (7)$$

де $\tilde{U}$ – квадратна матриця розміром $M-c$, яка утворена із матриці U шляхом вилучення останнього стовпця; $b_{A,i} = u_{i,M-c+1}$, $i = \overline{1, M-c}$.

Рівняння (4) розв'язується методом зворотної прогонки, починаючи з останнього рівняння. Очевидно, що

$$a_{A,M-c} = b_{A,M-c}. \qquad (8)$$

Інші значення $a_{A,i}$ обчислюються за такою ітераційною процедурою:

$$a_{A,i} = b_{A,i} - \sum_{j=i+1}^{M-c} u_{ij} a_{A,j},\ i = \overline{M-c-1, 1}. \qquad (9)$$

На відміну від класичного методу Гауса [10], де приведення до діагонального вигляду матриці U відбувається шляхом відніманням одних рівнянь, помножених на відповідні числа, із інших рівнянь. У результаті отримують верхню діагональну матрицю з ненульовими коефіцієнтами на головній діагоналі, тобто $u_{ii} \neq 0$, $i = \overline{1, M-c}$. У такому випадку для отримання розв'язку рівняння (4) необхідно праву частину співвідношення (9) розділити на u_{ii}.

Отже, після приведення системи рівнянь (4) до виду (7) відпадає необхідність в операції ділення для обчислення $a_{A,i}$. Як відомо, на виконання операції ділення витрачається найбільше машинного часу у порівнянні з іншими алгебраїчними операціями. Тому обчислення параметрів моделі (4) за рекурентними співвідношеннями (8) і (9) дає відчутну економію машинного часу у порівнянні з класичним методом Гауса.

Сказане справедливо і для LU – методу, у відповідності з яким матриця $\tilde{A}$ подається у вигляді добутку двох матриць L і U_R. Перша із них – нижня діагональна з одиницями на головній діагоналі, а друга – верхня діагональна матриця. Потім прямою і зворотною прогонками розв'язуються трикутні системи

$$L\bar{\gamma} = b_A,\ U_R \bar{a}_A = \bar{\gamma}.$$

Іншим способом зменшення затрат машинного часу є розпаралелення алгоритму приведення матриці $\tilde{A}$ до верхньої трикутної матриці U. На цю операцію витрачається більша частина часу з всієї частки часу, яка витрачається на розв'язування системи рівнянь (4).

Суть такого алгоритму у наступному. На стадії ініціалізації (початковій стадії) вся матриця $\tilde{A}$ розміщається у робочому просторі першого процесу – майстра. Перший крок – майстер відправляє, по можливості, рівні шари матриці $\tilde{A}$ іншим процесам (робітникам) так, що кожний із процесів, включаючи і перший отримує підматриці $\tilde{A}_i^{(1)}$, $i = \overline{0, q-1}$, де q – загальна кількість процесів. Після відправки майстер нормує перший рядок своєї підматриці $\tilde{A}_0^{(1)}$ за формулою (5) і тут же відсилає значення $\tilde{a}_{1\bullet}^{(1)}$ іншим робітникам. Робітники, отримавши рядок $\tilde{a}_{1\bullet}^{(1)}$, кожний із них у відповідності з формулою (6), де $k = \overline{1, M-c}$, перераховують елементи своїх підматриць, включаючи і елемент $\tilde{a}_{qi,M-c+1}$. Одночасно майстер обчислює нові значення елементів за формулою (6), починаючи з другого рядка. У результаті такого перерахунку робітники отримують підматриці $\tilde{A}_i^{(1)}$, $i = \overline{1, q-1}$, у яких нульовими будуть перші стовпці. У матриці $\tilde{A}_1^{(0)}$ перший стовпець буде

мати нульові елементи, крім першого $\tilde{a}_{11}^{(1)}$, який дорівнюватиме одиниці.

Другий крок – майстер нормує другий рядок своєї підматриці і значення $\tilde{a}_{2\cdot}^{(2)}$, яке обчислене за формулою (5), посилає всім робітникам. Ті, одночасно з майстром, модифікують свої підматриці у відповідності з процедурою (6). Причому майстер змінює індекс k від 3 до $M-c$, а робітники відповідно від 2 до $M-c$. У підсумку довжина другого рядка з ненульовими елементами скоротилася на одиницю (перший елемент дорівнює нулю, а другий – одиниці). Підматриці, які розміщені у робочих просторах робітників, будуть вміщувати по два нульових стовпці.

Процес модифікації елементів підматриць здійснюється циклічно за наведеною схемою до тих пір, поки майстер не завершить приводити матрицю до верхнього діагонального вигляду. У підсумку після закінчення першого циклу у робочому просторі майстра буде зберігатись верхня прямокутна матриця з одиницями на головній діагоналі, а у робочих просторах робітників отримаємо підматриці, число нульових стовпців яких визначається числом рядків матриці майстра.

На початку другого циклу робітники надсилають свої підматриці майстру, де відбувається їх об'єднання. Потім майстер об'єднану підматрицю розділяє на q частин і розсилає їх всім робітникам, включаючи і себе. Майстер нормує перший рядок своєї підматриці. При цьому провідним елементом буде перший ненульовий елемент першого рядка підматриці майстра. Модифікований перший рядок, у відповідності з формулою (5), майстер розсилає всім робітникам, які модифікують свої рядки за формулою (6). У подальшому обчислення відбуваються так, як це було у першому циклі. Алгоритм продовжує працювати до тих пір, поки розмір чергового шару, що підлягає відправленню, не перевищуватиме кількості процесів. Тоді майстер приводить підматрицю, що залишилась, до верхньої прямокутної матриці з одиницями на головній діагоналі. Завершальний етап – це об'єднання всіх матриць, які були збережені у робочому просторі майстра.

Обчислимо кількість операцій на кожному кроці обчислень, які виконуються майстром і кожним робітником при їх паралельній роботі.

Виконання майстром операції нормування першого рядка підматриці $\tilde{A}_0^{(1)}$ у відповідності з формулою (5) вимагає $z=M-c$ операцій ділення (діагональним елементам підматриці $\tilde{A}_0^{(1)}$ присвоюється значення одиниці). На перерахування елементів решти $s_0^{(1)}$ рядків підматриці $\tilde{A}_0^{(1)}$ за формулою (6) буде затрачено $\left(s_0^{(1)}-1\right)z$ операцій множення і $\left(s_0^{(1)}-1\right)z$ операцій віднімання. Загальна кількість операцій, що виконана майстром на першому кроці, становитиме

$$N_{10}^{(1)}=\left(2s_0^{(1)}-1\right)z.$$

Кожний i-тий робітник, отримавши $\tilde{A}_i^{(1)}$ підматрицю з $s_i^{(1)}$ рядками, виконує $N_i^{(1)}$ операцій перерахування своїх елементів за формулою (6). Всього буде $s_i^{(1)}z$ операцій множення і $s_i^{(1)}z$ операцій віднімання. Загальна кількість операцій, що виконує i-тий робітник, буде такою:

$$N_{1i}^{(1)}=2s_i^{(1)}z,\ i=\overline{1,q-1}.$$

На другому кроці обчислень майстер нормує другий рядок матриці $\tilde{A}_0^{(1)}$, затративши $z-1$ операцій ділення. Інші рядки матриці $\tilde{A}_0^{(1)}$ перераховуються за формулою (6). При цьому буде виконано $\left(s_0^{(1)}-2\right)\left(z-1\right)$ операцій множення і така ж кількість операцій віднімання. Отже, на другому кроці обчислень майстер виконає

$$N_{10}^{(2)}=\left(2s_0^{(1)}-3\right)\left(z-1\right)$$

арифметичних операцій.

Оскільки у робочому просторі кожного i-го робітника маємо матриці, в яких на один стовпець ненульових елементів стало менше, то загальна кількість операцій ділення, множення і віднімання, що виконує i-тий робітник, вже буде такою:

$$N_{1i}^{(2)}=2s_i^{(1)}\left(z-1\right),\ i=\overline{1,q-1}.$$

На третьому кроці обчислень майстер нормує третій рядок, виконавши $z-2$ операцію ділення. Починаючи з четвертого рядка, майстер перераховує всі елементи підматриці $\tilde{A}_0^{(1)}$ за формулою (6), витрачаючи таку кількість операцій ділення, множення і віднімання:

$$N_{10}^{(3)} = \left(2s_0^{(1)} - 5\right)\left(z-2\right).$$

Інші $q-1$ робітників перераховують елементи своїх підматриць $\tilde{A}_i^{(1)}$ за формулою (6), виконавши при цьому

$$N_{1i}^{(3)} = 2s_i^{(1)}\left(z-2\right),\ i=\overline{1,q-1}$$

операцій множення і віднімання.

Узагальнюючи отримані результати, можна стверджувати, що на k-ому кроці обчислень майстер та i-тий робітник виконують відповідно таку кількість операцій ділення і віднімання:

$$N_{10}^{(k)} = \left(2\left(s_0^{(1)} - k\right)+1\right)\left(z-k+1\right), \qquad (10)$$

$$N_{1i}^{(k)} = 2s_i^{(1)}\left(z-k+1\right),\ i=\overline{1,q-1}. \qquad (11)$$

Перший цикл паралельних обчислень закінчується тоді, коли виконається умова $k = s_0^{(1)}$, тобто у формулах (10), (11) $k=\overline{1,s_0^{(1)}}$.

Таким чином, після першого циклу обчислень отримуємо верхню прямокутну матрицю, в якій на головній діагоналі будуть розміщені одиниці. Розмір такої матриці $s_0^{(1)}\times\left(z+1\right)$.

Після об'єднання майстром $q-1$ матриць у його пам'яті буде зберігатись матриця розміром $\left(z-s_0^{(1)}\right)\times\left(z-s_0^{(1)}+1\right)$, в якій вилучені всі нульові елементи. Отриману матрицю майстер ділить на q шарів, потужність кожного із них складає $s_i^{(2)}$, $i=\overline{1,q-1}$ рядків. У робочому просторі майстра буде знаходитись матриця $\tilde{A}_0^{(2)}$ з $s_0^{(2)}$ рядками.

Кількість операцій ділення, множення і віднімання, які виконує майстер, та кількість операцій множення і віднімання, які виконує i-тий робітник, обчислимо за формулами (10) і (11), врахувавши, що кількість стовпців підматриць, які розміщені у робочих просторах майстра і робітників, становить $z-s_0^{(1)}+1$. Це означає, що у формулах (10) і (11) необхідно z замінити на $z-s_0^{(1)}$, а $s_0^{(1)}$ і $s_i^{(1)}$ відповідно на $s_0^{(2)}$ і $s_i^{(2)}$ У результаті такої заміни отримаємо, що

$$N_{20}^{(k)} = \left(2\left(s_0^{(2)} - k\right)+1\right)\left(z-s_0^{(1)}-k+1\right),$$

$$N_{2i}^{(k)} = 2s_i^{(2)}\left(z-s_0^{(1)}-k+1\right),\ i=\overline{1,q-1}.$$

Другий цикл обчислень закінчиться за виконання умови $k = s_0^{(2)}$.

Після закінчення другого циклу обчислень у пам'яті майстра буде зберігатись прямокутна матриця розміром $\left(s_0^{(1)}+s_0^{(2)}\right)\times\left(z+1\right)$, на головній діагоналі якої будуть розміщені одиниці.

На початку третього циклу обчислень майстер об'єднує $q-1$ підматрицю в одну матрицю розміром $\left(z-s_0^{(1)}-s_0^{(2)}\right)\times\left(z-s_0^{(1)}-s_0^{(2)}+1\right)$, яка не вміщує нульових елементів. Отриману матрицю майстер ділить на q підматриць. Перша підматриця, яка матиме $s_0^{(3)}$ рядків, залишається у робочому просторі майстра, а інші підматриці відправляються $q-1$ робітникам.

Як і у другому циклі, кількість операцій ділення і віднімання, що виконують у третьому циклі відповідно майстер і робітники, обчислимо за формулами (10) і (11), замінивши z на $z-s_0^{(1)}-s_0^{(2)}$. При цьому необхідно врахувати, що потужність підматриці $\tilde{A}_0^{(3)}$ становить $s_0^{(3)}$ рядків, а потужність підматриць $\tilde{A}_i^{(3)}$ має величину $s_i^{(3)}$, $i=\overline{1,q-1}$ рядків. Отже

$$N_{30}^{(k)} = \left(2\left(s_0^{(3)} - k\right)+1\right)\left(z-s_0^{(1)}-s_0^{(2)}-k+1\right), \qquad (12)$$

$$N_{3i}^{(k)} = 2s_i^{(3)}\left(z-s_0^{(1)}-s_0^{(2)}-k+1\right),\ i=\overline{1,q-1}. \qquad (13)$$

Третій цикл закінчується за умови, що $k = s_0^{(3)}$.

Нехай виконано r циклів обчислень. У результаті у пам'яті майстра буде розміщена прямокутна матриця розміром $\left(\sum_{j=1}^{r-1} s_0^{(j)}\right)\times\left(z+1\right)$, на головній діагоналі якої будуть розміщені одиниці. У робочому просторі майстра буде знаходитись матриця розміром $\left(z-\sum_{j=1}^{r-1} s_0^{(j)}\right)\times\left(z-\sum_{j=1}^{r-1} s_0^{(j)}+1\right)$ з ненульовими елементами, яку майстер ділить на q шарів. У результаті отримаємо q підматриць, кожна із яких має потужність $s_i^{(r)}$, $i=\overline{0,q-1}$.

Враховуючи (12) і (13), можемо стверджувати, що після закінчення r-того циклу обчислень майстер і кожний i-тий робітник виконають таку кількість операцій ділення і віднімання:

$$N_{r0}^{(k)} = \left(2\left(s_0^{(r)} - k\right)+1\right)\left(z - \sum_{j=1}^{r-1} s_0^{(j)} - k + 1\right), \quad (14)$$

$$N_{ri}^{(k)} = 2s_i^{(r)}\left(z - \sum_{j=1}^{r-1} s_0^{(j)} - k + 1\right),$$

$$r = \overline{1, N_z - 1},\ i = \overline{1, q-1}. \quad (15)$$

Закінченням r-того циклу є виконання умови $k = s_0^{(r)}$.

Алгоритм приведення матриці $\tilde{A}$ до верхнього трикутного вигляду з одиницями на головній діагоналі закінчує свою роботу тоді, коли виконається умова

$$z - \sum_{j=1}^{N_z - 1} s_0^{(j)} \le q, \quad (16)$$

де N_z – загальна кількість циклів обчислень.

Тепер обчислимо кількість операцій ділення, множення і віднімання, які виконуються протягом r-того циклу. Майстер виконує таку кількість вказаних операцій:

$$N_{M,r} = \sum_{k=1}^{s_0^{(r)}} N_{r0}^{(k)},\ r = \overline{1, N_z - 1}, \quad (17)$$

а на долю кожного i-того робітника припадає така кількість операцій множення і віднімання:

$$N_{w,r}^{(i)} = \sum_{k=1}^{s_0^{(r)}} N_{ri}^{(k)},\ r = \overline{1, N_z - 1},\ i = \overline{1, q-1}, \quad (18)$$

де $N_{r0}^{(k)}$ і $N_{ri}^{(k)}$ – відповідно визначаються за формулами (14) і (15).

Оскільки у загальному випадку розмір матриці $\tilde{A}$ не кратний кількості паралельних процесів, то неоднакова кількість рядків приводить до різного обчислювального навантаження процесів і закінчення обчислень буде визначатись часом роботи найбільш завантаженого процесу.

Якщо врахувати операції на пересилку $\tilde{a}_{ij}^{(i)}$ елементу у кожному k-тому кроці обчислень, то загальна кількість операцій ділення і віднімання, яка виконується в r-тому циклі, буде такою:

$$N_r = \max_{1 \le i \le q-1}\left\{N_{M,r}, N_{w,r}^{(i)}\right\} + s_0^{(r)},\ r = \overline{1, N_z - 1}. \quad (19)$$

Всі N_r операцій в r-тому циклі обчислень виконуються паралельно. Допустимо, що на виконання всіх операцій ділення і віднімання в r-тому циклі обчислень витрачається τ_r одиниць часу, а на операції пересилки – $\tau_{t,r}$ одиниць часу. Тоді загальні затрати часу на реалізацію паралельного алгоритму приведення матриці $\tilde{A}$ до верхнього діагонального виду слід обчислювати за такою формулою:

$$T = \sum_{r=1}^{N_z - 1}\left(\tau_r \max_{1 \le i \le q-1}\left\{N_{M,r}, N_{w,r}^{(i)}\right\} + \tau_{t,r} s_0^{(r)}\right) + T_f, \quad (20)$$

де T_f – затрати часу на приведення підматриці на останньому кроці обчислень до верхньої діагональної форми.

На останньому етапі обчислень, коли виконана умова (16), майстер підматрицю $\tilde{A}_z$ розміром $\left(z - \sum_{j=1}^{r-1} s_0^{(j)}\right) \times \left(z - \sum_{j=1}^{r-1} s_0^{(j)} + 1\right)$ приводить до верхнього діагонального вигляду у відповідності з формулами (5) і (6).

Обчислимо кількість операцій ділення і віднімання, які виконуються на останньому циклі обчислень. Введемо такі позначення: $z_\alpha = z - \sum_{j=1}^{r-1} s_0^{(j)}$. Матрицю $\tilde{A}_z$ можна розглядати як квадратну матрицю розміром z_α, до якої приєднаний стовпець вільних членів системи алгебраїчних лінійних рівнянь. Приведення такої матриці до верхнього діагонального вигляду з одиницями на головній діагоналі потребує [11]

$$N_f = \frac{4z_\alpha^3 + 3z_\alpha^2 - z_\alpha}{6} \quad (21)$$

операцій ділення, множення і віднімання.

З врахуванням значення N_f формулу (20) запишемо у такому вигляді:

$$T = \sum_{r=1}^{N_z - 1}\left(\tau_r \max_{1 \le i \le q-1}\left\{N_{M,r}, N_{w,r}^{(i)}\right\} + \tau_{t,r} s_0^{(r)}\right) + \tau_f N_f,$$

де τ_f – затрати часу на виконання операцій на завершальному етапі обчислень.

Знайдемо верхню оцінку для величини T. Для цього допустимо, що початкова матриця $\tilde{A}$, яка підлягає перетворенню до верхнього трикутного вигляду за формулами (5) і (6), розбивається на q рівних шарів, тобто $s_i^{(r)}$, $i=\overline{0,q-1}$, $r=\overline{1,N_z}$. Оскільки операція пересилки елементу $\tilde{a}_{ij}^{(i)}$ виконується значно швидше, ніж арифметичні операції множення, ділення і віднімання, то в останньому виразі можна знехтувати величиною $s_0^{(r)}$. Також допустимо, що $\tau = \max\limits_{1\le r\le N_z-1}\left\{\tau_r,\tau_{t,r}\right\}$. Тоді

$$T=\tau\left(\sum_{r=1}^{N_z-1}\max_{1\le i\le q-1}\left\{N_{M,r},N_{w,r}^{(i)}\right\}+N_f\right). \qquad (22)$$

Із формули (22) випливає, що оцінка значення T зводиться до визначення загального числа операцій, які необхідно здійснити для приведення матриці $\tilde{A}$ до верхнього трикутного вигляду.

На першому циклі обчислень потужність кожного із $s_i^{(1)}$ шарів $s_i^{(1)}=\frac{z}{q}$, $i=\overline{0,q-1}$, а матриця $\tilde{A}_1=\tilde{A}$ має $n_1=n$ рядків.

Після закінчення першого циклу обчислень кількість рядків матриці $\tilde{A}$ зменшилась на величину $s_0^{(1)}$. Нова матриця $\tilde{A}_2$ матиме таку кількість рядків: $z_2=z-s_0^{(1)}$.Враховуючи значення $s_0^{(1)}$, знаходимо, що $z_2=\frac{z}{q}(q-1)$.

На початку другого етапу обчислень майстер ділить матрицю $\tilde{A}_2$ на q рівних шарів, що у результаті дає $s_i^{(2)}=\frac{z}{q^2}(q-1)$, $i=\overline{0,q-1}$.

Виконання другого етапу обчислень приводить до скорочення матриці $\tilde{A}_2$ на $s_0^{(2)}$ рядків. У результаті отримуємо матрицю $\tilde{A}_3$, яка буде мати таку кількість рядків: $z_3=z-s_0^{(1)}-s_0^{(2)}$.

Якщо прийняти до уваги значення $s_0^{(1)}$ і $s_0^{(2)}$, то отримаємо, що $z_3=\frac{z}{q^2}(q-1)^2$.

Після поділу матриці $\tilde{A}_3$ на q рівних шарів будемо мати, що $s_i^{(3)}=\frac{z}{q^3}(q-1)^2$, $i=\overline{0,q-1}$.

У результаті виконання третього етапу обчислень отримаємо матрицю $\tilde{A}_4$ з такою кількістю шарів: $z_4=z-s_0^{(1)}-s_0^{(2)}-s_0^{(3)}$. З врахуванням значень $s_0^{(1)}$, $s_0^{(2)}$ і $s_0^{(3)}$ будемо мати, що $z_4=\frac{z}{q^3}(q-1)^3$.

Якщо тепер матрицю $\tilde{A}_4$ поділити на q рівних шарів, то $s_i^{(4)}=\frac{z}{q^4}(q-1)^3$, $i=\overline{0,q-1}$.

Отже, після закінчення r-того циклу обчислень отримаємо матрицю $\tilde{A}_r$ з такою кількістю рядків: $z_r=\frac{z}{q^r}(q-1)^r$.

Для продовження роботи алгоритму на $r+1$ циклі майстер ділить матрицю $\tilde{A}_r$ на q рівних шарів, так що

$$s_i^{(r)}=\frac{z}{q^r}(q-1)^{r-1},\ i=\overline{0,q-1},\ r=\overline{1,N_z-1}. \qquad (23)$$

Допустимо, що на останньому циклі обчислень умова (16) набуде такого вигляду: $z-\sum\limits_{j=1}^{N_z-1}s_0^{(j)}=q$. У відповідності з формулою (19) $s_0^{(j)}=\frac{z}{q^j}(q-1)^{j-1}$. Тому

$$1-\frac{1}{q}\sum_{j=1}^{N_z-1}\left(\frac{q-1}{q}\right)^{j-1}=\frac{q}{z}.$$

Вираз $\sum\limits_{j=1}^{N_z-1}\left(\frac{q-1}{q}\right)^{j-1}$ є сумою геометричної прогресії, у якій перший член одиниця, а знаменник це $\frac{q-1}{q}$. Обчисливши суму геометричної прогресії, отримаємо $\left(\frac{q-1}{q}\right)^{N_z-1}=\frac{q}{z}$.

Отримане рівняння дає змогу знайти кількість циклів обчислень, які є необхідними для приведення матриці $\tilde{A}$ до верхнього трикутного вигляду, тобто $N_z=ln\frac{q-1}{z}\cdot\left(ln\frac{q-1}{q}\right)^{-1}$. У загальному випадку величина N_z не належить до множини

натуральних чисел. Тому її слід заокруглити з надлишком до цілого числа.

Обчислення загальне число операцій буде таким:

$$N_p = \sum_{r=1}^{N_z-1}\left(\max_{1\le i\le q-1}\left\{ N_{M,r}, N_{w,r}^{(i)} \right\} + s_0^{(r)} \right) + N_f \quad (24)$$

і його обчислення відбувається за таким алгоритмом. За формулами (17) і (18) знаходимо значення $N_{M,r}$ та $N_{w,r}^{(i)}$. Оскільки у кожному циклі обчислень чергова матриця $\tilde{A}_i$ розбивається на q рівних шарів, то $N_{w,r}^{(i)}$ залежить тільки від індексу r, тобто $N_{w,r} = N_{w,r}^{(i)}$.

Якщо врахувати формулу (23), то співвідношення (14) і (15), які входять до формул (17) і (18), набудуть такого вигляду:

$$N_{r0}^{(k)} = \left(2\left(\frac{z}{q} L(q) - k \right) + 1 \right)\left(zL(q) - k + 1 \right),$$

$$N_r^{(k)} = 2\frac{z}{q} L(q)\left(zL(q) - k + 1 \right),\ r = \overline{1, N_z - 1},$$

де $L(q) = \left(1 - \frac{1}{q} \right)^{r-1}$.

Знаючи $N_{M,r}$ і $N_{w,r}$, за формулою (24) обчислюємо N, де $s_0^{(r)}$ знаходимо згідно співвідношення (23), а N_f – за формулою (21).

ВИСНОВКИ

Синтез емпіричних моделей оптимальної складності на засадах генетичних алгоритмів вимагає багаторазового розв'язання системи лінійних алгебраїчних рівнянь, на що затрачається більша частина машинного часу. Застосування паралельного алгоритму для вирішення поставленої задачі дає значний виграш у часі у порівнянні з послідовним алгоритмом.

Слід відзначити, що приведені розрахунки не враховують втрати часу на обмін даними, синхронізацію і конфлікти пам'яті. Врахування таких втрат дещо збільшить затрати часу на реалізацію паралельного алгоритму, але вони будуть значно менші від затрат часу для реалізації послідовного алгоритму розв'язання рівняння (4).

1. Ермаков С. М. Математическая теория оптимального эксперимента: учебное пособие / С. М. Ермаков, А. А. Жиглявский. – М.: Наука, 1987. – 320 с. 2. Горбійчук М. І. Метод синтезу емпіричних моделей на засадах генетичних алгоритмів / М. І. Горбійчук, М. І. Когутяк, О. Б. Василенко, І. В. Щупак // Розвідка та розробка нафтових і газових родовищ. – 2009. - № 4(33). – С. 72-79. 3. Курейчик В. М. Генетические алгоритмы / В. М. Курейчик // Перспективные информационные технологии и интеллектуальные системы. –2000. – № 1. – С. 18-22. 4. Воеводин Вл. В. Вычислительное дело и кластерные системы / Вл. В. Воеводин, С. А. Жуматий. – М.: Изд. МГУ, 2007. – 150 с. 5. Valkovskii V. A. An optimal algorithm for solving the problem of digital filtering / V.A. Valkovskii // Pattern Recognition and Image Analysis. – 1994 – 4, vol. 3. – P. 241 – 247. 6. Тютюнник М. Паралельні алгоритми та засоби розв'язання деяких задач масових обчислень / М. Тютюнник // Конференція молодих вчених «Підстригачівські читання – 20010», м. Львів, 25 – 26 червня 2010 року. – Львів: Ін-т прикл. пр. мех. і мат., 2010. 7. Полищук А. Д. Оптимизация оценки качества функционирования сложных динамических систем / А. Д. Полищук // Проблемы управления и информатики. – 2004. – № 4. – С. 39 – 44. 8. Хіміч О. Блочно-циклічні паралельні алгоритми методу Гауса розв'язування систем лінійних алгебричних рівнянь з розрідженими матрицями / О. Хіміч, В. Полянко // Вісник Львів. ун-ту. Серія прикл. матем. інформ. – 2009. – Вип. 15. – С. 109–116. 9. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польск. И. Д. Рудинского. – М.: Горячая линия-Телеком, 2004. – 452 с. 10. Вержбицкий В. М. Основы численных методов: учебник для вузов / В. М. Вержбицкий. – М.: Высшая школа, 2002. – 840 с. 11. Волков Е. А. Численные методы: учебное пособие для вузов / Е. А. Волков.– 2-е изд., исп. – М.: Наука, 1987. – 248 с.

Поступила в редакцію

Рекомендували до друку: докт. техн. наук Юрчишин В. М. та докт. техн. наук Семенцов Г. Н.