УДК 519.163 + 681.5.015

КОМБІНАТОРНИЙ АЛГОРИТМ МГУА З ПОСЛІДОВНИМ УСКЛАДНЕННЯМ СТРУКТУР МОДЕЛЕЙ НА ОСНОВІ РЕКУРЕНТНО-ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

С.М.Єфіменко
Міжнародний науково-навчальний центр інформаційних технологій та систем (МННЦ ІТС) НАН та МОН України,

syefim@ukr.net

Розроблено теоретичні основи рекурентно-паралельних обчислень у комбінаторному алгоритмі МГУА на основі генератора послідовно ускладнюваних структур моделей. Ефективність розробленого алгоритму досліджено за допомогою обчислювальних експериментів на персональному комп'ютері та кластерному багатопроцесорному комплексі.
Ключові слова: структурно-параметрична ідентифікація, МГУА, комбінаторний алгоритм, рекурентно-паралельні обчислення, кластерна система.

Theoretical grounds of recurrent parallel computing in combinatorial GMDH algorithm with sequentially complicated structures are developed. The effectiveness of constructed algorithm is experimentally investigated using personal computer and multiprocessor cluster system.
Keywords: structural parametric identification, GMDH, combinatorial algorithm, recurrent parallel computing, cluster system.

Разработаны теоретические основы рекуррентно-параллельных вычислений в комбинаторном алгоритме МГУА на основе генератора последовательно усложняемых структур моделей. Эффективность разработанного алгоритма исследована с помощью вычислительных экспериментов на персональном компьютере и кластерном многопроцессорном комплексе.
Ключевые слова: структурно-параметрическая идентификация, МГУА, комбинаторный алгоритм, рекуррентно-параллельные вычисления, кластерная система.

1. Задача структурно-параметричної ідентифікації

Розглядаємо задачу структурно-параметричної ідентифікації у такому вигляді. Необхідно сформувати за вибіркою експериментальних даних деяку множину моделей-кандидатів $\Im$ та знайти оптимальну з них за заданим критерієм селекції *CR*:

$$f^* = \arg\min_{f\in\Im} CR(y, f(X, \hat{\theta}_f)). \qquad (1)$$

Оцінки параметрів $\hat{\theta}_f$ для кожної моделі $f \in \Im$ в (1) є розв'язком задачі виду

$$\hat{\theta}_f = \arg\min_{f\in\Im} Q(y, X, \theta_f, s_f), \qquad (2)$$

де $Q \neq CR$ – критерій якості розв'язання задачі параметричної ідентифікації кожної згенерованої моделі, а s_f – складність моделі f (кількість її ненульових параметрів).

2. Комбінаторний алгоритм зі стандартним двійковим генератором на основі рекурентно-паралельних обчислень

Використання комбінаторного алгоритму COMBI МГУА [1] передбачає повний перебір всіх можливих моделей та вибір найкращої за значенням критерію.

В процесі перебору порівнюються моделі лінійного об'єкта з m входами

$$\hat{y}_v = X_v \hat{\theta}_v,\ v = 1,...,2^m - 1 \qquad (3)$$

де десятковому числу v ставиться у відповідність двійкове число d_v.

Зважаючи на експоненційну складність алгоритму, при його використанні доцільно оптимально поєднувати рекурентні обчислення [2] з їх розпаралелюванням на кластерних комплексах [3].

В [4] описується розроблена схема розпаралелювання комбінаторного алгоритму зі стандартним двійковим генератором та рекурентним обчисленням параметрів моделей за допомогою модифікованого алгоритму Гаусса розв'язування систем лінійних рівнянь. У цій схемі зміну станів двійкового структурного вектора $d = \{d_i\},\ \ i = \overline{1, m}$ з елементами 0 або 1 (включення або не включення в модель відповідного аргументу) організовано за принципом двійкового лічильника.

Зокрема, для випадку трьох аргументів послідовність можливих комбінацій має такий вигляд (поряд у дужках – відповідний двійковий структурний вектор):

$$y_1 = a_1 x_1 \qquad \{1, 0, 0\}$$
$$y_2 = a_2 x_2 \qquad \{0, 1, 0\}$$
$$y_3 = a_1 x_1 + a_2 x_2 \qquad \{1, 1, 0\}$$
$$y_4 = a_3 x_3 \qquad \{0, 0, 1\}$$
$$y_5 = a_1 x_1 + a_3 x_3 \qquad \{1, 0, 1\}$$
$$y_6 = a_2 x_2 + a_3 x_3 \qquad \{0, 1, 1\}$$
$$y_7 = a_1 x_1 + a_2 x_2 + a_3 x_3 \qquad \{1, 1, 1\}$$

В таблиці 1 наведено орієнтовний час моделювання за допомогою розробленого алгоритму. Вже при кількості аргументів, рівній 50, повний перебір за прийнятний час моделювання стає неможливим навіть при використанні кластерної системи зі ста процесорами. Якимось чином виконати ефективне скорочення повного перебору для такої схеми не передбачається можливим через особливості стандартного двійкового генератора (складність структурних векторів змінюється не послідовно). Для цього випадку розроблено таку схему розпаралелювання алгоритму COMBI.

Таблиця 1.

Оцінка часу повного перебору

Кількість аргументів	Кількість моделей	Час моделювання	
		1 процесор	100 процесорів
20	1 048 575	1 сек	0,01 сек
21	2 097 151	2 сек	0,02 сек
…	…	…	…
40	1,1E+12	~ 12 днів	~ 3 год
…	…	…	…
50	1,1E+15	~ 34 роки	~ 124 дні

3. Комбінаторний алгоритм з послідовним ускладненням структур моделей та схема його розпаралелювання

Ця схема використовує таку послідовність генерації двійкових чисел, при якій спочатку утворюються всі сполучення з однією одиницею у складі структурного вектора (усього генерується $C_m^1 = m$ можливих варіантів), потім – з двома одиницями ($C_m^2 = \frac{m(m-1)}{2}$ можливих варіантів), і т.д. до одного варіанту ($C_m^m = 1$) включення в модель усіх аргументів.

Послідовність усіх варіантів та структурних векторів для випадку трьох аргументів виглядає так:

$$y_1 = a_1 x_1 \qquad \{1, 0, 0\}$$
$$y_2 = a_2 x_2 \qquad \{0, 1, 0\}$$
$$y_3 = a_3 x_3 \qquad \{0, 0, 1\}$$
$$y_4 = a_1 x_1 + a_2 x_2 \qquad \{1, 1, 0\}$$
$$y_5 = a_1 x_1 + a_3 x_3 \qquad \{1, 0, 1\}$$
$$y_6 = a_2 x_2 + a_3 x_3 \qquad \{0, 1, 1\}$$
$$y_7 = a_1 x_1 + a_2 x_2 + a_3 x_3 \qquad \{1, 1, 1\}$$

Таку схему можна досить легко застосувати для розпаралелювання комбінаторного алгоритму на задану кількість процесорів. Ідея рівномірного розбиття загальної кількості моделей на всі процесори кластерної системи за умови, що однакова загальна кількість аргументів припадатиме на кожен процесор, полягає у наступному. Кількість моделей $C_m^i = \frac{m!}{(m-i)! \times i!}$ складності i, $i = \overline{1, m}$, рівномірно розподіляється між усіма процесорами p, тобто на кожен

процесор припадає $\frac{m!}{p \times (m-i)! \times i!}$ моделей. Таким чином, необхідно для кожної множини двійкових структурних векторів, що відповідають моделям складності i, $i = \overline{1, m}$, визначити початкову точку (структурний вектор, з якого починає будувати моделі) для кожного процесора

$$\frac{m!}{p \times (m-i)! \times i!}(j-1), \quad i = \overline{1, m}, \quad j = \overline{1, p}. \tag{4}$$

Алгоритм визначення початкового стану двійкового структурного вектора за його позицією при послідовному ускладненні для кожного процесора описано в [5].

Нехай маємо m аргументів та k процесорів кластерної системи. Запишемо послідовність кроків для моделей складності i, $i = \overline{1, m}$:

1. Визначення кількості сполучень – $C_m^i - 1$.

2. Визначення початкового стану двійкового вектора d для кожного процесора j, $j = \overline{1, k}$ у вигляді десяткового числа – $\left[\frac{C_m^i - 1}{k}\right](j-1) + 1$.

3. Перехід від визначеного на попередньому кроці десяткового числа до відповідного двійкового числа (за схемою із послідовним ускладненням) для кожного процесора:

позиція=$\left[\frac{C_m^i - 1}{k}\right](j-1) + 1$;

u=i-1, b=m-1, $C = C_b^u$;

Цикл по l, $l = \overline{1, m}$

якщо позиція<=C, то $d[l]$=1, u= u -1, b= b -1, $C = C_b^u$;

інакше $d[i]$=0, позиція=позиція-C, u= u-1, $C = C_b^u$.

Головною перевагою розробленої альтернативної схеми розпаралелювання алгоритму COMBI є те, що вона дозволяє частково розв'язувати задачу повного перебору у випадку, коли кількість аргументів для перебору перевищує можливості алгоритму зі стандартним двійковим генератором, і повний перебір за прийнятний час моделювання стає неможливим навіть із розпаралелюванням (орієнтовно більше п'ятдесяти аргументів). У такому випадку повний перебір доцільно виконувати не серед усіх можливих моделей, а лише моделей обмеженої складності.

Виконаємо оцінку часу моделювання з використанням алгоритму COMBI з послідовним ускладненням структур моделей на основі рекурентно-паралельних обчислень. Для спрощення обчислень (як і раніше) виходитимемо з того, що 2^{20}-1=1048575 моделей відповідна програма будуватиме за одну секунду. Визначимося також із прийнятним часом моделювання – нехай він становитиме не більше, ніж десять годин при розпаралелюванні на сто процесорів. За таких обмежень можна перебрати всі моделі складності не більше, ніж 15, для загальної кількості аргументів 50 (тобто побудувати всі моделі, двійкові 50-елементні структурні вектори яких містять від 1 до 15 одиниць). Для 100 аргументів для перебору можна дійти до складності 9, для 150 та 200 аргументів – до складності 7. Час моделювання при цьому можна знайти в таблиці 2.

Таблиця 2.

Оцінка часу перебору з послідовним ускладненням структур

Кількість аргументів, m	Обмеження складності, l	Кількість моделей, $\sum_{i=1}^{l} C_m^i$	Час моделювання, години	
			1 процесор	100 процесорів
50	15	3,7E+12	984	~ 10
100	9	2,1E+12	558	~ 6
150	7	3,1E+11	81,8	~ 1
200	7	2,4E+12	628	~ 6

4. Результати експериментів

4.1 Експерименти на персональному компютері

Експеримент 1.

Для експериментального визначення ефективності розпаралелювання комбінаторного алгоритму з послідовним ускладненням структур було виконано тестовий експеримент по розв'язанню задачі структурно-параметричної ідентифікації з повним перебором серед 25 аргументів. Обчислення було розподілено на п'ять потоків і послідовно виконано на персональному комп'ютері з процесором Intel Pentium M з частотою 1.73 ГГц. Таким чином отримуємо результат, наближений до теоретичного через виключення втрат, пов'язаних із міжпроцесорною взаємодією.

Під час експерименту генерувалася матриця плану X розміром 45×25 (45 точок для 25 аргументів) для системи умовних рівнянь $X\theta = y$. Вектор y формувався у вигляді лінійної комбінації п'яти послідовних аргументів: $y = x_{11} + x_{12} + x_{13} + x_{14} + x_{15}$. Краща модель відбиралася за критерієм регулярності [1]. Вимірювався час виконання кожного з п'яти потоків та час роботи програми без розпаралелювання. Результат експерименту у вигляді діаграми часу виконання представлено на рисунку 1.

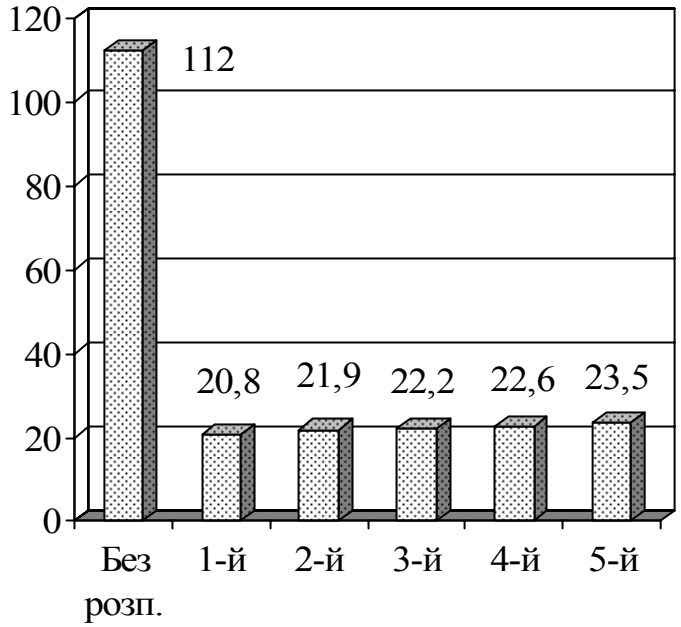

Рис. 1. Час виконання в секундах комбінаторного алгоритму з послідовним ускладненням структур

Результати експерименту можна використати для обчислення ефективності розпаралелювання

$$E = \frac{T_1}{5 \times T_{5\max}} \times 100\% \qquad (5)$$

та рівномірності навантаження на обчислювачі

$$P = (1 - \frac{T_{5\max} - T_{5\min}}{T_{5\max}}) \times 100\%, \qquad (6)$$

де T_1 – час виконання алгоритму з одним потоком (тобто без розпаралелювання), T_{5max} – час виконання алгоритму з розпаралелюванням на 5 потоків (визначається як максимальний серед п'яти потоків час виконання програми), T_{5min} – мінімальний серед п'яти потоків час виконання програми.

Суть формули (5) полягає в тому, що, якщо при використанні розпаралелювання на k потоків (у даному випадку п'яти) час моделювання зменшується у k разів, то ефективність розпаралелювання становить 100%. Відповідно до (6), для забезпечення стовідсоткової рівномірності навантаження всі обчислювачі (ядра, процесори, процеси) мають виконувати моделювання за один і той же час.

Ця тестова задача була розв'язана також в [6] за допомогою комбінаторного алгоритму на основі рекурентно-паралельних обчислень зі стандартним двійковим генератором. В таблиці 3 порівнюються результати роботи двох алгоритмів.

Таблиця 3.

Експериментальні результати

	Алгоритм COMBI	
	зі стандартним двійковим генератором	з послідовним ускладненням структур
Ефективність розпаралелювання, %	99,7	95,3
Рівномірність навантаження, %	99,2	88,5
Повний перебір для 25 аргументів на 1 процесорі, сек.	48,4	112

Двократна перевага алгоритму COMBI зі стандартним двійковим генератором пояснюється тим, що параметри всіх моделей оцінюються рекурентно, в той час, як для схеми з послідовним ускладненням структур – лише половини (що пов’язано з конструктивними особливостями відповідних генераторів структур). До того ж серед усієї послідовності структур генератора з послідовним ускладненням спочатку формується більше моделей, у яких параметри оцінюються рекурентно, а до кінця послідовності таких моделей стає все менше. З цим пов’язана і гірша рівномірність навантаження.

4.1 Експерименти на суперкомп'ютері Інституту кібернетики

Експеримент 2.

Експеримент 1 було виконано також на обчислювальному комплексі СКІТ ІК НАН України [7] на базі багатоядерних центральних процесорів Intel Xeon E5-2600 з частотою 2.6 ГГц.

Отримано такі результати (табл. 4, рис. 2):

Таблиця 4.

Експериментальні результати

Ефективність розпаралелювання, %	94,2
Рівномірність навантаження, %	88,4

Як бачимо, показники ефективності у реальному експерименті близькі до „теоретичних” (табл. 3), вищою є швидкодія програми за рахунок більш потужних процесорів.

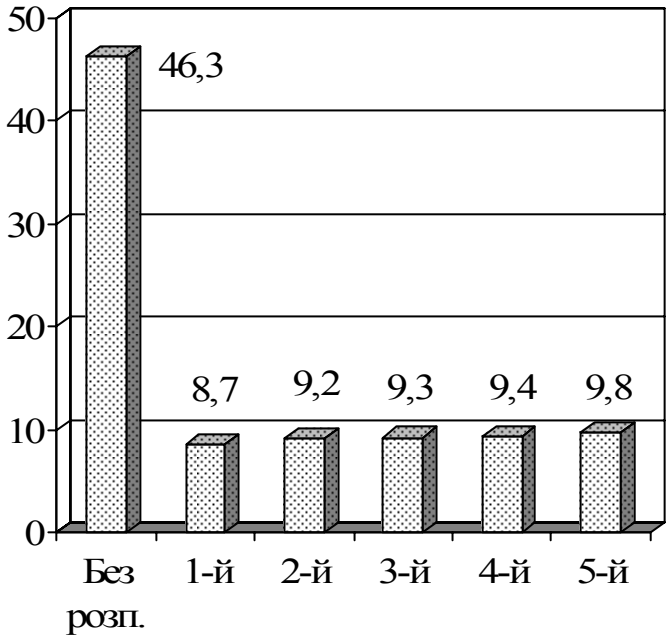

Рис. 2. Час виконання в секундах комбінаторного алгоритму з послідовним ускладненням структур

Експеримент 3.

Тестову задачу було сформовано таким чином: згенеровано матрицю плану X розміром 70×50 (70 точок для 50 аргументів) для системи умовних рівнянь $X\theta = y$. Вектор y формувався у вигляді лінійної комбінації п'яти аргументів:

$$y = x_{10} + x_{20} + x_{30} + x_{40} + x_{50}. \qquad (7)$$

Час повного перебору комбінаторним алгоритмом зі стандартним двійковим генератором на основі рекурентних обчислень для 50 аргументів триватиме близько 34 років (див. табл. 1). Однак цю задачу можна розв'язати за допомогою алгоритму з послідовним ускладненням структур, обмежившись моделями складності 7 (тобто розглянувши всі моделі, що містять не більше, ніж 7 аргументів).

Для структурно-параметричної ідентифікації було використано 6 вузлів, що мали 24 ядра, обчислювального кластера СКІТ – 4. Модель (7) такою обчислювальною системою було отримано менше ніж за 2 секунди.

Висновки

Розроблено принцип розпаралелювання операцій у комбінаторному алгоритмі COMBI МГУА з послідовним ускладненням структур моделей. Кожен процесор автономно обчислює початковий двійковий структурний вектор та кількість моделей, які він будуватиме, а також гарантується неповторюваність структур в різних процесорах. Це значно підвищує ефективність розпаралелювання, оскільки немає втрат часу на міжпроцесорну взаємодію.

Головною перевагою розробленої альтернативної схеми розпаралелювання алгоритму COMBI є те, що вона дозволяє частково розв'язувати задачу повного перебору у випадку, коли кількість аргументів для перебору перевищує можливості алгоритму зі стандартним двійковим генератором, і повний перебір доцільно виконувати не серед усіх можливих моделей, а лише моделей обмеженої складності. Оптимальне поєднання двох розглянутих схем дозволяє створити ефективну інтелектуальну технологію індуктивного моделювання на основі рекурентно-паралельних обчислень.

Виконано дослідження алгоритму COMBI з послідовно ускладнюваними структурами за допомогою обчислювальних експериментів на персональному комп'ютері та кластерному багатопроцесорному комплексі. Розроблена схема дозволила істотно підвищити ефективність комбінаторного алгоритму МГУА за рахунок виконання рекурентно-паралельних обчислень.

Література

1. Ивахненко А.Г., Степашко В.С. Помехоустойчивость моделирования. – Киев: Наукова думка, 1985. – 216 с..

2. Stepashko V. S. , and Efimenko S. N. . Sequential Estimation of the Parameters of Regression Model // Cybernetics and Systems Analysis, Springer New York, July, 2005, Vol. 41, Num. 4, pp.631-634.

3. Stepashko V.S., Yefimenko S.M. Optimal paralleling for solving combinatorial modelling problems // Proceedings of the 2nd International Conference on Inductive Modelling ICIM 2008. – Kyiv, 2008. – P. 172-175.

4. Єфіменко С.М., Степашко В.С. Рекурентно-паралельні обчислення в комбінаторному алгоритмі МГУА для задач індуктивного моделювання // Матеріали 21-ї Міжнародної конференції з автоматичного управління „Автоматика-2014", Київ, 23-27 вересня 2014 р. – К.: Вид-во НТУУ „КПІ" ВПІ ВПК „Політехніка", 2014. – С. 200-201.

5. Єфіменко С.М., Степашко В.С. Застосування паралельних обчислень при моделюванні з використанням комбінаторного алгоритму МГУА // Праці міжнародної наукової конференції «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту (ISDMCI'2008)». – Євпаторія, 2008. – Том 3 (частина 1). – С. 121-124..

6. Єфіменко С.М., Степашко В.С. Організація рекурентно-паралельних обчислень в комбінаторному алгоритмі МГУА для задач індуктивної побудови моделей // Праці 12-ї Всеукраїнської Міжнар. конф. «Оброблення сигналів і зображень та розпізнавання образів» (УкрОБРАЗ'2014), Київ, 3-7 листопада 2014 р. – Київ, УАсОІРО, 2014. – 168 с. / С. 43-46 / ISBN 978-966-479-069-4.

7. http://icybcluster.org.ua.