

Т.Б. МАРТИНЮК

Вінницький національний технічний університет, Вінниця, Україна,
e-mail: martyniuk.t.b@gmail.com.

Б.І. КРУКІВСЬКИЙ

Вінницький національний технічний університет, Вінниця, Україна,
e-mail: smiletex11@gmail.com.

УДОСКОНАЛЕНА МОДЕЛЬ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ З ФОРМУВАННЯМ РАНГІВ

Анотація. Удосконалено модель паралельного сортування масиву чисел з ранжуванням на базі одночасного застосування швидкісних операцій декременту/інкременту відповідно до масиву чисел і масиву їхніх рангів. Прискорення запропонованого алгоритму досягається завдяки фіксуванню результату обнуління $(n-1)$ елементів масиву чисел замість повного його обнуління. Наведено опис алгоритму паралельного сортування з формуванням рангів у компактній формі з використанням базису системи алгоритмічних алгебр Глушкова.

Ключові слова: система алгоритмічних алгебр, паралельне сортування, маска, ранг, декремент/інкремент.

ВСТУП

Для розв'язання складних задач діагностування, класифікації та прогнозування, що складають базис експертних технологій прийняття рішень, широко застосовуваними в процесі експертного оцінювання є методи ранжування об'єктів [1, 2]. Крім того, власне методи ранжування ефективно використовують «м'які обчислення», до яких відносять еволюційне моделювання, методи теорії нечітких множин, нейромережні та гібридні технології [1, 3, 4]. При цьому область застосування методів ранжування об'єктів має значний діапазон розв'язання практичних задач [5, 6].

В обчислювальних засобах, де оперують числовими даними, методи ранжування тісно пов'язані з такою процедурою, як сортування масивів даних [7, 8]. У цьому випадку сортування виконується без безпосереднього переміщення елементів числового масиву, але з формуванням відповідного індексного масиву [9]. Метою такого сортування є перевпорядкування масиву індексів (рангів), за допомогою яких виконуються операції попарного порівняння [9–11]. Результатом такого підходу є гнучкість процесу сортування з ранжуванням через можливість оброблення даних зі збереженням їхнього початкового списку, а також уникнення значних часових та апаратних витрат на переміщення (перекомутацію) елементів у масиві даних [7, 9, 10]. При цьому активно використовуються особливості асоціативної пам'яті [10, 12] для розв'язання задач різного призначення, оскільки сортування, ранжування та пошук належать асоціативно-логічним операціям [13].

У цій роботі розглядається паралельний алгоритм сортування, в якому ранги елементів числового масиву формуються після завершення процесу сортування і не використовуються модулі порівняння–обміну (compare-exchange modules) [14].

Метою роботи є аналіз можливості прискорення процесу сортування з ранжуванням з відповідним відображенням результату в базисі системи алгоритмічних алгебр (САА) Глушкова.

ОСОБЛИВОСТІ ПРОЦЕСУ ПАРАЛЕЛЬНОГО СОРТУВАННЯ З РАНЖУВАННЯМ

Для представлення послідовних та паралельних (синхронних) алгоритмів сортування числових масивів у компактній та наочній формі добре себе зарекомендував опис у базисі САА Глушкова [15]. Такий апарат паралельного програмування [16–18] розвивається та поширюється не тільки внаслідок застосування для подання у формалізованому вигляді відомих алгоритмів. Він також завдяки таким властивостям, як універсальність, компактність та наочність дає змогу класифікувати, синтезувати нові ефективні алгоритми та проводити їхній порівняльний аналіз. Як підтвердження цьому можна навести приклад опису у базисі САА Глушкова алгоритмів сортування від відомих послідовних [15] і паралельних методів [19] до нового варіанта паралельного сортування з ранжуванням [14].

У [14] наведено приклад сортування з ранжуванням масиву з чотирьох чисел, яке виконано за п'ять циклів. Цей приклад підтверджує залежність тривалості процесу запропонованого методу сортування від значення максимального числа у масиві, а не від кількості n елементів масиву. Це зумовлено відмовою від застосування класичного способу сортування на базі модулів порівняння–обміну [7, 9, 20] і заміною їх операцією декременту, яка виконується одночасно над усіма числами масиву, що є ознакою позрілого оброблення даних [14]. Такий підхід також можна розглядати як варіант вертикально-паралельного методу оброблення векторного масиву даних на відміну від відомого вертикального або послідовно-порозрядного методу [13].

Отже, в основі запропонованого методу сортування [14] лежать такі базові принципи:

- установлення у початковий (одичинний) стан рангів і елементів маски для всіх елементів масиву чисел;
- виконання операції декременту (зменшення на одиницю) одночасно над усіма елементами масиву чисел;
- використання ознаки обнуління відповідного елемента масиву чисел для маскування його рангу;
- виконання операції інкременту (збільшення на одиницю) одночасно над усіма незамаскованими рангами елементів масиву чисел.

Таким чином, основними умовами (ознаками) цього процесу сортування з ранжуванням [14] є такі:

- хоча б один i -й елемент a_i масиву чисел у поточний момент часу дорівнює нулю, тобто

$$\exists a_i = 0, i = \overline{1, n}; \quad (1)$$

- всі n елементи a_1, \dots, a_n вхідного масиву чисел у поточний момент часу дорівнюють нулю, тобто

$$\forall a_i = 0. \quad (2)$$

Водночас для наведеного методу [14] існує можливість зменшити час процесу сортування з ранжуванням не тільки завдяки використанню швидкісних операцій декременту/інкременту. Саме такий варіант ілюструє приклад сортування з ранжуванням масиву п'яти чисел $\{2, 5, 3, 4, 8\}$, що наведено у табл. 1.

З табл. 1 видно, що на п'ятому циклі, коли обнулюється передостанній елемент, вже сформовано остаточний масив рангів. Тому усі наступні цикли з шостого по восьмий не є обов'язковими. Отже, завдяки зупинки процесу сортування з ранжуванням на п'ятому циклі можна досягти зменшення часу тривалості цієї операції, оскільки всі наступні цикли є порожніми для остаточного формування рангів. Таким чином, залишаючи умову (1), потрібно змінити умову (2) для цього процесу так: кількість $(n-1)$ елементів a_i , крім найбільшого за величиною, у поточний момент часу дорівнює нулю.

Таблиця 1. Процес сортування з ранжуванням

0-й цикл			1-й цикл			2-й цикл		
Масив чисел	Елементи маски	Масив рангів	Масив чисел	Елементи маски	Масив рангів	Масив чисел	Елементи маски	Масив рангів
2	1	1	1	1	1	0	0	1
5	1	1	4	1	1	3	1	2
3	1	1	2	1	1	1	1	2
4	1	1	3	1	1	2	1	2
8	1	1	7	1	1	6	1	2
3-й цикл			4-й цикл			5-й цикл		
Масив чисел	Елементи маски	Масив рангів	Масив чисел	Елементи маски	Масив рангів	Масив чисел	Елементи маски	Масив рангів
0	0	1	0	0	1	0	0	1
2	1	3	1	1	4	0	0	4
0	0	2	0	0	2	0	0	2
1	1	3	0	0	3	0	0	3
5	1	3	4	1	4	3	1	5
6-й цикл			7-й цикл			8-й цикл		
Масив чисел	Елементи маски	Масив рангів	Масив чисел	Елементи маски	Масив рангів	Масив чисел	Елементи маски	Масив рангів
0	0	1	0	0	1	0	0	1
0	0	4	0	0	4	0	0	4
0	0	2	0	0	2	0	0	2
0	0	3	0	0	3	0	0	3
2	1	5	1	1	5	0	0	5

Для наочності у табл. 1 виокремлено моменти, коли одне число з масиву обнулюється, його ранг маскується і при цьому не змінюється, тоді як всі інші ранги збільшуються на одиницю. Отже, процес сортування з ранжуванням виконано за п'ять циклів попри те, що максимальним у масиві є число 8. Це пояснюється тим, що на початку сортування всім числам масиву було надано ранг 1, тобто було сформовано ранг найменшого числа у масиві. Правомірність початкового надання рангу 1 зумовлена базовим принципом строгого компромісного ранжування, коли ранги (натуральні числа в інтервалі від 1 до n) не мають збігатися [1].

ЗАПИС ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ З РАНЖУВАННЯМ У БАЗИСІ САА ГЛУШКОВА

Для опису розглянутого паралельного алгоритму сортування з урахуванням його особливостей використано розширений базис САА Глушкова [16–18]. Початковими даними є три масиви однакової довжини n :

- масив $M = a_1, a_2, \dots, a_n$ елементів, які сортуються;
- масив $R = r_1, r_2, \dots, r_n$ рангів відповідних елементів, які сортуються;
- масив $P = m_1, m_2, \dots, m_n$ елементів маски.

При цьому серед зауважень слід виокремити такі:

- масив P елементів маски введено для безпосереднього формування масиву R рангів, які визначають місцезнаходження конкретного елемента з масиву M у відсортованому масиві;

— покажчики і маркери для розмітки не застосовуються, оскільки в процесі паралельного сортування задіяно одночасно всі елементи кожного масиву.

У результаті з урахуванням особливостей удосконаленого алгоритму на позначених масивах використовуються такі базисні умови (предикати) і базисні оператори:

- $Z(i)$ істинне за умови (1), тобто $\exists a_i = 0$;
- $Z(n-1)$ істинне за умови, що обнулено кількість $(n-1)$ чисел у масиві;
- $p(i)$ істинне за виконання $\exists m_i = 0$;
- оператор виведення результату $OUT(R)$;
- оператор завершення роботи FIN регулярної схеми.

Крім того, особливістю розглянутого паралельного алгоритму сортування є те, що зона предикатів і операторів поширюється на всі елементи у кожному з трьох масивів, оскільки виконання операторів і перевірка істинності предикатів здійснюється одночасно над усіма елементами кожного масиву.

Для опису алгоритму сортування доцільно задіяти всі основні операції, що входять до сигнатури розширених САА-М Глушкова [15, 16], а саме:

- диз'юнкція $\alpha \vee \beta$;
- композиція $A \times B$, тобто послідовне застосування операторів A і B ;
- альтернатива $[\alpha](A \vee B)$, тобто якщо α істинне, то A , інакше B ;
- цикл $[\alpha]\{A\}$, тобто виконувати A поки α хибне, якщо α істинне, кінець циклу.

Крім того, позначення $A + B$ використовується для паралельного застосування операторів A і B [19].

Для початку виконання операцій декременту, інкременту і маскування вводяться базисні оператори вигляду:

- установлення паралельно початкових (одичних) станів для n елементів масиву рангів R

$$SET_n(\overline{r_1, r_n}) ::= (\overline{r_1, r_n} = 1);$$

- установлення паралельно початкових (одичних) станів для n елементів маски масиву P

$$SET_n(\overline{m_1, m_n}) ::= (\overline{m_1, m_n} = 1).$$

У результаті складові оператори із застосуванням операції альтернатива будуть мати такий вигляд:

- операція декременту над n елементами масиву M

$$DEC_n(\overline{a_1, a_n}) ::= [Z(i)](a_i - 0 \vee a_i - 1);$$

- операція інкременту над n елементами масиву R

$$INC_n(\overline{r_1, r_n}) ::= [p(i)](r_i + 0 \vee r_i + 1);$$

- операція маскування над n елементами масиву P

$$MASK_n(\overline{m_1, m_n}) ::= [Z(i)](m_i = 0 \vee m_i = 1).$$

Таким чином, удосконалений паралельний алгоритм сортування з формуванням рангів у термінах САА Глушкова можна представити в такий спосіб:

$$SORT^n(\overline{a_1, a_n}) ::= SET_n(\overline{r_1, r_n}) + SET_n(\overline{m_1, m_n}) \times [Z(n-1)]$$

$$\{DEC_n(\overline{a_1, a_n}) \times MASK_n(\overline{m_1, m_n}) \times INC_n(\overline{r_1, r_n})\} \times OUT(R) \times FIN. \quad (3)$$

З аналізу (3) видно, що основною частиною паралельного алгоритму є цикл,

який завершується після обнуління передостаннього за величиною елемента сортуваного масиву. У базовому циклі виконуються операції декременту і маскуванню рангів з урахуванням обнуління конкретного елемента числового масиву, а за результатом сформованих елементів маски виконується операція інкременту рангів.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Запропонований підхід до сортування з ранжуванням відрізняється від наведеного у [14] тим, що у розглядуваному випадку ранги відсортованих елементів масиву чисел остаточно формуються вже після обнуління передостаннього за величиною числа у масиві, а не після завершення повного обнуління числового масиву. Це зумовлено тим, що на початку процесу сортування усім рангам було надано значення 1, у результаті в процесі обнуління кожного числа його ранг залишився незмінним, а усі інші незамасковані ранги, тобто більші за нього за величиною, збільшилися на одиницю.

Для обчислення часових характеристик алгоритмів сортування, для яких характерним є випадковий розподіл значень елементів числового масиву, визначальним показником є відповідна залежність від розмірності числового масиву або максимальне значення тривалості процесу сортування [7, 13]. Так, для відомих методів та способів сортування цей показник варіюється від $O(n^2)$ до $O(n \cdot \log n)$ [7, 9, 20], при цьому найкращим вважається досягнення показника $O(n)$ [10, 19], де n — кількість елементів числового масиву.

Для наведеного у [14] алгоритму сортування його тривалість має залежність $O(a_i^{\max})$, де a_i^{\max} — максимальний за величиною елемент a_i . Причому максимальне значення цього елемента може дорівнювати величині 2^{m-1} для двійково-кодованих m -розрядних чисел. Отже, в цьому випадку тривалість кожного циклу сортування визначається тактом операції декременту t_{dec} , а весь час сортування

$$T_s^{\max} = t_{\text{dec}} \cdot (2^m - 1). \quad (4)$$

Таким чином, максимальна кількість циклів для запропонованого удосконаленого алгоритму сортування визначається числовим значенням найближчого до максимального числа у масиві, тобто завжди менше величини $2^m - 1$, а весь час сортування менше T_s^{\max} (4) від одного циклу і більше.

ВИСНОВКИ

У цій статті запропоновано паралельний алгоритм сортування, в якому для формування рангів елементів числового масиву не використовуються модулі порівняння–обміну (compare-exchange modules), а застосовуються швидкісні операції декременту/інкременту відповідно до елементів числового масиву і масиву їхніх рангів.

Зменшення кількості циклів процесу сортування з ранжуванням досягається завдяки зупинки процесу формування масиву рангів, коли кількість $(n - 1)$ елементів числового масиву обнулено, крім останнього елемента, найбільшого за величиною.

Результат відображення вдосконаленого паралельного алгоритму сортування з формуванням рангів у базисі САА Глушкова свідчить про функціональну потужність САА Глушкова через різноманітність застосувань та компактність представлення конкретного алгоритму.

СПИСОК ЛІТЕРАТУРИ

1. Гнатієнко Г.М., Снитюк В.Є. Експертні технології прийняття рішень. Київ: ТОВ «Маклаут», 2008. 444 с.
2. Шлезингер М., Главач В. Десять лекцій по статистическому и структурному распознаванию. Киев: Наук. думка, 2004. 548 с.

3. George F. Luger. Artificial intelligence: Structures and strategies for complex problem solving. 6th ed. Publisher: Addison-Wesley, 2008. 792 p.
4. Lucas V. Answer ranking in community question answering: A deep learning approach. Cornell University, 2022. 72 p. <https://doi.org/10.48550/arXiv.2212.01218>.
5. Shehata M., Abdelnaeem M., Mokhiamar O. Integrated multiple criteria decision-making framework for ranking Pareto optimal solutions of the multiobjective optimization problem of tuned mass dampers. *Ocean Engineering*. 2023. Vol. 278, N 114440. <https://doi.org/10.1016/j.oceaneng.2023.114440>.
6. Chambon T., Guillaume Jean-Loup, Lallement J. Information complexity ranking: A new method of ranking images by algorithmic complexity. *Entropy*. 2023. Vol. 25, N 3. P. 439. <https://doi.org/10.3390/e25030439>.
7. Knuth D.E. The art of computer programming. V.3, Sorting and Searching. Reading: Addison-Wesley Longman, Inc., 1998. 800 p.
8. Garland M. Sorting Programming massively parallel processors. (Fourth Edition). Morgan Kaufmann, 2023. 551 p. <https://doi.org/10.1016/B978-0-323-91231-0.00019-7>.
9. Sedgewick R. Algorithms in C++: Fundamentals, data structures, sorting, searching. Addison-Wesley, 1998. 716 p.
10. Martyniuk T.B. Structure of associative processor with bitwise serial processing of data. *Engineering Simulation*. 1997. Vol. 14. P. 383–389.
11. Martyniuk T., Vasilyeva T., Suprigan V., AL-Heyari M. Features of sorting memory realization. *Proceedings of SPIE-The International Society for Optical Engineering*. 2001. Vol. 4425. P. 89–91.
12. Martyniuk T., Krukivskyi B., Kupershtein L., Lukichov V. Neural network model of heteroassociative memory for the classification task. *Radioelectronic and Computer Systems*. 2022. № 2(102). P. 108–117. <https://doi.org/10.32620/reks.2022.2.09>.
13. Kohonen T. Content-addressable memories. Berlin; Heidelberg: Springer-Verlag, 1987. 388 p.
14. Martyniuk T.B., Krukivskyi B.I. Peculiarities of the parallel sorting algorithm with rank formation. *Cybernetics and Systems Analysis*. 2022. Vol. 58, N 1. P. 24–28. <https://doi.org/10.1007/s10559-022-00431-8>.
15. Цейтлин Г.Е. Проектирование последовательных алгоритмов сортировки: классификация, трансформация, синтез. *Программирование*. 1989. №3. С. 3–24.
16. Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А. Алгеброалгоритмические модели и методы параллельного программирования. Киев: Академперіодика, 2007. 631 с.
17. Andon P.I., Doroshenko A.Yu., Zhereb K.A., Yatsenko O.A. Algebra-algorithmic models and methods of parallel programming. Kyiv: Akadempriodyka, 2018. 192 p.
18. Doroshenko A., Yatsenko O. Formal and adaptive methods for automation of parallel programs construction: emerging research and opportunities. Hershey: IGI Global, 2021. 279 p. <https://doi.org/10.4018/978-1-5225-9384-3>.
19. Kozhemiako V.P., Martyniuk T.B., Khomyuk V.V. Distinctive features of structural programming of synchronous sorting algorithms. *Cybernetics and Systems Analysis*. 2006. Vol. 42, N 5. P. 714–723.
20. Lorin H. Sorting and sort systems. Mass.: Addison-Wesley Publishing Company, 1975. 373 p.

T.B. Martyniuk, B.I. Krukivskyi

ADVANCED MODEL OF PARALLEL SORTING ALGORITHM WITH RANK FORMATION

Abstract. The model of parallel sorting of a number array with ranking based on the simultaneous application of high-speed decrement/increment operations according to the array of numbers and the array of their ranks is improved. Acceleration of the proposed algorithm is achieved by fixing the result of zeroing (n-1) elements of the number array instead of its complete zeroing. The description of the algorithm of parallel sorting with the formation of ranks in a compact form using the basis Glushkov's system of algorithmic algebras (SAA) is given.

Keywords: system of algorithmic algebras, parallel sorting, mask, rank, decrement/increment.

Надійшла до редакції 03.05.2023