DOI 10.1007/s10559-022-00431-8
Cybernetics and Systems Analysis, Vol. 58, No. 1, January, 2022

PECULIARITIES OF THE PARALLEL SORTING ALGORITHM WITH RANK FORMATION

T. B. Martyniuk[1†] **and B. I. Krukivskyi**[1‡]

UDC 004.94.2

Abstract. *A new approach to parallel sorting of an array of numbers with formation of their ranks is analyzed. In the sorting process, operations, such as a decrement operation to process elements of a numerical array, and an increment operation to form their ranks are performed. A description of the parallel sorting algorithm with rank formation in the basis of Glushkov's System of Algorithmic Algebras (SAA) is proposed.*

Keywords: *system of algorithmic algebras, parallel sorting, numerical slice, rank, mask.*

INTRODUCTION

There is practical interest in improving the methods and means of sorting data sets [1], associated with a wide range of their application, which includes, in particular, Internet search engines, database management systems (DBMs), and signal and image processing (median and rank filtering) [2]. In addition, the modern element base (PLD and optoelectronic matrices of smart pixels) allows the hardware to implement, for example, network methods for sorting large data sets with acceptable speed [3].

However, for other software and hardware implementation of the developed algorithms, it is desirable to clearly see their description in a compact and simple form. Formalized descriptions of a wide range of algorithms and, in particular, Glushkov's systems of algorithmic algebras (SAA) [4–7] have proved themselves well in this. Glushkov's SAA basis [8], in turn, is expanded by modifications [9], and the process of parallel programming based on it is improved [10–12]. This article considers the peculiarities of the parallel algorithm representation in terms of Glushkov's SAA.

The aim of the work is to confirm the functionality of a compact description, in terms of Glushkov's SAA, of varieties of parallel sorting methods.

PECULIARITIES OF SLICE PROCESSING OF A NUMBER ARRAY

A peculiarity of this algorithm for sorting a linear array of numbers is the parallel slice processing. There are several options for slice processing, where a sequence of slices is formed. For example, bit slice processing is used for parallel-sequential processing of data arrays in associative processors [13–16].

There is another type of slice processing, where the slice is taken as the vector of difference, which is formed by simultaneously subtracting, from all the elements of the current vector array, the numbers of the smallest (non-zero) of them. In [17], such a slice is commonly referred to as a difference slice. This principle of difference-slice processing was substantiated and implemented during the modeling of the threshold neuron [17], and also described in the materials of Glushkov's SAA for multiprocessing of vector data sets [18] and classification of objects based on discriminant functions [19].

---

[1]Vinnytsia National Technical University, Vinnytsia, Ukraine, [†]*martyniuk.t.b@gmail.com*; [‡]*smiletex11@gmail.com*.
Translated from Kibernetyka ta Systemnyi Analiz, No. 1, January–February, 2022, pp. 31–36. Original article submitted July 12, 2021.

At the same time, it should be noted that the use of bit slice processing for sorting allows to form ranks (indices) of elements of the sorted array [15, 16]. In this case, a linear array of ranks is formed as a result of sorting. Each rank is located at a position that corresponds to the position of the element in the input array. In addition, the rank value corresponds to the position of the element in the sorted array [20].

The generated ranks can also be considered as addresses, which are convenient for reading the necessary array elements [13] from the associative memory. The application of ranking elements during their sorting is especially relevant for median image filtering [2, 21], as well as for determining extreme (minimum or maximum) elements in an array of numbers [13].

ALGORITHM FOR SLICE SORTING OF A LINEAR NUMBER ARRAY

In [22], a modification of slice processing is described and investigated, in which the decrement operation (reduction by one) is applied simultaneously to all elements of an array of numbers with their successive zeroing. It is the moment of fixing the sign of zeroing of a certain element of the array that is used to form the result of associative processing, i.e., to search for extreme numbers and to search by key among the elements of the input array of numbers [22].

In this paper, we consider the sorting process, in which the decrement operation is applied to the elements of a numerical array, as well as the increment operation (increment by one) relative to the ranks of the corresponding elements. Therefore, the sequential zeroing of elements of the input array of numbers $M = \{a_i\}$ takes place, as well as "growing" of their respective ranks $R = \{r_i\}$ and $i = \overline{1, n}$. The process of the proposed slice sorting with the formation of ranks includes the following operations:

a) setting in the initial (single) state of the ranks and the mask elements for all elements of the array;

b) decrementing over all elements of the array;

c) masking the rank of the corresponding zeroed element of the array;

d) incrementing over all unmasked ranks of array elements.

The main conditions (signs) of this process are as follows:

a) at least one ith element of the input array at the current time is equal to zero, i.e., $\exists\, a_i = 0$; $i = \overline{1, n}$;

b) all n elements of the input array at the current time are equal to zero, i.e., $\forall a_i = 0$.

The first condition is used for zeroing the corresponding ith element of the mask and masking ith rank. The second condition indicates the completion of the sorting process.

An example of sorting with an array of four numbers {2, 5, 3, 4} is presented in Table 1. In the zero cycle, unit is fixed in the rank array. During the first cycle, the elements of the array of numbers are reduced by one (decrement operation), and the elements of the array of ranks, in turn, remain unchanged. In the second cycle, one of the elements of the array of numbers is zeroed, namely, the first element. As a result, its rank remains unchanged (with a value of 1), and the ranks of all other numbers increase by one and take the value of 2 (operation of increment). The sorting process takes place until all the elements in the array of numbers are converted to zero, and as a result an array of ranks is formed.

Table 1 highlights the moments when one number from the array is zeroed and masked, remaining with the same rank; at the same time, all other ranks increase by one. The process of sorting with ranking is performed in five cycles, as the maximum value in the array is 5. The size of the array of numbers in this case does not affect the duration of the process, because the processing of numbers is performed under all of them at the same time.

The peculiarity of the proposed slice sorting is the absence of the operation of pairwise comparison of array elements and a complex network of switching elements according to the results of their comparisons. In addition, the duration of the sorting process does not depend on the quantity n of elements of the sorted array, but is determined by the value of the maximum element in the array of numbers, i.e., depends on the duration of decrement operation to complete zeroing of all elements of the array [22, 23]. Therefore, the maximum duration of the sorting process in this case is determined by the value 2^n for binary-coded numbers.

TABLE 1. The Process of Sorting with Ranking

0th Cycle			**1st Cycle**			**2nd Cycle**		
Number Array	Mask Elements	Array of Ranks	Number Array	Mask Elements	Array of Ranks	Number Array	Mask Elements	Array of Ranks
2	1	1	1	1	1	$\boxed{0}$	**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
3rd Cycle			**4th Cycle**			**5th Cycle**		
Number Array	Mask Elements	Array of Ranks	Number array	Mask Elements	Array of Ranks	Number Array	Mask Elements	Array of Ranks
0	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**
2	1	3	1	1	4	$\boxed{0}$	**0**	**4**
$\boxed{0}$	**0**	**2**	**0**	**0**	**2**	**0**	**0**	**2**
1	1	3	$\boxed{0}$	**0**	**3**	**0**	**0**	**3**

RECORDING THE PARALLEL SLICE SORTING ALGORITHM IN THE GLUSHKOV'S SAA BASIS

To describe the considered parallel sorting algorithm, taking into account its features, it is necessary to expand the Glushkov's SAA basis [5, 9]. This is connected to the necessity to consider the three following arrays of the same length n:

(i) the array of elements $M = a_1, a_2, \ldots, a_n$ that are sorted;

(ii) the array of ranks of corresponding elements $R = r_1, r_2, \ldots, r_n$ that are sorted;

(iii) the array of mask elements $P = m_1, m_2, \ldots, m_n$.

Introduction of the array P is required to determine, by using a masked array of ranks R of the location of a particular element of the array M in the sorted array.

Since all the elements of each array are involved in the sorting process simultaneously, pointers and markers are not used for their markup. On the marked arrays, the following base conditions (predicates) and operators with orientation on the known basis of regular schemes (RSs) for sorting algorithms in the Glushkov's SAA [5] are used:

(i) $Z(i)$ is true if the condition $\exists a_i = 0$ is satisfied;

(ii) $Z(n)$ is true if the condition $\forall a_i = 0$ is satisfied;

(iii) $p(i)$ is true if the condition $\exists m_i = 0$ is satisfied;

(iv) $\mathrm{OUT}(R)$ is the result output operator;

(v) FIN is the RS shutdown operator.

In the sorting algorithm under study, all elements of three arrays are considered as the zone of predicates and operators, because executing operators and checking whether predicates are true is carried out simultaneously on all elements of each array. In what follows, to determine the basic operators that are introduced to describe the sorting algorithm, we need to use the following basic operations related to the signature of the extended SAA-M [4, 5]:

(i) disjunction $\alpha \vee \beta$;

(ii) composition $A \times B$, i.e., successive use of operators A and B;

(iii) alternative $[\alpha]\ (A \vee B)$, i.e., if α is true, then A, otherwise, B;

(iv) cycle $[\alpha]\ \{A\}$, i.e., execute A until α is false, and if α is true, then it is the end of the cycle.

We can also use the notation $A + B$ for parallel application of operators A and B [7]. Therefore, to perform decrement, increment, and masking operations, we should represent the base operators as follows:

(i) parallel establishment of initial (single) states of n elements of an array of ranks R as follows:

$$\mathrm{SET}_n(\overline{r_1, r_n})::=(\overline{r_1, r_n}=1),$$

(ii) parallel establishment of initial (single) states of n elements of a mask of the array P as follows:

$$\mathrm{SET}_n(\overline{m_1, m_n})::=(\overline{m_1, m_n}=1).$$

Component operators, in this case, take their form by using the following alternatives:

(iii) the following decrement operation on n elements of the array M:

$$\mathrm{DEC}_n(\overline{a_1, a_n})::=[a_i=0]\ \ (a_i-0 \vee a_i-1)$$

or

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

(iv) the following increment operation on n elements of the array R:

$$\mathrm{INC}_n(\overline{r_1, r_n})::=[m_i=0]\ \ (r_i+0 \vee r_i+1)$$

or

$$\mathrm{INC}_n(\overline{r_1, r_n})::=[p(i)]\ \ (r_i+0 \vee r_i+1); \tag{2}$$

(v) the following masking operation on n elements of the array P:

$$\mathrm{MASK}_n(\overline{m_1, m_n})::=[a_i=0]\ \ (m_i=0 \vee m_i=1)$$

or

$$\mathrm{MASK}_n(\overline{m_1, m_n})::=[Z(i)]\ \ (m_i=0 \vee m_i=1). \tag{3}$$

As a result, the parallel algorithm of slice sorting with the formation of ranks in terms of the Glushkov's SAA can be written as follows:

$$\mathrm{SORT}^n(\overline{a_1, a_n})::=\mathrm{SET}_n(\overline{r_1, r_n})+\mathrm{SET}_n(\overline{m_1, m_n})$$
$$\times[Z(n)]\{\mathrm{DEC}_n(\overline{a_1, a_n})\times\mathrm{MASK}_n(\overline{m_1, m_n})\times\mathrm{INC}_n(\overline{r_1, r_n})\}\times\mathrm{OUT}(R)\times\mathrm{FIN}. \tag{4}$$

Taking into account the description of the component operators in the form of (1)–(3), this sorting algorithm takes a more detailed form

$$\mathrm{SORT}^n(\overline{a_1, a_n})::=\mathrm{SET}_n(\overline{r_1, r_n})+\mathrm{SET}_n(\overline{m_1, m_n})$$
$$\times[Z(n)]\{[Z(i)](a_i-0 \vee a_i-1)\times[Z(i)](m_i=0 \vee m_i=1)$$
$$\times[p(i)](r_i+0 \vee r_i+1)\}\mathrm{OUT}(R)\times\mathrm{FIN}. \tag{5}$$

Analyzing (4) and (5), we see that the main part of the parallel algorithm of slice processing is a cycle that ends after zeroing of all the elements of the sorted array. In the cycle itself, operations of decrement and masking of ranks by digital value of mask elements are performed, taking into account zeroing of a specific element of the numerical array, which is sorted. After each zeroing of the sorted array element, the rank increment operation is performed taking into account the generated mask elements.

CONCLUSIONS

Using the apparatus of basic predicates and operations that make up the signature of the extended Glushkov's SAA, we can describe any kind of sequential and parallel algorithms, which is illustrated on the example of sorting algorithms. At the same time, a characteristic feature of the Glushkov's SAA basis is its flexibility and adaptability, which confirms its inherent functional power for various applications.

REFERENCES

1. D. E. Knuth, The Art of Computer Programming, Vol. 3, Sorting and Searching, Addison-Wesley Longman, Inc., Reading (1998).
2. W. K. Pratt, Introduction to Digital Image Processing, CRC Press (2014).
3. V. Y. Osynskyi, T. B. Martyniuk, A. A. Kozlov, and M. S. Nasser Mokhamed, “Features of the optoelectronic realization of the sorting neural network,” Optoelectronic Information-Power Technologies, Vol. 18, No. 2, 58–67 (2009).
4. G. E. Tseitlin, “Design of serial sorting algorithms: classification, transformation, synthesis,” Programmirovanie, No. 3, 3–24 (1989).
5. G. E. Tseitlin, “Parallelization of sorting algorithms,” Cybern. Syst. Analysis, Vol. 25, No. 6, 790–798 (1989). https://doi.org/10.1007/BF01069781.
6. E. A. Yatsenko, “Regular schemes of algorithms of address sorting and search,” Upravl. Sistemy i Mashiny, No. 5, 61–66 (2004).
7. V. P. Kozhemiako, T. B. Martyniuk, and V. V. Khomyuk, “Distinctive features of structural programming of synchronous sorting algorithms,” Cybern. Syst. Analysis, Vol. 42, No. 5, 714–723 (2006). https://doi.org/10.1007/s10559-006-0110-7.
8. G. E. Tseitlin, “Structured programming in symbolic multiprocessing,” Cybern. Syst. Analysis, Vol. 19, No. 5, 614–625 (1983). https://doi.org/10.1007/BF01068755.
9. G. E. Tseitlin, “Glushkov algebras and clone theory,” Cybern. Syst. Analysis, Vol. 39, No. 4, 509–516 (2003). https://doi.org/10.1023/B:CASA.0000003501.97990.e5.
10. F. I. Andon, A. E. Doroshenko, G. E. Tseitlin, and E. A. Yatsenko, Algebraic–Algorithmic Models and Methods of Parallel Programming [in Russian], Akademperiodika, Kyiv (2007).
11. P. I. Andon, A. Y. Doroshenko, and K. A. Zhereb, ”Programming high-performance parallel computations: formal models and graphics processing units,” Cybern. Syst. Analysis, Vol. 47, No. 4, 659–668 (2011). https://doi.org/10.1007/s10559-011-9346-y.
12. F. I. Andon, A. E. Doroshenko, A. G. Beketov, V. A. Iovchev, and E. A. Yatsenko, “Software tools for automation of parallel programming on the basis of algebra of algorithms,” Cybern. Syst. Analysis, Vol. 51, No. 1, 142–149 (2015). https://doi.org/10.1007/s10559-015-9706-0.
13. T. Kohonen, Content-Addressable Memories, Springer-Verlag, Berlin–Heidelberg (1987).
14. K. J. Thurber, Large Scale Computer Architecture: Parallel and Associative Processors, Hayden Book Company, Rochelle Park, N.J. (1976).
15. T. B. Martyniuk, “Structure of associative processor with bitwise serial processing of data,” Engineering Simulation, Vol. 14, 383–389 (1997).
16. T. Martyniuk, T. Vasilyeva, V. Suprigan, and M. AL-Heyari, “Features of sorting memory realization,” Proc. SPIE, Vol. 4425 (Selected Papers from the Intern. Conf. on Optoelectronic Information Technologies), 89–91 (2001).
17. T. Martyniuk, L. Kupershtein, and A. Kozhemiako, Aspects of Difference-Slice Data Processing in Neural Network Structures [in Russian], LAMBERT Academic Publishing (2018).
18. T. B. Martyniuk and V. V. Khomyuk, “Data array multiprocessing by difference slices,” Cybern. Syst. Analysis, Vol. 47, No. 6, 948–953 (2011). https://doi.org/10.1007/s10559-011-9374-7.
19. T. B. Martyniuk, A. V. Kozhemiako, and L. M. Kupershtein, “Formalization of the object classification algorithm,” Cybern. Syst Analysis, Vol. 51, No. 5, 751–756 (2015). https://doi.org/10.1007/s10559-015-9767-0.
20. H. Lorin, Sorting and Sort Systems, Addison-Wesley, Reading, MA (1975).
21. K. I. Kucherenko and E. F. Ochin, “Two-dimensional median image filtering processors based on sorting networks,” Avtometriya, No. 2, 13–19 (1988).
22. T. B. Martyniuk, N. O. Denysiuk, and B. I. Krukivskyi, “Associative processors with parallel-serial data processing,” Information Technology and Computer Engineering, Vol. 44, No. 1, 27–36 (2019). https://doi.org/10.31649/1999-9941-2019-44-1-27-36.
23. T. B. Martyniuk and B. I. Krukivsky, “Model of parallel sorter for associative processor,” Visnyk of Vinnytsia Polytechnical Institute, No. 5 (152), 49–55 (2020). https://doi.org/10.31649/1997-9266-2020-152-5-49-55.