Design of Parallel Sorting System Using Discrete-Time Neural Circuit Model

Pavlo V Tymoshchuk
CAD Systems Department, Lviv Polytechnic National University, Lviv, 79000, Ukraine
Email: pavlo.v.tymoshchuk@lpnu.ua

Abstract. A parallel sorting system based on discrete-time K-Winners-Take-All (KWTA) neural circuit (NC) model is designed. The system is described by a set of difference equations and by step functions. The KWTA NC model of discrete-time is given by difference equation and by step functions. Corresponding functional block-diagrams of the system are presented. The upper limit on the number of iterations required to achieve a convergence of the sorting process to the steady state is defined. The system does not need a knowledge of range of input data change. In order to use the system, the minimal difference between the inputs should be known. The system is suitable for processing unknown inputs of finite values located in arbitrary unknown finite range. The system is characterized by arbitrary finite resolution of inputs, high speed, moderate computational complexity and complexity of software implementation. The results of computer simulations illustrating efficiency of the system are provided.

1. Introduction
Sorting is an operation of arranging input data in an ascending ordered sequence, descending ordered sequence, alphabetic ordered sequence, etc. Over 25 percent of all computing time is spent on sorting. Sorting operation is widely applied in data and signal processing, telecommunication, navigation, mobile devices, artificial intelligence, very large scale integration design etc. [1]. Therefore, a problem of designing efficient sorting algorithms to minimize time and storage of sorting is extremely important.

A number of sequential sorting approaches are applied [1]. A sequential sorting algorithm may not be efficient sufficiently if a large volume of data should be sorted since they have quadratic computational complexity \( \Theta(N^2) \). Therefore, parallel sorting algorithms of linear computational complexity \( \Theta(N) \) can be used in this case. Various parallel sorting algorithms and architectures are discribed in [2]. Parallel sorting systems based on continuous-time neural networks (NNs) are applied in [3] – [10] and in references herein. Hardware implementations of sorting systems in VLSI technology are presented, for instance, in [8], [9].

Discrete-time neural networks are more accurate, reliable, more appropriate for processing real noised data. Such type of the networks are more convenient for implementation in parallel software. Moreover, they can be implemented in an up-to-date parallel digital hardware [11].
Let us design a parallel sorting system based on discrete-time neural circuit (NC) model presented in [12]. The system should possess arbitrary finite high resolution of input data, high speed of data processing, and linear computational complexity. Moreover, the system should have moderate hardware implementation complexity and be capable to process any finite unknown input data located in any finite unknown range. Computer simulations confirming theoretical derivations should be provided.

2. Basic discrete-time neural circuit
Consider a vector \( a=(a_1,a_2,\ldots,a_N)^T \in \mathbb{R}^N, \ 1 < N < \infty \) of real input data with unknown values of its elements. The inputs are distinct and finite, i.e.
\[
a_i \neq a_j, \ -\infty < a_i, a_j < \infty, \tag{1}
\]
where \( i \neq j = 1,2,\ldots,N \). The simple and fast discrete-time K-winners-take-all (KWTA) NC presented in [12] is capable to determine the largest \( K \) of these inputs, which are appealed to as the winners. In particular, the circuit processes the vector \( a \) of these inputs to get a such vector \( b=(b_1,b_2,\ldots,b_N)^T \) of corresponding outputs that the following KWTA property is met:
\[
b_i > 0, i=1,2,\ldots,K; \ b_j < 0, \ j=K+1,K+2,\ldots,N. \tag{2}
\]
We simplify the NC presented in [12] having described it by the following difference equation:
\[
y(l+1) = y(l) + r \text{sgn}(D(y)), \tag{3}
\]
where \( y(l) \) is a discrete-time state variable with an initial condition \(-\infty < y(l) < \infty\), \( D(y) = \sum S_k(y) - K \) is a difference function between obtained and required numbers of positive outputs,
\[
S_k(y) = \begin{cases} 1, & \text{if } a_k - y > 0; \\ 0, & \text{otherwise} \end{cases} \tag{4}
\]
is a step function,
\[
\text{sgn}(D(y)) = \begin{cases} 1, & \text{if } D(y) > 0; \\ 0, & \text{if } D(y) = 0; \\ -1, & \text{if } D(y) < 0 \end{cases} \tag{5}
\]
is a signum function, \( r \) is a parameter that should have a value less than minimal difference between inputs, i.e., the inequalities \( 0 < r \leq \min|a_i - a_j|, \ i \neq j = 1,2,\ldots,N \) should be met. The quantity of iterations required to reach a convergence of the search process to the steady state KWTA operation is limited from above as follows:
\[
m \leq \left| y(l) - y^* \right| / r, \tag{6}
\]
where \( y^* \) is a steady state value of \( y(l) \). As it can be seen, the NC described by difference equation (3) and by output equations (4) requires setting a value of the parameter \( r \) that should be smaller than minimal difference between inputs. This means that the minimal difference between inputs should be known.

3. A discrete-time model of neural circuit based parallel sorting system
In parallel sorting, an order of sorting can be represented as a permutation matrix. In such a matrix, “1” in the row marked with \( a_i \) and column labeled with \( g_j \) can be determined as the ith item in an unsorted list and jth item in a sorted list [8] – [10]. For instance, in the case \( i=1,2,\ldots,10 \) an unordered list of inputs \{\( a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10} \)\} and their sorted list \{\( a_2,a_4,a_1,a_3,a_5,a_7,a_6,a_{10},a_8,a_9 \)\} are described by the following corresponding permutation matrix:
\[ d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7 \ d_8 \ d_9 \ d_{10} \ \text{rank} \\
\begin{array}{cccccccccc}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} \quad (7)

The matrix (7) can be converted to the sorting matrix given by
\[ S^1 \ S^2 \ S^3 \ S^4 \ S^5 \ S^6 \ S^7 \ S^8 \ S^9 \ S^{10} \ \text{rank} \\
\begin{array}{cccccccccccc}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 5 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 7 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 8 \\
\end{array} \quad (8)

In general case the sorting results can be presented as follows:
\[ d_j = a^T S^j, \hspace{1cm} d_{K+i} = a^T (S^{K-i} - S^K), \]
where the elements of the \( K \)th column \( S^K = [S^K_1, S^K_2, ..., S^K_N]^T \), \( K=1,2,\ldots,N \) of the sorting matrix are determined by the step functions (4) using the difference equation (3).

Since \( S^K = [I, I, \ldots, I]^T, S^N = [I, I, \ldots, I]^T \), only \( N-I \) equations (3) are necessary and each of these equations computes one column of the sorting matrix from left to right with \( k \) increasing from \( I \) to \( N-I \). Therefore, only \( N-I \) neurons are required when compared with the other sorting systems that use \( N^2 \) neurons [9]. Specifically, the 1WTA network is necessary to define the largest element of the list. The 1WTA network and the 2WTA network are required for computing the second item in the list in a parallel mode without recaluation the first item. The 2WTA network and the 3WTA network are necessary for determining the third item in the list in a parallel mode without recalculating the second item and so on. As such, the whole list of \( N \) items can be sorted by using the KWTA, \( K=1,2,\ldots,N-I \) networks without the need to calculate the last item by setting \( S^K = [I, I, \ldots, I]^T [3], [8] \).

4. Functional block-diagram of the system

A functional block-diagram of the discrete-time NC described by difference equation (3) and by output equations (4) is shown in figure 1(a). Figure 1(b) presents a functional block-diagram of part of parallel sorting system described by the first equation of (9). A functional block-diagram of part of parallel sorting system described by the second equation of (9) is shown in figure 1(c). The diagrams include inputs \( a_1, a_2, \ldots, a_N \), summers \( \sum \), blocks \( S_1, S_2, \ldots, S_N \) of step functions \( S_k(y), k=1,2,\ldots,N \), signum function \( \text{sgn}(5) \), multiplier \( \times \), time delay \( z^{-1} \), external sources of constant signals \( r_K, y(1) \), and outputs \( S_{1K}(y), S_{2K}(y), \ldots, S_{NK}(y) \).
Figure 1. (a) The functional block-diagram of the discrete-time NC described by difference equation (3) and by output equations (4); (b) The functional block-diagram of part of parallel sorting system based on NN described by first equation of (9); (c) The functional block-diagram of part of parallel sorting system based on NN described by second equation of (9).

The diagram shown in figure 1(a) consists of one multiplier, one time delay unit, three sources of constant signals (or two sources of constant signals if \( x(1)=0 \)), \( N+1 \) controlled switches, and \( N+2 \) summers. The diagram presented in figure 1(b) has \( N \) multipliers and one summer. The diagram shown in figure 1(c) consists of \( N \) multipliers and \( N+1 \) summers.

A functional block-diagram of one of the simplest comparative discrete-time parallel sorting system based on continuous-time KWTA NNs with the Heaviside step activation function and a single state variable described in [10] that was applied for parallel sorting in similar way needs one source of...
constant signals, one integrator, \( N \) switches, and \( N+1 \) summers. Thus, from the complexity point of view, the discrete-time parallel sorting system based on NC described by difference equation (3) and by output equations (4) is close to other comparable analogs.

The functional block-diagrams presented in figure 1 can be implemented in a parallel software or parallel digital VLSI hardware. Software implementation of the diagrams can be preferred if it is necessary to have an accurate, simple and flexible parallel realization. The diagrams can be implemented in a digital hardware if fast, compact, reliable, and noise insensitive realization is required [13].

5. Computer simulations of the system

Consider an example of computer simulations that demonstrate the performance of the discrete-time parallel sorting system described by equations (3), (4), and (9). Let us sort nine elements of uniformly distributed on the interval (-10.0, 10.0) random numbers, using parallel sorting system described by equations (3), (4), and (9). We set the initial values of the discrete-time state variables \( x(k) = 5.0 \), \( k = 1, 2, ..., N \). Using eight NCs described by difference equations (3) and by output equations (4) we need only eight neurons unlike to the 81 neurons in the analog sorting network in [9].

Figure 2 shows the dynamics of inputs and outputs of the system used for parallel sorting with \( r = 0.01 \), \( x(k) = 5.0 \), \( k = 1, 2, ..., N \). As it can be seen, the correct sorting of the inputs is reached in \( n < 40 < \frac{68 - 5}{0.01} = 500 \) iterations in accordance with estimate (6). Elapsed time necessary for completing this sorting in Matlab is approximately equal to 0.07 s.

Figure 3 presents the maximal numbers of iterations required for parallel sorting nine elements of uniformly distributed on the interval (-10.0, 10.0) random numbers described by equations (3), (4), and (9) for \( r = 0.01 \) and for different initial values of the discrete-time state variables \( x(k) \), \( k = 1, 2, ..., N \) chosen on the interval (-10.0, 10.0). As it can be seen in figure 3, the maximal number of iterations necessary to sort the inputs in this case is not larger than 50 for all values of \( x(k) \), \( k = 1, 2, ..., N \). Maximal elapsed time required for completing this sorting in Matlab is approximately equal to 0.05 s.

Figure 2. Trajectories of outputs of the parallel sorting system described by equations (3), (4), and (9) for \( N=9 \) uniformly distributed on the interval (-10.0, 10.0) random inputs \( a \), \( r = 0.01 \), and \( y(k) = 5.0 \), \( k = 1, 2, ..., N \).

Figure 3: Maximal numbers of iterations necessary for parallel sorting by the parallel sorting system described by equations (3), (4), and (9) for \( r = 0.01 \) and for different initial values of the discrete-time state variables \( x(k) \), \( k = 1, 2, ..., N \) chosen on the interval (-10.0, 10.0).
As it can be seen from the results of simulations, the parallel sorting system described by equations (3), (4), and (9) is accurate and fast. Moreover, this system does not need a knowledge of change range of inputs. In order to use the system a minimal difference between inputs should be known like close comparative discrete-time analog presented in [10]. However, in contrast to parallel sorting system described by equations (3), (4), and (9), this analog requires multiplications of large value difference functions by small value parameters in a right-hand side of the difference equation describing discrete-time KWTA NN model in the case of large number of inputs and small minimal difference between them. This can cause arising inaccuracies which lead to incorrect operation of such parallel sorting system. In addition, indicated multiplications can be difficult to implement in a corresponding hardware.

Conclusions
The discrete-time parallel sorting system of arbitrary unknown finite inputs located in unknown finite change range and corresponding functional block-diagrams are presented. The system possesses high operation speed and arbitrary finite resolution of inputs. Practical operation speed of the system and its resolution are limited by restrictions of software or hardware implementation of the system. The system can be implemented in a parallel software or parallel digital VLSI hardware. The software implementation can be chosen if an accurate, simple and flexible parallel realization is required. The system can be implemented in a digital hardware for parallel operation if fast, reliable, and noise insensitive realization is necessary. High performance of the system used for fast and accurate parallel sorting has been demonstrated by computer simulations. The system can be recommended to use if a large number of unknown finite inputs with known minimal difference between them located in unknown finite change range should be sorted with high resolution and speed.

References
[1] Knuth D E 1973 The Art of Computer Programming, Sorting and Searching (Reading: Addison-Wesley)
[2] Atkins M 1989 Proceedings of International Joint Conference on Neural Networks (Washington DC) (Washington DC: IEEE) p 65
[3] Takefuji Y and Lee K S 1990 IEEE Transactions on Circuits and Systems CAS-37 11 1425-29
[4] Chen W and Hsieh K 1990 Proceedings of International Joint Conference on Neural Networks (San Diego) vol. III (San Diego: IEEE) p 793
[5] Kwon T M and Zervakis M 1992 Proceedings of International Joint Conference on Neural Networks (Baltimore) vol. I (Baltimore: IEEE) p 701
[6] Tseng Y H and Wu J L 1992 Electronics Letters 28 10 906-08
[7] Wang J 1995 IEEE Transactions on Neural Networks 6 4 962–71
[8] Kwon T M and Zervakis M 1995 Multidimensional Systems and Signal Processing 6 4 333-46
[9] Wang J 2010 IEEE Transactions on Neural Networks 21 9 1496-06
[10] Cichocki A and Unbehauen R 1993 Neural Networks for Optimization and Signal Processing (New York: Wiley)
[11] Tymoshchuk P V 2009 Neurocomputing 72 3191-02
[12] Tymoshchuk P V and Shatnyi S V 2015 Proceedings of XX International Seminar/Workshop DIPED (L’viv: IAPMM Press) p 168