Analysis and Design of a k-Winners-Take-All Model With a Single State Variable and the Heaviside Step Activation Function

Jun Wang, *Fellow, IEEE*

***Abstract*—This paper presents a k-winners-take-all (kWTA) neural network with a single state variable and a hard-limiting activation function. First, following several kWTA problem formulations, related existing kWTA networks are reviewed. Then, the kWTA model model with a single state variable and a Heaviside step activation function is described and its global stability and finite-time convergence are proven with derived upper and lower bounds. In addition, the initial state estimation and a discrete-time version of the kWTA model are discussed. Furthermore, two selected applications to parallel sorting and rank-order filtering based on the kWTA model are discussed. Finally, simulation results show the effectiveness and performance of the kWTA model.**

***Index Terms*—Global stability, k-winners-take-all, optimization, recurrent neural network.**

I. Introduction

WINNERS-TAKE-ALL (WTA) is to select the maximum from a collection of inputs. It was shown theoretically to be computationally powerful to represent any continuous and logic functions [29], [30]. As a generalization of WTA to multiple selections, k-winners-take-all (kWTA) selects the k largest inputs out of n inputs ($1 \le k < n$) [31]. Besides their biological plausibility, WTA and kWTA have been widely used in various applications, such as decoding [8], sorting [21], filtering [5], [12], [21], feature extraction [51], clustering [19], classification [18], vision systems [17], [34], signal processing [14], image processing [11], associative memories [36], and mobile robot navigation [7], etc.

When the number of inputs is large or the selection process has to be operated in real time, parallel algorithms and hardware implementation are desirable. Over the past two decades, many WTA and kWTA networks have been proposed (see [2], [10], [13], [16], [22], [24], [26], [27], [31]–[33], [38], [39], [41]–[44], [46], [47], [49], [50]).

This paper presents a novel kWTA model based on previous results. Similar to a recent kWTA network [16], the present model has one single state variable, but a single design parameter only, which makes its model complexity close to (if not yet) minimum. Similar to another recent kWTA network [24], the present model uses the hard-limiting discontinuous (step) activation function, which enables and expedites its output convergence in finite time.

The rest of the paper is organized as follows. In Section II, the kWTA problem is formulated as several optimization problems with reducing complexity. In Section III, several existing kWTA networks developed recently based on a quadratic programming formulation are highlighted. In Section IV, the new kWTA model with a single state variable and Heaviside step activation function is proposed. In Sections V and VI, the global stability of the kWTA network is proven and the bounds of convergence time are derived. In Section VII, a discrete-time version of the kKTA model is described and the selection of its step-size is discussed. In Section VIII, to expedite the convergence, the statistical estimation of the initial state is discussed for random inputs with uniform and normal distributions. In Section IX, simulation results are discussed. In Sections X and XI, two selected applications of the kTWA model for parallel sorting and rank-order filtering are discussed, respectively. Finally, the conclusion is given in Section XII.

II. Problem Formulations

Generally, the kWTA operation can be defined or encoded as the following binary function [30]:

$$x_i = k\mathrm{WTA}(u_i) = \begin{cases} 1, & \text{if } u_i \in \{k \text{ largest elements of } u\} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $u = (u_1, u_2, \ldots, u_n)^T$ is the input vector and $x = (x_1, x_2, \ldots, x_n)^T$ is the output vector.

The kWTA solution can be determined by solving the following linear integer programming problem [43]:

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} u_i x_i \quad \text{or} \quad \text{minimize} - \sum_{i=1}^{n} u_i x_i \\ &\text{subject to} \quad \sum_{i=1}^{n} x_i = k \\ &\qquad\qquad\quad x_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{2}$$

Manuscript received January 15, 2010; revised April 9, 2010 and May 14, 2010; accepted May 15, 2010. Date of publication August 13, 2010; date of current version September 1, 2010. This paper was supported by the Research Grants Council of Hong Kong, under Project CUHK417608E.

The author is with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Shatin, Hong Kong, and also the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200052, China (e-mail: jwang@mae.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2052631

If the kth and $(k+1)$th largest elements of u are different (i.e., the solution is unique), according to its total modularity property [1], the above linear integer programming problem is equivalent to the following linear programming problem [10]:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} u_i x_i \quad \text{or} \quad \text{minimize} - \sum_{i=1}^{n} u_i x_i \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & 0 \le x_i \le 1, \quad i = 1, 2, \ldots, n. \end{array} \tag{3}$$

The dual problem of the above linear programming problem (3) can be described as follows:

$$\begin{array}{ll} \text{minimize} & ky + \sum_{i=1}^{n} v_i \\ \text{subject to} & y + v_i \ge u_i, \quad i = 1, 2, \ldots, n \\ & v_i \ge 0, \quad i = 1, 2, \ldots, n \end{array} \tag{4}$$

where $y, v_1, \ldots, v_n$ are dual decision variables.

It was proven in [27] that the kWTA problem (2) is also equivalent to the following quadratic integer programming problem:

$$\begin{array}{ll} \text{minimize} & \frac{a}{2} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} u_i x_i \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & x_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n. \end{array} \tag{5}$$

where a is a positive constant.

Furthermore, assuming that the kth and $(k+1)$th largest elements of u are different, the quadratic integer programming problem is equivalent to the following quadratic programming problem [27]:

$$\begin{array}{ll} \text{minimize} & \frac{a}{2} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} u_i x_i \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & 0 \le x_i \le 1, \quad i = 1, 2, \ldots, n \end{array} \tag{6}$$

where $a \le \bar{u}_k - \bar{u}_{k+1}$ represents the input resolution, and $\bar{u}_k$ and $\bar{u}_{k+1}$ denote respectively the kth and $(k+1)$th largest elements of u.

In its vector form, the kWTA problem can be described as

$$\begin{array}{ll} \text{minimize} & \frac{a}{2} x^T x - u^T x \\ \text{subject to} & e^T x = k \\ & x \in [0, 1]^n \end{array} \tag{7}$$

where $e = (1, 1, \ldots, 1)^T$.

III. Recent Models

In recent years, several effective kWTA networks were developed based on the quadratic programming problem formulation (6). For example, a kWTA neural network with n neurons was tailored from the simplified dual network for quadratic programming [27]

$$\text{state equation } \epsilon \frac{dv}{dt} = -Mv + g(Mv - v + s) - s \tag{8}$$

$$\text{output equation } x = Mv + s \tag{9}$$

where $\epsilon > 0$ is a scaling constant, $v \in \mathbb{R}^n$ is the state vector, $M = (I - ee^T/n)/a$, $s = Mu + ke/n$, I is an identity matrix, and $g(\cdot)$ is a piecewise linear activation function defined as

$$g(\rho) = \begin{cases} 0, & \text{if } \rho < 0 \\ \rho, & \text{if } 0 \le \rho \le 1 \\ 1, & \text{if } \rho > 1. \end{cases} \tag{10}$$

The state equations of a kWTA network with $n+1$ neurons [24] based on (6) and tailored from the projection network [48] are written as

$$\epsilon \frac{dx}{dt} = -x + g(x - \eta(ax - u - ze)) \tag{11}$$

$$\epsilon \frac{dz}{dt} = -e^T x + k \tag{12}$$

where $x \in \mathbb{R}^n$ and $z \in \mathbb{R}$ are respectively the state vector and variable of output and hidden layers, η is any positive constant. When $a = 0$ [i.e., the quadratic programming problem (6) is reduced to the linear programming problem (3)], the kWTA model (11) becomes the one in [13].

A different kWTA network with a hard-limiting discontinuous activation function is presented in [24] developed based on a one-layer recurrent neural network for quadratic programming in [25]

$$\text{state equation } \epsilon \frac{dv}{dt} = -(I - P)v - [aI + (1-a)P]g_\infty(v) + s \tag{13}$$

$$\text{output equation } x = -\frac{(I - P)v}{a} + \frac{s}{a} + \frac{k(a-1)}{na} e \tag{14}$$

where $P = ee^T/n$, $s = (I - P)u + ke/n$, $g_\infty(\rho)$ is a hard-limiting discontinuous activation function (i.e., Heaviside step function) defined as

$$g_\infty(\rho) = \begin{cases} 0, & \text{if } \rho < 0 \\ [0, 1], & \text{if } \rho = 0 \\ 1, & \text{if } \rho > 0. \end{cases} \tag{15}$$

A much simpler kWTA network with a single state variable only was developed based on the improved dual neural network [16]

$$\text{state equation } \epsilon \frac{dz}{dt} = -e^T x + k \tag{16}$$

$$\text{output equation } x = g\left(ez + \frac{u}{a}\right) \tag{17}$$

where $z \in \mathbb{R}$ is the state variable. The exact same model was reinvented one year after in [49].

Note that (12) and (16) are identical. It is interesting to see that, if $\eta = 1/a$, then the stability of state equation (11) results in the output equation (17).

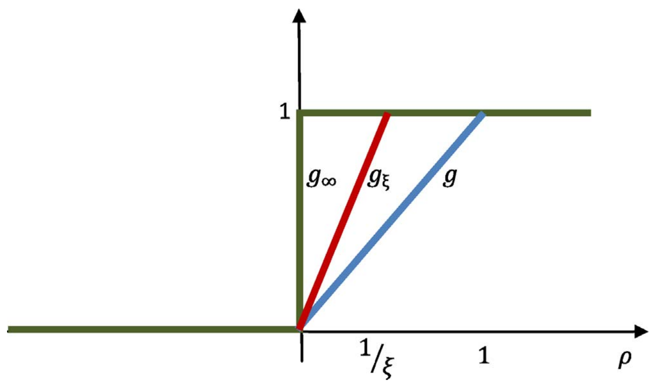

Fig. 1. Piecewise linear activation function with a positive gain parameter ξ as its slope and the Heaviside step activation function as $\xi \to +\infty$.

To ensure correct winner selections, according to the definition of the variables in (1), we must have

$$g\left(z + \frac{\bar{u}_k}{a}\right) = 1 \tag{18}$$

$$g\left(z + \frac{\bar{u}_{k+1}}{a}\right) = 0. \tag{19}$$

According to the definition of the activation function (10), we have

$$z + \frac{\bar{u}_k}{a} \geq 1 \tag{20}$$

$$z + \frac{\bar{u}_{k+1}}{a} \leq 0. \tag{21}$$

Combining (20) and (21) results in $\bar{u}_k - \bar{u}_{k+1} \geq a$, which echoes the result in [27].

A discrete-time counterpart of the above kWTA network [16] was presented in [26] and [49]

$$\text{state equation} \quad z(t+1) = z(t) - \beta\left(\sum_{i=1}^{n} x_i(t) - k\right) \tag{22}$$

$$\text{output equation} \quad x_i(t) = g\left(ez(t) + \frac{u}{a}\right) \tag{23}$$

where $\beta > 0$ is the step size. It was proven that the discrete-time kWTA network is globally convergent if $0 < \beta < 2/n$ [26] or $0 < \beta < 2/\sqrt{n}$ [49].

IV. Model Description

The piecewise linear activation function $g(\cdot)$ can be modified by adding a positive gain parameter ξ as shown in Fig. 1 and the following equation:

$$g_\xi(\rho) = \begin{cases} 0, & \text{if } \rho < 0 \\ \rho, & \text{if } 0 \leq \rho \leq 1/\xi \\ 1, & \text{if } \rho > 1/\xi. \end{cases} \tag{24}$$

When the gain parameter is unity (i.e., $\xi = 1$), then $g_1(\rho) = g(\rho)$. When the gain parameter approaches to positive infinity (i.e., $\xi \to +\infty$), the activation function becomes identical with that in the one of one-layer neural networks with hard-limiting (Heaviside step) activation function (15) in [24]. It will be shown that a kWTA model with positive infinity gain parameter ξ can work better hereafter.

If we use this high-gain activation function in the kWTA model in (16), then the output equation becomes

$$x_i = g_\xi\left(z + \frac{u_i}{a}\right), \quad i = 1, \ldots, n. \tag{25}$$

Similar to the case with the piecewise linear activation function (10) in (20) and (21), to ensure correct selections, we must have

$$z + \frac{\bar{u}_k}{a} \geq \frac{1}{\xi} \tag{26}$$

$$z + \frac{\bar{u}_{k+1}}{a} \leq 0. \tag{27}$$

Combining the above inequalities results in

$$\bar{u}_k - \bar{u}_{k+1} \geq \frac{a}{\xi}. \tag{28}$$

If $\xi \to +\infty$, then the above inequality always holds regardless of the value of a. Without loss of generality, we let $a = 1$.

In view that the output variables x_i are supposed to be binary by their definition, the infinity-gain activation function $g_\infty(\cdot)$ in (15) may be used. Substituting $y = -z$ and $a = 1$ into (25) and let $\xi \to +\infty$, we have a kWTA model with the Heaviside step activation function

$$\text{state equation} \quad \epsilon\frac{dy}{dt} = \sum_{i=1}^{n} x_i - k \tag{29}$$

$$\text{output equation} \quad x_i = g_\infty(u_i - y) \quad i = 1, \ldots, n \tag{30}$$

where $y \in \mathbb{R}$ is the state variable, and $g_\infty(\cdot)$ is the infinity-gain activation function (15) as $\xi \to +\infty$ as defined in (15) and shown in Fig. 1. In view that x_i is a discontinuous function of y, the state equation (29) is an ordinary differential equation with a discontinuous right-hand side.

It is obvious that any equilibrium of the state equation in (29) makes the equality constraint in (5) satisfied. A closer look can reveal that the output variables in (30) corresponding to a stable equilibrium must be a solution to the kWTA problem (1). The remaining question is the global stability of the state equation, which will be proven in the next Section.

Note that the kWTA model has one design parameter (a scaling constant ϵ) only and its role is straightforward. Compared with the existing model with piecewise linear activation function in (10), the present one has three advantages: 1) The present model is independent of a through it is derived based on the quadratic programming formulation with a in its quadratic term; 2) it has the highest input resolution as $\xi \to +\infty$; and 3) it converges faster due to its higher activation gain, as shown in Section IX.

V. Global Stability

Before we analyze the stability, let us consider the kWTA problem from another perspective: the key issue is simply to find a threshold between the kth and $(k+1)$th largest element of u such that any inputs above the threshold become winners and those bellow it are losers. From the output equation in (30), it can be seen easily that the state variable y plays the role of threshold exactly. Following this lead, we can reformulate the kWTA problem as an 1-D unconstrained minimization problem:

$$\text{minimize} \quad f(y) = ky + \sum_{i=1}^{n} h(u_i - y) \tag{31}$$

where $y \in \mathbb{R}$

$$h(\rho) = \max\{0, \rho\} = \begin{cases} 0, & \rho \leq 0 \\ \rho, & \rho > 0. \end{cases} \tag{32}$$

Although the equivalence between the unconstrained minimization problem (31) in the new kWTA formulation and the existing ones is not obvious at first, it becomes apparent if we examine its following values:

$$f(y) = \begin{cases} (k-n)y + \sum_{i=1}^{n} u_i, & y < \bar{u}_n \\ \vdots & \vdots \\ \sum_{i=1}^{k+2} \bar{u}_i - 2y, & y \in (\bar{u}_{k+3}, \bar{u}_{k+2}] \\ \sum_{i=1}^{k+1} \bar{u}_i - y, & y \in (\bar{u}_{k+2}, \bar{u}_{k+1}] \\ \sum_{i=1}^{k} \bar{u}_i, & y \in (\bar{u}_{k+1}, \bar{u}_{k}] \\ y + \sum_{i=1}^{k-1} \bar{u}_i, & y \in (\bar{u}_{k}, \bar{u}_{k-1}] \\ 2y + \sum_{i=1}^{k-2} \bar{u}_i, & y \in (\bar{u}_{k-1}, \bar{u}_{k-2}] \\ \vdots & \vdots \\ ky, & y \geq \bar{u}_1 \end{cases} \tag{33}$$

where $\bar{u}_i$ is defined as the ith largest element of u ($i = 1, 2, \ldots, n$). Among the seven values (five of them are in the adjacent neighborhoods), $\sum_{i=1}^{k} \bar{u}_i$ is the smallest. Fig. 2 depicts several examples of the objective function (31) with different values of n and k, where $u \in \{1, 2, \ldots, n\}$, $n = 10$, and $k = 2, 3, 4, 5$, respectively, in each subplot. Obviously, the minimum of the objective function in each subplot falls between $n - k$ and $n - k + 1$.

Now let us revisit the dual problem (4). Reorganizing and combining the two inequality constraints results in

$$v_i \geq u_i - y \geq 0, \quad i = 1, 2, \ldots, n. \tag{34}$$

By comparing the objective functions in (4) and (31), it can be seen that $h(u_i - y)$ in (31) plays the role of v_i above. As such, a solution to (31) is the same as that to (4). It is interesting to note that the state dynamics of the present kWTA model (29) virtually computes the optimal solution to the dual problem (4) and generate the optimal solutions to the primal problem (3) in its outputs (30).

As each of the $n + 1$ terms in (31) is continuous and convex, $f(y)$ is a continuous and convex function. According to the state and output equations in (29), the upper-right dini-derivative of $f(y)$ can be derived as follows:

$$\begin{aligned} D^+ f(y) &= \sum_{i=1}^{n} D^+ h(u_i - y) + k = -\sum_{i=1}^{n} g_\infty(u_i - y) + k \\ &= -\sum_{i=1}^{n} x_i + k = -\epsilon \frac{dy}{dt}. \end{aligned}$$

Consequently

$$\frac{df(y)}{dt} = D^+ f(y) \frac{dy}{dt} = -\epsilon \left(\frac{dy}{dt}\right)^2 \begin{cases} < 0, & dy/dt \neq 0 \\ = 0, & dy/dt = 0. \end{cases} \tag{35}$$

As the objective function $f(y)$ in (31) is bounded below by $\sum_{i=1}^{k} \bar{u}_i$, radially unbounded (i.e., $f(y) \to +\infty$ as $|y| \to +\infty$) as shown in (33), and descending over time before being stable as shown in (35), according to the Lyapunov stability theory, the state of the kWTA model is globally stable.

VI. Convergence Time

In this section, the lower and upper bounds of the convergence time are derived.

Let us denote y_0 as the initial state, $\bar{y}$ as an equilibrium, and $\bar{t}$ as the time to reach the equilibrium, i.e., $y_0 = y(0)$ and $y(\bar{t}) = \bar{y}$.

Integrating both side of the state equation (29), we have

$$\epsilon(y(t) - y_0) = \int_0^t \left(\sum_{i=1}^{n} x_i(\tau) - k\right) d\tau. \tag{36}$$

In view that $\sum_{i=1}^{n} x_i - k$ is an integer with a descending absolute value, if $y(t)$ is not an equilibrium, then

$$\sum_{i=1}^{n} x_i - k \begin{cases} \geq +1, & \text{if } y_0 < \bar{y} \\ \leq -1, & \text{if } y_0 > \bar{y}. \end{cases}$$

Therefore

$$\epsilon(y(t) - y_0) = \begin{cases} \geq +t, & \text{if } y_0 < \bar{y} \\ \leq -t, & \text{if } y_0 > \bar{y}. \end{cases} \tag{37}$$

Equation (37) gives an upper bound of the convergence time

$$\bar{t} \leq \epsilon |\bar{y} - y_0|. \tag{38}$$

Now, let us derive a lower bound of the convergence time. Taking the absolute value of both sides of (36), we have

$$\epsilon |y(t) - y_0| = \left| \int_0^t \left(\sum_{i=1}^{n} x_i(\tau) - k\right) d\tau \right| \leq \int_0^t \left| \sum_{i=1}^{n} x_i(\tau) - k \right| d\tau \leq \max\{n-k, k\} t. \tag{39}$$

From (39), a lower bound of the convergence time can be derived as

$$\bar{t} \geq \frac{\epsilon |\bar{y} - y_0|}{\max\{n-k, k\}}. \tag{40}$$

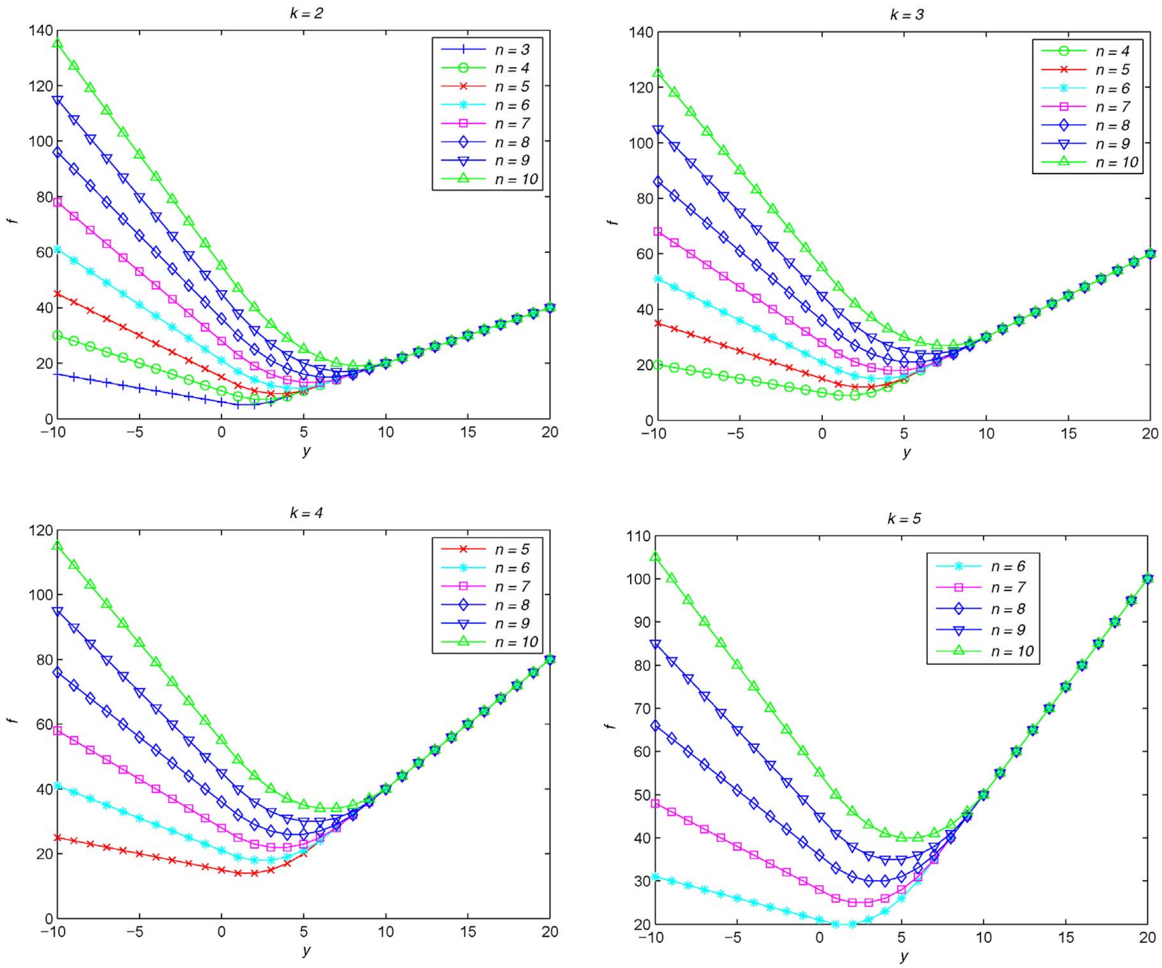

Fig. 2. Objective function in (31) with $u \in \{1, 2, \ldots, n\}$ and $k = 2, 3, 4, 5$.

From (36), we have

$$\epsilon(y(t) - y_0) = \sum_{i=1}^{n} \int_0^t x_i(\tau)d\tau - kt \begin{cases} \leq (n-k)t, & \text{if } y_0 < \bar{y} \\ \geq -kt, & \text{if } y_0 > \bar{y}. \end{cases} \tag{41}$$

Equation (41) gives the following improved lower bound of the convergence time:

$$\bar{t} \geq \begin{cases} \frac{\epsilon}{n-k}(\bar{y} - y_0), & \text{if } y_0 < \bar{y} \\ \frac{\epsilon}{k}(y_0 - \bar{y}), & \text{if } y_0 > \bar{y}. \end{cases} \tag{42}$$

Although $\bar{y}$ in the upper bound (38) and lower bounds (40) and (42) is unknown, the results still carry important information as follows.

1) The upper bound in (38) implies that the convergence time of the continuous-time kWTA model in (29) is finite.
2) The upper bound (38) and lower bounds (40) and (42) are proportional to ϵ. It means that the convergence can be expedited by using small value of ϵ.
3) In addition, the lower bounds in (40) and (42) are inversely proportional to $n-k$ or k. It implies that kWTA problems with larger values of $n-k$ and k could have faster convergence.

VII. Discrete-time Version

A discrete-time counterpart of the continuous-time kWTA model is as follows:

$$\text{state equation} \quad y(t+1) = y(t) + \beta\left(\sum_{i=1}^{n} x_i(t) - k\right) \tag{43}$$

$$\text{output equation} \quad x(t) = g_\infty(u - ey(t)) \tag{44}$$

where $\beta > 0$ is the step size.

Obviously, when β takes a small positive value, the discrete-time kWTA model is globally convergent as well. The value of the step size actually depends on the distribution and dispersion of inputs u. Since the state variable y is the threshold and supposed to fall in $(\bar{u}_{k+1}, \bar{u}_k]$ and $\sum_{i=1}^{n} x_i(t) - k$ is an integer and the difference between the number of winners and given k, a closer look tells us that the discrete-time dynamics (43) is guaranteed to be globally convergent without any overshot if the step size β is no bigger than the minimum difference of inputs $\min_{i\in\{1,2,n-1\}}(\bar{u}_i - \bar{u}_{i+1})$. In fact, the above sufficient condition for upper bound of the step size is over conservative. In the event of the minimum resolution being zero or unknown, two less conservative choices of the upper bound are the standard deviation and mean difference of inputs. In such cases, the global convergence cannot be guaranteed and sometimes an oscillation may occur at the worst case. Nevertheless, a gradual value reduction of β will always work well out any persistent oscillation and reach convergence eventually.

Similar to those in its continuous-time counterpart, the lower and upper bounds of the finite convergence iterations of the discrete-time kWTA model (43) can be derived as

$$\frac{|\bar{y}-y_0|}{\beta\max\{n-k,k\}} \le \bar{t} \le \frac{|\bar{y}-y_0|}{\beta}. \tag{45}$$

VIII. State Initialization

As shown in the previous two sections, the upper and lower bounds of convergence time of the kWTA model are proportional to $|\bar{y}-y_0|$. Although the state y of kWTA model is guaranteed to be globally convergent to $\bar{y}$ in finite time from any y_0, prior information about $\bar{y}$ is helpful to initialize y_0 closely to $\bar{y}$. Obviously, the value of $\bar{y} \in (\bar{u}_{k+1}, \bar{u}_k]$ depends on the distribution of $u_1, u_2, \ldots, u_n$, as well as the values of k and n. In this section, to further enhance the efficiency of the kWTA selection process, proper initial state initialization will be discussed, based on the statistical properties of inputs u with uniform and normal distributions. Initial states for inputs from other probability distributions may be similarly derived.

A. Uniform Distribution

Assume that $u_1, u_2, \ldots, u_n$ are random variables uniformly distributed on $[u_{\inf}, u_{\sup}]$, where $u_{\inf}$ and $u_{\sup}$ are respectively the supremum and infimum of u_i $(i = 1, 2, \ldots, n)$. By dividing $[u_{\inf}, u_{\sup}]$ into n equal intervals, the initial value y_0 should be set as follows:

$$\begin{aligned} y_0 &= u_{\inf} + \frac{n-k}{n}(u_{\sup}-u_{\inf}) = u_{\sup} - \frac{k}{n}(u_{\sup}-u_{\inf}) \\ &= \frac{n-k}{n}u_{\sup} + \frac{k}{n}u_{\inf}. \end{aligned} \tag{46}$$

Fig. 3 illustrates the linear relationship between k/n and y_0 for a uniform distribution, where $[u_{\inf}, u_{\sup}] = [-3, 3]$. In fact, (46) can also be derived by inverting $F_c(y_0) = k/n$; i.e.,

$$y_0 = F_c^{-1}(k/n) \tag{47}$$

where $F_c^{-1}(k/n)$ is the inverse of $F_c(y_0) = P(u_i > y_0)$ $(i = 1, 2, \ldots, n)$ that is the complementary cumulative distribution function of the continuous uniform distribution defined as follows:

$$F_c(y) = \begin{cases} 1, & \text{if } y < u_{\inf} \\ \frac{u_{\sup}-y}{u_{\sup}-u_{\inf}}, & \text{if } y \in [u_{\inf}, u_{\sup}] \\ 0, & \text{if } y > u_{\sup}. \end{cases} \tag{48}$$

Assume that the sample mean of u $(\bar{u} = \sum_{i=1}^{n} u_i/n)$ and the sample standard deviation $(\hat{\sigma})$ are available. If $u_{\inf}$ or $u_{\sup}$ is unknown, then they may be estimated using the method of moments in statistics with the sample mean and standard deviation as follows [35]:

$$\hat{u}_{\inf} = \bar{u} - \sqrt{3}\hat{\sigma}, \quad \hat{u}_{\sup} = \bar{u} + \sqrt{3}\hat{\sigma}. \tag{49}$$

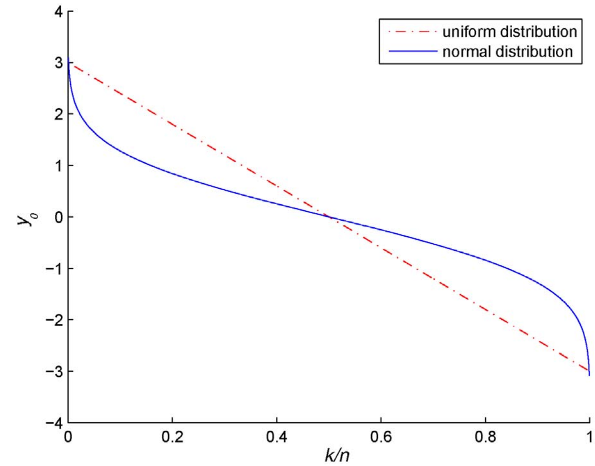

Fig. 3. Relationships between k/n and y_0 for uniform and normal distributions.

Substituting the above estimates into (46), we have the following state initialization rule:

$$y_0 = \bar{u} + \sqrt{3}(n-2k)\hat{\sigma}/n. \tag{50}$$

B. Normal Distribution

Assume that $u_1, u_2, \ldots, u_n$ are normally distributed around their mean $\bar{u}$ with the sample standard deviation $\hat{\sigma}$. Similar to the case of uniform distribution, the initial state y_0 can be estimated according to (47), that is

$$y_0 = \bar{u} + \sqrt{2}\hat{\sigma}\operatorname{erf}^{-1}\left(\frac{n-2k}{n}\right) \tag{51}$$

where $\operatorname{erf}(\cdot)$ is the Gauss error function defined as follows:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z \exp(-\rho^2)d\rho. \tag{52}$$

Several ways for numerical approximations of the inverse error function $\operatorname{erf}^{-1}(\cdot)$ can be found in [40]. Fig. 3 depicts the nonlinear relationship between k/n and y_0 in the standard normal distribution with zero-mean and unity standard deviation.

IX. Simulation Results

In this section, many simulation results of the kWTA model are presented.

A. Integer Inputs

First, let us consider kWTA problems with 100 random integer inputs $u_i \in \{1, 2, \ldots, n\}$. Fig. 4 depicts the transient state of the continuous-time kWTA model in (29) starting from 100 uniform random initial values in $[-100, 100]$, where $n = 100$, $k = n/2$, and $\epsilon = 10^{-6}$. It shows that the state variable is globally stable at an equilibrium from every initial value. Fig. 5 shows the monotone descending value of the objective function $f(y)$ in (31) with respect to time t. The simulation results also show that the convergence time is consistent with the estimated upper and lower bounds in (38) and (40) or (42).

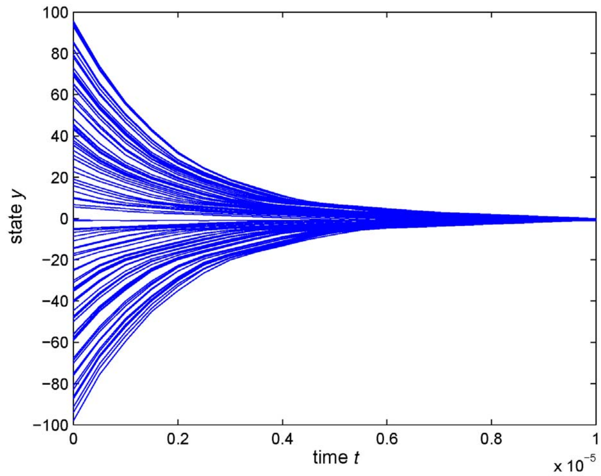

Fig. 4. Transient state of the continuous-time kWTA model (29) starting from 100 random initial values within [−100, 100] for randomized integer inputs.

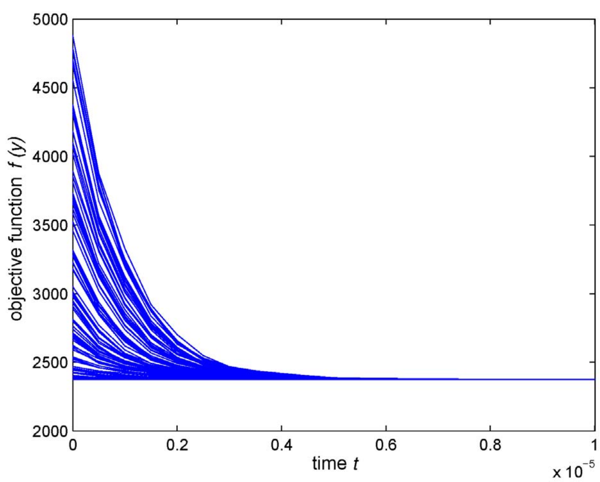

Fig. 5. Descending value of the objective function (31) minimized by using the continuous-time kWTA model (29) starting from 100 random initial values for randomized integer inputs.

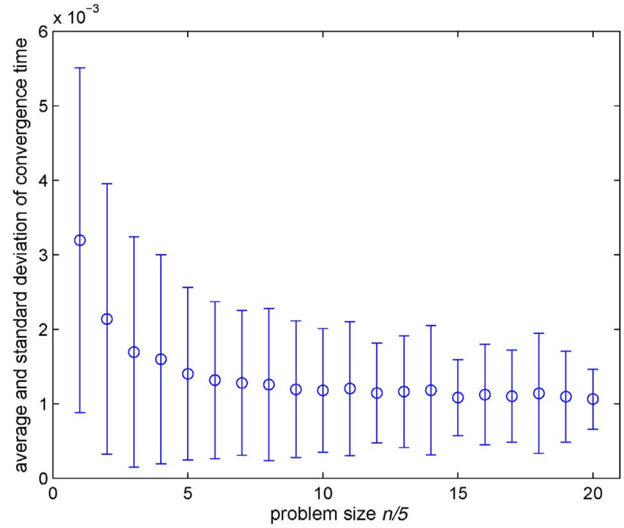

Fig. 6. Average and standard deviation of convergence time in the continuous-time kWTA model (29) with 100 randomized integer inputs u of increasing size n and random integer k between 1 and $n-1$ for randomized integer inputs.

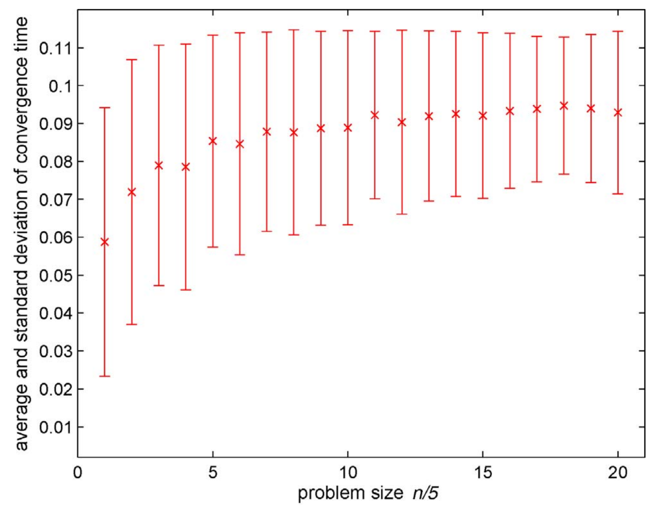

Fig. 7. Average and standard deviation of convergence time in the existing kWTA model (16) (where a = 1) with 100 randomized integer inputs u of increasing size n and random integer k between 1 and $n-1$ for randomized integer inputs.

To demonstrate the scalability of the kWTA model, Monte Carlo simulation results are shown in Figs. 6 and 7 which depict the average and standard deviation of state convergence times of the continuous-time kWTA model with the Heaviside step activation function in (29) and those with the piecewise linear activation function in (10) (a = 1) with 1000 random integer inputs $u_i \in \{1, 2, \ldots, n\}$ of increasing sizes (i.e., $n = 5, 10, \ldots, 100$), random $k \in \{1, 2, \ldots, n-1\}$, and random initial states in $[-10\,000, 10\,000]$, where the convergence times are recorded as the times elapsed before $dz < 10^{-6}$. The results show that the present kWTA model with the Heaviside step activation function converges faster than that with the piecewise linear activation function by one order on average. It also shows that the statistical mean and standard deviation of convergence time of the state variable y decrease as the problem size n increases.

Fig. 8 depicts the transient state of the discrete-time kWTA model (43) with several different values of β, where $n = 10^6$, $k = n/2$, and u is an n-vector of random integers. It shows that the convergence is reached within ten iterations for one million input data. Fig. 9 shows the descending value of the objective function $f(y)$ in (31) with respect to iteration number t.

B. Low-Resolution Inputs

To demonstrate the high-precision performance of the kWTA network, let us now consider a kWTA problem with inputs of tiny difference $u_i = (i - n/2)\delta$ for $i = 1, 2, \ldots, n$, where δ is a small number. Fig. 10 depicts the transient state of the continuous-time kWTA model (29) starting from 100 uniform random initial values in [−100, 100], where $n = 1000$, $k = n/2, \delta = 10^{-6}$, and $\epsilon = 10^{-6}$. Fig. 11 shows the descending value of the objective function $f(y)$ in (31) with respect to time t. According to (38) and (40) or (42), $2 \times 10^{-9}|\bar{y} - y_0| \leq \bar{t} \leq 10^{-6}|\bar{y} - y_0|$. In view of

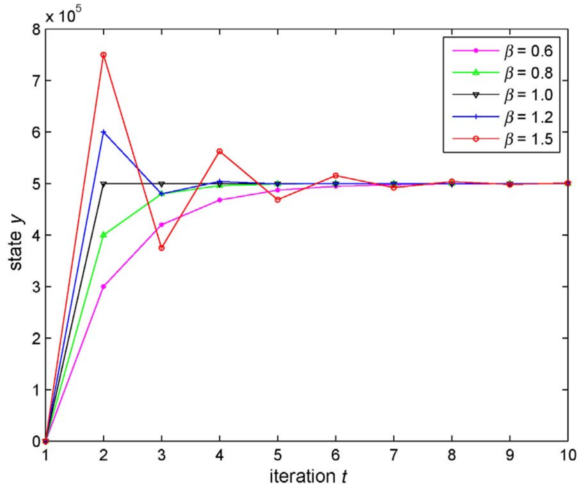

Fig. 8. Transient state of the discrete-time kWTA model (43) starting from the origin with several different values of β for randomized integer inputs.

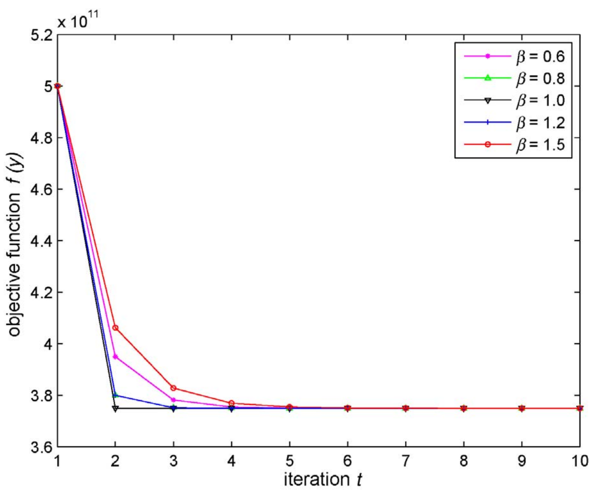

Fig. 9. Descending value of the objective function (31) minimized by using the discrete-time kWTA model (43) for randomized integer inputs.

$0 \leq |\bar{y} - y_0| \leq 100$, at the worst case, $2 \times 10^{-7} \leq \bar{t} \leq 10^{-4}$, as shown in Fig. 10.

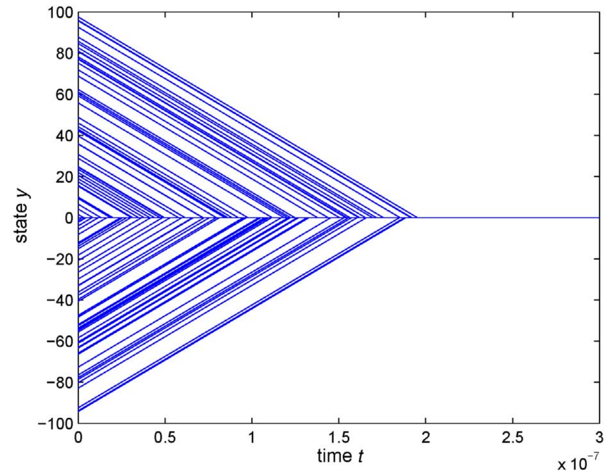

Fig. 10. Transient state of the continuous-time kWTA model (29) starting from the 100 random initial values for low-resolution inputs.

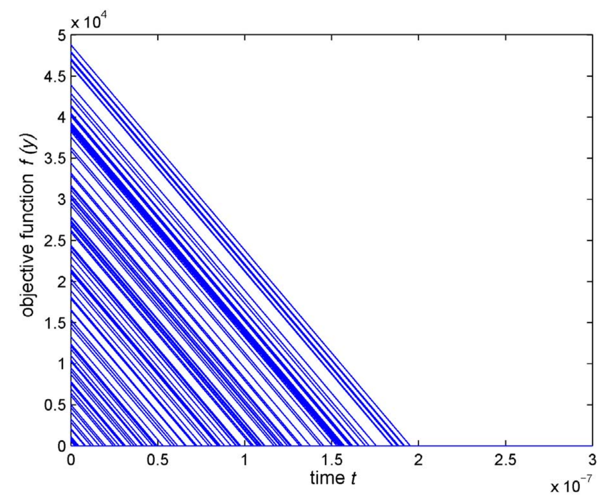

Fig. 11. Descending value of the objective function (31) minimized by using the continuous-time kWTA model (29) starting from 100 random initial values for low-resolution inputs.

C. Random Inputs

Now let us consider kWTA with random inputs having uniform and normal distributions. Figs. 12 and 13 depict the transient states of the continuous-time kWTA model in (29) for random inputs with uniform and standard normal distributions starting from the estimated initial states according to (47) and (51), respectively, where $n = 100$, $k = 1, 2, \ldots, n-1$, and $\epsilon = 10^{-6}$.

To further demonstrate the effectiveness of the state initialization proposed in the preceding section, Monte Carlo simulation results are presented below. Let us define a normalized average error function as

$$e(y_0) = \frac{\sum_{k=1}^{n-1} \max\{\max\{y_0 - \bar{u}_k, 0\}, \max\{\bar{u}_{k+1} - y_0, 0\}\}}{(n-1)(u_{\sup} - u_{\inf})}.$$

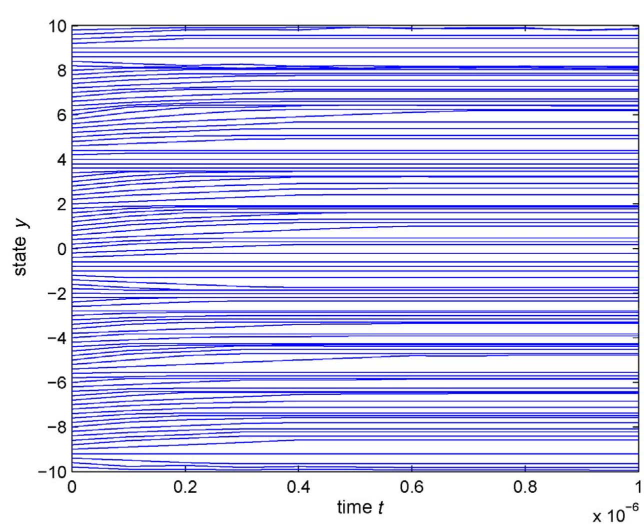

Fig. 12. Transient state of the continuous-time kWTA model (29) with 1000 random inputs u having uniform distribution, where $u_{\inf} = -10$ and $u_{\sup} = 10$.

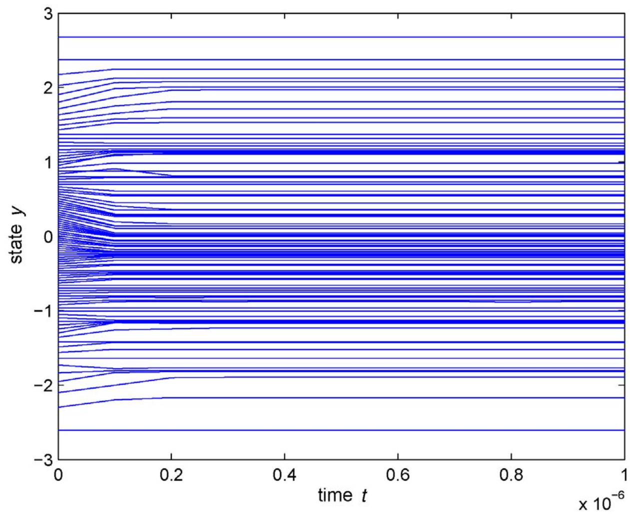

Fig. 13. Transient state of the continuous-time kWTA model (29) with 1000 random inputs u having standard normal distribution.

For random u with uniform distribution, $u_{\inf}$ and $u_{\sup}$ were randomly generated from uniform distributions in $[-1, 0]$ and $[0, 1]$, respectively, $n = 10^7$, and $k = 1, 2, \ldots, n-1$. By using the initialization approaches in (47) and (50), the values of the above normalized average error function $e(y_0)$ are 0.0061% and 0.0034%, respectively. For random u with the standard normal distribution, the normalized average error from the initialization approach in (51) with $n = 10^6$ is 0.0179%.

X. kWTA-Based Parallel Sorting

Sorting is a fundamental operation accounting for 25% of data processing time [20]. Besides many sorting algorithms, many hardware-based sorters were designed over decades (see [37], [45]). For parallel sorting, a sorting order can be represented in a permutation matrix, where "1" in row labeled with r_i and column with c_j defined as the ith item in a unsorted list and jth item in a sorted list [45]. For example

$$\begin{array}{ccccccccc} & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & \text{rank} \\ r_1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\ r_2 & 0 & 0 & 0 & 1 & 0 & 0 & 4 \\ r_3 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ r_4 & 0 & 0 & 0 & 0 & 0 & 1 & 6 \\ r_5 & 0 & 0 & 0 & 0 & 1 & 0 & 5 \\ r_6 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \end{array} \tag{53}$$

represents an unsorted list of $\{r_1, r_2, r_3, r_4, r_5, r_6\}$ and its ordered list $\{r_3, r_1, r_6, r_2, r_5, r_4\}$. A generalization of the order representation by the permutation matrix by keeping "1"s in rows is

$$\begin{array}{ccccccccc} & c'_1 & c'_2 & c'_3 & c'_4 & c'_5 & c'_6 & \text{rank} \\ r_1 & 0 & 1 & 1 & 1 & 1 & 1 & 2 \\ r_2 & 0 & 0 & 0 & 1 & 1 & 1 & 4 \\ r_3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ r_4 & 0 & 0 & 0 & 0 & 0 & 1 & 6 \\ r_5 & 0 & 0 & 0 & 0 & 1 & 1 & 5 \\ r_6 & 0 & 0 & 1 & 1 & 1 & 1 & 3 \end{array} \tag{54}$$

which represents the same sorted list above represented in (53). To decode the sorting results, a simple logic can be used to flip over the redundant "1" elements after the first "1" in each row in (54), that is

$$c_{i+1} = \overline{c_i \cdot c'_i} \cdot c'_{i+1}, \quad i = 1, \ldots, n-1$$

where "$\cdot$" is the logic AND operator and $^-$ is logic NOT operator.

If we use $n-1$ single-state kWTA models and each kWTA model computes one column of the above sorting matrix from left to right with k increasing from 1 to $n-1$, then only $n-1$ neurons will be needed with a substantial reduction of the model complexity compared with the analog sorting networks with n^2 neurons in [21] and [45]. Specifically, a WTA network with a single state variable (i.e., $k = 1$) is adopted to determined the largest element of the list. Next, a kWTA model with k being 2 computes the second item in the list without recounting the first item. As such, the whole list of n items can be sorted using $n-1$ kWTA networks without the need for computing the last item.

Let $n = 6, \{r_1, r_2, r_3, r_4, r_5, r_6\} = \{5, 3, 6, 1, 2, 4\}$. In this case, only five (5) neurons are needed by using five kWTA models (29), in contrast to 36 neurons in the analog sorting network in [45]. Fig. 14 shows the convergent behaviors of the state variables in the five kWTA models for parallel sorting, from one initial state $y(0) = 0$. The complete transient behaviors of the output variables x in the five kWTA networks are recorded at the following time instants: $t_0 = 0$, $t_1 = 2.51 \times 10^{-7}$, $t_2 = 3.16 \times 10^{-7}$, $t_3 = 3.98 \times 10^{-7}$, $t_4 = 5.01 \times 10^{-7}$, $t_5 = 6.31 \times 10^{-7}$, $t_6 = 1.00 \times 10^{-6}$, $t_7 = 1.26 \times 10^{-6}$, $t_8 = 1.59 \times 10^{-6}$, $t_9 = 2.00 \times 10^{-6}$, $t_{10} = 2.51 \times 10^{-6}$, $t_{11} = 1.00 \times 10^{-5}$

$$\begin{array}{ccccc} & & t_0 & & \\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_1 & & \\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 0&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_2 & & \\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 0&0&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1 \end{array}$$

$$\begin{array}{ccccc} & & t_3 & & \\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 0&0&0&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_4 & & \\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 0&0&0&1&1\\ 0&1&1&1&1\\ 1&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_5 & & \\ 1&1&1&1&1\\ 1&1&1&1&1\\ 1&1&1&1&1\\ 0&0&0&0&1\\ 0&0&1&1&1\\ 1&1&1&1&1 \end{array}$$

$$\begin{array}{ccccc} & & t_6 & & \\ 1&1&1&1&1\\ 0&1&1&1&1\\ 1&1&1&1&1\\ 0&0&0&0&1\\ 0&0&0&1&1\\ 1&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_7 & & \\ 1&1&1&1&1\\ 0&0&1&1&1\\ 1&1&1&1&1\\ 0&0&0&0&0\\ 0&0&0&1&1\\ 1&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_8 & & \\ 1&1&1&1&1\\ 0&0&1&1&1\\ 1&1&1&1&1\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&1&1&1&1 \end{array}$$

$$\begin{array}{ccccc} & & t_9 & & \\ 1&1&1&1&1\\ 0&0&0&1&1\\ 1&1&1&1&1\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&1&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_{10} & & \\ 0&1&1&1&1\\ 0&0&0&1&1\\ 1&1&1&1&1\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&0&1&1&1 \end{array} \qquad \begin{array}{ccccc} & & t_{11} & & \\ 0&1&1&1&1\\ 0&0&0&1&1\\ 1&1&1&1&1\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&0&1&1&1. \end{array}$$

XI. kWTA-Based Rank Order Filtering

Rank order filters are nonlinear filters with many applications including digital image processing, speech processing, coding and digital TV, etc. A rank order filter functions by working by selecting its input with a certain rank as its output.

Rank order filters entails substantial processing power to implement, which limits their real-time signal processing applications. Nevertheless, rank order filters can benefit from their parallelist realizations. Numerous approaches have been

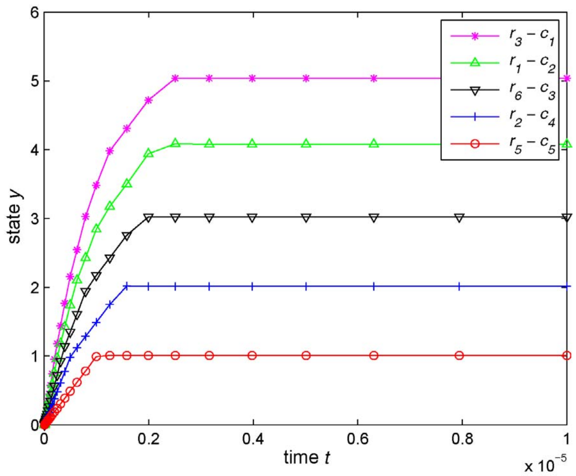

Fig. 14. Transient states of the five kWTA models (29) for parallel sorting.

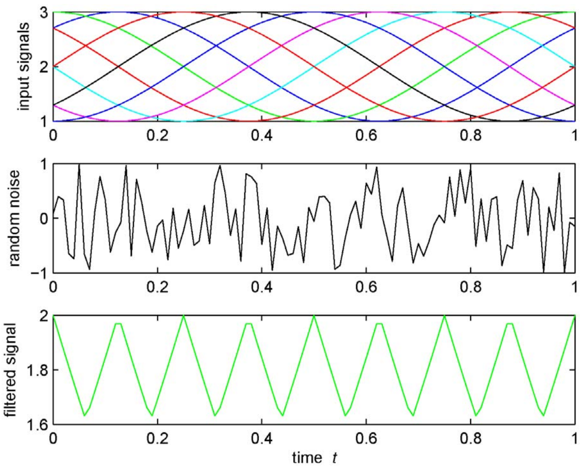

Fig. 15. Sinusoidal input signals and filtered output signal from the rank order filter based on the kWTA model (29) for rank order filtering.

developed to design rank-order filters using hardware (see [3]–[5], [28]). In particular, a rank order filter design based on two kWTA models are proposed in [5]. Specifically, a kWTA model with $k = r$ is used in parallel to another kWTA model with $k = r - 1$ to select the input with its rank order being r. Specifically, the filtered output signal is defined as

$$\tilde{u} = u^T(k\text{WTA}(u)_{k=r} - k\text{WTA}(u)_{k=r-1}) \tag{55}$$

where $k\text{WTA}(u)$ is the kWTA function as defined in (1) and can be realized by using two kWTA models defined in (29).

Consider $n - 1$ sinusoidal signals defined as $u_i = \sin(\omega t + i\phi) + \bar{u}$ $(i = 1, \ldots, n - 1)$ and a random noise u_n, where ω is the angular frequency, ϕ is the phase shift, and $\bar{u}$ is the bias. The first subplot in Fig. 15 depicts the six input sinusoidal signals and a random noise u and the last subplot depicts the filtered output signal $\tilde{u}$, where $n = 9, k = 5, \bar{u} = 2, \omega = 2\pi/10$, and $\phi = \pi/4$ (i.e., it is a median filter).

XII. Conclusion

In this paper, a kWTA model with a single state variable and a Heaviside step activation function is presented. It is shown theoretically that the state variable of the present kWTA model is globally stable and output variables are globally convergent in finite time. Lower and upper bounds of the convergence time and initial state estimates for random inputs are also derived. In addition, it is shown experimentally by Monte Carlo simulations that the convergence of the present kWTA model is substantially faster than that of the one with piecewise linear activation function. The superior performance and characteristics of the kWTA model were also demonstrated by using simulation results. In addition, two selected applications of the kWTA model for parallel sorting and rank order filtering were also delineated. The kWTA network may serve as a building block embedded in many computational models in a variety of application settings. Further investigations are directed to other applications of the kWTA model such as k-medoid clustering, k-neighborhood classification, and top-k queries, etc.

References

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. New York: Wiley, 1993.

[2] B. A. Calvert and C. Marinov, "Another k-winners-take-all analog neural network," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 829–838, Jul. 2000.

[3] C. Chakrabarti, "Sorting network based architectures for median filters," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 11, pp. 723–727, Nov. 1993.

[4] C. Chakrabarti and L.-Y. Wang, "Novel sorting network-based architecture for rank order filters," *IEEE Trans. VLSI Syst.*, vol. 2, no. 4, pp. 502–507, Dec. 1994.

[5] U. Cilingiroglu and T. L. E. Dake, "Rank-order filter design with a sampled-analog multiple-winners-take-all core," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 978–984, Aug. 2002.

[6] G. L. Dempsey and E. S. McVey, "Circuit implementation of a peak detector neural network," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 9, pp. 585–591, Sep. 1993.

[7] G. N. deSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.

[8] R. Erlanson and Y. Abu-Mostafa, "Analog neural networks as decoders," in *Advances in Neural Information Processing Systems*, vol. 1, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 585–588.

[9] Y. Fang, M. Cohen, and M. Kincaid, "Dynamics of a winner-take-all neural network," *Neural Netw.*, vol. 9, no. 7, pp. 1141–1154, 1996.

[10] L. V. Ferreira, E. Kaszkurewicz, and A. Bhaya, "Synthesis of a k-winners-take-all neural network using linear programming with bounded variables," in *Proc. Int. Joint Conf. Neural Netw.*, 2003, pp. 2360–2365.

[11] A. Fish, D. Akselrod, and O. Yadid-Pecht, "High-precision image centroid computation via an adaptive k-winner-take-all circuit in conjunction with a dynamic element matching algorithm for star tracking applications," *Analog Integr. Circuits Signal Process.*, vol. 39, no. 3, pp. 251–266, 2004.

[12] A. Fish, D. Turchin, and O. Yadid-Pecht, "An APS with 2-D winner-take-all selection employing adaptive spatial filtering and false alarm reduction," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 159–165, Jan. 2003.

[13] S. Gu and J. Wang, "A k-winners-take-all neural network based on linear programming formulation," in *Proc. Int. Joint Conf. Neural Netw.*, 2007, pp. 1022–1025.

[14] A. K. J. Hertz and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley, 1991.

[15] X. Hu and J. Wang, "Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems," *IEEE Trans. Syst., Man Cybern. B*, vol. 37, no. 5, pp. 1414–1421, Oct. 2007.

[16] X. Hu and J. Wang, "An improved dual neural network for solving a class of quadratic programming problems and its k winners-take-all application," *IEEE Trans. Neural Netw.*, vol. 19, no. 12, pp. 2022–2031, Dec. 2008.

[17] L. Itti, C. Koch, E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[18] B. J. Jain and F. Wysotzki, "A k-winner-takes-all classifier for structured data," in *Proc. 26th Ann. German Conf. AI*, LNCS 2821. 2003, pp. 342–354.

[19] B. J. Jain and F. Wysotzki, "Central clustering of attributed graphs," *Mach. Learn.*, vol. 56, nos. 1–3, pp. 169–207, 2004.

[20] D. E. Knuth, *The Art of Computer Programming, Sorting and Searching*. Reading, MA: Addison-Wesley, 1973.

[21] T. M. Kwon and M. Zervakls, "KWTA networks and their applications," *Multidimensional Syst. Signal Process.*, vol. 6, no. 4, pp. 333–346, 1995.

[22] J. Lazzaro, S. Ryckebusch, M. Mahowald, C. Mead, "Winner-take-all networks of O(N) complexity," in *Advances in Neural Information Processing Systems*, vol. 1, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 703–711.

[23] C. C. Lin and C. J. Kuo, "2-D rank-order filter by using max–min sorting network," *IEEE Trans. Circuits Systems Video Technology*, vol. 8, no. 8, pp. 941–946, Dec. 1998.

[24] Q. Liu and J. Wang, "Two *k*-winners-take-all networks with discontinuous activation functions," *Neural Netw.*, vol. 21, nos. 2–3, pp. 406–413, 2008.

[25] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 558–570, Apr. 2008.

[26] Q. Liu, J. Cao, and J. Liang, "A discrete-time recurrent neural network with one neuron for *k*-winners-take-all operation," in *Proc. 6th Int. Symp. Neural Netw.*, LNCS 5551. 2009, pp. 272–278.

[27] S. Liu and J. Wang, "A simplified dual neural network for quadratic programming with its KWTA application," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1500–1510, Nov. 2006.

[28] L. E. Lucke and K. K. Parhi, "Parallel processing architectures for rank order and stack filters," *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1178–1189, May 1994.

[29] W. Maass, "On the computational power of winner-take-all," *Neural Comput.*, vol. 12, no. 11, pp. 2519–2535, 2000.

[30] W. Maass, "Neural computation with winner-take-all as the only nonlinear operation," in *Advances in Information Processing Systems*, vol. 12, S. A. Solla, T. K. Leen, and K.-R. Mueller, Eds. Cambridge, MA: MIT Press, 2000, pp. 293–299.

[31] E. Majani, R. Erlanson, and Y. Abu-Mostafa, "On the *k*-winners-take-all network," in *Advances in Neural Information Processing Systems*, vol. 1, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 634–642.

[32] C. Marinov and B. Calvert, "Performance analysis for a *k*-winners-take-all analog neural network: Basic theory," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 766–780, Jul. 2003.

[33] C. A. Marinov and J. J. Hopfield, "Stable computational dynamics for a class of circuits with $O(N)$ interconnections capable of KWTA and rank extractions," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 5, pp. 949–959, May 2005.

[34] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," *Science*, vol. 194, pp. 283–287, Oct. 1976.

[35] NIST/SEMATECH. (2010). *e-Handbook of Statistical Methods* [Online]. Available: http://www.itl.nist.gov/div898/handbook/

[36] P. O. Pouliquen, A. G. Andreou, and K. Strohbehn, "Winner-takes-all associative memory: A hamming distance vector quantizer," *Analog Integr. Circuits Signal Process.*, vol. 13, nos. 1–2, pp. 211–222, 1997.

[37] S. Rovetta and R. Zunino, "Minimal-connectivity programmable circuit for analog sorting," *IEE Proc. Circuits, Devices Syst.*, vol. 146, no. 3, pp. 108–110, Aug. 1999.

[38] B. Sekerkiran and U. Cilingiroglu, "A CMOS *k*-winners-take-all circuit with O(n) complexity," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 1, pp. 1–5, Jan. 1999.

[39] J. P. F. Sum, C. S. Leung, P. K. S. Tam, G. H. Young, W. K. Kan, and L. W. Chan, "Analysis for a class of winner-take-all model," *IEEE Trans. Neural Netw.*, vol. 10, no. 1, pp. 64–71, Jan. 1999.

[40] N. M. Temme, "Error functions, Dawson's and Fresnel integrals," in *NIST Handbook of Mathematical Functions*, F. W. J. Olver, D. M. Lozier, R. F. Boisvert, C. W. Clark, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[41] P. V. Tymoshchuk, "A dynamic K-winners take all analog neural circuit," in *Proc. 4th Int. Conf. Perspect. Technol. Methods MEMS Design*, 2008, pp. 13–18.

[42] P. V. Tymoshchuk, "A discrete-time dynamic *k*-winners-take-all neural circuit," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3191–3202, 2009.

[43] K. Urahama and T. Nagao, "*k*-winners-take-all circuit with O(n) complexity," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 776–778, May 1995.

[44] J. Wang, "Analogue winner-take-all neural networks for determining maximum and minimum signals," *Int. J. Electron.*, vol. 77, no. 3, pp. 355–367, 1994.

[45] J. Wang, "Analysis and design of an analog sorting network," *IEEE Trans. Neural Netw.*, vol. 6, no. 4, pp. 962–971, Jul. 1995.

[46] W. Wolfe, D. Mathis, C. Anderson, J. Rothman, M. Gottler, G. Brady, R. Walker, G. Duane, and G. Algghband, "K-winner networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 310–315, Mar. 1991.

[47] J.-L. Wu and Y.-H. Tseng, "On a constant-time, low-complexity winner-take-all neural network," *IEEE Trans. Comput.*, vol. 44, no. 4, pp. 601–604, Apr. 1995.

[48] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Trans. Circuits Syst. I*, vol. 49, no. 4, pp. 447–458, Apr. 2002.

[49] Y. Xia and C. Sun, "A novel neural dynamical approach to convex quadratic program and its efficient applications," *Neural Netw.*, vol. 22, no. 10, pp. 1463–1470, 2009.

[50] J. Yen, J. Guo, and H. Chen, "A new *k*-winners-take-all neural network and its array architecture," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 901–912, Sep. 1998.

[51] A. Yuille and D. Geiger, "Winner-take-all networks," in *The Handbook of Brain Theory and Neural Networks*, 2nd ed. Cambridge, MA: MIT Press, 2003, pp. 1228–1231.

Jun Wang (S'89–M'90–SM'93–F'07) received the B.S. degree in electrical engineering and the M.S. degree in systems engineering from Dalian University of Technology, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree in systems engineering from Case Western Reserve University, Cleveland, OH, in 1991.

He is currently a Professor with the Department of Mechanical and Automation Engineering, Shatin, Hong Kong. Prior to this position, he held various academic positions at Dalian University of Technology, Case Western Reserve University, and University of North Dakota, Grand Forks. He also held various short-term visiting positions at the U.S. Air Force Armstrong Laboratory, Dayton, OH, in 1995, RIKEN Brain Science Institute, Wako Saitama, Japan, in 2001, Universite Catholique de Louvain, Ottignies, Belgium, in 2001, Chinese Academy of Sciences, Beijing, China, in 2002, and Huazhong University of Science and Technology, Hubei, China, from 2006 to 2007. He has also held a Cheung Kong Chair Professorship with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, since 2008. His current research interests include neural networks and their applications.

Dr. Wang has been an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics, Part B since 2003, and an Editorial Advisory Board Member of the *International Journal of Neural Systems* since 2006. He also served as an Associate Editor of the IEEE Transactions on Neural Networks from 1999 to 2009 and IEEE Transactions on Systems, Man, and Cybernetics, Part C from 2002 to 2005, and a Guest Editor of the special issues of *European Journal of Operational Research* in 1996, *International Journal of Neural Systems* in 2007, and *Neurocomputing* in 2008. He held organizational positions with several international conferences, such as the General Chair of the 13th International Conference on Neural Information Processing in 2006, and the 2008 IEEE World Congress on Computational Intelligence held in Hong Kong. He has served as the President of the Asia Pacific Neural Network Assembly in 2006. He is an IEEE Distinguished Lecturer for 2010 to 2012 and a recipient of the Research Excellence Award from the Chinese University of Hong Kong for 2008 to 2009 and the Shanghai Natural Science Award (first class) for 2009.