Invited Paper

Realization of desired digital spike-trains by a simple evolutionary algorithm

Toshimichi Saito¹a), Kei Yamaoka¹, and Tomoki Hamaguchi¹

¹ EE Dept., Hosei University
Koganei, Tokyo 184-8584, Japan

a) tsaito@hosei.ac.jp

Received March 1, 2017; Revised March 30, 2017; Published October 1, 2017

Abstract: This paper studies realization of desired digital spike-trains based on a simple evolutionary algorithm. First, the dynamics of spike-trains is visualized by a digital spike map. The map is defined on a set of points and is represented by a characteristic vector of integers. Second, in order to realize desired spike-trains, we present a simple evolutionary algorithm that aims at optimization of the characteristic vector for a cost function. We also present a simple method to super-stabilize desired spike-trains. Third, in order to implement the digital spike map, we introduce a digital spiking neuron consisting of two shift registers and a wiring between them. An elementary FPGA-based circuit is presented and a super-stable spike-train is demonstrated.

Key Words: dynamic digital circuits, evolutionary computation, spike-trains

1. Introduction

This paper studies realization of desired spike-trains in dynamic digital circuits based on a simple evolutionary algorithm. Spike-trains have played important roles in various systems including spiking neuron models, image processing systems, and communication systems [1–5]. For example, in image processing systems, synchronization of artificial spiking neurons is used in segmentation [4]. In communication systems, spike-position modulation and spike-pattern-division multiplexing have been used [3, 5]. Realization of desired spike-trains is important not only in fundamental study of nonlinear dynamics but also in engineering applications.

First, in order to visualize dynamics of spike-trains, we introduce a digital spike map (Dmap [6–8]). The Dmap is defined on a set of points and can be regarded as a digital version of analog one-dimensional maps represented by the logistic map [9]. The Dmap is represented by a characteristic vector of integers. Since the domain of Dmap consists of a finite number of points, the steady state must be a periodic spike-train (PST). Depending on initial conditions and parameters, the Dmap can generate a variety of PSTs.

Second, in order to realize desired PSTs, we present a simple evolutionary algorithm (SEA). Basically, the evolutionary algorithms use individuals which are potential solutions of an optimization problem for a cost function [10–13]. The individuals refer to their past history, communicate to each other, and try to find a solution. In the SEA, each individual corresponds to a characteristic vector of a Dmap. The individual gives a Dmap that can generate a PST and a desired feature of the PST is

Nonlinear Theory and Its Applications, IEICE, vol. 8, no. 4, pp. 267–278 ©IEICE 2017 DOI: 10.1587/nolta.8.267
evaluated by a cost function. An individual with the best evaluation is declared as the global best that is a candidate for preservation in the next generation. Plural candidates can exist and the number of individuals can vary flexibly. After a desired PST is obtained, the PST can be super-stabilized by a simple replacement of the characteristic vector. The super-stability means that all the initial points fall directly into the PST. It is suitable for robust and reliable operation of a PST generator.

Third, in order to implement the Dmap, we introduce a digital spiking neuron (DSN [2, 3]). The DSN is constructed by two shift registers and a wiring between them. Depending on a wiring pattern, the DSN can generate various PSTs.

The DSN is applicable to spike-pattern-division multiplexing communication [3]. The dynamics of the DSN can be described by a Dmap and the wiring pattern corresponds to the characteristic vector of the Dmap. An elementary FPGA-based circuit is presented and a super-stable spike-train is demonstrated.

Our results of the Dmap and SEA will be developed into evolutionary algorithm based design of various dynamic digital circuits. As is well known, various evolutionary algorithms have been presented and have been applied to optimization problems in various systems. For example, analog/digital filters, switching power converters, and neural networks [11–19]. The design of dynamic digital circuit is an important/effective object in the applications. It should be noted that the Dmap is related not only to the digital dynamic circuits but also to various digital dynamical systems and their applications. That is, a Dmap is derived from cellular automata [20] which are applicable to sound data description [21] and image processing [22]. A Dmap is derived from dynamic binary neural networks which are applicable to control of switching power converters [17–19]. Our results provide basic information to consider such applications.

2. Digital spike map

Figure 1 shows a Dmap and spike-train. Let \( \tau \) denoted a normalized time and let \( \tau_n \) denote the \( n \)-th spike-position. Let a clock signal with period 1 exist and let \( \theta_n \) denote the \( n \)-th spike-phase: \( \theta_n = \tau_n \mod 1 \). The Dmap is defined by

\[
\theta_{n+1} = f(\theta_n), \; \theta_n \in L_M, \; L_M = \{l_1, \cdots, l_M\}, \; l_i = i/M, \; i \in \{1, 2, \cdots, M\}.
\]

The Dmap is represented by the characteristic vector

\[
d \equiv (d_1, \cdots, d_M), \; d_i = Mf(l_i) \in \{1, 2, \cdots, M\}
\]

where \( d_i \) is the numerator of \( f(l_i) \). For example, a Dmap in Fig. 1 is represented by \( d = (2, 4, 4, 6, 8, 3, 2, 4) \). As an initial spike-position \( \tau_1 \in [0, 1) \) is given, the Dmap outputs a sequence of spike-phases \( \{\theta_n\} \) corresponding to a spike-train

\[
Y(\tau) = \begin{cases} 
1 & \text{for } \tau = \tau_n \\
0 & \text{for } \tau \neq \tau_n 
\end{cases}, \; \tau_n = \theta_n + (n - 1).
\]

The \( n \)-th spike-position appears in the \( n \)-th interval: \( \tau_n \in [n-1, n) \). The spike-train can be represented by the sequence of spike-phases \( \{\theta_n\} \). This Dmap can be implemented by the DSN in Section 5. Since the domain \( L_M \) of the Dmap consists of a finite number of points, the steady state must be a PST. The Dmap cannot generate a chaotic spike-train but a variety of PSTs. Here we give basic definitions.

**Definition 1:** A point \( p \in L_M \) is said to be a periodic point (PEP) with period \( k \) if \( p = f^k(p) \) and \( f(p) \) to \( f^k(p) \) are all different where \( f^k \) is the \( k \)-fold composition of \( f \). A PEP with period 1 is referred to as a fixed point. A sequence of the PEPs \( \{p, f(p), \cdots, f^{k-1}(p)\} \) is said to be a periodic orbit (PEO) with period \( k \). Since a PEO with period \( k \) is equivalent to a PST with period \( k \), let the term PEO mean both PEO and PST hereafter. For example, in Fig. 1, the PEO with period 3 is equivalent to the PST with period 3.

**Definition 2:** A point \( q \in L_M \) is said to be an eventually periodic point (EPP) with step \( k \) if \( q \) is not a PEP but falls into some PEP \( p \) after \( k \) steps: \( f^k(q) = p \). An EPP with step 1 is referred to as a direct eventually periodic point (DEPP): \( f(q) = p \). An EPP corresponds to an initial spike-position.
Fig. 1. Dmap of \( d = (2, 4, 6, 8, 3, 2, 4), (M = 8) \). The PEO with period 3 (red points) corresponds to the PST with period 3 where \( \theta_4 = \theta_1 \).

Fig. 2. Dmap of \( d = (4, 4, 6, 6, 3, 3), (M = 8) \). Super-stable PEO with period 3 and DEPPs (blue points).

of a transient spike-train to a PST. For example, in Fig 1, the Dmap has one PEO and all other points are EPPs falling into the PEO. The EPPs represent transient phenomena. Note that if the Dmap has only one PEO, all the other points are EPPs falling into the PEO.

Definition 3: A PEO is said to be stable if at least one EPP fall into the PEO. A PEO is said to be a super-stable periodic orbit (SSPO) if all the other points (except for the PEO) are DEPPs to the PEO. That is, all the initial points fall instantaneously into the PEO. Figure 2 shows an example of super-stable PEO with period 3: all the other 16 – 4 points are DEPPs to the PEO.
3. Simple evolutionary algorithm

In order to realize a Dmap that can generate a desired PST, we present the SEA. As preparations, we give several definitions. The SEA has individuals each of which corresponds to a characteristic vector \( \mathbf{d} \) of a Dmap. The \( i \)-th individual at generation \( g \) is denoted by

\[
\delta^i(g) = (\delta^i_1(g), \ldots, \delta^i_M(g)), \quad i \in \{1, 2, \ldots, K(g)\}, \quad g \in \{0, 1, 2, \ldots, g_{max}\}, \quad K(g) \leq K_{max}
\]

where \( K(g) \) is the number of individuals at generation \( g \). Let \( g_{max} \) be a parameter of the maximum limit of generation. The number of individuals can vary. Let \( K_{max} \) be a parameter of the maximum limit number of individuals. Each individual gives one Dmap. For simplicity, we consider realization of a PEO with period \( p \) having a desired characteristics. The PEO is evaluated by a positive definite cost function

\[
F_c(\delta^i(g)) \geq 0, \quad i \in \{1, 2, \ldots, K(g)\}.
\]  

Let \( G_b(g) \) be the minimum value of \( F_c(\delta^i) \) for all \( i \) at generation \( g \). \( G_b(g) \) is referred to as the global best at generation \( g \). The SEA is defined by the following 4 steps.

**Step 1** (Initialization): Let \( g = 0 \) and let \( K(g) = 1 \). We give an initial individual \( \delta^1(0) \) whose Dmap has a PEO with period \( p \). Let \( G_b(0) = F_c(\delta^1(0)) \).

**Step 2** (Making candidates): In the \( i \)-th individual \( \delta^i(g) \), \( i = 1 \sim K(g) \), one element is changed into other integers in \( \{1, \ldots, M\} \). It makes \( M \times (M-1) \times K(g) \) candidates of individuals for the next generation. Let \( \beta^j(g) \) be the \( j \)-th candidate where \( j = 1 \sim M(M-1)K(g) \). For example, in \( \delta \) of Dmap in Fig. 1, we obtain \( 8 \times (8-1) \) candidates.

**Step 3** (Evaluation of candidates): All the candidates are evaluated by the cost function \( F_c \) and the global best is updated:

\[
\begin{cases}
G_b(g) \leftarrow F_c(\beta^j(g)) & \text{if } F_c(\beta^j(g)) < G_b(g) \\
G_b(g) \leftarrow G_b(g) & \text{otherwise}
\end{cases}
\]

where \( j = 1 \sim M(M-1)K(g) \). If \( G_b(g) \) is the minimum value of the cost function \( F_c \) then the algorithm is terminated. If \( G_b(g) \) is not the minimum value then a candidate of the global best is declared as a new individual. If the number of the new individuals is \( K_n \) then \( K(g) \leftarrow K_n \). If \( K_n \) exceeds \( K_{max} \), then \( K_{max} \) individuals are selected randomly from the \( K_n \) individuals and \( K(g) \leftarrow K_{max} \).

**Step 4**: Let \( g \leftarrow g + 1 \), go to Step 2, and repeat until the maximum generation \( g_{max} \).

It should be noted that the number of individual evaluations in Step 3 is at most \( M(M-1)K_{max}g_{max} \). It is much smaller\(^1\) than the number of brute force \( M^M \).

If a Dmap with a desired PST is given then we super-stabilize the PST. The super-stabilization is realized by a simple replacement of a characteristic vector. Let \( \mathbf{D} \) be a characteristic vector whose subset \( \mathbf{d} \) corresponds to the desired PST with period \( p \):

\[
\mathbf{D} = (D_1, \ldots, D_M), \quad \mathbf{d} = (d_1, \ldots, d_p), \quad \mathbf{d} \subset \mathbf{D}.
\]

Let \( \mathbf{d}^c \) be a complementary set of \( \mathbf{d} \): \( \mathbf{d} \cup \mathbf{d}^c = \mathbf{D} \) and \( \mathbf{d} \cap \mathbf{d}^c = \emptyset \). The subset \( \mathbf{d}^c \) consists of all the elements of \( \mathbf{D} \) except for \( \mathbf{d} \). Replacing value of all the elements of \( \mathbf{d}^c \) with value of either element of \( \mathbf{d} \), the PST is super-stabilized. For example, applying this replacement to the PEO in Fig. 1, the PEO is super-stabilized as shown in Fig. 2.

\(^1\)If \( M \) increases and the computation becomes hard then some kind of countermeasure is required, e.g., sampling of candidates in Step 2 and an approximation of the cost function in Step 3.
4. Numerical experiments

We apply the SEA to a basic optimization problem of a Dmap. First, we define a cost function for realization of a PST with low autocorrelation. A PST with period \( p \) is denoted by

\[
\begin{aligned}
Y_p(\tau + n) &= Y_p(\tau) & \text{for } n \mod p = 0 \\
Y_p(\tau + n) &\neq Y_p(\tau) & \text{for } n \mod p \neq 0.
\end{aligned}
\]  

(6)

Autocorrelation of the PST is defined by

\[
R_{YY}(d) = \sum_{\tau=1}^{p} Y_p(\tau)Y_p(\tau + d) \quad \text{for } d \in \{0, \ldots, p\}
\]  

(7)

The cost function represents the second peak of autocorrelation

\[ F_c(\boldsymbol{d}) = \max_d R_{YY}(d) \quad \text{for } d \in \{1, \ldots, p-1\}, \quad F_c(\boldsymbol{d}) \geq 1 \]  

(8)

where \( \boldsymbol{d} \) is a characteristic vector whose Dmap has the PST with period \( p \). The minimum value is \( F_c(\boldsymbol{d}) = 1 \). We apply the SEA to this cost function with the following parameters

\[ M \in \{30, 32, 34\}, \quad K_{\text{max}} \in \{5, 10, 20\}, \quad g_{\text{max}} = 100 \]

Figure 3 shows Dmaps in example 1. As generation \( g \) increases, the Dmap evolves and the global best decreases and reaches the minimum value:

\[ G_b(0) = 7, \quad G_b(1) = 4, \quad G_b(2) = 2, \quad G_b(3) = 2, \quad G_b(4) = 1; \quad (g \in \{0, 1, 2, 3, 4\}) \]

The best individual in each step is as the following:

\[
\delta(0) = (2, 3, 4, 5, 6, 7, 8, 1, 25, 13, 6, 28, 29, 9, 4, 30, 14, 31, 30, 10, 27, 1, 20, 28, 12, 2, 4, 31, 14, 27, 8)
\]

\[
\delta(1) = (2, 3, 4, 11, 6, 7, 8, 1, 25, 13, 6, 28, 29, 9, 4, 30, 14, 31, 30, 10, 27, 1, 20, 28, 12, 5, 2, 4, 31, 14, 27, 8)
\]

\[
\delta(2) = (24, 3, 4, 11, 6, 7, 8, 1, 25, 13, 6, 28, 29, 9, 4, 30, 14, 31, 30, 10, 27, 1, 20, 28, 12, 5, 2, 4, 31, 14, 27, 8)
\]

\[
\delta(3) = (12, 3, 4, 11, 6, 7, 8, 1, 25, 13, 6, 28, 29, 9, 4, 30, 14, 31, 30, 10, 27, 1, 20, 28, 12, 5, 2, 4, 31, 14, 27, 8)
\]

\[
\delta(4) = (12, 7, 4, 11, 6, 7, 8, 1, 25, 13, 6, 18, 29, 9, 4, 30, 14, 31, 30, 10, 27, 1, 20, 28, 12, 5, 2, 4, 31, 14, 27, 8)
\]

where green elements denote the changed elements. Applying an appropriate replacement of the characteristic vector, we obtain the following characteristic vector that gives the Dmap in Fig. 3(f) where the desired PEO (\( F_c(\delta) = 1 \)) is super-stabilized.

\[
\delta = (12, 7, 1, 1, 7, 8, 1, 7, 7, 8, 18, 8, 8, 12, 12, 12, 31, 18, 18, 18, 31, 31, 31, 2, 2, 2, 2, 27, 27, 27, 27)
\]

Figure 4 shows three examples of evolution processes. In example 1, the global best reaches the optimal value at \( g = 4 \) as mentioned above. The number of individuals varies. Especially, at \( g = 3, 10 \) individuals are selected from best individuals for next evolution. The evolution process depends on random individual initialization and the random selection of individuals in Step 3. In example 2, the global best reaches the optimal value speedily at \( g = 2 \). In example 3, the global best reaches the optimal value slowly at \( g = 10 \). For \( g = 3 \sim 9 \), the upper limit number (10 = \( K_{\text{max}} \)) of individuals are selected randomly. The global best cannot change until \( g = 9 \) and can be the optimal value at \( g = 10 \). If \( K_{\text{max}} \) is set to be larger then larger number of individuals could be selected and global best would be improved faster. However, as \( K_{\text{max}} \) increases, the SEA approaches the brute force and the computation cost increases.

We have executed 50 trials where different random numbers are used in individual initialization in Step 1 and selection of individuals in Step 3. Basically, the algorithm performance is evaluated by two measures.
Fig. 3. Dmaps in evolution process example 1 of SEA. (a) $g = 0$, $G_b(0) = 7$. (b) $g = 1$, $G_b(1) = 4$. (c) $g = 2$, $G_b(2) = 2$. (d) $g = 3$, $G_b(3) = 2$. (e) $g = 4$, $G_b(4) = 1$. (f) Super-stabilized PEO with period 8 (red points) and DEPPs (blue points).
Fig. 4. Examples of evolution process. Gb: Global best. K: the number of individuals.

| M  | Kmax | #IND | #ITE | #IND × #ITE |
|----|------|------|------|-------------|
| 30 | 5    | 3.47 | 7.50 | 29.18        |
|    | 10   | 4.43 | 4.15 | 21.15        |
|    | 20   | 6.81 | 3.55 | 27.93        |
| 32 | 5    | 3.50 | 6.40 | 25.00        |
|    | 10   | 4.89 | 4.48 | 25.05        |
|    | 20   | 6.54 | 3.58 | 26.65        |
| 34 | 5    | 3.40 | 5.90 | 22.75        |
|    | 10   | 4.68 | 3.63 | 19.20        |
|    | 20   | 7.24 | 3.25 | 26.80        |

#IND: The average number of individuals per one generation.
#ITE: The average number of iterations until $F_c$ is optimized.

The computation cost can be evaluated by the number of individuals multiplied by iterations. Removing top 5 and bottom 5 trials in the computation cost, the results for the 40 trials are summarized in Table I. In the right column, #IND × #ITE means the average number of individuals multiplied by iterations. We have obtained the optimal value of $F_c$ in all the trials. The results suggest existence of a suitable value of parameter $K_{max}$ (the upper limit number of individuals). In these results, $K_{max} = 10$ seems to be suitable.

5. Digital spiking neurons

In order to implement the Dmap, we introduce the digital spiking neuron (DSN [2]). Connecting two shift registers, the DSN is constructed as shown in Fig. 5. The left and right shift resistors are referred to as P-cells and X-cells, respectively. The P-cells consist of $M$ elements and operate as a pacemaker.
Fig. 5. Digital spiking neurons. (a) \( \mathbf{a} = (7, 6, 7, 6, 5, 11, 13, 12) \) for \( \text{Dmap in Fig. 1} \). (b) \( \mathbf{a} = (5, 6, 7, 6, 11, 12, 13) \) for \( \text{Dmap in Fig. 2} \).

Only one element can be 1 with period \( M \) and all the other elements are 0:

\[
P(\tau) = (P_1(\tau), \ldots, P_M(\tau)), \quad P_i = 1 \iff \tau \mod M = i, \quad i \in \{1, 2, \ldots, M\}
\]

where \( \tau \) denotes discrete time and is represented by positive integers. Actual time step depends on clock speed in real circuits. Applying replacement \( \tau \rightarrow \tau/M \), the discrete time is identical with that in previous sections. The X-cells consists of \( N \) elements and construct a state variable vector corresponding to the membrane potential in analog neuron models. Only one element can be 1 and all the other elements are 0:

\[
X(\tau) = (X_1(\tau), \ldots, X_M(\tau))
\]

The P-cells and X-cells are connected by a wiring represented by the wiring vector

\[
\mathbf{a} = (a_1, \ldots, a_M), \quad a_i = j \iff P_i \text{ is connected to } X_j
\]

For example, the wiring vector of the DSN in Fig. 5(a) is \( \mathbf{a} = (5, 6, 7, 6, 7, 11, 12, 13) \) and that in Fig. 5(b) is \( \mathbf{a} = (7, 6, 7, 6, 5, 11, 13, 12) \). Each wiring activates either element of the X-cells. The activated elements construct a base signal where only one element can be 1 and all the other elements are 0:

\[
B(\tau) = (B_1(\tau), \ldots, B_N(\tau)), \quad B_j(\tau) = 1 \iff P_i(\tau) = 1 \text{ and } a_i = j, \quad j \in \{1, 2, \ldots, N\}.
\]

In order to define the dynamics of the DSN, we initialize X-cells such that \( X_k(1) = 1 \) for some \( k \). Let \( \tau \) denote positive discrete times. For \( \tau \geq 1 \), the dynamics is defined as the following:

If \( X_j(\tau) = 1 \) for some \( j \in \{1, 2, \ldots, N-1\} \), then \( X_{j+1}(\tau + 1) = 1 \).
If \( X_N(\tau) = 1 \) then DSN output a spike \( Y(\tau) = 1 \) and X-cell is reset to the base signal:
If \( X_N(\tau) = 1 \) and \( B_j(\tau) = 1 \) then \( X_j(\tau + 1) = 1 \) where \( j \in \{1, 2, \ldots, N-1\} \).

As illustrated in Fig. 6, the DSN generates a spike-train.

\[
Y(\tau) = \begin{cases} 
1 & \text{if } X_N(\tau) = 1 \\
0 & \text{otherwise}
\end{cases}
\]
Let $\tau_n$ denote the $n$-th spike-position and let $\theta_n = \tau_n \mod M$ denote the $n$-th spike-phase. Since the $n$-th spike determines the $(n + 1)$-th spike, a Dmap can be defined.

$$\theta_{n+1} = F(\theta_n), \; \theta_n \in \{1, 2, \cdots, M\} \tag{14}$$

Applying replacement $\tau \leftarrow \tau/M$ and $\theta \leftarrow \theta/M$, the spike-phase is identical with those in Eq. (1).

For simplicity, we set

$$N = 2M - 1, \; 0 \leq a_i - i \leq M, \; i \in \{1, 2, \cdots, M\} \tag{15}$$

In this case, one spike appears once per one period $M$ and the $n$-th spike appears in the $n$-th interval $\tau_n \in \{(n-1)M, \cdots, nM\}$.

The Dmap is determined by the wiring vector $\mathbf{a}$ as the following. Let $X_N(\tau_n) = N$ and $Y(\tau_n) = 1$ for $\tau_n = i, \; i = 1 \sim M$. At time $\tau_n + 1$, the X-cell is reset to the base $B_j$: $X_i(\tau_n + 1) = j$ provided $a_i = j$ and $P_i(\tau_n) = 1$. The X-cell reaches the threshold and output the next spike at time

$$\tau_{n+1} = \tau_n + N - a_i + 1$$

Noting $0 \leq a_i - i \leq M$, we obtain

$$d_i = \tau_{n+1} \mod M = (i + 2M - 1 - a_i + 1) \mod M = M - (a_i - i)$$

That is, if a wiring vector $\mathbf{a}$ is given, the characteristic vector of the Dmap given by

$$\mathbf{d} = (d_1, \cdots, d_M), \; d_i = M - (a_i - i), \; i = 1 \sim M \tag{16}$$

For example, the wiring vector of DSN in Fig. 5(a) gives the characteristic vector of Dmap in Fig. 1:

$$\mathbf{d} = (2, 4, 4, 6, 8, 3, 2, 4) \leftarrow \mathbf{a} = (7, 6, 7, 6, 5, 11, 13, 12)$$

The Dmap generates stable PEO and the DSN generates a stable PST as shown in Fig. 6(a). The wiring vector of DSN in Fig. 5(b) gives the characteristic vector of Dmap in Fig. 2:
This Dmap generates a super-stable PEO and the DSN generates a super-stable PST as shown in Fig. 6(b). In order to observe this super-stable PST experimentally, we have designed a digital circuit as shown in Fig. 7. Figure 8 shows Verilog simulation of the super-stable PST. This circuit has been implemented in FPGA and Fig. 9 shows a laboratory measurement. In the experiment, we have used a FPGA board (DIGILENT Inc. NEXYS3 Spartan-6) and a Logic Analyzer (DIGILENT Inc. Analog Discovery2). This elementary FPGA-based circuit is a first step to implement Dmaps in Section 4 and to implement larger scale DSNs that can generate various PSTs.

\[
d = (4, 4, 4, 6, 3, 3, 3) \quad \leftrightarrow \quad a = (5, 6, 7, 6, 7, 11, 12, 13)
\]
6. Conclusions

The SEA based realization of desired PSTs has been considered in this paper. This is a basic problem in application of evolutionary computation to design of dynamic digital circuits. The dynamics of spike-trains is visualized by the Dmap and the Dmap is represented by a characteristic vector of integers. In the SEA, individuals correspond to the characteristic vector and evaluated by a cost function. The individual(s) with the best evaluation are preserved in the next generation and the number of individuals can vary flexibly. Applying the SEA to a basic problem to optimize autocorrelation of a PST, the algorithm effectiveness is conformed. After a desired PST is obtained, the PST can be super-stabilized by simple replacement of the characteristic vector elements. The Dmap can be implemented in the DSN consisting of two shift registers connected by a wiring. An elementary FPGA-based circuit is presented and a super-stable PST is demonstrated.

In order to develop the results into more generalized design procedure and its applications, we should consider many problems including the following. First, setting of suitable parameter values (e.g., $K_{\text{max}}$) for a given cost-function. Table I suggests existence of suitable parameter values. Second, analysis of evolution process. If individuals are trapped into local optima, efficient escape scheme is required. Third, implementation of larger scale efficient circuit and its engineering applications. The elementary FPGA-based circuit is a first step to implement the effective circuits.

References

[1] E.M. Izhikevich, *Dynamical systems in neuroscience*, MIT Press, 2006.
[2] H. Torikai, A. Funew, and T. Saito, “Digital spiking neuron and its learning for approximation of various spike-trains,” *Neural Networks*, vol. 21, pp. 140–149, 2008.
[3] T. Iguchi, A. Hirata, and H. Torikai, “Theoretical and heuristic synthesis of digital spiking neurons for spike-pattern-division multiplexing,” *IEICE Trans. Fundamentals*, vol. E93-A, no. 8, pp. 1486–1496, 2010.
[4] S.R. Campbell, D. Wang, and C. Jayaprakash, “Synchrony and desynchrony in integrate-and-fire oscillators,” *Neural Comput.*, vol. 11, pp. 1595–1619, 1999.
[5] N.F. Rulkov, M.M. Sushchik, L.S. Tsimring, and A.R. Volkovskii, “Digital communication using chaotic-pulse-position modulation,” *IEEE Trans. CAS-I*, vol. 48, no. 12, pp. 1436–1444, 2001.
[6] N. Horimoto and T. Saito, “Analysis of digital spike maps based on bifurcating neurons,” *NOLTA*, vol. 3, no. 4, pp. 596–605, 2012.
[7] A. Matoba, N. Horimoto, and T. Saito, “Basic dynamics of the digital logistic map,” *IEICE Trans. Fundamentals*, vol. E96-A, no. 8, pp. 1808–1811, 2013.
[8] H. Yamada and T. Saito, “Steady-versus-transient plot for analysis of digital maps,” *IEICE Trans. Fundamentals*, vol. E99-A, no. 10, pp. 1806–1812, 2016.
[9] E. Ott, *Chaos in dynamical systems*, Cambridge, 1993.
[10] A.P. Engelbrecht, *Fundamentals of computational swarm intelligence*, Willey, 2005.
[11] W. Ertel, *Introduction to artificial intelligence*, Springer, 2009.
[12] C. Couvreur and Y. Bresler, “On the optimality of the backward greedy algorithm for the subset selection problem,” *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 3, pp. 797–808, 2000.
[13] K. Yamada and T. Saito, “Realization of desired super-stable spike-trains based on digital spiking neurons,” *Proc. IJCNN*, pp. 198–205, 2016.
[14] K.L. Shi and H. Li, “Optimized random PWM strategy based on genetic algorithms,” *Proc. IEEE/IECON*, pp. 7–11, 2003.
[15] H. Matsushita and T. Saito, “Application of particle swarm optimization to parameter search in dynamical systems,” *NOLTA*, vol. 2, no. 10, pp. 458–471, 2011.
[16] R.A. Vural, T. Yildirim, T. Kadioglu, and A. Basargan, “Performance evaluation of evolutionary algorithms for optimal filter design,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 135–147, 2012.
[17] R. Kouzuki and T. Saito, “Learning of simple dynamic binary neural networks,” *IEICE Trans. Fundamentals*, vol. E96-A, no. 8, pp. 1775–1782, 2013.
[18] J. Moriyasu and T. Saito, “A cascade system of dynamic binary neural networks and learning of periodic orbit,” *IEICE Trans. Inf. & Syst.*, vol. E98-D, no. 9, pp. 1622–1629, 2015.

[19] R. Sato and T. Saito, “Stabilization of desired periodic orbits in dynamic binary neural networks,” *Neurocomputing 248*, pp. 19–27, 2017.

[20] L.O. Chua, “A nonlinear dynamics perspective of Wolfram’s new kind of science,” I, II. World Scientific, 2005.

[21] W. Wada, J. Kuroiwa, and S. Nara, “Completely reproducible description of digital sound data with cellular automata,” *Physics Letters A 306*, pp. 110–115, 2002.

[22] P.L. Rosin, “Training cellular automata for image processing,” *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2076–2087, 2006.