Connection Sparsification and Orbit Stabilization of Dynamic Binary Neural Networks based on Multiobjective Evolutionary Algorithms

Tomoyuki Togawa and Toshimichi Saito
EE Dept., HOSEI University, Koganei, Tokyo, 184-8584 Japan
tomoyuki.togawa.8e@stu.hosei.ac.jp, tsaito@hosei.ac.jp

***Abstract*—0 A dynamic binary neural network is characterized by ternary connection parameters and can generate various binary periodic orbits. This paper studies a two-objective problem for sparsity of the connection parameters and stability of the binary periodic orbits. In order to optimize the two-objective problem, we present a simple algorithm (ALG/M) based on the multiobjective evolutionary algorithm based on decomposition. The ALG/M decomposes the two-objective problem into multiple subproblems and can optimize the problem effectively. Performing elementary numerical experiments for typical examples of binary periodic orbits, it is confirmed that the ALG/M realizes both appropriate connection sparsity and strong orbit stability. It is also confirmed that the ALG/M outperforms another algorithm based on the regularization algorithms such as the Lasso.**

***Index Terms*—binary neural networks, stability, sparsification, periodic orbits, multiobjective evolutionary algorithms.**

I. Introduction

A dynamic binary neural network (DBNN) is a recurrent-type neural network characterized by the signum activation function and ternary connection parameters [1]-[4]. Depending on the connection parameters, the DBNN can generate various binary periodic orbits. The DBNN is suitable for precise analysis and simple hardware implementation [3]-[7]. The dynamics of DBNNs can be integrated into the digital return map defined on a set of points. The digital return map can be regarded as a digital version of one-dimensional maps [8] and is useful to visualize/analyze the dynamics. On the other hand, the DBNN can be implemented into FPGA based hardware that is basic to realize engineering applications. The applications include associative memories [9], control of switching circuits [10], and central pattern generators [11].

In the analysis of the DBNN, storage of a target binary periodic orbit (TBPO) and stability of the stored TBPO are basic problem. We have obtained a parameter condition that guarantees storage of a TBPO [1]. We have also obtained experimental results that suggest appropriate connection sparsification can reinforce stability of a TBPO. The stability corresponds robust system operation and the sparsity can reduce power consumption in hardware [2] [3]. That is, optimization of the two objectives (the connection sparsity and orbit stability) can realize robust and low-power system operation in engineering applications. However, a trade-off may exist between the two objectives and optimization of the two-objectives is not easy.

In order to optimize the two-objective problem, this paper presents a simple algorithm (ALG/M) based on the multiobjective evolutionary algorithm based on decomposition (MOEA/D [12]). Although uniobjective optimization problems require the optimization of only one objective, multiobjective optimization problems require the simultaneous optimization of multiple objectives [12]-[15]. If the simultaneous optimization is achieved, we can obtain the optimal solution; however, usually the optimal solution is not feasible. We often encounter various difficulties such as the presence of conflicting objectives, where an improvement in one objective may cause a deterioration in another objective. The task is to find Pareto optimal solutions which balance trade-offs. The Pareto solutions in a decision space correspond to the Pareto front in the objective space. In order to optimize multiobjective problems, various evolutionary algorithms (incl. the MOEA/D) have been studied.

The ALG/M aims at optimization of the two-objective problem. The first objective is given by the ratio of initial points falling directory into a TBPO. It evaluates stability of the TBPO. The second objective is given by the ratio of zero elements in connection parameters. It evaluates connection sparsity of the DBNN. Referring to the MOEA/D, the ALG/M decomposes the two-objective problem effectively into multiple subproblems. Using simple mutation operators, the ALG/M evolves potential solutions and can optimize the two-objective problem. Performing elementary numerical experiments for typical examples of TBPOs, we have confirmed that the ALG/M realizes appropriate connection sparsity of the DBNN and strong stability of TBPOs. The TBPO examples include a control signal of central pattern generators. We have also confirmed that the ALG/M outperforms another algorithm (ALG/L) based on regularization algorithms such as the Lasso [16] [17]. The ALG/L tries to minimize one cost function given by weighted sum of the two objectives for connection sparsity and orbit stability.

As novelty of this paper, it should be noted that this is the first paper of MOEA/D based parameters optimization for two-objective problems in recurrent-type neural networks. Since trade-off problems are inevitable in parameters optimization in various neural networks, the ALG/M will be developed into an efficient synthesis algorithm of neural networks.

II. Dynamic Binary Neural Networks

Here we define the DBNN and TBPO [1] [2]. The dynamics is described by

$$
\begin{aligned}
x_i^{t+1} &= F\left(\sum_{j=1}^{N} w_{ij} x_j^t - T_i\right),\ i = 1 \sim N \\
F(x) &= \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}
\end{aligned} \tag{1}
$$

where $x_i^t \in \{-1, +1\} \equiv \boldsymbol{B}$ is the i-th binary state at discrete time t. The DBNN is characterized by signum activation function F and ternary connection parameters $w_{ij} \in \{-1, 0, +1\}$ as shown in Fig. 1. Threshold parameters $T_i \in \{0, \pm 1, \pm 2, \cdots\}$ are integer. For simplicity, we arrange these parameters into the connection matrix W and the threshold vector T:

$$
W \equiv \begin{pmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{NN} \end{pmatrix} \quad T \equiv \begin{pmatrix} T_1 \\ \vdots \\ T_N \end{pmatrix} \tag{2}
$$

The DBNN in Eq. (1) is abbreviated by $\boldsymbol{x}^{t+1} = \boldsymbol{F}_D(\boldsymbol{x}^t)$ where $\boldsymbol{x}^t \equiv (x_1^t, \cdots, x_N^t)$. A TBPO with period p is a periodic sequence of binary vectors

$$
\boldsymbol{z}^1, \cdots, \boldsymbol{z}^p, \cdots,\ \begin{cases} \boldsymbol{z}^t = \boldsymbol{z}^s & \text{for } |t - s| = n \cdot p \\ \boldsymbol{z}^t \neq \boldsymbol{z}^s & \text{for } |t - s| \neq n \cdot p \end{cases} \tag{3}
$$

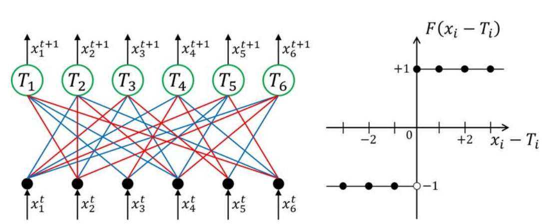

Fig. 1. Dynamic binary neural networks and signum activation function. Red and blue branches correspond to positive (w_{ij}=+1) and negative (w_{ij}=-1) connections, respectively.

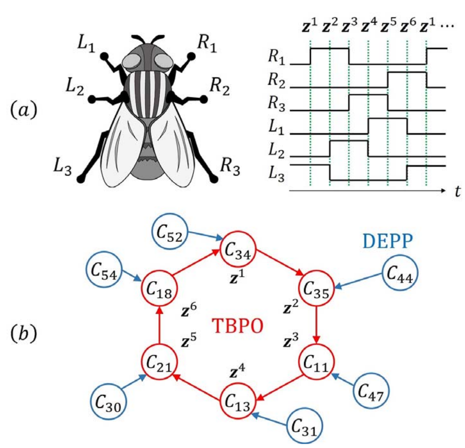

Fig. 2. TBPO1: an example of target binary periodic orbit ($\boldsymbol{z}^7 = \boldsymbol{z}^1$). (a) An insect gait pattern. (b) DEPPs to the TBPO1.

where $\boldsymbol{z}^t \equiv (z_1^t, \cdots, z_N^t)$ and $z_i^t \in \boldsymbol{B}$. For simplicity, we consider the period p in the order of N ($p = O(N)$). A TBPO with period p is said to be stored into the DBNN if $\boldsymbol{z}^{t+1} = \boldsymbol{F_D}(\boldsymbol{z}^t)$ for all $t = 1 \sim p$, ($\boldsymbol{z}^{p+1} = \boldsymbol{z}^1$). Storage of a TBPO is guaranteed if the following storage condition is satisfied:

$$
\begin{aligned}
&L(i) < T_i \leq R(i) \text{ for all } i \\
&R(i) = \min_{\tau}\left(\sum_{j=1}^{N} w_{ij} z_j^\tau\right) \text{ for } \tau \text{ such that } z_i^{\tau+1} = +1 \\
&L(i) = \max_{\tau}\left(\sum_{j=1}^{N} w_{ij} z_j^\tau\right) \text{ for } \tau \text{ such that } z_i^{\tau+1} = -1
\end{aligned} \tag{4}
$$

For storage of a TBPO, we have presented a simple correlation-based learning in [1] that can set connection parameters w_i to satisfy $L(i) \leq R(i) + 1$ for a class of TBPO and can set integer parameters T_i. As a typical example of TBPO, we introduce the 6-dimensional TBPO1 with period 6:

$$
\begin{aligned}
\boldsymbol{z}^1 &= (+1, -1, -1, -1, -1, +1) \\
\boldsymbol{z}^2 &= (+1, -1, -1, -1, +1, -1) \\
\boldsymbol{z}^3 &= (-1, -1, +1, -1, +1, -1) \\
\boldsymbol{z}^4 &= (-1, -1, +1, +1, -1, -1) \\
\boldsymbol{z}^5 &= (-1, +1, -1, +1, -1, -1) \\
\boldsymbol{z}^6 &= (-1, +1, -1, -1, -1, +1)
\end{aligned} \tag{5}
$$

This TBPO1 corresponds to an insect gait pattern as shown in Fig. 2 (a). Applying the correlation-based learning to the TBPO1, we obtain the following full binary connection matrix and threshold vector that guarantee storage of the TBPO1.

$$
W_B = \begin{pmatrix} +1 & +1 & -1 & -1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & -1 & -1 & -1 & +1 & +1 \\ -1 & +1 & -1 & +1 & -1 & +1 \end{pmatrix} \quad T_B = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \tag{6}
$$

Since $\boldsymbol{B}^N$ is equivalent to a set of 2^N points, dynamics of the DBNN is integrated into a Dmap

$$
\theta^{t+1} = f_D(\theta^t),\ \theta^t \in L_N \equiv \{C_1, \cdots, C_{2^N}\}, \tag{7}
$$

where $C_i \equiv \frac{i}{2^N}$, $i = 1 \sim 2^N$. The domain of the Dmap is $L_6 = \{C_1, \cdots, C_{2^6}\}$ where C_1 to C_{2^6} are equivalent to $(-1, -1, -1, -1, -1, -1)$ to $(+1, +1, +1, +1, +1, +1)$. Since the number of points in the domain is finite, the steady state must be a periodic orbit. Fig. 3(a) shows a Dmap of the DBNN with parameters in Eq. (6). The Dmap is useful in analysis/visualization of the dynamics.

Using the Dmap, we give basic definitions. A point $\theta_p \in L_D$ is said to be a periodic point with period p if $f_D^p(\theta_p) = \theta_p$ and $f_D(\theta_p)$ to $f_D^p(\theta_p)$ are all different where f_D^k is the k-fold composition of f_D. A sequence of periodic points $\{f_D(\theta_p), \cdots, f_D^p(\theta_p)\}$, is said to be a periodic orbit. A periodic orbit in Dmap is equivalent to a BPO in the DBNN. A point $\theta_e \in L_D$ is said to be an eventually periodic point (EPP) with step q if θ_e is not a periodic point but falls into some PEO after q steps. An eventually periodic point with step 1 is referred to as a direct eventually periodic point (DEPP). As the number of DEPPs increases, stability of the TBPO becomes strong. Fig. 2 (b) illustrates a TBPO and DEPPs to it. Fig. 3 (a) shows DBNN and Dmap for full binary connection W_B in Eq. (6).

III. Stabilization and Sparsification

Here we consider sparsification of connection matrix W and stabilization of a TBPO. In order to evaluate a DBNN, we define two objectives. The first objective evaluates stability of a TBPO with period p:

$$F_1(W) = 1 - \frac{\text{The number of DEPPs falling into a TBPO}}{2^N - p} \tag{8}$$

where $0 \le F_1(W) \le 1$. As $F_1(W)$ decreases, stability is reinforced and the DBNN becomes robust. The second objective evaluates sparsity of the connection matrix

$$F_2(W) = 1 - \frac{\text{The number of zeros in connection matrix}}{N^2 - N} \tag{9}$$

where $0 \le F_2(W) \le 1$. For convenience, we assume that a connection matrix with N non-zero elements is sparse limit. As $F_2(W)$ decreases, the connection sparsity increases. If the connection sparsity in a hardware increases, then power consumption is reduced. For the DBNN with full binary connection matrix W_B in Eq. (6), $F_1(W_B) = 52/58$ and $F_2(W_B) = 30/30$. As zero elements are inserted appropriately into the W_B, the stability of the TBPO1 can be reinforced [1]. We show an example of the sparsification:

$$W_S = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & +1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & +1 \end{pmatrix} \quad T_S = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \tag{10}$$

These parameters satisfy storage condition in Eq. (4) and storage condition of TBPO1 is guaranteed. Fig. 3 (b) shows DBNN and Dmap for the connection matrix W_S. The two objectives are given by $F_1(W_S) = 22/58$ and $F_2(W_S) = 6/30$. In this example, the connection sparsification can reinforce stability of TBPO (F_1 reduces from 52/58 to 22/58). In order to optimize both connection sparsity and TBPO stability, we consider two algorithms.

A. ALG/M: MOEA/D based algorithm

The first algorithm ALG/M is based on the MOEA/D [13]. It tries to optimize the two-objective problem:

$$\text{Minimize } F(W) = (F_1, F_2) \in S_O, \text{ subject to } W \in S_D \tag{11}$$

where $S_O = \{(F_1, F_2) | 0 \le F_1 \le 1,\ 0 \le F_2 \le 1\}$ and $S_D \equiv \{W | w_{ij} \in \{-1, 0, +1\}, i = 1 \sim N, j = 1 \sim N\}$. In the real-valued objective functions $F : S_D \to S_O$, S_D and S_O are referred to as the design space and the objective space, respectively. In this problem, we define domination relationship. Connection (matrix) parameters $W_1 \in S_D$ are said to dominate other connection parameters $W_2 \in S_D$ if either of the following is satisfied:

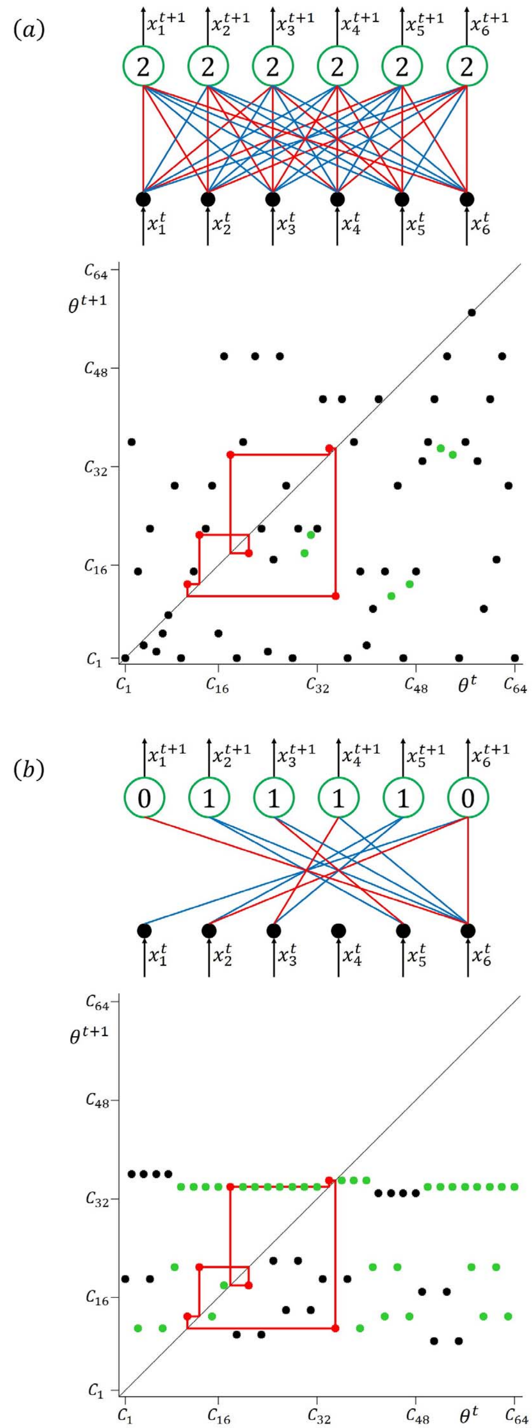

Fig. 3. DBNN and Dmaps. (a) W_B. 6 green points in Dmap denote DEPPs. ($F_1(W_B) = 52/58, F_2(W_B) = 30/30$). (b) W_S. 36 green points denote DEPPs. ($F_1(W_S) = 22/58, F_2(W_S) = 6/30$).

$$F_1(W_1) < F_1(W_2) \quad \text{and} \quad F_2(W_1) < F_2(W_2) \tag{12}$$

$$F_1(W_1) < F_1(W_2) \quad \text{and} \quad F_2(W_1) = F_2(W_2) \tag{13}$$

$$F_2(W_1) < F_2(W_2) \quad \text{and} \quad F_1(W_1) = F_1(W_2). \tag{14}$$

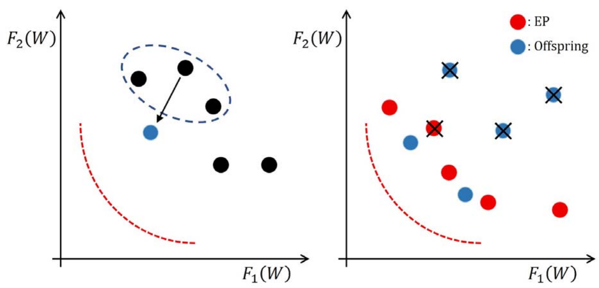

Fig. 4. Mutation (left) and EP update (right).

A set of parameters W_p is referred to as a Pareto optimal solution if W_p is not dominated by any other parameters. The set of all Pareto solutions is named a Pareto set while its image in the objective space is named Pareto front.

In the ALG/M, the two-objective problem is decomposed into multiple uni-objective subproblems by the weighted sum decomposition approach [12] [14]. The k-th subproblem is represented by the k-th adaptability function

$$\text{Minimize } F^k(W|\boldsymbol{\lambda}^k) = \lambda_1^k F_1(W) + \lambda_2^k F_2(W),\ k = 1 \sim M_1 \quad (15)$$

where the weight vectors $\boldsymbol{\lambda}^k \equiv (\lambda_1^k, \lambda_2^k)$, $k \in \{1, \cdots, M_1\}$, are given by $\lambda_1^k = (k-1)/(M_1-1)$ and $\lambda_1^k + \lambda_2^k = 1$. Each weight vector gives one uni-objective problem by the adaptability function. The ALG/M tries to minimize the adaptability functions $F^k(W|\boldsymbol{\lambda}^k)$ for W. Let W^k denote the k-th individual (potential solution) of the k-th subproblem $F^k(W|\boldsymbol{\lambda}^k)$. The W^k corresponds to the connection parameters. Correspondence between a weight vector and an individual is one-to-one. A set of M_1 individuals, $\{W^1 \cdots, W^{M_1}\}$, is referred to as a population. An external population (EP) is used to store nondominated individuals at each generation as illustrated in Fig. 4. Let λ^{l_1} and λ^{l_2} $(l_1, l_2 \in B(k))$ denote closest weight vectors to λ^k where $B(k) = \{k-1, k, k+1\}$ for $k = 2 \sim M_1 - 1$, $B(1) = \{1, 2, 3\}$ for $k = 1$, and $B(M_1) = \{M_1-2, M_1-1, M_1\}$ for $k = M_1$. Let g denote the generation of the algorithm evolution. The ALG/M is defined as the following.

Step 1 (Initialization):
Let $g = 1$. Set EP$= \emptyset$. Generate an initial population $\{W^1 \cdots, W^{M_1}\}$.
Step 2(Update): For $k = 1, \cdots, M_1$, do.

- One individual is selected randomly from neighbor set $B(k)$. Applying a mutation, a new individual is generated, where the mutation means inserting one zero element randomly into the connection matrix. If the new individual does not satisfy the storage condition in Eq. (4), then the individual is ignored and the mutation operator is applied again. If the new individual satisfies the storage condition until M_1 times application of the mutation, then the individual is declared as an offspring, otherwise the original individual is declared as the offspring and a threshold vector T is determined.
- Using the adaptability function with the elitism, the neighbor set $B(k)$ is updated.
- Remove from external population (EP) all elements dominated by the offspring. Add the offspring to EP if no elements in EP dominate offspring. Fig. 4 illustrates mutation and the EP update.

Step 3 (Termination):
Let $g \leftarrow g + 1$, go to Step 2, and repeat until the maximum generation g_{max}. After the algorithm is terminated, the EP gives the Pareto optimal solutions.

Experiment 1: We have applied the ALG/M to the TBPO1 where the algorithm parameters are

$$M_1 = 50,\ g_{max} = 30. \quad (16)$$

In Step 1, we prepare 50 copies of full binary connection matrix W_B as the fundamental individuals. At $g = 1$, the initial population is given by inserting one zero element randomly into the each individual. An evolution process is shown in Fig. 5. The ALG/M tries to optimize the two-objective problem and finds three optimal solutions by $g = g_{max}$. The three solutions give three DBNNs one of which has connection matrix W_S in Eq. (10). These DBNNs can realize appropriate connection sparsity and strong TBPO stability.

B. ALG/L: Lasso based algorithm

The second algorithm ALG/L is based on the Lasso [16] [17]. It tries to minimize cost function given by weighted sum of two terms

$$G(W|\boldsymbol{\lambda}) = \lambda_1 F_1(W) + \lambda_2 F_2(W),\ \lambda_1 + \lambda_2 = 1. \quad (17)$$

As is well known, the Lasso is one of regularization algorithms in machine learning. In the regularization, the cost function is given by weighted sum of error term and an L1 norm of connection parameters. The addition of the norm term is effective to reduce learning error by parameters sparsification (dimension reduction of the parameters space). In the ALG/L, the first and the second terms F_1 and F_2 correspond to the error term and the L1 norm term, respectively. Selection of the weight parameter λ_1 is important and we try various values of it. In the regularization algorithms, the cost function is usually minimized by a gradient method. The ALG/L uses a population of M_2 individuals and tries to minimize $G(W|\boldsymbol{\lambda})$ by mutation as the following.

Step 1 (Initialization):
Generate initial population as is the ALG/M.
Step 2(Update): For $k = 1, \cdots, M_2$, do.

- Reproduce offspring by using a mutation operator.
- Each offspring is evaluated by $G(W|\boldsymbol{\lambda})$. M_2 offspring are selected by the elitism and are preserved in the next generation.

Step 3 (Termination):
Let $g \leftarrow g + 1$, go to Step 2, and repeat until the maximum generation g_{max}.

Experiment 2: We have applied the ALG/L to the TBPO1 where the algorithm parameters are

$$\lambda_1 = \lambda_2 = 0.5,\ M_2 = 200,\ g_{max} = 30. \tag{18}$$

The ALG/L finds one optimal solution by $g = g_{max}$. An evolution process is shown in Fig. 5. As g increases, the best value of the cost function G decreases with plateau-like shape.

C. Comparison of the two algorithms

In the ALG/M, the best value of the adaptability functions $F \equiv \min_k F^k$ decreases almost monotone. For $g \geq 5$, F of ALG/M becomes lower than G of ALG/L. These results suggest that the ALG/M outperforms the ALG/L. The reasons may include effective selection ability of MOEA/D and flexible search process. In the ALG/M, individuals can move wide range in the objective space. Each individual can make various offspring by cross-interaction of neighbor sets $B(k)$. On the other hand, in ALG/L, territory of each individual is restricted by the weighted sum $\lambda_1 F_1 + \lambda_2 F_2$ in the objective space.

In order to confirm the algorithm efficiency, we have performed numerical experiments for 10 examples of TBPO with period 6. The 10 TBPOs are generated randomly and the following three algorithms are applied: ALG/M with $M_1 = 50$ individuals, ALG/L with $M_2 = 50$ individuals, and ALG/L with $M_2 = 200$ individuals. At $g = 1$, initial individuals are generated from the full binary connection by the correlation-based algorithm as is W_B in Eq. (6). The maximum generation is $g_{max} = 30$.

The ALG/M is evaluated the best value of the adaptability functions $F \equiv \min_k F^k$ for $1 \leq g \leq g_{max}$. The average values for 10 trials (10 TBPOs) are shown in Table I. The ALG/L is evaluated the best value of the cost function G for $1 \leq g \leq g_{max} = 30$. In each of 10 trials, the best, average, and worst values for all the value of λ are calculated. The average of the best, average, and worst values for 10 trials are shown in Table I. In the Table, we can see that the average solution of ALG/M smaller than the best solution of ALG/L: the ALG/M outperforms the ALG/L. However, these results are given by only 10 simple examples. For accurate evaluation of the ALG/M, more detailed consideration is necessary for various TBPOs.

TABLE I
ALGORITHMS EVALUATION FOR 10 TBPO EXAMPLES WITH PERIOD 6.

Generation		5	10	15	20	25	30
ALG/M	Avg	0.558	0.523	0.450	0.308	0.141	0.125
ALG/L	Best	0.574	0.545	0.500	0.333	0.158	0.129
$M_2 =$	Avg	0.725	0.647	0.565	0.476	0.394	0.381
50	Worst	0.833	0.671	0.610	0.610	0.598	0.597
ALG/L	Best	0.562	0.536	0.466	0.332	0.158	0.129
$M_2 =$	Avg	0.725	0.645	0.565	0.476	0.391	0.379
200	Worst	0.834	0.710	0.646	0.616	0.601	0.601

IV. CONCLUSIONS

Connection sparsity of the DBNN and stability of a TBPO have been considered in this paper. In order to optimize the two-objective problem, a MOEA/D based simple algorithm ALG/M is presented. Performing basic numerical experiments of typical TBPOs, it is confirmed that the ALG/M can find optimal solutions: suitable sparse connection of the DBNN and strong stability of a TBPO can be realized. It is also confirmed that the ALG/M outperforms the Lasso based algorithm ALG/L.

Future problems include numerical experiments of various TBPOs, application to various objectives, analysis of evolution process, and hardware implementation for engineering applications.

REFERENCES

[1] R. Sato and T. Saito, Stabilization of desired periodic orbits in dynamic binary neural networks, Neurocomputing 248, pp. 19-27, 2017.
[2] R. Sato, S. Aoki, and T. Saito, Connection sparsity versus orbit stability in dynamic binary neural networks, Proc. IJCNN, pp. 4482-4487, 2017.
[3] S. Aoki, S. Koyama, and T. Saito, Theoretical analysis of dynamic binary neural networks with simple sparse connection, Neurocomputing, 341, pp. 149-155, 2019.
[4] S. Koyama and T. Saito, Guaranteed storage and stabilization of desired binary periodic orbits in three-layer dynamic binary neural networks, Neurocomputing, in press, https://doi.org/10.1016/j.neucom.2020.01.105
[5] D. L. Gray, A. N. Michel, A training algorithm for binary feed forward neural networks, IEEE Trans. Neural Netw. 3, 2, 176-194, 1992.
[6] M. Courbariaux, Y. Bengio, J.-P. David, Binary Connect: training deep neural networks with binary weights during propagations, in NIPS, pp. 3105-3113, 2015.
[7] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio., Binarized neural networks: training neural networks with weights and activations constrained to +1 or -1, arXiv:1602.02830, 2016.
[8] E. Ott, Chaos in dynamical systems, Cambridge, 1993.
[9] J. J. Hopfield, Neural networks and physical systems with emergent collective computation abilities, Proc. of the Nat. Acad. Sci., 79, pp. 2554-2558, 1982.
[10] B. K. Bose, Neural network applications in power electronics and motor drives - an introduction and perspective, IEEE Trans. Ind. Electron. 54, 1, pp. 14-33, 2007.
[11] A. Lozano, M. Rodriguez, R. Roberto Barrio, Control strategies of 3-cell central pattern generator via global stimuli, Sci. Rep. 6:23622, 2016.
[12] Q. Zhang and L. Hui, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput., 11, 6, pp. 712-731, 2007.
[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II IEEE Trans. Evol. Comput. 6, 2, pp. 182-197, 2002.
[14] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, Localized weighted sum method for many-objective optimization, IEEE Trans. Evol. Comput., 11, 1, pp. 3-18, 2018.
[15] C. He , L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin and X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, IEEE Trans. Evol. Comput., 23, 6, pp. 949-961, 2019.
[16] R. Tibshirani, Regression shrinkage and selection via Lasso, J. Roy. Stat. Soc. B, 58, 1, pp. 267-288, 1996.
[17] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist., 32, 2, pp. 407-499, 2004.

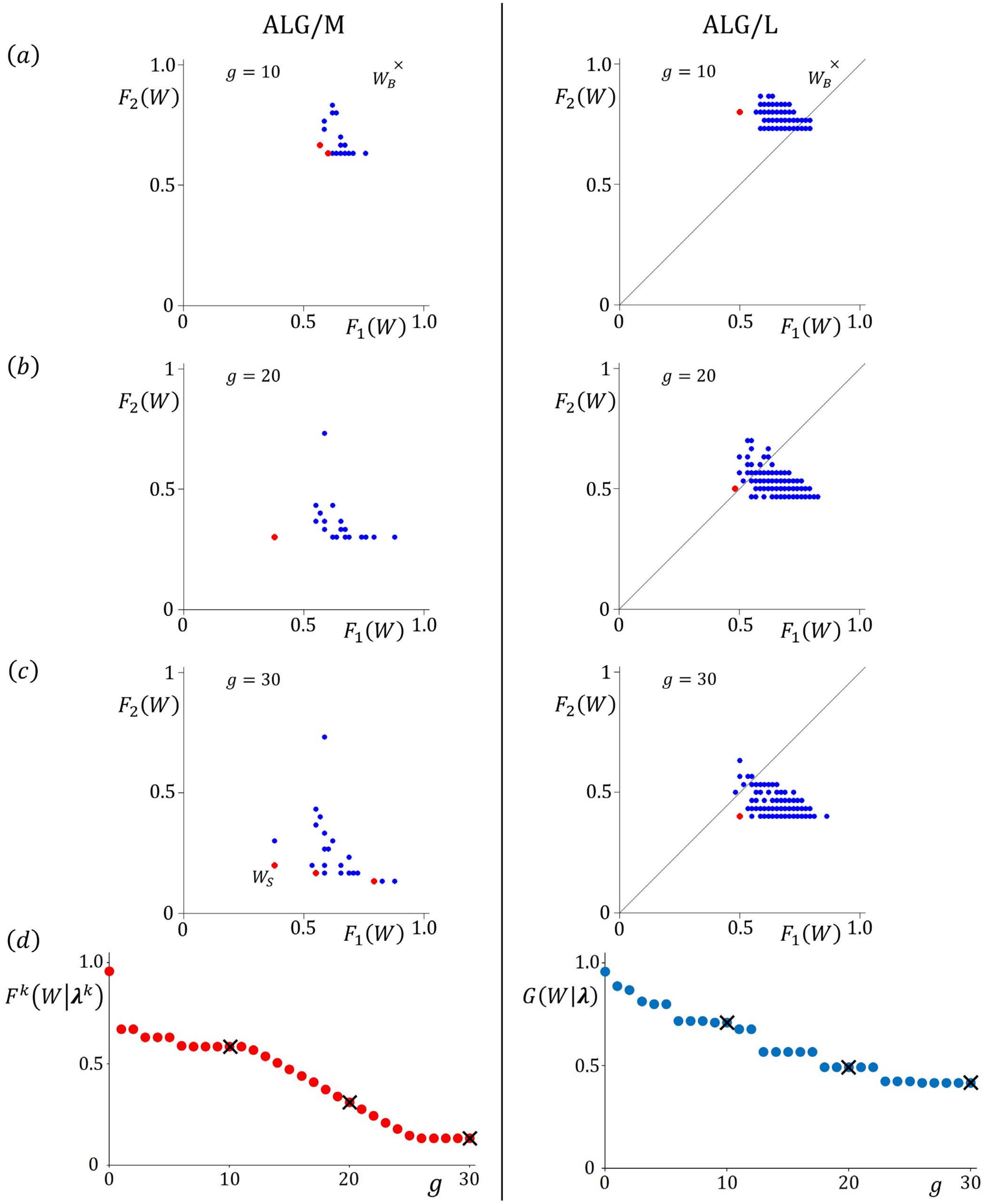

Fig. 5. Evolution process of MOEA/D based algorithm (ALG/M, left) and Lasso based algorithm (ALG/L, right). (a)-(c): Snapshots of individuals. Red individuals are optimal solutions at each generation. (d) Evolution of $\min_k F^k(W|\boldsymbol{\lambda^k})$ and $G(W|\boldsymbol{\lambda})$. At $g = g_{max}$, $F^k(W|\boldsymbol{\lambda^k}) = 0.133$ ($F_1(W) = 22/58$, $F_2(W) = 6/30$) for ALG/M and $G(W|\boldsymbol{\lambda}) = 0.450$ ($F_1(W) = 29/58$, $F_2(W) = 12/30$) for ALG/L.