A simple multiobjective evolutionary algorithm for design of digital maps

Hayate Arai¹ and Toshimichi Saito¹a)

¹ EE Dept., Hosei University, Koganei, Tokyo 184-8584, Japan
a) tsaito@hosei.ac.jp

Received November 10, 2018; Revised February 13, 2019; Published July 1, 2019

Abstract: This paper presents a simple multiobjective evolutionary algorithm and considers its application to two-objective optimization problems in design of digital maps. The algorithm is based on the MOEA/D and uses mutation operators in reproduction of potential solutions. The digital map is defined on a set of points and can generate various periodic spike-trains. The algorithm is applied to two simple problems that require optimization of autocorrelation of periodic spike-trains and distribution of inter spike intervals. The first problem has a trade-off between two objective functions and the algorithm can find an approximated Pareto front. The second problem does not have a trade-off and the algorithm can find an approximated utopia point. The algorithm performance is confirmed in elementary numerical experiments.

Key Words: multiobjective optimization, MOEA/D, digital maps, periodic orbits

1. Introduction

Multi-objective optimization problems (MOPs, [1, 2]) have been studies in various fields of science. Although uniobjective optimization problems require the optimization of only one objective function, the MOPs require the simultaneous optimization of multiple objective functions, e.g., cost and performance, accuracy and speed, or efficiency and stability in various engineering problems. If the simultaneous optimization is achieved, we can obtain the optimal solution (utopia point); however, usually the optimal solution is not feasible. In the MOPs, we encounter various difficulties. One typical problem is the presence of conflicting objectives, where an improvement in one objective may cause a deterioration in another objective. The task is to find solutions (Pareto set) which balance trade-offs. The solutions in a decision space correspond to the Pareto front in the objective space. In order to find the Pareto front effectively, various evolutionary algorithms have been presented [1–6]. The MOEA/D (multiobjective evolutionary algorithm based on decomposition [3]) is known as one of the most efficient algorithms. The algorithms performance has been investigated in benchmarks such as multiobjective 0-1 knapsack problems [3].

This paper considers multiobjective evolutionary algorithms in applications to design of nonlinear dynamical systems. Such applications have not been considered in previous works. In the design, it is not easy to tune the system parameters for desired operation in stability, power spectrum, efficiency, and so on. The parameters tuning usually spends enormous computation cost. In order to reduce the computation cost, the multiobjective evolutionary algorithms seem to play important roles.
general discussion is hard, this paper gives a basic discussion: a simple multiobjective evolutionary algorithm (SMOEA) is presented and is applied to two-objective optimization problems in design of digital maps (Dmaps [7–10]).

The SMOEA is a simplified version of the MOEA/D and is characterized by two points. First, the decision variable is given as an integer vector that represents a Dmap. Second, as genetic operators, only simple mutations are used. The mutation operators reproduce new individuals (potential solutions) for next generation. The MOEA/D uses various genetic operators such as crossover operators.

The Dmap is defined on a set of points and can generate various periodic spike-trains (PSTs). The PSTs are applicable to various systems including modeling of spiking neurons [11], spike-based communications [12, 13], central pattern generators [14], and time-series approximation [8]. The Dmap is suitable in FPGA-based hardware implementations and several circuit designs can be found in [15, 16].

The SMOEA is applied to two elementary problems that requires optimization of a PST and its inter-spike intervals (ISIs). In the problem 1, the first and second objective functions evaluate low autocorrelation of a PST and narrowness of ISI distribution, respectively. In this case, a trade-off exists and the SMOEA can find an approximated Pareto front. In the problem 2, the first and second objective functions evaluate low autocorrelation of a PST and spread of ISI distribution, respectively. In this case, a trade-off does not exist and the SMOEA can find an approximated utopia point. The algorithm performance is confirmed in typical Dmaps. It should be noted that this is the first paper and is the first step in applications of MOEA/D to design of nonlinear dynamical systems.

2. Simple multiobjective evolutionary algorithm

In this section, we present the SMOEA. First, the two-objective problems aim at minimizing two objective functions \( f_1 \) and \( f_2 \):

\[
\text{Minimize } f(d) = (f_1(d), f_2(d)) \in \mathbb{R}^2, \text{ subject to } d = (d_1, \cdots, d_M) \in \mathbb{Z}^M \equiv D
\]

(1)

where \( d \) is an \( M \)-dimensional integer decision variable vector. Concrete examples of \( d \) and \( f(d) \) are shown in Section 3. In the real-valued objective functions \( f : D \rightarrow \mathbb{R}^2 \), \( D \) and \( \mathbb{R}^2 \) are referred to as the decision space and the objective space, respectively. Replacing the \( M \) dimensional integer space \( \mathbb{Z}^M \equiv D \) with \( M \)-dimensional real space \( \mathbb{R}^M \), the discussions are generalized. Figure 1 illustrates the decision space and the objective space.

Second, a decision variable \( d_a \in D \) is said to dominate another decision variable \( d_b \in D \) if either of the following is satisfied:

\[
\begin{align*}
\text{If } f_1(d_a) &< f_1(d_b) \text{ and } f_2(d_a) < f_2(d_b), \\
\text{or } f_1(d_a) &> f_1(d_b) \text{ and } f_2(d_a) > f_2(d_b), \\
\text{then } d_a \text{ dominates } d_b.
\end{align*}
\]

Fig. 1. Objective space and decision space (conceptual scheme).

\footnote{It is trivial that low autocorrelation of analog signals means spread power spectrum, however, it is not trivial that low autocorrelation of PSTs means spread of ISI distributions.}

281
A solution \( d_p \) is referred to as a Pareto optimal solution if \( d_p \) is not dominated by any other potential solution. The set of all Pareto optimal solutions is named a Pareto set while its image in the objective space is named Pareto front. A solution \( d_a \) is referred to as a utopia point if the \( d_a \) gives the minimal of all the objectives. The utopia point is not a feasible point if a trade-off exists in the objective problem. A Pareto front (convex front) and utopia point are illustrated in Fig. 1.

We then present the SMOEA for two-objective problems in Eq. (1). Let \( d_i(g) \) be the \( i \)-th individual (potential solution) at the generation \( g \).

\[
d_i(g) = (d_{i1}(g), \cdots, d_{iM}(g)) \in D, \ i \in \{1, 2, \cdots, N\}
\]

where \( N \) is the number of the individuals. In order to divide the objective space, we introduce weight vectors

\[
\lambda_j = (\lambda_j^1, \lambda_j^2), \ \lambda_j^1 + \lambda_j^2 = 1, \ \lambda_j^1 = \frac{j}{H}, \ j \in \{0, \cdots, H\}.
\]

where \( H \) is a division number. The \( H + 1 \) weight vectors are used to decompose the MOP into \( H + 1 \) scalar problems. Let the number of individuals be \( N = H + 1 \). We apply the weighted sum approach [3] to the optimization problems. In this case, a scalar adaptability function is defined by

\[
h_j(d) = \lambda_j^1 f_1(d) + \lambda_j^2 f_2(d), \ j \in \{0, 1, \cdots, N - 1\}
\]

where \( h_j(d) \) is referred to as the \( j \)-th adaptability function. We prepare an external population EP for storage of nondominated potential solutions. The SMOEA is defined by the following steps.

**Step 1** (Initialization): Let \( g = 0 \) and let \( EP = \emptyset \). The initial individuals are generated randomly.

**Step 2** (Update): For \( j = 0, \cdots, N - 1 \), do

- Step 2-1 (Reproduction): Using genetic operator(s) (e.g., mutation), candidate individuals are generated.
- Step 2-2 (Evaluation): All the candidates are evaluated by the \( j \)-th adaptability function \( h_j \). The best evaluated candidate is declared as the \( j \)-th potential solution \( y_j \), is removed from candidates at generation \( g \) and is used as an individual in the next generation \( g + 1 \).
- Step 2-3 (Update of EP): Remove from EP all elements dominated by \( f(y_j) \). Add \( f(y_j) \) to EP if no elements in EP dominate \( f(y_j) \).

**Step 3** (Termination): If EP is not updated then the algorithm is terminated. Otherwise, \( g \leftarrow g + 1 \), go to Step 2, and repeat until the maximum generation \( g_{\text{max}} \).

When the SMOEA is terminated, the EP gives an approximation of Pareto front or utopia point. The SMOEA is a simplified version of the MOEA/D [3] and is characterized by two points: only simple mutations are used as genetic operators in Step 2-1, and each individual in Eq. (5) corresponds to an integer decision variable vector \( d \). In the existing MOEA/D based algorithms [3–6], the reproduction (Step 2-1) includes various genetic operators such as crossover operators and evaluation (Step 2-2) is applied to various benchmark problems in a higher-dimensional objective space.

### 3. Digital maps and cost functions

Figure 2 shows a Dmap defined by

\[
\theta_{n+1} = F(\theta_n), \ \theta_n \in \{1, 2, \cdots, M\} \equiv L_M.
\]
The Dmap is represented by the characteristic vector
\[ d = (d_1, \ldots, d_M), \quad d_k = F(k) \in \{1, 2, \ldots, M\}, \quad k \in \{1, 2, \ldots, M\}. \tag{9} \]

For example, a Dmap in Fig. 2(a) is represented by \( d = (4, 6, 5, 10, 1, 1, 3, 4, 6) \). Since the domain consists of a finite number of integer points, the steady state of the Dmap must be a periodic orbit. In this paper, this characteristic vector is used as an \( M \)-dimensional integer decision variable vector \( d \) in two-objective optimization problem of Eq. (1). Although the Dmap is related to various digital dynamical systems such as cellular automata, we consider the Dmap in relation to spike-trains. Let \( \tau \) denote a normalized time and let \( \tau_n \) denote the \( n \)-th spike-position. Let a clock signal with period \( M \) exist and let \( \theta_n = \tau_n \mod M \) denote the \( n \)-th spike-phase. As an initial spike-phase \( \theta_1 \in (1, M) \) is given, the Dmap outputs a sequence of spike-phases \( \{\theta_n\} \) that gives a spike-train
\[ S(\tau) = \begin{cases} 1 & \text{for } \tau = \tau_n, \\ 0 & \text{for } \tau \neq \tau_n. \end{cases} \quad \tau_n = \theta_n + (n - 1)M. \tag{10} \]

The \( n \)-th spike-position appears in the \( n \)-th interval: \( \tau_n \in [(n-1)M, nM). \) The spike-train can be represented by a sequence of spike-phases \( \{\theta_n\} \) governed by a Dmap. As stated earlier, since the domain the Dmap consists of a finite number of points, the steady state must be a periodic spike-train (PST). The Dmap cannot generate a chaotic spike-train but a variety of PSTs. Here we give basic definitions.

A point \( p \in L_M \) is said to be a periodic point (PEP) with period \( l \) if \( p = F_l(p) \) and \( F(p) \) to \( F_l(p) \) are all different where \( F_l \) is the \( l \)-fold composition of \( F \). A sequence of the PEPs \( \{F(p), \ldots, F_l(p)\} \) is said to be a periodic orbit (PEO) with period \( l \). A PEO with period \( l \) is equivalent to a PST with period \( l \). For example, in Fig. 2, the PEO with period 3 is equivalent to the PST with period 3.

A point \( q \in L_M \) is said to be an direct eventually periodic point (DEPP) if \( q \) is not a PEP but falls directly into the PEO: \( F(q) = p \). A PEO is said to be direct-stable if all the points (except for the PEO) are DEPPs to the PEO. Figure 2(b) shows an example of direct-stable PEO with period 3: all the other \((10 - 3)\) points are DEPPs to the PEO.

If a desired PEO (PST) is given then the PEO can be direct-stabilized by a simple replacement of a characteristic vector. Let us assume that a subset \( \varphi \) of a characteristic vector \( d \) constructs a PEO with period \( p \):
\[ \varphi = \{\varphi_1, \ldots, \varphi_p\} \subset d, \quad \varphi_2 = F(\varphi_1), \varphi_3 = F(\varphi_2), \ldots, \varphi_1 = F(\varphi_p). \tag{11} \]

In Fig. 2(a), \( \varphi = \{5, 8, 3\} \) constructs a PEO with period 3. Except for the PEO, replacing each element of \( D \) with either element of \( \varphi \), the PEP can be direct-stabilized. In Fig. 2(b), the characteristic vector consists of 5, 8 or 3: the PEO is direct-stable.
In order to synthesize desired Dmaps, we introduce three cost functions two of which are used in two-objective optimization problems. The first cost function $f_1$ measures the second peak of the normalized autocorrelation:

$$f_1(d) = \max_s R_{yy}(s) \geq 1, \quad R_{yy}(s) = \frac{1}{p^M - 1} \sum_{\tau=0}^{p^M-1} \sum_{\tau=0}^{p^M-1} S(\tau)S(\tau + s) \quad \text{for } s \in \{1, \ldots, pM-1\}$$

where $S(\tau)$ is a PST with period $pM$ corresponding to a PEO with period $p$ of a Dmap. $R_{yy}(s)$ is the autocorrelation of $S(\tau)$. Figure 3 shows a PST from a Dmap in Fig. 2 and its autocorrelation. In this example, the evaluation value is $f_1(d) = 1/(3 - 1)$.

The second cost function $f_2$ is proportional to variation of ISIs that evaluates spread of the ISI distribution.

$$f_2(d) = \frac{1}{N_v p} \sum_{i=1}^{p} (\Delta_i - \Delta_a)^2$$

where $\Delta_i$ denotes the $i$-th inter-spike interval between the $i$-th and $(i + 1)$-th spike positions (see Fig. 2(a)): $\Delta_i = (M - \theta_i) + \theta_{i+1}$. $\Delta_a$ denotes the average of $\Delta_i$. $N_v$ is a normalized factor and we set $N_v = 50p = 3200$ after trial-and-errors. Figure 3(a) shows ISIs of a PST with period 3 corresponding to the PEO with period 3 in Fig. 2. In this example, the evaluation value is $f_2(d) = \frac{1}{3 \times 3200} \{(13 - 10)^2 + (5 - 10)^2 + (12 - 10)^2\} \approx 3.96E^{-3}$.

The third cost function $f_3$ is inversely proportional to $f_2$ that evaluates narrowness of the ISI distribution.

$$f_3(d) = \frac{N_v p}{\sum_{i=1}^{p} (\Delta_i - \Delta_a)^2}$$

where $N_v$ is a normalized factor and we set $N_v = 60$ after trial-and-errors. These cost functions are basic to consider engineering applications such as spike-based communications [12] and pseudo random number generators [10]. Using digital spiking neurons, the Dmaps can be implemented on a FPGA board [16].

4. Numerical experiments

We consider the capability of SMOEA not only in search of a Pareto front when a trade-off exists but also in search of a utopia point when a trade-off does not exist. In order to investigate the capability, we consider the following two problems of the Dmaps.

$$P_1: \quad \text{Minimize } f(d) = (f_1(d), f_2(d)), \text{ subject to } d = (d_1, \ldots, d_M) \in D$$

$$P_2: \quad \text{Minimize } f(d) = (f_1(d), f_3(d)), \text{ subject to } d = (d_1, \ldots, d_M) \in D$$

If $f_1$ decreases then the second peak of autocorrelation decreases. If $f_2$ decreases then the ISI distribution becomes narrow. If $f_3$ decreases then the ISI distribution becomes wide. As shown afterward, a trade-off exits in P1 whereas does not exist in P2. In the application to P1 and P2, individuals of SMOEA correspond to the characteristic vector $d$ of Dmaps that is the decision variable vector.
Let each individual have a PEO with period $p$. If an element of an individual corresponds to a periodic point then the element is referred to as a periodic element, otherwise a non periodic element. As a genetic operator in Step 2-1, a mutation subroutine is used. First, the mutation subroutine selects $P_m\%$ of individuals randomly. In the selected individuals, non periodic elements are replaced randomly. Next, the mutation subroutine replace one periodic element for another integer. Figure 4 shows examples of the mutation for a PEO with period $p = 4$. In Fig. 4(a), the PEO with period $p = 4$ corresponds to periodic elements $\{2, 3, 4, 1\}$. If the periodic element 3 is replaced for 9 then another PEO with period 4 is obtained as shown in Fig 4(b). If the replaced element cannot make a PEO with period $p$ then the mutation subroutine replaces another element further to obtain a PEO with period $p$. In this replacement, another PEO with $p$ must be made. For example, if the periodic element 4 in Fig. 4(a) is replaced for 10 then the element cannot make a PEO with period $p = 4$. If the element 6 is further replaced for 1 then another PEO with period 4 is made as shown in Fig. 4(c). After the mutations, at most $N \times p \times (M - p) \times N \times g_{max}$ candidates of individuals are obtained. All the candidates are evaluated in Step 2-2 and $N$ candidates survive as individuals in the next generation.

In order to apply the SMOEA to $P_1$ and $P_2$, we have selected the following parameter values: $M = 128$, $p = 64$, $N = 30$, $g_{max} = 50$, $P_m = 0.5$. (17)

As a criterion to consider computation cost, we show the number of evaluations by the scalar adaptability function $h_l(d)$.

**Brute force:** $M^M = 128^{128}$

**SMOEA:** at most $N \times p \times (M - p) \times N \times g_{max} = 30 \times 64 \times 64 \times 30 \times 50$.

It goes without saying that the SMOEA is much more efficient than the brute force. Note that the computation becomes easy as $M$ and/or $p$ decrease whereas becomes impossible as $M$ increases (curse of dimensionality). The parameter values in Eq. (17) are selected after trial-and-errors.

At the initial generation $g = 0$, the SMOEA generates $N = 30$ initial individuals such that the Dmaps have the same PEO with period $p = 64$ and other points are arranged randomly. Figure 5 shows typical evolution process on the objective spaces. In the two-objective optimization problem $P_1$, the SMOEA can find an approximated Pareto front at generation $g = 45$, as the red points in Fig. 5(c). A trade-off exists between $f_1$ (low autocorrelation) and $f_2$ (narrowness of ISI distributions).

In the two-objective optimization problem $P_2$, the SMOEA can find an approximated utopia point at $g = 26$ as the red cross in Fig. 5(f). At $g = 26$, the SMOEA is terminated and the EP includes...

---

2Crossover operators usually erase the PEO with period $p$ and are not suitable for design of Dmaps.
Fig. 5. Evolution process on the objective space. Red points: elements in EP. Black points: dominated variables. P1: $f_1 = 1.000$, $f_2 = 0.0197$ at initial generation $g = 0$. (a) On-going status at $g = 10$. (b) On-going status at $g = 20$. (c) An approximated Pareto front (red points). P2: $f_1 = 1.000$, $f_3 = 0.952$ at initial generation $g = 0$. (a) On-going status at $g = 10$. (b) On-going status at $g = 20$. (f) An approximated utopia point (red cross).
two points that approximate the utopia point. The SOMEA can simultaneously optimize the two objective functions $f_1$ (low autocorrelation) and $f_3$ (spread of ISI distributions): trade-off does not exist between $f_1$ and $f_3$. Figure 6 shows typical examples of the Dmaps in the evolution process and
that after the termination. When a desired PEO is given, applying the simple replacement as shown in Fig. 2(b), the PEO can be direct-stabilized as shown in the Fig. 6(c) and (f).

5. Conclusions
The SMOEA is presented and is applied to simple two-objective optimization problems in design of Dmaps. The SMOEA divides the objective space by weighted sum approach and uses mutation operators that reproduce a periodic orbit in the Dmap. The SMOEA is applied to two optimization problems for autocorrelation of PSTs and distribution of ISIs. In the first problem, trade-off exists between two objective functions and the SMOEA can find an approximated Pareto front. In the second problem, trade-off does not exist and the SMOEA can find an approximated utopia point. Performance of the SMOEA is investigated in basic numerical experiments.

In order to develop the SMOEA into more effective algorithm, we have many problems in our future works. The problems include analysis of the performance for the algorithm parameters, analysis of the search process, analysis of the computation cost, development into higher-dimensional MOPs, and applications to various engineering systems.

References
[1] A.P. Engelbrecht, *Fundamentals of computational swarm intelligence*, Willey, 2005.
[2] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization: A short review,” *Proc. IEEE-CEC*, pp. 2424–2431, 2008.
[3] Q. Zhang and L. Hui, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
[4] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, “Localized weighted sum method for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 3–18, 2018.
[5] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, “On tchebycheff decomposition approaches for Multiobjective evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226–244, 2018.
[6] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, “Adaptively allocating search effort in challenging many-objective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 433–448, 2018.
[7] H. Torikai, H. Hamanaka, and T. Saito, “Reconfigurable spiking neuron and its pulse-coupled networks: Basic characteristics and potential applications,” *IEEE Trans. Circuits Syst. II*, vol. 53, no. 8, pp. 734–738, 2006.
[8] 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.
[9] T. Saito, K. Yamaoka, and T. Hamaguchi, “Realization of desired digital spike-trains by a simple evolutionary algorithm,” *NOLTA*, vol. 8, no. 4, pp. 267–278, 2017.
[10] H. Arai and T. Saito, “A simple evolutionary algorithm for design of desired digital spike maps,” *Prof. NOLTA*, pp. 359–362, 2018.
[11] E.M. Izhikevich, *Dynamical systems in neuroscience*, MIT Press, 2006.
[12] 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.
[13] 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.
[14] A. Lozano, M. Rodriguez, and R. Roberto Barrio, “Control strategies of 3-cell central pattern generator via global stimuli,” Sci. Rep. 6, 23622; doi: 10.1038/srep23622, 2016.
[15] K. Takeda and H. Torikai, “A novel spike-train generator suitable for QCA implementation towards UWB-IR applications,” *NOLTA*, vol. 9, no. 4, pp. 436–452, 2018.
[16] H. Uchida and T. Saito, “Multi-phase synchronization phenomena in a ring-coupled system of digital spiking neurons,” *IEICE Trans. Fundamentals*, vol. E102-A, no. 1, pp. 235–241, 2019.