2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504771

An Evolutionary Algorithm Taking Account of Epistasis among Parameters for Black-Box Discrete Optimization

Sho Shimazu
School of Computing
Tokyo Institute of Technology
Yokohama, Japan
shimadu.s@ic.dis.titech.ac.jp

Isao Ono
School of Computing
Tokyo Institute of Technology
Yokohama, Japan
isao@c.titech.ac.jp

***Abstract*—We propose an evolutionary algorithm that takes account of epistasis among parameters for black-box discrete optimization problems. The black-box discrete optimization (BB-DO) is an important problem that appears in various real-world problems such as hyper-parameter optimization of machine learning and is a difficult class of optimization problems to which optimization methods that require derivative of an objective function cannot be applied. In addition, epistasis among parameters, or the dependencies among variables, makes BB-DO problems more difficult. The bayesian optimization algorithm (BOA) has been proposed as a promising method to address epistasis in BB-DO problems. However, BOA suffers from a serious problem. The problem is that the diversity of a population is likely to be lost. Therefore, BOA requires a large population size for optimization. In order to remedy the problem of BOA, we introduce three schemes to maintain the diversity of the population into BOA. In experiments, we use two benchmark problems, a 3-deceptive function and a NK-landscape, and a structural optimization of neural networks to show the effectiveness of the proposed method. The experimental results showed that the proposed method improved the number of evaluations by 31.5% and the population size by 96.9% in a 180-dimensional 3-deceptive function and found comparable or better solutions in all NK-landscape settings compared to BOA. In the structural optimization of neural networks, the proposed method improved the number of evaluations by 21.3% and the population size by 92.3% compared to BOA. In addition, the proposed method was superior to conventional optimization methods used in this field, ASNG-NAS, regularized evolution, reinforcement learning, TPE, and random search, in terms of the evaluation value.**

***Index Terms*—Black-box discrete optimization, Epistasis among parameters, Bayesian optimization algorithm, Neural architecture search**

I. Introduction

A black-box discrete optimization (BB-DO) [1] is an important problem that appears in various real-world problems such as hyper-parameter optimization of machine learning. BB-DO is a difficult class of optimization problems to which optimization methods that require the derivative of an objective function cannot be applied because the representation of the objective function is not explicitly given.

Estimation of distribution algorithms (EDAs) [2] is a framework for BB-DO problems. In EDAs, multiple individuals are generated and evaluated on the search space, and a probabilistic model is built using the individuals chosen based on the evaluation values to generate individuals for the next generation. There have been proposed various EDAs such as the population-based incremental learning (PBIL) [3], the compact genetic algorithm (cGA) [4], the mutual-information-maximizing input clustering (MIMIC) [5], and the bayesian optimization algorithm (BOA) [6] so far, and their effectiveness has been shown in BB-DO problems.

One of the factors that make BB-DO difficult is epistasis among parameters, i.e., dependencies among variables [7]. Epistasis among parameters is a property of an objective function where there are dependencies among certain variables, and the evaluation value is affected by it. Therefore, in problems with epistasis among parameters, optimizing each variable independently may cause inefficient search or fail to find the global optimum.

The bayesian optimization algorithm (BOA) [6] has been proposed as a promising EDA to address epistasis among parameters. BOA employs Bayesian networks to represent dependencies among variables and builds a Bayesian network using superior individuals in the population. However, BOA has a problem in that the diversity of the population is likely to be lost. Therefore, BOA requires a large population size to maintain the diversity of the population and to build a Bayesian network that expresses the appropriate dependencies among variables. This could increase the number of evaluations of the objective function because unnecessary individuals are generated and evaluated, especially in high dimensional problems.

In this paper, we propose a method that addresses the problem of BOA. To do so, we introduce three schemes into BOA to maintain the diversity of the population and to build an appropriate Bayesian network with a small population size. In order to show the effectiveness of the proposed method, we compare its performance with that of BOA on two benchmark

978-1-7281-8393-0/21/$31.00 ©2021 IEEE

problems, a 3-deceptive function and a NK-landscape, and a structural optimization of neural networks. In the structural optimization of neural networks, we also compare the performance of the proposed method and that of conventional methods used in this field, ASNG-NAS, regularized evolution, reinforcement learning, TPE, and random search.

In the following, Section II explains BOA. In Section III, we introduce the proposed method. The experiment results are shown in Section IV. Section V is conclusion.

II. Bayesian Optimization Algorithm

The bayesian optimization algorithm (BOA) [6] is one of the most powerful methods in estimation of distribution algorithms (EDAs) for BB-DO problems with epistasis among parameters. EDA is a framework that builds an explicit probabilistic model using superior individuals in a population and generates individuals from the probabilistic model, instead of generating individuals by crossover in genetic algorithms (GAs) [8]. BOA builds a Bayesian network as the probabilistic model to represent the dependencies among variables.

In the following, we briefly explain Bayesian networks in Section II-A, the construction of network structures of Bayesian networks in Section II-B, the estimation of conditional probabilities of nodes in Section II-C, the generation of individuals using a Bayesian network in Section II-D, and the algorithm of BOA in Section II-E and finally point out problems of BOA in Section II-F.

A. Bayesian Networks

The Bayesian network represents dependencies between variables as a tree structure. A variable in the i-th dimension, or the i-th node, is denoted by X_i. The Bayesian network is assumed to be a directed acyclic graph and is represented as a joint probability distribution shown in Eq. (1).

$$Pr(X) = \prod_{i=1}^{D} Pr(X_i|\Pi_{X_i}), \tag{1}$$

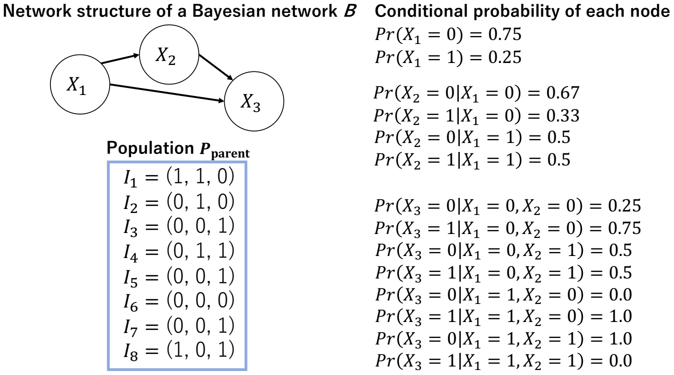
where $X = (X_1, ..., X_D)$, Π_{X_i} is the set of nodes that are the parents of node X_i, and D is the dimension of the problem.

Fig. 1 shows an example of a Bayesian network in the case of 3-bit strings. The Bayesian network consists of a network structure as shown in Fig. 1 (upper-left) and a conditional probability of each node as shown in Fig. 1 (right). The conditional probabilities are calculated by using the population P_{parent} shown in Fig. 1 (bottom-left) and the network structure.

In order to build the Bayesian network, it is necessary to construct the network structure and estimate the conditional probabilities under the constructed network structure.

B. Construction of Network Structures

In BOA, the K2 algorithm [9] is used to construct a network structure B. The K2 algorithm uses an arbitrary evaluation metric $\text{Score}(X_i, \Pi_{X_i})$ which measures the degree of the dependency between a node X_i and a set of parent nodes Π_{X_i} of X_i. The algorithm of the K2 algorithm is as follows.



Network structure of a Bayesian network B
X_1
X_2
X_3
Population P_{parent}
$I_1 = (1, 1, 0)$
$I_2 = (0, 1, 0)$
$I_3 = (0, 0, 1)$
$I_4 = (0, 1, 1)$
$I_5 = (0, 0, 1)$
$I_6 = (0, 0, 0)$
$I_7 = (0, 0, 1)$
$I_8 = (1, 0, 1)$
Conditional probability of each node
$Pr(X_1 = 0) = 0.75$
$Pr(X_1 = 1) = 0.25$
$Pr(X_2 = 0|X_1 = 0) = 0.67$
$Pr(X_2 = 1|X_1 = 0) = 0.33$
$Pr(X_2 = 0|X_1 = 1) = 0.5$
$Pr(X_2 = 1|X_1 = 1) = 0.5$
$Pr(X_3 = 0|X_1 = 0, X_2 = 0) = 0.25$
$Pr(X_3 = 1|X_1 = 0, X_2 = 0) = 0.75$
$Pr(X_3 = 0|X_1 = 0, X_2 = 1) = 0.5$
$Pr(X_3 = 1|X_1 = 0, X_2 = 1) = 0.5$
$Pr(X_3 = 0|X_1 = 1, X_2 = 0) = 0.0$
$Pr(X_3 = 1|X_1 = 1, X_2 = 0) = 1.0$
$Pr(X_3 = 0|X_1 = 1, X_2 = 1) = 1.0$
$Pr(X_3 = 1|X_1 = 1, X_2 = 1) = 0.0$

Fig. 1. A network structure of a Bayesian network B with three nodes (upper-left). A popoulation P_{parent} where each node takes a binary value (bottom-left). A conditional probability of each node which is estimated using the network structure B and the population P_{parent} (right).

(1) Initialize a set of parent nodes of each node with an empty set, i.e., $\Pi_{X_i} = \emptyset$.
(2) Assuming that X_j is a parent node of X_i, i.e., $\Pi^j_{X_i} = \Pi_{X_i} \cup \{X_j\}$, calculate $\text{Score}(X_i, \Pi^j_{X_i})$ for each node and its set of parent nodes.
(3) If there is no improvement in the scores of all tuples $(X_i, \Pi^j_{X_i})$, then the algorithm is terminated.
(4) Set $\Pi^l_{X_k}$ to Π_{X_k}, where k and l are defined as

$$k, l \leftarrow \underset{i,j}{\text{argmax}}\, \text{Score}(X_i, \Pi^j_{X_i}). \tag{2}$$

Go to (2).

In BOA, the Bayesian information criterion (BIC) metric [10] is used as the score and is calculated as follows.

$$\begin{aligned} \text{BIC}(X_i, \Pi_{X_i}) &= -H(X_i|\Pi_{X_i})\lambda - 2^{|\Pi_i|}\frac{\log_2(\lambda)}{2}, \\ H(X_i|\Pi_{X_i}) &= -\sum_{x_i, \pi_{x_i}} Pr(x_i, \pi_{x_i}) \log_2 Pr(x_i|\pi_{x_i}), \end{aligned} \tag{3}$$

where λ is a population size, $H(X_i|\Pi_{X_i})$ is a conditional entropy of X_i under Π_{X_i}, x_i is possible values of X_i, and π_{x_i} is possible values of Π_{X_i}.

The network structure B is constructed by putting an arrow from each node of Π_{X_i} to the node X_i, where Π_{X_i} is determined using the K2 algorithm.

C. Estimation of Conditional Probabilities of Nodes

The conditional probability of each node $Pr(x_i|\pi_{x_i})$ is calculated by the maximum likelihood estimation.

First, BOA estimates the probability of the root node of the network structure B constructed according to Section II-B. The probability of the root node is estimated using the frequency of the values of the root node in the population. If X_1 is a root node and there are six individuals with $X_1 = 0$ and two individuals with $X_1 = 1$ in the population, then $Pr(X_1 = 0) = 0.75$ and $Pr(X_1 = 1) = 0.25$, as shown in Fig. 1.

Next, BOA estimates the conditional probabilities of the nodes that are connected to the root node. If X_2 is a child node of the root node X_1 and there are four individuals with $(X_1 = 0, X_2 = 0)$ and two individuals with $(X_1 = 0, X_2 = 1)$ in the population, then $Pr(X_2 = 0|X_1 = 0) = 0.67$ and $Pr(X_2 = 1|X_1 = 0) = 0.33$, as shown in Fig. 1. Similarly, $Pr(X_2 = 0|X_1 = 1)$ and $Pr(X_2 = 1|X_1 = 1)$ are calculated from the frequency of $(X_1 = 1, X_2 = 0)$ and $(X_1 = 1, X_2 = 1)$ in the population.

Finally, BOA estimates the conditional probabilities of the remaining nodes in order up to the leaf nodes as with the above example.

D. Generation of Individuals Using a Bayesian Network

The value of each dimension in an individual is determined using conditional probabilities Pr estimated according to Section II-C in order from the root node to the leaf nodes of the network structure B constructed according to Section II-B.

As an example, we use a network structure B shown in Fig. 1 (upper-left) and conditional probabilities Pr shown in Fig. 1 (right) to explain how to generate an individual. First, a value of the root node X_1 is determined using $Pr(X_1)$. In this example, assume that $X_1 = 0$. Next, a value of X_2 is determined using $Pr(X_2|X_1 = 0)$. In this example, assume that $X_2 = 0$. Finally, a value of X_3 is determined using $Pr(X_3|X_1 = 0, X_2 = 0)$. In this example, assume that $X_3 = 1$. As a result, a generated individual is $X = (0, 0, 1)$.

E. Algorithm

The algorithm of BOA is shown in Algorithm 1. In line 1, BOA generates uniformly random individuals to initialize a population P whose size is λ. In line 3, BOA selects the best $r\%$ individuals from the population P to construct a population P_{parent}, where r is a user parameter that determines a trade-off between the goodness and the diversity in the population P_{parent}. In line 4, BOA constructs a network structure B using P_{parent}, BIC metric, and the K2 algorithm. In line 5, BOA estimates the conditional probability of each node in the network structure B according to Section II-C. In line 6, using the network structure B and the conditional probability $Pr(B)$, BOA generates new $\lambda_{\text{candidate}}$ individuals for the population $P_{\text{candidate}}$ according to Section II-D. In line 7, the truncation function *truncate*(P, $P_{\text{candidate}}$) is used to replace the worst $\lambda_{\text{candidate}}$ individuals in the population P with the ones in the population $P_{\text{candidate}}$.

Algorithm 1 BOA

```
1: Initialize the population $P$ whose size is $\lambda$
2: repeat
3:     $P_{\text{parent}} \leftarrow$ Select best $r\%$ individuals from $P$ based on their evaluation values
4:     $B \leftarrow$ Construct a network structure using BIC metric and the K2 algorithm described in Section II-B
5:     $Pr(B) \leftarrow$ Estimate a conditional probability of each node according to Section II-C
6:     $P_{\text{candidate}} \leftarrow$ Generate new $\lambda_{\text{candidate}}$ individuals with $B$ and $Pr(B)$ according to Section II-D
7:     $P \leftarrow \text{truncate}(P, P_{\text{candidate}})$
8: until termination conditions are met
```

F. Problems

We believe that BOA has three problems that cause the loss of the diversity of the population. If the diversity of the population is likely to be lost, the large population size is required to maintain the diversity of the population and, as a result, the number of evaluations would increase to find a global optimum, especially in high dimensional problems.

The first problem is that the diversity of the population $P_{\text{candidate}}$ is likely to be lost. As described in Section II-C, the maximum likelihood estimation is employed to estimate the conditional probability of each node with the frequency of the values of each variable in the population P_{parent} at each generation. Consequently, if a combination of values of certain variables does not exist in the population P_{parent}, the combination is never generated. For example, in Fig. 1, since the population P_{parent} does not include a combination $(X_1 = 1, X_2 = 0, X_3 = 0)$, $Pr(X_3 = 0|X_1 = 1, X_2 = 0)$ is zero. As a result, the combination $(X_1 = 1, X_2 = 0, X_3 = 0)$ will never be generated. A combination $(X_1 = 1, X_2 = 1, X_3 = 1)$ is also the same case as the above example. Therefore, if there are many duplicated individuals in the population P_{parent}, the estimated conditional probability is likely to converge to a certain value, and only similar individuals will be generated.

The second problem is that the diversity of the population P_{parent} is likely to be lost. In BOA, the best $r\%$ individuals in the population P are selected as P_{parent}, which is called the top selection. When the best $r\%$ individuals in the population P include duplicated ones, the number of unique individuals in the population P_{parent} decreases, and the diversity of P_{parent} will be reduced.

The third problem is that the diversity of the population P is likely to be lost. BOA replaces the worst $\lambda_{\text{candidate}}$ individuals in the population P with the ones in the population $P_{\text{candidate}}$, which is called the truncation replacement. The truncation replacement does not take account of similarity between individuals in the population P and in the population $P_{\text{candidate}}$. As a result, individuals with rare combination of values of variables are likely to be lost from the population P, and the diversity of P will be lost. In addition, in the truncation replacement, an individual in the population P is replaced with one in the population $P_{\text{candidate}}$ even if the individual in the population P is superior to the one in the population $P_{\text{candidate}}$.

III. Proposed Method

In this section, we propose a method that addresses the three problems described in Section II-F.

In order to address the first problem, we propose a method for estimating the conditional probability using the exponential moving average of the conditional probability estimated by the maximum likelihood estimation at each generation. This prevents the estimated conditional probability from depending on each generation's population P_{parent} only, and alleviates the rapid convergence of the conditional probability.

In order to address the second problem, we replace the top selection with the tournament selection [11]. In the tournament selection, individuals with worse evaluation values in the population P can be chosen. Therefore, the tournament selection maintains the diversity of the population P_{parent} compared to the top selection.

In order to address the third problem, we replace the truncation replacement with the restricted tournament replacement (RTR) [12]. RTR takes into account similarity between individuals in the replacement in order to maintain the diversity of the population P and the replacement occurs only when the evaluation value improves.

In the following, we explain the proposed estimation method for the conditional probabilities of nodes in Section III-A, the selection method in Section III-B, and the replacement method in Section III-C and summarize the algorithm of the proposed method in Section III-D.

A. Estimation of Conditional Probabilities of Nodes

We propose a method for estimating the conditional probability using the exponential moving average of the conditional probability estimated by the maximum likelihood estimation at each generation.

The proposed estimation method takes the population P_{parent} and the network structure B as an input. It consists of the following procedures.

(1) Estimate the temporary conditional probability Pr^t at generation t by the maximum likelihood estimation described in Section II-C.

(2) Update the conditional probability Pr according to Eq. (4).

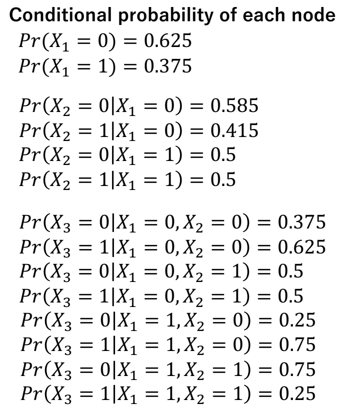
$$\begin{aligned} &Pr(X_i|\Pi_{X_i}) \\ &\quad = (1-\eta) \times Pr(X_i|\Pi_{X_i}) + \eta \times Pr^t(X_i|\Pi_{X_i}), \end{aligned} \tag{4}$$

where η is a user parameter which is called the *update rate* that adjusts the degree of utilization between Pr^t and Pr.

Procedure (1) is the same as BOA, and Procedure (2) updates the conditional probability Pr estimated in the previous generation using Pr^t estimated in generation t. When $\eta = 1$, it is identical with the estimation method of BOA.

The initial value of $Pr(X_i|\Pi_{X_i})$ is set uniformly on the search space. For example, if X_i is binary, the initial values of $Pr(X_i = 0|\Pi_{X_i})$ and $Pr(X_i = 1|\Pi_{X_i})$ are 0.5 and 0.5, respectively.

Fig. 2 shows the conditional probabilities estimated by the above procedures where a network structure B is the one shown in Fig. 1 (upper-left), a population P_{parent} is the one shown in Fig. 1 (bottom-left), the initial values of the conditional probabilities are 0.5, and the *update rate* is $\eta = 0.5$. The proposed estimation method alleviates the rapid convergence of the conditional probability compared to the maximum likelihood estimation, as shown in Fig. 2. Note that the proposed estimation method prevents $Pr(X_3 = 0|X_1 = 1, X_2 = 0)$ and $Pr(X_3 = 1|X_1 = 1, X_2 = 1)$ from being zero.



Conditional probability of each node

$Pr(X_1 = 0) = 0.625$
$Pr(X_1 = 1) = 0.375$

$Pr(X_2 = 0|X_1 = 0) = 0.585$
$Pr(X_2 = 1|X_1 = 0) = 0.415$
$Pr(X_2 = 0|X_1 = 1) = 0.5$
$Pr(X_2 = 1|X_1 = 1) = 0.5$

$Pr(X_3 = 0|X_1 = 0, X_2 = 0) = 0.375$
$Pr(X_3 = 1|X_1 = 0, X_2 = 0) = 0.625$
$Pr(X_3 = 0|X_1 = 0, X_2 = 1) = 0.5$
$Pr(X_3 = 1|X_1 = 0, X_2 = 1) = 0.5$
$Pr(X_3 = 0|X_1 = 1, X_2 = 0) = 0.25$
$Pr(X_3 = 1|X_1 = 1, X_2 = 0) = 0.75$
$Pr(X_3 = 0|X_1 = 1, X_2 = 1) = 0.75$
$Pr(X_3 = 1|X_1 = 1, X_2 = 1) = 0.25$

Fig. 2. Conditional probabilities estimated by the proposed estimation method. The proposed estimation method uses a network structure B and a population P_{parent} shown in Fig. 1 to estimate the conditional probabilities. The initial values of the conditional probabilities are 0.5, and the *update rate* is $\eta = 0.5$.

B. Selection Method

In the proposed method, we use the tournament selection [11] for constructing the population P_{parent}. The tournament selection is a method that randomly chooses s individuals from the population P and selects the best individual among them, where s is called the tournament size.

C. Replacement Method

We employ the restricted tournament replacement (RTR) [12] instead of the truncation replacement for replacing individuals in P with those of $P_{\text{candidate}}$.

The algorithm of RTR is shown in Algorithm 2. The populations P and $P_{\text{candidate}}$ are taken as an input, and the population P after individuals are replaced is returned as an output. In the loop of lines 1-14, RTR determines whether or not to replace each individual $I_{\text{candidate}}$ in the population $P_{\text{candidate}}$ with an individual in the population P and does the replacement if necessary. In the loop of lines 3-10, RTR randomly chooses n individuals in the population P, calculates the Hamming distance between each of the individuals and $I_{\text{candidate}}$, and selects the index, i_{min}, so that the individual indexed by i_{min}, $P[i_{\text{min}}]$, has the smallest Hamming distance. Here, n in line 3 is a user parameter, *rand* in line 4 is a function that returns a uniformly random integer value in a given range, and *hamming_dist* in line 5 is a function that measures the Hamming distance between two individuals. Finally, RTR compares the evaluation values of $I_{\text{candidate}}$ and $P[i_{\text{min}}]$ in line 11, and if $I_{\text{candidate}}$ has a better evaluation value, $P[i_{\text{min}}]$ is replaced with $I_{\text{candidate}}$ in line 12. Here, *eval* in line 11 is a function that returns the evaluation value of the individual.

D. Algorithm

The algorithm of the proposed method is shown in Algorithm 3. The outline of the algorithm is the same as BOA.

Algorithm 2 restrictedTournamentReplacement

```
Input: $P, P_{\text{candidate}}$
Output: $P$
for each $I_{\text{candidate}} \in P_{\text{candidate}}$ do
    $d_{\text{min}} \leftarrow \infty$
    for $j = 1$ to $n$ do
        $i \leftarrow$ rand$(1, |P|)$
        $d \leftarrow$ hamming_dist$(P[i], I_{\text{candidate}})$
        if $d < d_{\text{min}}$ then
            $d_{\text{min}} \leftarrow d$
            $i_{\text{min}} \leftarrow i$
        end if
    end for
    if eval$(I_{\text{candidate}}) <$ eval$(P[i_{\text{min}}])$ then
        $P[i_{\text{min}}] \leftarrow I_{\text{candidate}}$
    end if
end for
```

Algorithm 3 Proposed method

```
1: Initialize the population $P$ whose size is $\lambda$
2: Initialize the conditional probability $Pr$
3: repeat
4:     $P_{\text{parent}} \leftarrow$ Select $r\%$ individuals from $P$ by the tournament selection based on their evaluation values
5:     $B \leftarrow$ Construct a network structure using BIC metric and the K2 algorithm described in Section II-B
6:     $Pr(B) \leftarrow$ Estimate a conditional probability of each node according to Section III-A
7:     $P_{\text{candidate}} \leftarrow$ Generate new $\lambda_{\text{candidate}}$ individuals with $B$ and $Pr(B)$ according to Section II-D
8:     $P \leftarrow$ restrictedTournamentReplacement$(P, P_{\text{candidate}})$
9: until termination conditions are met
```

The differences are that the proposed method initializes the conditional probability Pr in line 2, that the proposed method uses the tournament selection in line 4, that the proposed method estimates the conditional probability according to Section III-A in line 6, and that the proposed method uses RTR in line 8.

IV. Experiments

In order to show the effectiveness of the proposed method, we compare its performance with that of BOA on benchmark problems in Section IV-A and on neural architecture search (NAS) [13] in Section IV-B. In Section IV-B, we also compare the performance of the proposed method and that of conventional methods used in NAS.

A. Evaluation on Benchmarks

1) Benchmark problems: In this experiment, we use two benchmark problems, the 3-deceptive function [7] and the NK-landscape [14].

The 3-deceptive function is defined as

$$f(\boldsymbol{x}) = \sum_{i=0}^{D/3-1} g(x_{3i+1}, x_{3i+2}, x_{3i+3}), \tag{5}$$

$$g(x_1, x_2, x_3) = \begin{cases} 0.9 & \sum_i x_i = 0 \\ 0.8 & \sum_i x_i = 1 \\ 0 & \sum_i x_i = 2 \\ 1 & \sum_i x_i = 3, \end{cases} \tag{6}$$

where $\boldsymbol{x} \in \{0,1\}^D$ is D-dimensional bit strings. The 3-deceptive function is a benchmark problem with dependencies among the three bits each and with a deceptive function landscape. We use the 3-deceptive function with the dimensions $D = \{30, 60, 90, 120, 150, 180\}$.

The NK-landscape is a problem with an adjustable rugged function landscape for D-dimensional bit strings. The NK-landscape has a parameter k which specifies that how many bits each bit depends on and determines the ruggedness of the function landscape. The evaluation value of an individual is calculated by averaging the evaluation values of all bits. The evaluation value of a bit is obtained from a lookup table by using the values of the bit and other k bits, i.e., the bit depends on the other k bits. Here, the k bits and the lookup table are randomly generated in advance. Please refer to [14] for more details of the NK-landscape. We use the NK-landscape with the dimensions $D = \{30, 60\}$ and the parameters $k = \{2, 3, 4, 5\}$.

2) Settings: The parameters of the proposed method and those of BOA are given as follows: the selection rate is $r = 50\%$, the tournament size is $s = 2$, the number of individuals generated in each generation is $\lambda_{\text{candidate}} = 0.5\lambda$, the user parameter in the restricted tournament replacement is $n = 5$, and the *update rate* is $\eta = 0.5$. In the 3-deceptive function, we perform 30 independent trials and search for the smallest population size λ where the global optimum can be found in all the trials. In the NK-landscape, we perform 30 independent trials and search for the population size λ where the average evaluation value is the best when the number of evaluations reaches 10^5.

3) Results: Figure 3 shows the number of evaluations for the proposed method and BOA to find the optimum on the 3-deceptive function when the dimension of the 3-deceptive function is varied. The proposed method outperforms BOA in all dimensions, and the larger the dimension becomes, the larger the difference is. Figure 4 shows the population size for the proposed method and BOA to need to find the optimum in all the trials when the dimension of the 3-deceptive function is varied. The population size increases in proportion to the dimension in BOA while it is almost constant in the proposed method. In the problem with $D = 180$, the proposed method succeeded in reducing the number of evaluations by 31.5% and the population size by 96.9%.

Table I shows the average evaluation values obtained by the proposed method and BOA on the NK-landscape when the dimension and the user parameter k are varied. The proposed method succeeded in obtaining comparable or better evaluation values in all settings compared to BOA. The larger the dimension and the user parameter k become, the larger the difference is.

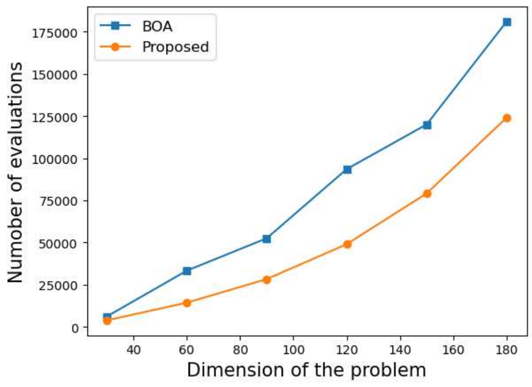


Fig. 3. Comparison of the number of evaluations between the proposed method (orange) and BOA (blue) in the 3-deceptive function. The x-axis shows the dimension of the problem and the y-axis shows the number of evaluations.

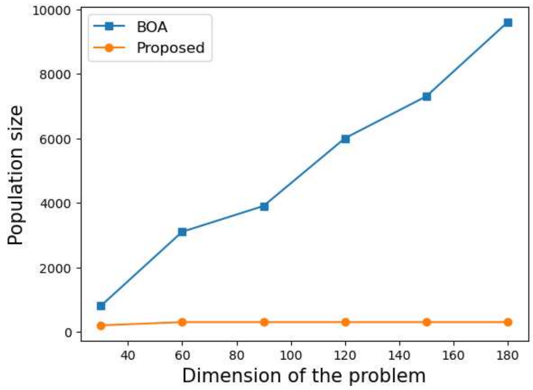


Fig. 4. Comparison of the population size between the proposed method (orange) and BOA (blue) in the 3-deceptive function. The x-axis shows the dimension of the problem and the y-axis shows the population size.

4) Discussions: As shown in Fig. 4, in the 3-deceptive function, BOA needs a large population size as the dimension of the problem increases. Hence, BOA generates and evaluates many individuals in each generation even if the probabilistic model almost converges. On the other hand, the proposed method requires less population sizes. As a result, as shown in Fig. 3, as the dimension of the problem increases, the difference in the number of evaluations between the proposed method and BOA becomes larger.

Figure 5 shows a Bayesian network which the proposed method built when the proposed method was applied to a 30-dimensional 3-deceptive function. The x- and y-axes correspond to dimension, and the bright area means that the dependencies between the two variables were detected. In the 3-deceptive function, since a bit string of an individual is divided into sub-strings of three bits each to compute the evaluation values, the dependencies among the variables should be detected in blocks of three dimensions each on the diagonal. As shown in Fig. 5, the proposed method succeeded in detecting the dependencies.

TABLE I
RESULTS FOR THE NK-LANDSCAPE. D IS THE DIMENSION OF THE PROBLEM AND k IS THE USER PARAMETER OF THE PROBLEM. BOA AND PROPOSED METHOD ARE THE AVERAGE EVALUATION VALUE AND THE STANDARD DEVIATION OVER 30 INDEPENDENT TRIALS WHEN THE NUMBER OF EVALUATIONS REACHES 10^5. THE LARGER THE EVALUATION VALUE IS, THE BETTER IT IS. DIFF IS THE DIFFERENCE BETWEEN THE EVALUATION VALUE OF THE PROPOSED METHOD AND THAT OF BOA.

D	k	Proposed method	BOA	diff
30	2	$\mathbf{0.781 \pm 0.000}$	$\mathbf{0.781 \pm 0.000}$	0.000
30	3	$\mathbf{0.798 \pm 0.000}$	0.789±0.010	+0.009
30	4	$\mathbf{0.812 \pm 0.000}$	$\mathbf{0.812 \pm 0.002}$	0.000
30	5	$\mathbf{0.779 \pm 0.004}$	0.771±0.005	+0.008
60	2	$\mathbf{0.812 \pm 0.001}$	0.811±0.001	+0.001
60	3	$\mathbf{0.805 \pm 0.003}$	0.801±0.003	+0.004
60	4	$\mathbf{0.780 \pm 0.005}$	0.769±0.005	+0.011
60	5	$\mathbf{0.782 \pm 0.012}$	0.761±0.011	+0.021

Table II shows the results when the three schemes introduced in the proposed method are removed one by one. In Table II, *Proposed w/o Updating of CPTs* is a method where the proposed estimation method of the conditional probability of a node described in Section III-A is replaced with BOA's estimation method described in Section II-C, *Proposed w/o Tournament* is a method where the tournament selection is replaced with the top selection, and *Proposed w/o RTR* is a method where the restricted tournament replacement is replaced with the truncation replacement. We performed 30 independent trials and searched for the smallest population size where the global optimum was found in all the trials. Table II suggests that all three schemes introduced in the proposed method are effective and that a scheme for estimating the conditional probability has the most significant impact on performance, which reduces the population size from 600 to 200. This is because the scheme described in Section III-A can generate a combination of values of variables that does not exist in the population P_{parent} at each generation, unlike the maximum likelihood estimation described in Section II-C. Therefore, the proposed method could preserve the diversity of the population and build a Bayesian network that appropriately represents the problem structure even with a small population size.

As shown in Table I, the proposed method is superior to BOA when the dimension and the user parameter k are large. In the NK-landscape, the larger the dimension and the user parameter k are, the more difficult and the more complex the problem is. Since the diversity of the population is likely to be lost in BOA, it is difficult for BOA to build a Bayesian network that appropriately represents the problem structure when the problem becomes complex. On the other hand, the proposed method could maintain the diversity of the population due to the three schemes, which could construct a more appropriate Bayesian network than BOA.

B. Evaluation on Neural Architecture Search

1) Neural architecture search: Neural architecture search (NAS) [13] is a network structure optimization problem for neural networks, which is one of the real-world problems of

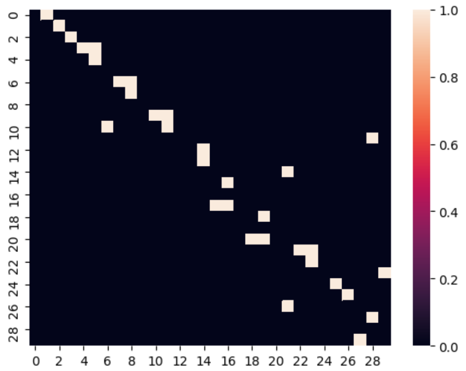


Fig. 5. Bayesian network which the proposed method built when the proposed method was applied to a 30-dimensional 3-deceptive function. The x- and y-axes correspond to dimension, and the bright area means the dependencies between the row and column variables.

TABLE II
EFFECTIVENESS OF THE THREE SCHEMES INTRODUCED IN THE PROPOSED METHOD ON THE 30-DIMENSIONAL 3-DECEPTIVE FUNCTION. #EVALUATIONS IS THE AVERAGE NUMBER OF EVALUATIONS ($\times 10^3$) AND THE STANDARD DEVIATION ($\times 10^3$) TO BE REQUIRED TO FIND THE GLOBAL OPTIMUM OVER 30 INDEPENDENT TRIALS. THE POPULATION SIZE λ IS THE SMALLEST ONE WHERE THE GLOBAL OPTIMUM WAS FOUND IN 30 INDEPENDENT TRIALS.

Method	# Evaluations	Population Size λ
Proposed w/o Updating of CPTs	5.33±0.509	600
Proposed w/o Tournament	4.22±0.559	300
Proposed w/o RTR	4.54±0.597	300
Proposed	$\mathbf{3.84 \pm 0.666}$	**200**

TABLE III
RESULTS FOR THE NAS-BENCH-101. #EVALUATIONS IS THE AVERAGE NUMBER OF EVALUATIONS ($\times 10^3$) AND THE STANDARD DEVIATION ($\times 10^3$) TO BE REQUIRED TO FIND THE GLOBAL OPTIMUM OVER 30 INDEPENDENT TRIALS. THE POPULATION SIZE λ IS THE SMALLEST ONE WHERE THE GLOBAL OPTIMUM WAS FOUND IN 30 INDEPENDENT TRIALS.

Method	#Evaluations	Population Size λ
Proposed Method	$\mathbf{14.4 \pm 3.62}$	**100**
BOA	18.3±5.40	1300

TABLE IV
COMPARISON OF THE AVERAGE BEST REGRET($\times 10^{-3}$) OF VARIOUS METHODS OVER 30 INDEPENDENT TRIALS IN THE NAS-BENCH-101. THE SMALLER THE VALUE IS, THE BETTER IT IS.

Method	Best regret
Proposed Method	**2.16**
BOA	3.51
ASNG-NAS	4.19
Regularized Evolution	4.38
Reinforcement Learning	4.96
TPE	8.69
Random Search	7.39

the black-box discrete optimization. In this experiment, we use the NAS-Bench-101 [15] which is a benchmark problem of NAS. The NAS-Bench-101 has a tabular dataset which maps neural network architectures to their training and evaluation metrics. A neural network architecture is a series of multiple units, each of which consists of N normal cells and a down-sampling layer. The search space is the internal structure of the normal cell, which is represented by a directed acyclic graph (DAG) with seven nodes, an input node, an output node, and five processing nodes. The DAG is represented by adjacency matrix which is an upper triangular matrix of 7×7. Each processing node takes one of three labels, 3×3 convolution, 1×1 convolution, and 3×3 max-pool. Therefore, the NAS-Bench-101 has 26 dimensions: 21 dimensions for the upper triangular matrix and 5 dimensions for determining the node labels. For the evaluation value, we use a score called regret: $r(A_i) = f(A_i) - f(A^*)$, where $f(A_i)$ is the test error for an architecture A_i and A^* is the architecture with the lowest error of all the architectures.

2) Settings: The parameters of the proposed method and those of BOA are given as follows: the selection rate is $r = 50\%$, the tournament size is $s = 2$, the number of individuals generated in each generation is $\lambda_{\text{candidate}} = 0.5\lambda$, the user parameter in the restricted tournament replacement is $n = 5$, and the *update rate* is $\eta = 0.1$. We perform 30 independent trials and search for the smallest population size λ where the global optimum can be found in all the trials.

3) Results: Table III shows the result of each method in NAS-Bench-101. The proposed method succeeded in reducing the number of evaluations by 21.3% and the population size by 92.3% compared to BOA.

4) Comparison with conventional methods in NAS: Table IV shows the average best evaluation values or regrets of ASNG-NAS [16], regularized evolution [17], reinforcement learning [13], TPE [18], and random search [19] in addition to the proposed method and BOA on 30 independent trials. The user parameters of ASNG-NAS are set to the recommended values [16], and those of remaining methods are set to the recommended values [15]. The termination condition is that the *estimated wall-clock time* reaches 10^7 seconds [15]. The *estimated wall-clock time* can be calculated by summing up training time, given by the NAS-Bench-101, of each architecture. As shown in Table IV, the proposed method and BOA outperform the other methods, especially the proposed method is the best performance.

5) Discussions: Fig. 6 and 7 show the Bayesian networks obtained in the first and second trials of the proposed method. The bright area means that the dependencies between the two variables were detected. In spite of different trials, the same dependencies were detected in several areas. It suggests that dependencies among the variables would exist in the search space of NAS-Bench-101 and, therefore, the proposed method and BOA outperformed the other conventional methods as shown in Table IV.

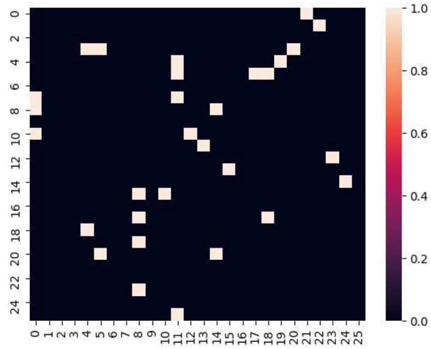


Fig. 6. Bayesian network which the proposed method built in the first trial when the proposed method was applied to the NAS-Bench-101. The x- and y-axes correspond to dimension, and the bright area means the dependencies between the two variables.

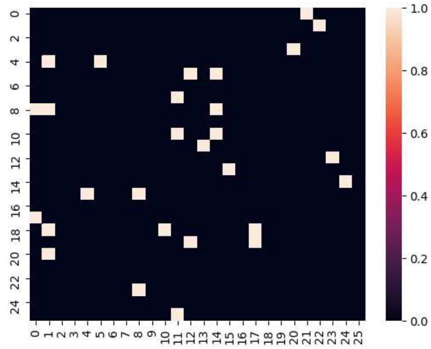


Fig. 7. Bayesian network which the proposed method built in the second trial when the proposed method was applied to the NAS-Bench-101. The x- and y-axes correspond to dimension, and the bright area means the dependencies between the two variables.

V. Conclusion

In this paper, we proposed a method that introduced three schemes into the bayesian optimization algorithm (BOA) to preserve the diversity of the population for black-box discrete optimization (BB-DO) problems with epistasis among parameters. The proposed method addresses a problem of BOA in that the diversity of the population is likely to be lost, especially in high dimensional problems. The experiment results showed that the proposed method found the global optimum with the smaller population size than BOA and was superior to BOA in terms of the number of evaluations in the 3-deceptive function. In particular, the proposed method improved the number of evaluations by 31.5% and the population size by 96.9% in the 180-dimensional 3-deceptive function. Besides, the proposed method found comparable or better solutions in all NK-landscape settings compared to BOA. In the structural optimization problem of neural networks, the proposed method improved the number of evaluations by 21.3% and the population size by 92.3% compared to BOA. In addition, we compared the proposed method and the conventional methods used in this field, ASNG-NAS, regularized evolution, reinforcement learning, TPE, and random search, and showed that the proposed method outperformed all the methods in terms of the evaluation value.

In future work, we will propose an adaptive mechanism for the *update rate* for estimating conditional probabilities of Bayesian networks and improve the method to be able to optimize BB-DO problems with a smaller population size.

Acknowledgement

This work was partially supported by JSPS KAKENHI Grant Number JP20K11986.

References

[1] B. Doerr and F. Neumann, *Theory of Evolutionary Computation*. Springer International Publishing, 2019.

[2] M. Hauschild and M. Pelikan, “An introduction and survey of estimation of distribution algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111 – 128, 2011.

[3] S. Baluja, “Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning,” Carnegie Mellon University, Tech. Rep., 1994.

[4] G. R. Harik, F. G. Lobo, and D. E. Goldberg, “The Compact Genetic Algorithm,” *Trans. Evol. Comp*, vol. 3, no. 4, pp. 287–297, 1999.

[5] J. S. De Bonet, C. L. Isbell, Jr., and P. Viola, “MIMIC: Finding Optima by Estimating Probability Densities,” in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, ser. NIPS’96, 1996, pp. 424–430.

[6] M. Pelikan and D. E. Goldberg, “BOA: The Bayesian Optimization Algorithm.” Morgan Kaufmann, 1999, pp. 525–532.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[8] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, Jun. 1994.

[9] G. F. Cooper and E. Herskovits, “A Bayesian Method for the Induction of Probabilistic Networks from Data,” pp. 309–347, 1992.

[10] G. Schwarz, “Estimating the Dimension of a Model,” *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 03 1978.

[11] B. L. Miller, B. L. Miller, D. E. Goldberg, and D. E. Goldberg, “Genetic Algorithms, Tournament Selection, and the Effects of Noise,” *Complex Systems*, vol. 9, pp. 193–212, 1995.

[12] C. F. Lima, C. Fernandes, and F. G. Lobo, “Investigating Restricted Tournament Replacement in ECGA for Non-Stationary Environments,” in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: Association for Computing Machinery, 2008, pp. 439–446.

[13] B. Zoph and Q. V. Le, “Neural Architecture Search with Reinforcement Learning,” in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[14] S. Kauffman and E. Weinberger, “The NK model of rugged fitness landscapes and its application to maturation of the immune response,” *Journal of theoretical biology*, vol. 141 2, pp. 211–245, 1989.

[15] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, “NAS-Bench-101: Towards Reproducible Neural Architecture Search,” in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 7105–7114.

[16] Y. Akimoto, S. Shirakawa, N. Yoshinari, K. Uchida, S. Saito, and K. Nishida, “Adaptive Stochastic Natural Gradient Method for One-Shot Neural Architecture Search,” in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 171–180.

[17] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, “Regularized Evolution for Image Classifier Architecture Search,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4780–4789, Jul. 2019.

[18] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyper-Parameter Optimization,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2546–2554.

[19] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012.