The essence of Real-valued Characteristic Function for Pairwise Relation in Linkage Learning for EDAs

Jui-Ting Lee
Department of Electrical Eegineering
National Taiwan University
No.1, Sec. 4, Roosevelt Rd.,
Taipei, Taiwan
r98921050@ntu.edu.tw

Kai-Chun Fan
Department of Electrical Eegineering
National Taiwan University
No.1, Sec. 4, Roosevelt Rd.,
Taipei, Taiwan
r98921070@ntu.edu.tw

Tian-Li Yu
Department of Electrical Eegineering
National Taiwan University
No.1, Sec. 4, Roosevelt Rd.,
Taipei, Taiwan
tianliyu@cc.ee.ntu.edu.tw

ABSTRACT

Existing EDAs learn linkages starting from pairwise interactions. The characteristic function which indicates the relations among variables are binary. In other words, the characteristic function indicates that there exist or not interactions among variables. Empirically, it can occur that two variables should be sometimes related but sometimes not. This paper introduces a real-valued characteristic function to illustrate this property of fuzziness. We examine all the possible binary models and real-valued models on a test problem. The results show that the optimal real-valued model is better than all the binary models. This paper also proposes a crossover method which is able to utilize the real-valued information. Experiments show that the proposed crossover could reduce the number of function evaluations up to four times. Moreover, this paper proposes an effective method to find a threshold for entropy based interaction-detection metric and a method to learn real-valued models. Experiments show that the proposed crossover with the learned real-valued models works well.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Global optimization–Analyse.

General Terms

Algorithms.

Keywords

Building Blocks, Crossover, Linkage Learning.

1. INTRODUCTION

Estimation of distribution algorithms (EDAs) learn linkages among the variables of problems. There are at least two limitations of linkage-learning methods. The first one is that the linkage-learning methods consider the interactions between two variables (so-called pairwise linkages) rather than multi-variables. ecGA [5] learns linkages starting from two variables and then learns pairwise linkages between two groups of variables. LINC and LIMD [9] detect nonlinearity between two variables. Bayesian network based algorithms (*e.g.* EBNA, BOA, hBOA [1, 12, 10]) construct a graph model with causal edges among two variables. DSMGA [18] utilizes a dependency structure matrix which records dependencies between two variables. Because of computational burden – running time of linkage learning methods is desirable to be sub-quadratic, it is reasonable to consider only pairwise linkages, which yields $O(l^2)$ running time.

The second limitation of linkage-learning methods is that two variables are recognized as having interactions or having no interactions. In other words, the relations among variables are binary. ecGA divides variables into several groups. LINC and LIMD detect nonlinearity between two variables. DSMGA uses entropy as a criterion to detect interaction among variables and then a threshold is used to differentiate whether two variables have interactions or not. D^5 [15] utilizes a perturbation based method to construct linkage sets. All the above algorithms adopt a deterministic model during recombination.

As for Bayesian network based algorithms, the relations among the variables in a Bayesian network are either conditionally dependent or independent (simply and conditionally), so the relations are still binary. Moreover, given a Bayesian network, the form of the interpreted model is deterministic. In Section 2, the relations in a Bayesian network will be discussed in greater detail.

The binary property of all the above algorithms can be represented by a binary characteristic function which indicates the outcomes of a pairwise linkage. This function only outputs one and zero, and there does not exist a real number to illustrate the relations among variables. As a result, a question comes to our mind: Do GAs need a real-valued characteristic function to indicate the pairwise linkage? The outputs of the real-valued characteristic function indicate non-deterministic models. Here the non-deterministic models mean that the relations between variables is not fixed during recombination.

This paper discusses the possibility of the real-valued characteristic function and demonstrate its benefits. The goal of this paper is to demonstrate its importance. Experiments

GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0557-0/11/07 ...$10.00.

show that the optimal real-valued model is better than the optimal binary model. The concept of the real-valued characteristic function should be further discussed and could be useful for developing next-generation EDAs.

The rest of this paper is structured as follows. The formulation and discussions of the real-valued characteristic function is addressed in Section 2. In Section 3, a short experiment demonstrate its benefits. Section 4 proposes a crossover to realize its concept. In section 5, experiments show that the optimal real-valued model is better than the optimal binary model. Section 6 proposes a method to find a threshold for entropy based metric [14] to detect interactions among genes. Then a method is provided to learn real-valued models. Section 7 proposes a population-wise version of the proposed crossover. Finally, Section 8 concludes this paper.

2. REAL-VALUED CHARACTERISTIC FUNCTION

This section first discusses that commonly used EDAs adopt binary models, in which the relations among variables are binary and deterministic. This paper then addresses the formulation of binary characteristic function which represents the binary models. At last, the real-valued characteristic function, indicate the non-deterministic models, is proposed and explained carefully.

EDAs like ecGA, LIMD, LINC, D^5 and DSMGA perform building block (BB)-wise crossover. It can be easily observed that the relations among variables are either having interactions (in the same BB) or having no interactions (in the different BBs), so the relations are represented by a binary characteristic function. Moreover, the constructed models are fixed during recombination.

EBNA, BOA and hBOA sample offspring from Bayesian networks. After one variable is sampled, children of this variable are sampled according to their corresponding conditional probabilities, and this process is similar to population-wise shuffling. The relations among variables are either conditionally dependent (*e.g.,* $P(A, B) = P(A) \times P(B|A)$, or *vice versa*) if there exist a path from A to B or independent (*e.g.,* simple independence: $P(A, B) = P(A) \times P(B)$). After constructing the Bayesian networks, the models for generating offspring are fixed during recombination. As a result, the relations among variables can be represented by a binary characteristic function and the learned models are deterministic. In other words, the learned models are fixed during recombination.

For overlapping problems, D^5, LIEM [8] and DSMGA are known to have the ability to identify overlapping BBs. Yu *et al.* [19] proposed a crossover method which is able to effectively recombine overlapping BBs. The idea is using a minimal cut algorithm to disrupt minimal number of overlapping BBs. Tsuji *et al.* [16] then modified Yu's crossover method so that it could increase BB mixing rate without disrupting more BBs by considering identical allele values. The above crossover methods do not have static cross sites (some BBs will be disrupted). Note that the concept of relations is different form the cross sites. Take simple genetic algorithm for example, using one-point crossover could create dynamic cross sites. However, dynamic cross sites do not involve the information of the relations. This paper focuses on the concept of the relations and the models for recombination rather than the method to choose cross sites.

After discussing all the above algorithms, we found that the models utilized by these algorithms can all be indicated by a binary characteristic function and the models is deterministic during recombination. The formulation of the binary characteristic function is as follows:

$$\mathcal{R}_B(X_i, X_j) \in \{0, 1\}. \tag{1}$$

This characteristic function indicates that the relation between variable X_i and X_j is either zero or one. Zero indicates that there exist no interactions among X_i and X_j; while one indicates that X_i and X_j have interactions. As a result, we wonder that if there exists a optimal model which can not be represented by the binary characteristic function. Here the optimal model means the model which yields the fewest number of function evaluations for EDAs to reach the global optima.

For problems where the variables are not interacted with each others, these problems can be solved without respecting the interactions among the variables. OneMax is a typical example. For problems where the variables are strongly interacted with each others, they can be solved efficiently if the interactions are detected and the variables are properly decomposed into sub-problems. Trap function [2] is such an example. We wonder that if there exist problems that can not be categorized into the previous two types of problems.

This paper tries to investigate the possibility of the relations indicated by a real-valued characteristic function and proposes its formulation. The real-valued characteristic function is addressed as follows.

$$\mathcal{R}_r(X_i, X_j) \in [0, 1]\,. \tag{2}$$

The relation between X_i and X_j is now represented by a real number in between zero and one. Note that the relation $\mathcal{R}_B$ is a special case of $\mathcal{R}_r$. The value indicated by Equation 2 represent the certainty that X_i and X_j should be bound together. When the value approaches one, we tend to process X_i and X_j as interacting; while the value approaches zero, we tend to process X_i and X_j as not interacting. Consequently, fuzziness has been added into deterministic binary characteristic function.

Take $\mathcal{R}_r = 0.9$ as an example. During recombination, 10 percent of the population are treated as independent and 90 percent of the population are treated as dependent. By introducing the real-valued characteristic function, one could construct a non-deterministic model. In other words, the form of the model is not fixed.

Note that this is different from the concept of Bayesian networks. If we introduce the real-valued characteristic into Bayesian networks, the edges in the networks would become non-deterministic during recombination. The joint distribution over the involved variables would not be a fixed form while sampling the offspring.

To demonstrate the benefits of the real-valued characteristic function, next section show that utilizing the real-valued models could get a better performance than all the binary models on the test problem.

3. THE BENEFITS OF REAL-VALUED MODELS

Although the real-valued characteristic function has been

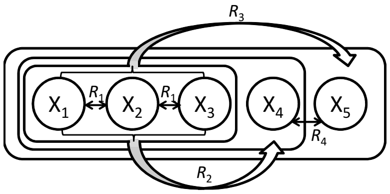

Figure 1: The structure of the problem with the fitness function defined by Equation 3, the circles denote the variables, the arrows denote the relationship types among the variables and the rectangles denote sub-problem.

proposed, its benefits have not been demonstrated – why do GAs need such a function to indicate relations? In this section, we enumerate all the possible binary models and some of the real-valued models for GAs to solve a found problem. If there exist real-valued models that require fewer number of function evaluations, it should draw more attention and be further studied. Here we sweep through the real values which indicate the models because we want to find the optimal real-valued model, and how to learn the real values will be discussed in Section 6.

The fitness function of the test problem is formulated as

$$\begin{aligned} f(\vec{x}) = &\, Trap_3(x_1,x_2,x_3) + Trap_4(x_1,x_2,x_3,x_4) \\ &+ Trap_5(x_1,x_2,x_3,x_4,x_5), \end{aligned} \tag{3}$$

where $Trap_k$ denotes trap function [2] of order k, and x_i denotes the value of gene on the position i. We classify the relations among variable into several groups according to the structure of the problems. For example, the relation between X_1 and X_4 is similar to the relation between X_2 and X_4 in this test problem, so they are considered as the same type of relation. We define the types of relation as the Cartesian product of two sets U and V:

$$\begin{aligned} &\mathcal{R}_k : U \times V && \forall u \in U, \forall v \in V \\ &\mathcal{R}(u,v) = 1 && ,\text{if}\, u = v. \end{aligned}$$

Table 1 shows the four types of relation in the test problem. As a result, one can only sweep through four values rather than $\binom{5}{2}$ values to enumerate the real-valued models. Figure 1 shows the types of relations of the test problem, where the circles denote the variables, the arrows denote the relationship types among the variables and the rectangles denote the structure of sub-function. In the rest of this paper, this representation will be used to show the types of relations.

To realize the concept of real-valued characteristic function, Equation 2 is modified as below:

$$\mathcal{P}(X_i,X_j) = \frac{1}{2} + \frac{\mathcal{R}_r(X_i,X_j)}{2}, \tag{4}$$

where $\mathcal{R}_r(X_i,X_j) \in [0,1]$, so $\mathcal{P}$ is a real number in between 0.5 and 1.0. $\mathcal{P}$ represents the probability that X_i and X_j come from the same parent while creating offspring. Pairs of variable having high values of $\mathcal{P}$ tend to be transferred together during recombination. By such transformation, one could implement the proposed concept with pairwise uniform crossover.

Take two variables for example, two variable will be treated as independent if $\mathcal{P}$ is equal to 0.5. In this situation, the crossover probability of one variable is 0.5 and is independent of the other variable. If $\mathcal{P}$ is equal to 1.0 and one variable is crossed, the conditional crossover probability of the other variable is $\mathcal{P}$, 1.0. In the same fashion, if $\mathcal{P}$ is equal to 1.0 and one variable is not crossed, the conditional crossover probability of the other variable is $1 - \mathcal{P}$, 0.0. As we can see, these two variables will be treated as a building block (BB) if $\mathcal{P}$ is equal to 1.0.

After determining the real values of all the four types of relations, the $\mathcal{P}$-value of any pair of variables can be calculated. Crossover models can be then created by calculating the joint probability of $\mathcal{P}$. For example, if X_1 has been transferred and X_2 has stayed, the conditional probability for X_3 to be transferred is calculated as

$$\frac{\mathcal{P}(X_3,X_1)\cdot\bar{\mathcal{P}}(X_3,X_2)}{\mathcal{P}(X_3,X_1)\cdot\bar{\mathcal{P}}(X_3,X_2) + \bar{\mathcal{P}}(X_3,X_1)\cdot\mathcal{P}(X_3,X_2)},$$

where $\bar{\mathcal{P}}$ denotes $1 - \mathcal{P}$ and the denominator is added as a normalizer. As we can see, if $\mathcal{P}(X_3,X_1)$ is equal to 1.0, the conditional probability for X_3 to be transferred is 1.0. Then the sub-solution consisting of X_3 and X_1 will not be disrupted. Moreover, because $\mathcal{P}$ is in between 0.5 and 1.0, one could construct a non-deterministic model that sometimes X_3 and X_1 are treated as a BB while sometimes not.

The experiments are performed with binary tournament selection and full replacement. The crossover method is similar to pairwise uniform crossover. The difference is that the crossover probability is the calculated conditional probability.

For all the enumerated 52 binary models and the real-valued models, bisection method is used to measure the minimal population size of GA with 10 consecutively fully population convergence. The 52 models is acquired by enumerating all the possible models of 5 bits. The number of possible models is equal to bell number [13]. The bisection method is repeated 100 times to get a stable number of function evaluations. Table 2 shows that both the optimal $\mathcal{R}_3$ and $\mathcal{R}_4$ are 0.2 rather than 0 or 1. In other words, for the pair of variables where the relations are indicated by $\mathcal{R}_3$ and $\mathcal{R}_4$, the probability for transferring these pairs of variables together should be slightly higher than considering they are not interacted. The optimal real-valued model outperforms the optimal binary model by approximately 13% number of function evaluations.

Table 1: The relationship types in the problem of Equation 3.

Set U	Set V	Relation
X_1, X_2, X_3	X_1, X_2, X_3	$\mathcal{R}_1$
X_1, X_2, X_3	X_4	$\mathcal{R}_2$
X_1, X_2, X_3	X_5	$\mathcal{R}_3$
X_4	X_5	$\mathcal{R}_4$

Table 2: By enumerating all the 52 binary models and some of the real-valued models, the experimental results show the benefits with the existence of the real-valued characteristic function.

	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	N_{fe}
Opt. real-valued model	1.0	1.0	0.2	0.2	457.82
Opt. binary model	1	1	0	1	528.55

These experiments demonstrate the benefits of the real-valued characteristic function. There exists a better way to represent the relations. Next section proposes a crossover which is able to recombine chromosomes according to the real-valued models.

4. A CROSSOVER FOR REAL-VALUED MODELS

This section proposes a crossover designed for dealing with $\mathcal{P}$ defined in Equation 4. Imagining an l-bits problem, an l-by-l adjacent matrix can be constructed, where the entry d_{ij} contains $\mathcal{P}(X_i, X_j)$, and how to learn the real values will be discussed in Section 6. This adjacent matrix is treated as the models in EDAs. The proposed crossover is able to utilize the information provided by the adjacent matrix and recombining two chromosomes. In other words, it will produce two children from two parents at a time according to the matrix. The algorithm is as follow:

Algorithm: The Proposed Crossover

1. Input two chromosomes as parents p_1 and p_2.
2. $\forall$ genes i in chromosome, if $p_1(i) = p_2(i)$, then $c_1(i) \leftarrow p_1(i)$, $c_2(i) \leftarrow p_2(i)$; else, then $G \leftarrow G$ joint $\{i\}$, where $p_1(i)$ denotes the i-th gene in p_1.
3. $\mathit{root} \leftarrow \arg\max_i \{\sum_{j \in G} \mathcal{P}(X_i, X_j) | i \in G\}$
4. $c_1(\mathit{root}) \leftarrow p_1(\mathit{root})$ and $c_2(\mathit{root}) \leftarrow p_2(\mathit{root})$. $G \leftarrow G - \{\mathit{root}\}$.
5. $\mathit{next} \leftarrow \arg\max_i \{\mathcal{P}(X_{root}, X_i) | i \in G\}$
6. Calculate probability Pr. If rand(0,1)< Pr, $c_1(\mathit{next}) \leftarrow p_2(\mathit{next})$ and $c_2(\mathit{next}) \leftarrow p_1(\mathit{next})$. Otherwise, $c_1(\mathit{next}) \leftarrow p_1(\mathit{next})$ and $c_2(\mathit{next}) \leftarrow p_2(\mathit{next})$.
7. $\mathit{root} \leftarrow \mathit{next}$, $G \leftarrow G - \{\mathit{next}\}$
8. Repeat steps 5 to 7 until $G = \emptyset$.
9. The two chromosomes, c_1 and c_2, are the results.

Firstly, two chromosomes must be input as parents. In step 2, compare the values of the alleles between two parents and assign the values of identical genes to the children, because it is futile to cross these genes. Record rest genes into a set G. In step 3, find the gene i that maximize $\sum_{j \in G} \mathcal{P}(X_i, X_j)$. The sum can be viewed as the relation of X_i between the remaining genes. A gene which has a large sum indicates that it is of significance for some level, so it should be treated firstly. One can also find the gene of the largest $\mathcal{P}(X_i, X_j)$ and even use the technique of roulette wheel to find the gene. What we provided is just one of the choices. In step 4, do not cross the gene of the largest sum, because it does not make a difference whether the first gene is crossed or not, and then move this gene out of the set G. In step 5, find a gene that maximize $\mathcal{P}$ with the previous assigned gene, then calculate the conditional probability to cross this gene with respect to $k-1$ previous genes, where k denotes bounded problem difficulty.

The following pseudo-code is used to calculate the conditional probability for crossover:

Pseudo-code for calculating the conditional probability

```
for all last $k-1$ assigned genes $i$ do
  if gene $i$ is crossed then
    $P_c \leftarrow P_c \times \mathcal{P}(X_{next}, X_i)$
    $P_{nc} \leftarrow P_{nc} \times \bar{\mathcal{P}}(X_{next}, X_i)$
  else
    $P_c \leftarrow P_c \times \bar{\mathcal{P}}(X_{next}, X_i)$
    $P_{nc} \leftarrow P_{nc} \times \mathcal{P}(X_{next}, X_i)$
  end if
end for
return $\frac{P_c}{(P_c + P_{nc})}$
```

$\bar{\mathcal{P}}(X_{next}, X_i)$ denotes $1 - \mathcal{P}(X_{next}, X_i)$. This algorithm returns a conditional probability to cross X_{next}. Make the decision that whether or not to cross X_{next} according to this probability, and then repeat steps 5 to 7 until all the genes are assigned. Finally, return the results.

The proposed crossover is able to utilize real-valued models and recombine chromosomes. This crossover has some properties as below:

- Pairwise crossover. The proposed method processes chromosomes pairwisely, so it is easy to be parallelized for shortening executing time.
- No clustering. This method do not have to adopt clustering method, so the clustering time is saved.
- Generalizing. The proposed crossover will reduce to a uniform crossover if all $\mathcal{P}$ is 0.5, and it will reduce to a BB-wise crossover if all $\mathcal{P}$ between the genes in the same BB is 1.0 and all the rest $\mathcal{P}$ is 0.5. The ability of the uniform crossover and the BB-wise crossover is preserved without losing the capability of real-valued models.

The proposed crossover provides a practical mechanism and a experimental platform to employ real-valued models, so one could perform experiments with the proposed crossover to explore the possibility of the real-valued characteristic function. In the next section, experiments show that the proposed crossover using real-valued models could reduce the number of function evaluations up to four times on the test problems.

5. EXPERIMENTS

In this section, we demonstrate three test problems which exist real-valued models that outperforms all the possible binary models. The experiments are similar to the experiments in Section 3. The performance is improved to nearly four times with the optimal real-valued model on the third test problem. A short discussion about the possible causes of the performance improvements is addressed.

The following experiments are performed with the proposed crossover. We sweep across values of $\mathcal{R}$ by 0.2 interval to find the optimal models that require fewest number of function evaluations. We choose 0.2 interval rather than 0.1

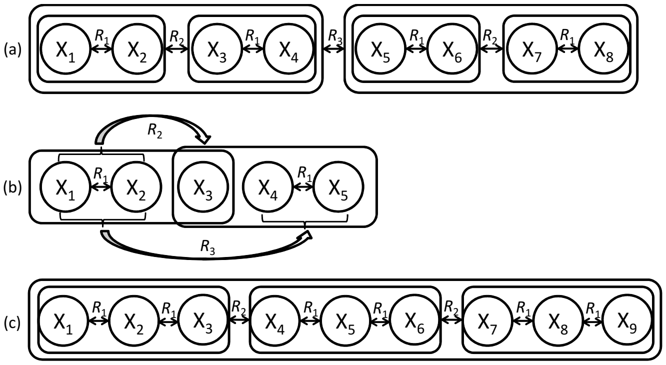
(a)
X_1 R_1 X_2 R_2 X_3 R_1 X_4 R_3 X_5 R_1 X_6 R_2 X_7 R_1 X_8
(b)
R_2
X_1 R_1 X_2 X_3 X_4 R_1 X_5
R_3
(c)
X_1 R_1 X_2 R_1 X_3 R_2 X_4 R_1 X_5 R_1 X_6 R_2 X_7 R_1 X_8 R_1 X_9

Figure 2: The structure of the problems with the fitness functions defined by Equation 6, Equation 7 and Equation 8, where the circles denote the variables, the arrows denote the types of relations and the rectangles denote sub-problem.

to decrease the time to enumerate the real-valued models, and the found optimal model is better enough to defeat the optimal binary model. To reduce the difficulty of sweeping $\mathcal{R}$ values, the relationships of the same type share the same $\mathcal{R}$ values as addressed in Section 3. For each models, 100 independent bisection runs are performed to find the minimal population size for 10 consecutively successes of finding the global optima. The averaged minimal population size of 100 bisection runs is used to yield the number of function evaluation that EDAs need for solving the problems.

5.1 Three Test Problems

The first test problem is hierarchical XOR [17]. The definition of hXOR is as follows:

$$h_{xor}(\vec{x}) = \begin{cases} 1 & \text{if } \lambda = 1 \\ 1 & \text{if } h_{xor}(L) = 1, h_{xor}(R) = 1, \text{and } L = \bar{R} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where λ is the number of hierarchical levels, $L = x_1 x_2 \cdots x_{2^{\lambda-1}}$, $R = x_{2^{\lambda-1}+1} x_{2^{\lambda-1}+2} \cdots x_{2^\lambda}$, and $\bar{R}$ is the bitwise negation of R. For $\lambda > 0$, the fitness of hXOR is defined as:

$$f_1(\vec{x}) = H_{xor}(L) + H_{xor}(R) + \begin{cases} \text{length}(\vec{x}) & \text{if } h_{xor}(\vec{x}) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The base case is when $\lambda = 0$, $Hxor(\vec{x}) = 1$. hXOR has two global optima and $2^{l/2}$ local optima at the lowest level for a problem size l. There are exactly half of 1's and half of 0's in the global optima.

Table 3: The experimental results of the three test problems show that the optimal real-valued models outperform the optimal binary models by approximately 39%, 83% and 362% number of function evaluations.

		$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	N_{fe}
(a)	Opt. real-valued model	1	0.2	0	238.24
(a)	Opt. binary model	1	1	0	327.22
(b)	Opt. real-valued model	1	1	0.2	6070.33
(b)	Opt. binary model	1	1	1	11104.41
(c)	Opt. real-valued model	1	0.4	–	19574.29
(c)	Opt. binary model	1	1	–	70923.66

An 8-bit hXOR is used, and its types of relations are shown in Figure 2(a). The meaning of this plot is addressed in Section 3. There exist three different types in an 8-bit hXOR. The first one is among genes comprising level 1 global optima. The second one is among genes comprising level 2 global optima and so on. Table 3(a) shows that the optimal real-valued model outperforms the optimal binary model by 27%. The optimal values of $\mathcal{R}_1$ and $\mathcal{R}_3$ are still binary; however, the optimal $\mathcal{R}_2$ is not binary.

The second test problem is two traps overlapping with one bit. Someone may wonder that why not using a problem with cyclically overlapping BBs used in [19]. Because the cyclically overlapping problem has much more types of relations to be determined, it is impractical to sweep across $\mathcal{R}$ and find the optimal model. As a result, a rather simple overlapping problem is used. The fitness function is written as

$$f_2(\vec{x}) = Trap_3(x_1, x_2, x_3) + Trap_3(x_3, x_4, x_5). \tag{7}$$

The structure and relation types of this test problem are shown in Figure 2(b). There exist three relation types in this test problem. The first one is among variables in the same trap without the overlapping bit. The second one is among variables in the trap and the overlapping bit. The third one is among variables in the different traps. 20 sequential sub-problems (100-bits totally) defined by Equation 7 are used. As for those relations among different sub-problems, we set $\mathcal{R}$ as zero.

Table 3(b) indicate a 45% improvement with the real-valued model. The relations among the variables in the same trap, $\mathcal{R}_1$ and $\mathcal{R}_2$, are reasonably one. The optimal binary $\mathcal{R}_3$ should be one to avoid too many BB disruptions or EDAs will fail to find the optima. However, the optimal non-binary $\mathcal{R}_3$ is 0.2. The results show that the relations indicated by $\mathcal{R}_3$ do not as strong as $\mathcal{R}_1$ and $\mathcal{R}_2$.

The fitness function of the last test problem is written as

$$f_3(\vec{x}) = Trap_3(x_1, x_2, x_3) + Trap_3(x_4, x_5, x_6) + \\ Trap_3(x_7, x_8, x_9) + Trap_9(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9). \tag{8}$$

The structure and the relation types are shown in Figure 2(c). $\mathcal{R}_1$ denotes the relations among the variables within the same $Trap_3$. $\mathcal{R}_2$ denotes the relations among the variables within the same $Trap_9$ but in different $Trap_3$. We use 10 sub-problems defined by Equation 8 in sequential (90-bits totally). Table 3(c) shows that the optimal real-valued model outperforms the optimal binary model by 72%.

It is reasonable to recognize the genes in the same $Trap_3$ as a BB, so the optimal $\mathcal{R}_1$ is equals to 1. However, GAs should also recognize the genes in the same $Trap_9$ as a whole BB in the case of binary model or it will fail to find the global optima. Larger BBs require much more number of function evaluations because GAs need a larger population size to meet the initial supply requirement [3]. However, the optimal real-valued model indicates that GAs do not need to respect the relation introduced by $Trap_9$ all the time. The optimal non-binary $\mathcal{R}_2$ is equal to 0.4. One of the possible reasons for the non-binary value is that the local optima of $Trap_3$ also comprises the local optima of $Trap_9$, so it is needless to consider the relation indicated by $\mathcal{R}_2$ all the time. The real-valued characteristic function provides

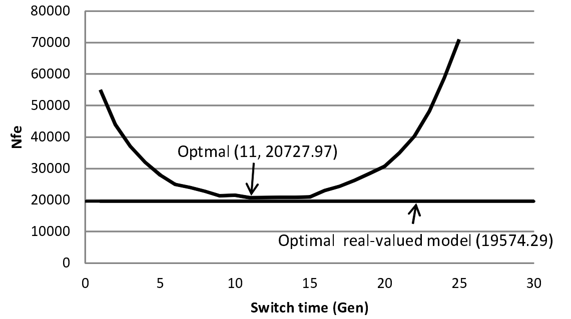

Figure 3: This figure shows that switching the BB information in the 11th generation yields the fewest number of function evaluations 20727.98 with the binary models.

a way to compromise between assuring the initial supply requirement and preserving the information of BB, and the experimental results show the improvements of performance.

5.2 Discussions

This subsection tries to further examine why the performance of the optimal real-valued model is much better than the optimal binary model.

Further experiments are performed with the third test problem by considering its characteristic. $\mathcal{R}_2$ is equal to 0 in the first few generations to reduce the population size guided by the initial supply requirement. $\mathcal{R}_2$ then switch to 1 to introduce the BB information of $Trap_9$. We sweep across the optimal time to switch $\mathcal{R}_2$ from the first generation to the 25th generation.

Figure 3 shows that switching $\mathcal{R}_2$ in the 11th generation yields the fewest number of function evaluations. Although this method could get a comparable performance to the optimal real-valued model, there exists no realistic mechanism to automatically detect the optimal time to switch the BB information. Moreover, applying the switching method does not outperform the optimal real-valued model.

We use ecGA to solve this problem so that we could observe the learned models generation by generation. In reality, ecGA needs a large population size (48000) to conquer the problem difficulty. The BB information of $Trap_3$ is recognized in the first few generations, then some BBs of $Trap_3$ will be merged into one 6-bits BB. In the end, all the BBs of $Trap_3$ will be merged into one 9-bits BB. In other words, GAs need a dynamic binary BB information in order to solve this problem. However, finding the optimal switching time is the key to efficiently solve this problem, and there exists no such mechanism to detect the optimal switching time. Using a static real-valued model could get a well performance without finding the awkward method to switch the BB information.

In conclusion, one can find that it is of significance to pay attention on the real-valued characteristic function based on the above experimental results. The real-valued models have the abilities to reduce the number of function evaluations to nearly four times on the third test problem. Further experiments show that even using a optimal dynamic binary model does not outperform the optimal static real-valued model. In the next section, a method to learn the real-valued model is proposed, so one can apply to the unknown problems.

6. INTERACTION-DETECTION

Although the method to recombine chromosomes is proposed in Section 4, how to learn the optimal $\mathcal{R}$ is yet unknown. In this section, we propose a method to find a threshold for identifying the binary models with entropy based metrics. Then a method is proposed to acquire the real-valued models.

6.1 For Binary Models

Entropy [14] is the most commonly used metrics in EDAs to detect interactions among variables. ecGA, EBNA, BOA, hBOA and DSMGA are typical examples. The loss in entropy is mutual information (MI) in the case of two variables. The form of mutual information shows as

$$\mathbb{I}(X;Y) \quad = \quad \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \qquad (9)$$

where X and Y are two random variables. x and y denote the outcomes of these two random variables respectively. Note that if X and Y are independent, $p(x,y) = p(x)p(y)$, and hence $\mathbb{I}(X;Y) = 0$. One can use this metric to detect interactions among variables, and MI performs well on most cases.

According to Kleiter [7] and Hutter and Zaffalon [6], if two variables X and Y are independent, the sampled mutual information $\mathbb{I}_n(X;Y)$ calculated over n samples can be approximate as Beta distribution with mean $\mu \approx \frac{1}{2n}$ and variance $\sigma^2 \approx \frac{1}{2n^2}$. Both μ and σ^2 tend to zero when the sample size n is large.

Take an $l = m \times k$-bits decomposable problem for example, one could calculate the pairwise MI among all variables, $\binom{l}{2}$ totally. m is the number of sub-problem, and k is the problem difficulty. The number of the independent pairs $(\binom{l}{2} - m \times \binom{k}{2})$ is greater than the dependent pairs $(m \times \binom{k}{2})$ unless k is greater than $\frac{l+1}{2}$. In real world, the problem difficulty is bounded, so the number of the dependent pairs is no more than the number of independent pairs. If the population is infinite, the median of all $\binom{l}{2}$ pairwise MI among l variables is reasonably considered as independent pair with bounded difficulty.

For the sake of picking up a sampled MI $\mathbb{I}(X;Y)$ where X and Y are independent, median of all $\binom{l}{2}$ values is chosen. However, the chosen pair may not be independent because of the sampling noise with finite population. According to the population sizing by [20], the model accuracy is $1 - \frac{1}{m}$. In reality, one could randomly pick up 5 candidates from the all pairwise MI values. The median of these 5 chosen candidates is then represented as the pairwise MI of independent pair. The probability that the median-of-5 does not come from independent pairs can be calculated as follows. The ratio of the number of dependent pairs to the number of all pairs is approximately $\frac{1}{m}$. In the condition that the median-of-5 is not the MI from independent pairs, three of the five candidates should be from dependent pairs, so the probability that the median-of-5 is not the MI from independent pairs is $\frac{1}{m^3}$, which is small enough. Moreover, as mentioned before, the variance of the distribution of sampled MI tends to zero when the sample size is large, so it is more likely

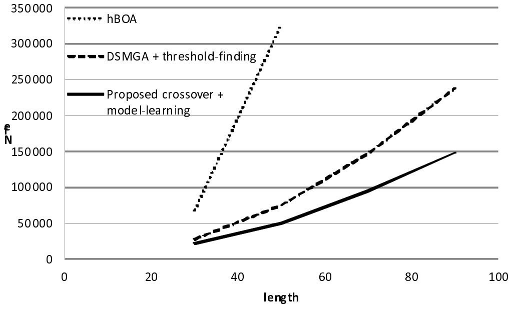

Figure 4: This figure shows that DSMGA with the threshold-finding method is comparable to hBOA and the prosed crossover with the model-learning method is better than DSMGA on the $(m,\ k)$-trap, where m is equal to 30 and k is equal to 5.

that the median is independent with larger population size. According to [6], once an sampled MI between two independent variables is given, one could calculate the μ and σ^2 of the approximated distribution.

According to [20], the decision error $1-F(x;\alpha,\beta)$ is bounded by $\frac{1}{\epsilon} \geq \Theta(m^3)$, where m denotes the number of BBs. Because m can not be determined for the unknown problems, one could use $\epsilon = \frac{1}{l^3}$ to derive the threshold x for interaction-detection.

The above procedure can be applied to EDAs so as to detect interactions among variables by classifying the pairwise mutual information into binary values. DSMGA with this threshold-finding method is comparable with hBOA. Figure 4 shows the experiments on $(m,\ k)$-trap.

6.2 For Real-valued Models

The previous section described how to calculate a threshold for binary models. Now we can simply utilize the knowledge of the distribution of MI to build real-valued models.

One could consider the values of $\mathcal{R}$ generated by this method as a confidence level. If the value of $F(\mathcal{M};\alpha,\beta)$ is nearly 100%, it is of high confidence to believe that this relationship is interacted, because it is nearly higher than all the independent MI from the estimated distribution. If the error rate, $1-F(\mathcal{M};\alpha,\beta)$, is larger than $\frac{1}{l^3}$, one could consider this pair as independent and map $\mathcal{R}$ to nearly 0.

In behalf of benchmarking the ability of the proposed crossover with this procedure, experiments are performed to test its performance and ability to solve problems. We implement the proposed crossover and the procedure to acquire the real-valued models with tournament selection and restricted tournament replacement (RTR) [4, 11]. For each test problem, 10 independent bisection runs are performed to find the minimal population size for 10 consecutively successes of finding the global optima. The averaged minimal population size of 10 bisection runs is used to yield the number of function evaluation that GAs need for solving the problems.

Table 4: The experimental results of 10 sub-problem of Equation 8 show the performance of different EDAs.

Methods	N_{fe}
Optimal real-valued model	19574.29
Optimal binary model	70923.66
DSMGA+threshold-finding	97470.12
The proposed XO+model-learning	36736.53
hBOA	33988.82

Figure 4 indicates that the prosed crossover with the method to acquire real-valued models is better than DSMGA even on the $(m,\ k)$-trap. The optimal model of the $(m,\ k)$-trap is considered as binary, but the proposed crossover has a better performance. As a result, there might exist some other benefits to utilize real-valued models, so the proposed crossover is better even on the $(m,\ k)$-trap.

Table 4 indicates that the proposed crossover with method to acquire real-valued models slightly lose to hBOA on the third test problem because the interaction-detection mechanism does not get the optimal real-valued model. Although the method to acquire real-valued models could get the value for the proposed crossover to use and perform well to some degree, the learned model is still not optimal.

In this section, a threshold-finding method is proposed for learning binary models, and another method is proposed for learning real-valued models. Although this method is incapable of acquiring the optimal real-valued models, the performance is comparable to hBOA. More experiments should be performed to investigate how to acquire the optimal real-valued models.

7. RECOMBINATION FROM PAIRWISE TO POPULATION-WISE

The proposed crossover mentioned in Section 4 produces two children from two parents at a time. This section provides a population-wise crossover method which utilizes the concept of the real-valued characteristic function because population-wise shuffle is known that having a better mixing rate than the one of uniform crossover.

This crossover method recombines chromosomes by shuffling population-wisely. First, find the locus (site of a gene) i that maximize $\sum_{j \in G} \mathcal{P}(X_i, X_j)$, and shuffle the genes on this locus population-wisely. Next, choose an unshuffled locus which maximizes $\mathcal{R}$ with the last shuffled locus, and decide whether to shuffle the genes on this locus according to the probability given by $1-\mathcal{R}$. For those unshuffled genes, assign the values which of the same parents with the genes on the last shuffled locus. In other words, some genes are BB-wisely shuffled and some genes are randomly shuffled.

With this crossover, one could utilize the real-valued models population-wisely. Up to now, the population-wise method works only with RTR, and the reasons are still being investigated. Table 5 shows that the population-wise crossover

Table 5: The experimental results of 10 sub-problems of Equation 8 show that the performance of the proposed population-wise crossover with RTR performs 2 times better than the proposed pairwise crossover with RTR. Both methods use the optimal real-valued models.

Methods	N_{fe}
pairwise XO + RTR	15997.31
population-wise XO + RTR	7052.58

with RTR performs 2 times better than the pairwise crossover on the third test function. This shows the potentialities of the population-wise crossover and it should be further researched.

8. CONCLUSIONS

This paper investigates the relations among variables and proposes a real-valued characteristic function that breaks the limitation of the binary characteristic function. Experiments demonstrate the benefits that the GAs with real-valued models perform better. This paper also propose a crossover which is able to utilize the real-valued models. Experiments show that the proposed crossover using real-valued models could reduce the number of function evaluation up to four times on the test problem. Moreover, this paper proposed an effective method to find a threshold for the entropy-based metrics and a method to provide real-valued models for the proposed crossover. Experiments show that the proposed crossover with the learned real-valued models works well.

This paper demonstrated that model building in EDAs can benefit from the utilization of relations with real-valued characteristic functions. As a first attempt, we used the beta distribution to estimate the relation. Even though our method may not be optimal, EDAs with those real-valued models consumed significantly fewer function evaluations on several test functions. More experiments need be conducted to investigate the property of real-valued relations so as the estimation of relation can be more accurate. Nevertheless, the idea of using a real-valued characteristic function may shed some light on developing next-generation EDAs.

9. REFERENCES

[1] R. Etxeberria and P. Larrañaga. Global optimization using bayesian networks. *Proceedings of the Second Symposium on Artificial Intelligence Adaptive Systems*, pages 332–339, 1999.

[2] D. E. Goldberg. Simple genetic algorithms and the minimal, deceptive problem. *Genetic Algorithms and Simulated Annealing*, pages 74–88, 1987.

[3] D. E. Goldberg, K. Sastry, and T. Latoza. On the supply of building blocks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 336–342. Morgan Kaufmann, 2001.

[4] G. R. Harik. Finding multiple solutions in problems of bounded difficulty. Technical Report IlliGAL Report No. 94002, University of Illinois at Urbana-Champaign, Urbana, IL, January 1994.

[5] G. R. Harik. Linkage learning via probabilistic modeling in the ecga. Technical Report IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Urbana, IL, February 1999.

[6] M. Hutter and M. Zaffalon. Distribution of mutual information from complete and incomplete data. *Computational Statistics & Data Analysis*, 48(3):633–657, 2005. to appear.

[7] G. D. Kleiter. The posterior probability of bayes nets with strong dependences. *Soft Computing*, 3:162–173, 1999.

[8] M. Munetomo. Linkage identification based on epistasis measures to realize efficient genetic algorithms. In *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*, CEC '02, pages 1332–1337, Washington, DC, USA, 2002. IEEE Computer Society.

[9] M. Munetomo and D. E. Goldberg. Identifying linkage by nonlinearity check. Technical Report IlliGAL Report No. 98012, University of Illinois at Urbana-Champaign, Urbana, IL, February 1998.

[10] M. Pelikan. Bayesian optimization algorithm: from single level to hierarchy. *Doctoral dissertation,*, University of Illinois at Urbana-Champaign, Champaign, IL, 2002.

[11] M. Pelikan and D. E. Goldberg. Hierarchical bayesian optimization algorithm = bayesian optimization algorithm + niching + local structures. pages 525–532. Morgan Kaufmann, 2001.

[12] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz. BOA: The bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, pages 525–532, 1999.

[13] G. C. Rota. The Number of Partitions of a Set. *The American Mathematical Monthly*, 71(5):498–504, 1964.

[14] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[15] M. Tsuji, M. Munetomo, and K. Akama. Modeling dependencies of loci with string classification according to fitness differences. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pages 246–257, 2004.

[16] M. Tsuji, M. Munetomo, and K. Akama. A crossover for complex building blocks overlapping. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 1337–1344, New York, NY, USA, 2006. ACM.

[17] R. Watson and J. B. Pollack. Hierarchically consistent test problems for genetic algorithms. 1999.

[18] T.-L. Yu, D. E. Goldberg, K. Sastry, C. F. Lima, and M. Pelikan. Dependency structure matrix, genetic algorithms ,and effective recombination. *Evolutionary Computation*, vol. 17, no. 4, pp. 595–626, 2009.

[19] T.-L. Yu, K. Sastry, and D. E. Goldberg. Linkage learning, overlapping building blocks, and systematic strategy for scalable recombination. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, pages 1217–1224, New York, NY, USA, 2005. ACM.

[20] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pelikan. Population sizing for entropy-based model building in discrete estimation of distribution algorithms. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 601–608, New York, NY, USA, 2007. ACM.