Higher-Order Linkage Learning in the ECGA

David Iclănzan
Babes-Bolyai University, Department of Computer Science
Str. Mihail Kogalniceanu no. 1, Cluj-Napoca, Romania
david.iclanzan@gmail.com

ABSTRACT
In this paper, we present a higher-order dependency identification algorithm for binary variables that employs a novel metric to detect linkages. The proposed method applies an entropy distillation operation over bounded groups of variables. Lack of increase in randomness signals an underlying statistical dependence between the inputs.

We incorporate the higher order linkage learning in the Extended Compact Genetic Algorithm (ECGA). Experiments show that the extended linkage learning enables the ECGA to correctly model and solve problems with bounded-order building-blocks that do not contain pairwise dependencies.

Categories and Subject Descriptors
I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search

General Terms
Algorithms, Design, Theory

Keywords
linkage learning, higher order statistics, entropy distillation

1. INTRODUCTION
Successful solving of nearly-decomposable problems by competent selecto-recombinative algorithms depends on the proper processing of the partial sub-solutions, also referred as building-blocks [5]. Estimation of Distribution Algorithms (EDAs) [14] [17] [10] or probabilistic model-building genetic algorithms (PMBGAs) [20] use interaction detection techniques designed to decompose the boundedly difficult problems on the fly by adapting the linkage model.

Always finding an accurate model is not a trivial endeavor as exploration of higher order dependencies is a computationally expensive task being exponential in the size of interactions covered. A common approach and efficient computational shortcut employed by linkage learning methods is to begin the construction of higher order models by exploiting pairwise dependencies. Therefore, decomposable problems lacking or exhibiting only a low degree of pairwise dependencies are hard for current linkage learning mechanisms and consequently can be very hard to optimize [10] [1] [2].

The “innovation time” (linkage learning) of recombinative algorithms must be less than the “takeover or convergence time” (mixing and decision making between competing building-blocks) [6]. Thus, the precision of the linkage model in early iterations is highly critical for problems where the repeated mixing of sub-solutions is required [10]. Without a proper linkage model, the takeover and convergence mechanism will operate on an incomplete set of building-blocks, leading to allelic premature convergence.

In this paper we introduce a novel higher order interaction detection technique for binary variables and incorporate it in the Extended Compact Genetic Algorithm (ECGA) [3][7]. Our method exploits the property of the exclusive or (XOR) operator to produce randomness from a non-deterministic source. The method searches for groups of variables where entropy distillation does not occur, signaling a non-determinism in the source - statistical dependence between the variables.

The ECGA with higher-order linkage learning is used to solve instances of problems containing bounded-order building-blocks exhibiting pairwise allelic independence.

Albeit a costly search for the groups of variables must be performed, this approach can solve problems that are intractable by pairwise linkage exploitation. For other cases it can facilitate a fast convergence, requiring an order of magnitudes less objective function evaluations, than approaches that do not correctly model the higher-order dependencies.

The rest of the paper is organized as follows: in section 2 we discuss the issue of detecting higher-order linkages for functions that do not contain pairwise low-order dependences. Section 3 introduces a novel multivariate linkage detection method, where a bounded order search is performed for variables where the XOR operation does not lead to entropy distillation. Section 4 presents the ECGA and the incorporation of the new linkage detection mechanism. The test functions are described in section 5, while section 6 presents the experimental results. Conclusions and further work is outlined in section 7.

2. DETECTING GROUPS OF DEPENDENT VARIABLES
Binary problems of real interest may have many variables with complicated multivariate interactions among them. For these problem pairwise dependencies might be very small
or lacking altogether. Finding the adequate linkage model for problems containing such dependencies is a hard task. Prospective methods must combine an extensive higher order model search guided by a criteria that evaluates the quality of the model in rapport with the evidence, like the Minimum Description Length (MDL) principle [21] or Bayesian-Dirichlet metric [18].

2.1 Complexity of multivariate linkage detection

The complexity of the model determination is a product of the complexity of the search and candidate model evaluation. For problems where statistical dependence can only be detected by considering at least \( k \) variables, the search must enumerate at least all combinations of variables taken \( k \) at the time. Thus the run-time cost of the model determination will be:

\[
T(M_{(n,k)}) = O(n^k)T_E(m_k)
\]

where \( n \) is the number of variables and \( T_E(m_k) \) is the run-time cost of the model quality discriminant function.

2.2 Model discriminant function

An ordered tuple of binary random variables \( X = (X_1, X_2, \ldots, X_n) \) is independent if the joint distribution \( Pr(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \) is equal for all \( x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n \).
Relative entropy or Kullback-Leibler divergence [3] can be used to measure the “distance” between these two distributions:

\[
D_{KL}(p||q) = \sum_{x \in X} p(x) \log_2 \left( \frac{p(x)}{q(x)} \right)
\]

Measuring the \( D_{KL} \) between the observed joint distribution of some variables and the product of independent joint distribution one can measure the information gain by considering a group of variables linked.

The complexity of calculating \( D_{KL} \) is exponential in \( k \) with a base equal to the cardinality of the random variables. Thus, for the binary case \( T_E(m_k) = O(2^k) \).

In the following we consider ways in which the complexity of the model discriminatory function can be heavily reduced for the binary case.

3. XOR BASED MULTIVARIATE LINKAGE DETECTION

Most practical sources of randomness, be it hardware or pseudo-random number generators, exhibit a certain level of imperfection or bias. A perfectly random bit has an entropy of one bit and bias of 0. To obtain a highly random bit, there are algorithms that combine multiple streams of imperfect random bits, each with entropy less than one, to create a single bit with entropy one and bias 0. This process is called entropy distillation or entropy extraction.

3.1 Exclusive OR and entropy distillation

Exclusive OR (XOR, also denoted by \( \oplus \)) is commonly used to reduce the bias from imperfectly random bits, provided that the random bits are statistically independent.

The reduction in bias by repeatedly applying the XOR on non-deterministic inputs can be computed using the Piling-Up Lemma [15].

**Lemma 3.1.** Let \( X_i \) for \( i \in \{1, n\} \) be statistically independent random binary variables, where \( p_i \) is the probability that \( X_i = 0 \) and \( \epsilon_i = p_i - 1/2 \) are the biases. Then the probability that \( X_1 \oplus X_2 \oplus \ldots \oplus X_n = 0 \) is

\[
\frac{1}{2} + 2^{n-1} \prod_{i=1}^n \epsilon_i
\]

Note that as biases \( \epsilon_i \in [0, 1] \) their product is a monotonically decreasing function. If any of the \( \epsilon \)'s is zero, that is, one of the binary variables is unbiased, the resulting probability function will be unbiased. Also, performing an XOR with a constant variable having \( p_i = 0 \) or \( p_i = 1 \) i.e a maximum bias of \( \epsilon_i = \pm 1/2 \) will not reduce the bias.

In our algorithm we will apply the XOR operation in a sequential manner, performing in each step the operation between a variable \( X_i \) and the result \( Y = X_1 \oplus X_2 \oplus \ldots \oplus X_{i-1} \) of the repeated XOR up to that variable \( i \). Therefore, we take a closer look on the expected result of XOR for two variables.

**Lemma 3.2.** If \( X \) and \( Y \) are independent random binary variables with expectations \( E(X) = \mu \) and \( E(Y) = \nu \) then

\[
E(X \oplus Y) = \mu + \nu - 2\mu\nu
\]

\[
= \frac{1}{2} - 2(\mu - \frac{1}{2})(\nu - \frac{1}{2})
\]

**Proof.** Following from the logical table of the XOR, for two bits \( a \) and \( b \), \( a \oplus b \) equals 1 if \( a = 0 \) and \( b = 1 \) or if \( a = 1 \) and \( b = 0 \).

Thus, \( E(X \oplus Y) \) can be written as

\[
E(X \oplus Y) = (1 - \mu)\nu + \mu(1 - \nu)
\]

\[
= \mu + \nu - 2\mu\nu
\]

\[
= \frac{1}{2} - 2(\mu - \frac{1}{2})(\nu - \frac{1}{2})
\]

\[
= \frac{1}{2} - 2(\mu - \frac{1}{2})(\nu - \frac{1}{2})
\]

☐

3.2 XOR based dependency detection

The Piling-Up Lemma is successfully used in linear cryptanalysis to construct linear approximation to the action of non-linear block ciphers. In this application, the \( X_i \)'s are approximations to the substitution-boxes of block ciphers for which the biases are trivial to measure. The attack relies on performing a costly search for finding combinations of input and output values that have very high biases i.e probabilities very close of zero or one.

In our vision a similar approach can also be used for linkage detection: a search is performed to find groups of variables for which the actual probability mass of the result obtained by performing the XOR greatly differs from the value predicted by the Piling-Up Lemma. For these cases the high bias must come from the fact that the assumption of non-determinism is not satisfied. Thus, there is an underlying statistical dependence between the inputs. If the absolute
The presented backtracking algorithm often recomputes large subtrees. However, trying to cache partial results in order to avoid recomputation would quickly hit the available memory barrier for large problems.

In a post processing step, in order to minimize disruption, we process the resulting building-block list as an elementary building-block. Thus, for 10 variables and remove the nodes containing building-blocks that are not relevant or not.

The algorithm explores all combination of variables up to level $k$. As the XOR operation is associative, each combination is generated and analyzed only once. This is achieved by enforcing an increasing order of the elements in the stack. Hereby, each dependent group of variables is detected and added to the linkage model only once.

The proposed metric has a great complexity advantage, as performing $k$ consecutive XOR operations in linear.

In order to enable the solving of boundedly difficult problems, we must be able detect multivariate dependencies from $n$ variables up to order $k$ ($k < n$). For this we perform a search over all combinations up to order $k$ and we use the measure in the previous subsection to detect dependencies.

The notations, parameters and variables used in the search algorithms are detailed in the following:

- The “:” denotes the colon operator as used in Matlab or Octave colon notation for selecting ranges of a dimension. Thus, $X(:,i)$ denotes the $i$th column of $X$.
- $n$, $k$ - $n$ is the number of variables while $k$ is the size limit on the order of dependencies explored.
- $st$, $idx$ - represent the stack and the actual level in the stack, variables needed for the recursive backtracking.
- $X$ - is a binary matrix of size $m \times n$, where $m$ is the number of observations. Thus each column of $X$ corresponds to one of the $n$ variables.
- $pX$ - is a vector containing the approximated probability mass functions of the $n$ variables. $pX(i)$ is just the number of ones in the $i$th column of $X$, divided by the number of observations $m$.
- $thr$ - is a predefined threshold used to decide if the detected amount of dependency should be deemed relevant or not.
- $LinkageModel$ - is a list that collects the detected building-blocks.
- $xr$ - is a binary matrix of $m$ rows and $idx$ columns. Each column $i$ contains the result of applying repeatedly XOR to the variables contained in $st$ up to the level $i$. For example, if $idx = 4$, thus the stack contains for values $st = 2, 6, 7, 9$, then the first column of $xr$ just equals $X_{st(0)} = X_2$ as there are no lower levels to XOR with. The second column will be $X_2 \otimes X_6$, the third $X_2 \otimes X_6 \otimes X_7$ and finally, the last column will be the result of $X_2 \otimes X_6 \otimes X_7 \otimes X_9$. There is no need to recompute the entire series of XOR's at each level, as for columns greater than one the $xr(:,i) = xr(:,i-1) \otimes X(:,st(i))$ recursive relation holds.
- $xi$ - is a vector of length $idx$, containing for each level $i$ the expected probability mass function of the result of applying repeatedly XOR to the variables contained in $st$ up to the level $i$ under the assumption of statistical independence.
- $f$ - is a vector of length $idx$, containing for each level $i$ the actual probability mass function of the result of applying repeatedly XOR to the variables contained in $st$ up to the level $i$.
- $diff$ - is the difference between the expected and actual probability mass functions of the result.

3.3 Searching for the linkage structure

In order to enable the solving of boundedly difficult problems, we must be able detect multivariate dependencies from $n$ variables up to order $k$ ($k < n$). For this we perform a search over all combinations up to order $k$ and we use the measure in the previous subsection to detect dependencies.

The proposed search algorithm is presented in Fig. 1.

Figure 1: The proposed search implemented as a recursive backtracking algorithm.

value of the resulting bias is greater than a threshold, then we will consider the variables as linked in our linkage model.

The proposed metric has a great complexity advantage, as performing $k$ consecutive XOR operations in linear.

While the approach is efficient in detecting multivariate dependencies, we still have to perform an ample search to find the groups of variables that are linked.

function
FindLinkages(st,idx,k,n,xr,xi,
f,X,pX,thr,LinkageModel)
{
  /* compute start index, and initialize XOR results on the 1'st level*/
  if (idx == 0)
    {
      si = 0; xr(:,0) = X(:,0); xi(0) = pX(0);
    }
  else
    si = st(idx-1) + 1;
  for (i = si; i < n; i++)
    {
      st(idx) = i;
      if (idx > 0)
        {
          // compute XOR result
          xr(:,idx) = xr(:,idx-1) \otimes X(:,i);
        }
      if variables are independent
        this should be the expected nr. of 0's as of Eq. ?? */
      xi(idx) = xi(idx-1) + pX(i) - 2*xi(idx-1)*pX(i);
      // compute the actual nr. of 0's
      f(idx) = nrZerosIn(xr(:,idx));
      f(idx) /= nrRowsIn(X);
      // compute the difference
      diff = xi(idx) - f(idx);
      /* if it is significant, retain the interdependent variables*/
      if (|diff| > thr)
        LinkageModel.Add(st,idx);
    }
    if (idx < k)
      // recursive call
      FindLinkages(st,idx+1,k,n,xr,xi,
f,X,pX,thr,LinkageModel);
}
(3,5,7,9) the post processing eliminates (1,3,5) and (5,7,9) as they are subsets of (1,3,5,7) respectively (3,5,7,9). Also we add (2) as an elementary building-block, as it is not contained in any linkage-group of the model. The resulting linkage model is \{(0,4,6,8),(1,3,5,7),(3,5,7,9),(2)\}.

In the following section we detail how the proposed higher order linkage detection can be incorporated in the ECGA.

4. THE EXTENDED COMPACT GENETIC ALGORITHM

The ECGA [7] is a multivariate extension of the Compact Genetic Algorithm [5] based on the key principle that learning a good probability distribution of the population is equivalent to the linkage learning process. The measure of a good distribution is quantified based on minimum description length (MDL) models. MDL is pilled on the concept that any regularity in a given set of data can be used to compress the data. The best hypothesis for a given set of data is the one that leads to the largest compression. Consequently, a tradeoff between model accuracy and complexity must be found.

MDL restriction reformulates the problem of finding a good distribution as minimizing both population representation (population complexity \(C_p\)) and the cost of representing the model itself (model complexity \(C_m\)). Hence the combined complexity criterion \(C_c\) to be minimized is given by:

\[
C_c = C_p + C_m
\]

The probability distribution used by the ECGA belongs to the Marginal Product Model (MPM), a class of probability model. Subsets of variables can be modeled jointly as partitions, providing a direct linkage map. Partitions together with the products of marginal distributions over them they form the MPMs.

The MPM concept is illustrated in Table 1 over a 3 bit problem with [1,3], [2] as partitions. The first and third bit are jointly distributed while variable [2] is independent. The compound partition [1,3] can have four settings: \{00, 01, 10, 11\}. The probability distribution for the partitions is given by the frequency of the individuals in the population with those bit values. Starting from a random population, the ECGA applies the process of evaluation, selection, MPMs based model-building and sampling until a halting criterion is met.

In its model-building phase, the ECGA greedily searches the space of possible partitions guided by the \(C_c\), evaluating all pairwise partition merges and always retaining the best one until no more improvements can be made. Given a partition configuration, their probability distribution are estimated by counting the frequencies of each different partition setting in the population.

The model building process is outlined in Algorithm 1. The method has \(O(n^3)\) complexity over the combined complexity criterion evaluations as line 5 iterates over pairs of variables. This can be intuitively seen as \(\binom{2}{2}\) has complexity \(O(n^2)\). Implementing a cache to avoid recomputation of MPMs, can decrease the model by one factor.

Enabling the higher order dependency search in the ECGA is simple: we only need to replace the initial model \(m\) formed by all variables as independent partitions, with the building-block list \(\text{LinkageModel}\) obtained by running the function \text{FindLinkages}\ described in Fig. 4 and performing the post processing step. This adds an \(O(n^3)\) overhead to the algorithm.

Another effect of the higher order dependency search is that the initial model will no longer be restricted to partitions, allowing an initial building-block setup containing overlaps. Therefore, when sampling overlapping building-blocks, in order to make the effect of the potential disruption non-deterministic, the sampling mechanism enumerates the building-blocks in a random order.

5. TEST FUNCTIONS

5.1 Pairwise independent test functions

One of the simplest ways to design an allelic-pairwise independent function for the binary case is to assign the same fitness to the individuals with the same parity of unitation. The parity function is the generalization of the exclusive or (XOR), computing the number of 1’s in a string:

\[
\text{parity}(X) = \begin{cases} 
C_{\text{even}} & \text{if } u(X) \text{ is even} \\
C_{\text{odd}} & \text{otherwise} 
\end{cases}
\]

where \(u(X)\) is the unitation or bit count of string \(X\), and \(C_{\text{even}}\) and \(C_{\text{odd}}\) are constants.

Coffin and Smith [2] defined the concatenated parity function (CPF) and the concatenated parity/trap function (CP/TF) in their work.

CPF is defined as a concatenation of parity functions:

\[
\text{CPF}(X) = \sum_{i=0}^{m-1} \text{parity}(X_{ik},X_{ik+1}\ldots X_{ik+(k-1)})
\]

<table>
<thead>
<tr>
<th>(X_1, X_2)</th>
<th>(X_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P(X_1 = 0 \text{ and } X_3 = 0) = 0.0)</td>
<td>(P(X_2 = 0) = 0.25)</td>
</tr>
<tr>
<td>(P(X_1 = 0 \text{ and } X_3 = 1) = 0.5)</td>
<td>(P(X_2 = 1) = 0.75)</td>
</tr>
<tr>
<td>(P(X_1 = 1 \text{ and } X_3 = 0) = 0.5)</td>
<td></td>
</tr>
<tr>
<td>(P(X_1 = 1 \text{ and } X_3 = 1) = 0.0)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: MPM over 3 variables. \(X_1\) and \(X_2\) are linked together defining a joint distribution where \(X_1\) having the same values as \(X_2\) has probability 0.0. This, together with the information that \(X_2\) is three times more likely to be 1 than 0 helps focusing the search.
where \( m \) is the number of concatenated parity functions.

A more difficult problem can be derived from the parity function by concatenating deceptive traps with the CPF, resulting in the CP/TF:

\[
CP/TF(X) = \sum_{i=0}^{m-1} \begin{cases} 
\text{parity}(x_{ik} \ldots x_{ik+(k-1)}) & \text{if } i \text{ is even} \\
\text{trap}(x_{ik} \ldots x_{ik+(k-1)}) & \text{otherwise}
\end{cases}
\]

(9)

In concatenated trap functions \[9\], the fitness gradient leads search away from the solution to each block. Each block of length \( k \) is maximally rewarded if it is contains all but ones; for the other cases, the fitness of the block is directly proportional with the number of zeros, the string of all zeros being a strong local optima:

\[
\text{trap}(X) = \begin{cases}
1 & \text{if } u(X) = k \\
0 & \text{otherwise}
\end{cases}
\]

(10)

Certain EDAs can solve these problems in polynomial time, without a correct linkage model by randomly stumbling upon one of the global optima \[1\] \[2\]. However, the search is inefficient, requesting many objective function evaluations as the problem structure is not exploited. Competent optimization algorithms must solve boundedly difficult problems with modularity, hierarchy, and overlap quickly, reliably, and accurately \[6\]. Here, we focus on achieving these goals, therefore the correct modelling and sampling of the building-blocks is critical.

5.2 Pairwise weakly dependent test functions

Relying the model building just on pairwise dependency exploitation is not problematic only for the cases when these dependencies are lacking. A higher-order dependency might be present at pairwise level but the signal it provides might be weak and be suppressed by the exogenous noise coming from other building-blocks.

To study this phenomena we introduce a test function called \( \text{halfones} \), which rewards a building-block if it has half of it genes set to one. For odd length building-block sizes, we overcome the fact that the half is not an integer by taking the ceiling, rounded-up value of the length divided by 2; thus for these cases, the building-block must have exactly one more gene set to one than there are genes set to zero.

The function is formalized as:

\[
\text{halfones}(X) = \begin{cases}
1 & \text{if } u(X) = \lceil \frac{\text{length}(X)}{2} \rceil \\
0 & \text{otherwise}
\end{cases}
\]

(11)

where \( u(X) \) is the unitation or bit count of the string \( X \).

While pairwise dependencies are present between the variables of \( \text{halfones} \), their strength is inversely proportional with the length of the building-block and the number of other building-blocks.

6. RESULTS

6.1 Linkage detection performance

In a first step, we tested how the higher-order linkage measurement can be used to detect and quantify multivariate dependencies. We were interested in seeing if it can (1) reliably detect higher-order linkage groups that do not contain pairwise dependencies and (2) aid performance for cases where pairwise dependencies are present but weak.

For the first scenario we relied on the CPF as the parity function is allelic pairwise independent. For the second test objective we used a concatenated \( \text{halfones} \) function.

To account for the effect of both building-block length and building-block numbers, we varied the parameters in the following ranges: \( k \) from 5 to 20 by increments of 1, \( m \) from 10 to 200 by increments of 10.

For each pair of \( k \) and \( m \) we randomly generated a population of \( n = 10000 \) individuals with uniform distribution and selected the top 2500. Next we measured the average pairwise dependence \( pw(bb) \) between genes belonging to the same building-block and the average pairwise dependence \( pw(non – bb) \) between genes belonging to different building-blocks. We also measured the average bias \( b\text{XOR}(bb) \) obtained by performing XOR based entropy distillation on the genes belonging to the same building-blocks. Similarly, \( b\text{XOR}(non – bb) \) averages the bias obtained by performing XOR based entropy distillation on genes belonging to different building-blocks. When calculating the statistics for non related genes, to be able to trace the results, we averaged the results obtained from 10000 randomly generated blocks containing genes from different linkage groups.

For the CPF, as expected, there was no significant difference between \( pw(bb) \) and \( pw(non – bb) \) since the parity function is pairwise allelic independent. However, the XOR based entropy distillation signaled a high bias for the building-blocks, while for the non-linked groups of variables the bias was less than 0.009. The biases on CPF for all \( k \)-s and \( m \)-s are presented in Fig. 2. For the building-blocks, the bias is almost as high as 0.2 on the easiest case. When the problem size is increased, this value decreases to about 0.04.

In the case of concatenated \( \text{halfones} \) there are pairwise dependencies, thus the values for \( pw(bb) \) and \( pw(non – bb) \) differ. Their value for the different pairs of \( k \) and \( m \) is presented in Fig. 3. As seen in the figure, the pairwise dependency measure might be useful only for limited problems sizes, where both \( k \) and \( m \) are small. Once these values increases the pairwise dependency measure quickly looses its ability to discriminate the real underlaying linkages. Again, as depicted in Fig. 4, the XOR based entropy distillation bias measure can discriminate between linked and non-linked groups of variables. Albeit now the bias for the hardest instance is about 0.02, this approach is much more robust to the exogenous noise coming from the other building-blocks in the problem. Increasing the selection pressure can reduce the inter building-block noise. For example by selecting only the top 500 out of 10000 individuals, the bias for the hardest problem increases from 0.02 to 0.0425.

From Fig. 2 and 3 results that the XOR based entropy distillation bias measure is mostly insensitive to the building-block size, as for the same \( m \) values the bias is almost constant for all \( k \)-s.

We conclude that the proposed linkage detection mechanism can detect allelic pairwise independent building-blocks and it also useful in amplifying and aggregating low pairwise dependency signals.

6.2 ECGA scalability

We conducted scalability experiments on the shuffled CPF and CP/TF for the ECGA with higher order linkage learning, where the minimal population sizes were determined with the bisection method \[19\], requiring 50 out of 50 suc-
cessful runs. As already discussed, we put emphasis on the correct linkage learning, thus we counted as successful run those instances where the global optima is found and the correct building-block structure is detected. A model is considered correct if the dependent alleles are modelled in same linkage groups.

The bias threshold used in the higher order linkage learning was set to 0.04;

Similarly to previous works \cite{1,2} we set the block size \( k = 5 \), \( C_{\text{odd}} = 5 \) and \( C_{\text{even}} = 0 \) for the parity function. In the ECGA we used truncation selection, where the top quarter of the population was selected for learning.

The ECGA with higher-order linkage detection easily solved all instances of the test suite detecting also the correct building-blocks. The minimal population sizes for the two test functions are presented in Fig. 5 while the scaling of objective function evaluations is presented in Fig. 6.

The inclusions of the trap function makes the task of linkage learning harder, making more difficult the separation of different building-blocks. Therefore, the minimal population sizes and number of objective function evaluations are significantly larger in the case of the CP/TF.

As it exploited the problem structure, the minimal population sizes and number of objective function evaluations are a magnitude of order smaller than those reported for EDAs that solve these problems without correct linkage structures. On the CP/TP with \( m = 100 \) building-blocks, the ECGA without higher-order linkage learning, requires more than double in population size and several hundred-thousand objective function evaluations to find the global optima \cite{1}.

For these problems, the computational investment for a higher-order linkage model yields a high return by requiring a much smaller number of objective function evaluations. This gain is especially important for complex problems where the objective functions can be very slow to evaluate, restricting the optimization process to only a few hundred or thousand objective calculations \cite{1,2}. Besides the increased speed in convergence, another important advantage is that the correct linkage structure is identified and explicitly modeled. This description of higher-order dynamics exhibited by the problem can help advancing knowledge by providing valuable insights about the problems and their structure.

7. CONCLUSIONS AND FUTURE WORK

Current multivariate EDAs rely on computationally efficient pairwise linkage detection mechanisms to identify higher order linkage blocks. This may lead to poor performance for problems where such pairwise dependencies are not available for exploitation.

In this paper we introduced a higher-order linkage detection mechanism, where we search for groups of variables from which randomness can not be extracted, signaling an underlaying dependency. Albeit we must perform the highly combinatorial search, the complexity of measuring the entropy distillation is just linear.

Experiments performed on various allelic-pairwise independent of bounded order confirms that the method is able to learn the correct linkage groups for these hard functions and optimize them efficiently. To best of our knowledge, so far no other EDA applies higher order model search, thus no method is known to infer the correct linkage model for these types of problems.

8. REFERENCES

\cite{1} S.-C. Chen and T.-L. Yu. Difficulty of linkage learning in estimation of distribution algorithms. In *Proceedings of the 11th Annual conference on Genetic...
Figure 6: Objective function evaluation scalability of the ECGA with higher order linkage learning on the CPF and CP/TP.
Figure 2: Average bias of the XOR based distillation for groups of dependent and independent variables on the CPF.

Figure 3: Average pairwise correlation coefficient for variables belonging to groups of dependent and independent variables on the concatenated halfones.

Figure 4: Average bias of the XOR based distillation for groups of dependent and independent variables on the concatenated halfones.