Hierarchical Allelic Pairwise Independent Functions

David Iclănzan
david.iclanzan@gmail.com
Department of Electrical Engineering
Sapientia Hungarian University of Transylvania
540485, Tg-Mureş, CP 4, OP 9, Romania

ABSTRACT

Current multivariate EDAs rely on computationally efficient pairwise linkage detection mechanisms to identify higher order linkage blocks. Historical attempts to exemplify the potential disadvantage of this computational shortcut were scarcely successful.

In this paper we introduce a new class of test functions to exemplify the inevitable weakness of the simplified linkage learning techniques. Specifically, we show that presently employed EDAs are not able to efficiently mix and decide between building-blocks with pairwise allelic independent components. These problems can be solved by EDAs only at the expense of exploring a vastly larger search space of multivariable linkages.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search

General Terms

Algorithms, Design, Theory

Keywords

linkage learning, hierarchical functions, pairwise independence

1. INTRODUCTION

Estimation of Distribution Algorithms (EDAs) [14, 19, 17] or probabilistic model-building genetic algorithms (PMBGAs) [23] extend the classical framework of Evolutionary Algorithms (EAs) [10] with a novel approach consisting in learning and exploiting information from selected individuals. Global statistical information is extracted from promising solutions and used to infer a probabilistic model. New solutions are then sampled from the probability distribution model in order to generate the next population.

EDAs can be grouped into various classes based on their linkage learning mechanisms. The most basic ones like the PBIL [1], cGA [13] and UMDA [19] assume all the variables in the problem are independent, thus they use univariate probability distributions and do not require any linkage learning mechanism. Bivariate EDAs, like MIMIC [2] and BMDA [24], rely only on the pairwise dependencies between variables. Finally, multivariate EDAs based on the maximum entropy principle such as ECGA [12], BOA [21] and its extension hBOA [20], FDA [18], EBNA [7], DSMGA [35] and DSMGA++ [34], search for the optimal distribution model among multiple variables. Having the greatest expressing power, the later, multivariate class of EDAs is the most suited for solving nearly decomposable problems [29].

Multivariate EDAs, which must exploit the linkage information to estimate joint probabilities and construct the models, generally rely on the exploitation of pairwise linkages to construct the higher order models, as the exploration of the whole possible search space of multivariable linkage is a computationally very expensive task – exponential in the size of interactions covered. For example, ECGA builds the marginal product models by merging building-blocks according to the Minimum Description Length principle [26]. As only the merging of two building-blocks is considered, in the very beginning of model building the method can only exploit pairwise dependency. BOA also relies on pairwise dependency as only one addition or deletion of an edge is considered in each step.

As 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) [11], the precision of the linkage model in early iterations is critical.

There is a vast literature on the influence of model accuracy on the performance of EDAs, considering both the underfitting (missing important dependencies) and overfitting (spurious dependencies) of the linkage models [33, 25, 3, 4, 16, 15, 27, 32, 6].

Although it is shown that both types of model inaccuracies more or less hinder EDAs performance, so far no problem could be pointed out for which these methods completely fail. Studies aimed at finding such problems [3, 4], used allelic pairwise independent functions like the parity function, where variables appear to be independent when observing only two of them, albeit there is a strong dependency among bigger groups of variables. Lacking pairwise dependency that can be immediately exploited, such functions were believed to be difficult for EDAs. However, ex-

GECCO'11, July 12–16, 2011, Dublin, Ireland.

periments demonstrated that the parity function and the other proposed pairwise allelic independent functions could still be solved by specific EDAs within a polynomial number of function evaluations to the problem size [3].

This paper focuses on settling the open question, whether or not the heavy reliance on pairwise exploitation in present EDAs implies a weakness on linkage learning for some nearly decomposable problems – a class of problems for which EDAs are considered well-suited.

We first review previous work regarding the performance of EDAs on pairwise allelic independent functions and present our opinion of why these proposed problems do not pose a real challenge to EDAs. Then, following these observations we introduce another allelic pairwise independent function and experiment on it. Finally, we have some analysis and conclusions on the empirical results and we also investigate under which conditions can be the proposed problem efficiently solved.

2. PREVIOUS WORK

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 operator and the Walsh transform, being:

$$parity(X) = \begin{cases} C_{even} & \text{if } u(X) \text{ is even} \\ C_{odd} & \text{otherwise} \end{cases} \tag{1}$$

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

Having the parity function as the elementary building-block, Coffin and Smith [4] described 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:

$$CPF(X) = \sum_{i=0}^{m-1} parity(X_{ik}, X_{ik+1} \dots X_{ik+(k-1)}) \tag{2}$$

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:

$$CP/TF(X) = \sum_{i=0}^{m-1} \begin{cases} parity(x_{ik} \dots x_{ik+(k-1)} & \text{if } i \text{ is even} \\ trap(x_{ik} \dots x_{ik+(k-1)} & \text{otherwise} \end{cases} \tag{3}$$

In concatenated trap functions [5], 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:

$$trap(X) = \begin{cases} k & \text{if } u(X) = k \\ k - 1 - u(X) & \text{otherwise} \end{cases} \tag{4}$$

Coffin and Smith [4] found that the hBOA, scales exponentially on these problems. As hBOA is a state-of-the-art multivariate EDA which have shown robust and reliable performance on other nearly decomposable problems, the authors concluded that CPF exemplifies the class of hard problems for EDAs. However, in a later study Chen and Yu [3] demonstrated that both the CPF and CP/TF can be solved by some EDAs in polynomial time. The CPF is solved by the cGA and ECGA, while CP/TF can be addressed only by the later method. They also empirically demonstrated that the overfitting phenomenon may be accountable for the exponential scalability of hBOA on CPF, as their experiment of artificially injecting spurious linkages into the ECGA resulted in an exponential runtime. Another work [6] had shown that the exact model building in Bayesian network based EDAs, results in correct linkage detection and efficient solving of the CPF.

j	l = 2	l = 3
0	1111	11111111
1	1010	10101010
2	1100	11001100
3	1001	10011001
4		11110000
5		10100101
6		11000011
7		10010110

Table 1: Walsh codes

In their attempt of designing a problem that deceives linkage learning and consequently allelic convergence as well, Chen and Yu [3] constructed a harder test problem, with fewer global optima, that is based on Walsh codes.

Walsh functions [8, 9] are defined as:

$$\psi(x) = \begin{cases} 1 & \text{if } u(x \wedge j) \text{ is even} \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

where x and j are bit-strings of length l and '$\wedge$' denotes bit-wise logical AND operation.

The i-th bit of a Walsh code W_j of length 2^l is given by the following formula:

$$W_j[i] = \frac{1}{2}(\psi(i-1) + 1) \tag{6}$$

which converts analog -1 in Walsh functions to digital bit 0.

Table 1 shows the Walsh codes of length 2^2 and 2^3. Walsh codes are allelic-pairwise independent except the first bit, which is fixed and so are their complements. For any block size $3 < k < 2^l$, one can choose from each W_j an almost arbitrary combination of k bits (the first, constant bit should not be considered), to obtain a group of allelic-pairwise independent strings. Since the number of Walsh codes of length k is much smaller than 2^{k-1} - the number of binary strings of same length, containing an even number of ones, Chen and Yu [3] reasoned that a concatenated function based on these functions should be much harder to solve. However, their experimental results showed that cGA still solves the proposed concatenated Walsh code function in polynomial time. The authors concluded, that the answer to the question regarding the existence of problems which require the $O(l^2k)$ time to build a exact distribution model, must be postponed.

2.1 EDAs and Problem Difficulty

cGA does not model linkages groups and in [3] it was shown that the ECGA is not able to learn the linkages in parity subproblems. However, these methods can solve these pairwise allelic independent functions as they have a quite large number of global optima. For example, half of the configurations are optimal in the case of the parity function. Therefore, CPF has $2^{m(k-1)}$ optima, where m is the number and k is the size of the parity subfunctions. The Hamming distance between any configuration and the closest global optima can not exceed m and on average a randomly generated binary string would require $m/2$ successful mutations.

Because the subproblems are completely separable, no matter which subproblem converges slower, it always requires only polynomial time. In this way the Walsh code function, where there are fewer optima in each block, can also be solved.

We think that these functions are not suited to exemplify hard problems for EDAs, because they target only one kind of difficulty, the one regarding linkage learning. Goldberg [11] identifies the three traits for a competent recombinative framework. Competitive algorithms should:

1. Ensure the identification (supply) of raw subsolutions or building-blocks.

2. Make good decision between competing building-blocks.

3. Mix subsolutions efficiently so the discovery of even more fit, higher order building-blocks happens before irreversible decisions are made at (2). Put in another way, "innovation time" should be less than "takeover or convergence time".

As multivariate EDAs try to incorporate all three traits, a proper test functions should challenge all these aspects. So far, the proposed functions targeted only (1), by making the identification of building-blocks difficult. We asses, that if hard linkage-learning is combined with te requirement to mix and decide between building-blocks, EDAs relying on pairwise independency exploitation will not be able to solve this kind of problems efficiently.

3. THE HIERARCHICAL PAIRWISE ALLELIC INDEPENDENT FUNCTION

By embedding pairwise allelic independent functions in a hierarchical structure, we can build a test function which is able to challenge the competitive recombinative framework at all three levels. Pairwise independence makes building-block identification hard, while the hierarchical structure demands both efficient mixing and correct decision making with regard to the building-blocks.

Although having a gross-scale building-block structure, hierarchical problems are hard to solve without proper problem decomposition as the blocks from these functions are not separable. The inherent difficulty of hierarchically decomposable problems arise from the fact that a building-block can have multiple context-optimal settings. In this way, there is always more than one way to solve a (sub-)problem [30], leading to the separation of building-blocks "fitness" i.e. contribution to the objective function from their meaning. This conceptual separation induces the non-linear dependencies between building-blocks: providing the same objective function contribution, a building-block might be completely suited for one context whilst completely wrong for another one. Therefore, the fitness of a building-block can be misleading if it is incompatible with its context. However, the contribution of the building-blocks indicate how can the dimensionality of the problem be reduced by expressing one block in a lower level as one variable in the upper level. The formation of higher order building-blocks from lower level ones, reduces the problem dimensionality. If a proper niching is applied and the promising subsolutions are kept until the method advances to upper levels (where a correct decision can be made) the hierarchical difficulty can be overcome. Methods employing multivariate linkage-learning and a form of niching, like the ECGA, hBOA, DSMGA++ can efficiently solve hierarchical problems like HIFF [30] and HXOR [31], where the basic building-blocks are the bits of the binary string.

In our proposed setup, the basic building-blocks for the hierarchical structure are based on Walsh codes of length k. This set of elementary building blocks is denoted by W_j^k. We design an another set of competing building blocks $\overline{W}_j^k$, which are obtained by bit-wise negation of the elements from W_j^k. From the definition of Walsh codes results that the elements from these sets are orthogonal, thus allelic-pairwise independent.

The generalized Walshcode rewarding functions is defined as:

$$\omega(x) = \begin{cases} 1 & \text{if } x \in W_j^k \\ 1 & \text{if } x \in \overline{W}_j^k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Using $\omega(x)$ we define a completely separable problem, which concatenates m Walsh code based subproblems:

$$CWCF(X) = \sum_{i=0}^{m-1} \omega(X_{ik}, X_{ik+1} \dots X_{ik+(k-1)}) \tag{8}$$

The $CWCF$ is hard for current model building strategies in EDAs, thus it should present an adequate challenge to step (1) of the recombinative framework - the identification of building-blocks.

For challenging step (2) and (3) we employ a hierarchical scheme based on the XOR relation, where joint building-blocks are only rewarded, iff the two composing valid building-blocks are bit-wise complemental.

However, the Walsh code based subfunctions can not be directly inputted in the hierarchical scheme. Each subfunctions may have a large number of optimal settings, which could lead to an exponential number of competing schemata, which in turn imply the need for an exponentially growing population. To avoid this, a mapping function is introduced, which reduces the number of competing schemata to two:

$$\xi(x) = \begin{cases} 0 & \text{if } x \in W_j^k \\ 1 & \text{if } x \in \overline{W}_j^k \\ - & \text{otherwise} \end{cases} \tag{9}$$

We assume ξ to be a vectorized function, where for multiple inputs it outputs a string, where the i-th character is the mapping for the i-th input.

Thus, schemata from W_j^k map to 0, schemata from $\overline{W}_j^k$ map to 1, while all other schemata are considered invalid and are mapped as '-'. Mapping the solutions to $CWCF$, one obtains a string of length m, just one symbol for each subproblem and two competing initial schemata, 0 and 1. The mapped string is evaluated according to HXOR. Invalid blocks, coded with the '-' are not taken into account.

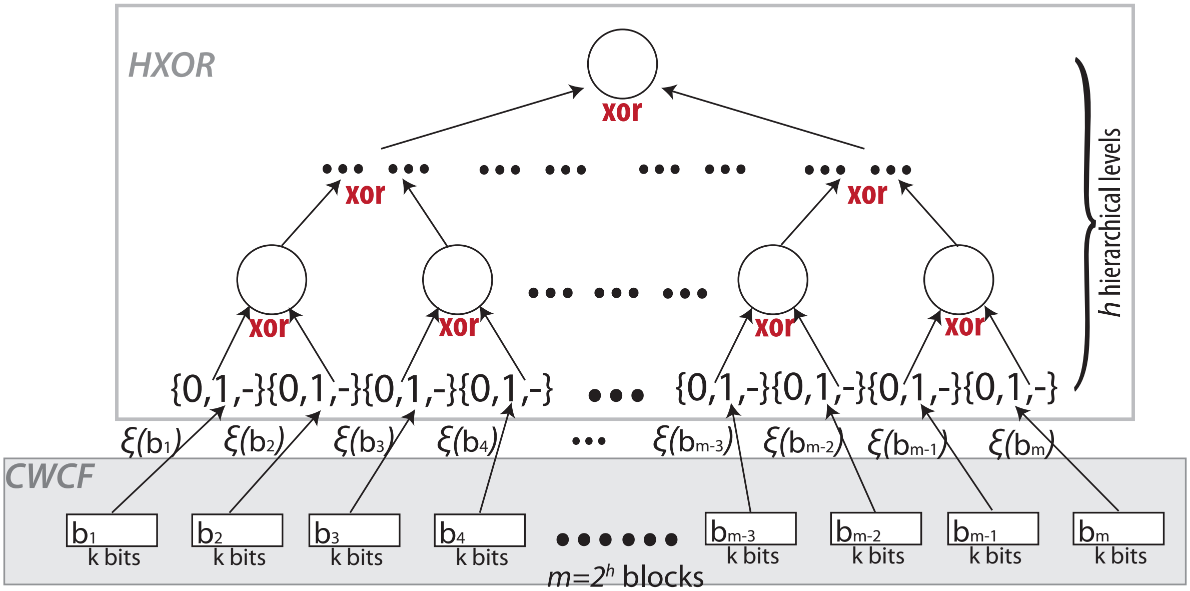

Figure 1: Hierarchical Pairwise Allelic Independent Function based on Walsh code subfunctions and HXOR.

Finally, we can formally define a hierarchical pairwise allelic function (HPAIF) as being:

$$HPAIF(X) = CWCF(X) + HXOR(\xi(X)) \qquad (10)$$

For a complete hierarchical evaluation, where at each level the number of building-blocks is halved, m must be a power of two. In this paper we experiment with HPAIF functions based on $m = 2^h$ Walsh code based subfunctions, $h = \{2, 3, 4\}$ hierarchical levels. The length k of the basic Walsh code subfunctions take values between 5 and 12.

HPAIF is visually depicted in Fig. 1.

The difficulty of HPAIF increases along both with the value of k and h. The fraction between W_j^k and 2^k diminishes as k increases, thus the task for step (1), discovery of good schemata, becomes harder and harder. The value h impacts the difficulty for step (2) and (3), good decision making and efficient mixing.

HPAIF is a hard problem if a correct linkage model is not used as there is a non-linear dependence between subfunctions, and without a correct mixing of building-blocks higher order configurations are very hard to discover.

The function has $2 * |W_j^k|$ global optima, where $|W_j^k|$ is the cardinality of the Walsh code set. This value is 8 for $k \in [5, 7]$ and 16 for $k \in [8, 12]$

4. PERFORMANCE OF ECGA ON HPAIF

4.1 The Extended Compact Genetic Algorithm

The ECGA [12] is a multivariate extension of the Compact Genetic Algorithm [13] 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 pillared 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:

$$C_c = C_p + C_m \qquad (11)$$

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 4.1 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 $\binom{n}{2}$ has complex-

Algorithm 1: Model-building in the ECGA

```
Build initial model $m$ where each variable is an independent partition;
repeat
    $m_{best} \leftarrow m$;
    foreach $[p, q]$ from the $\binom{|m|}{2}$ set of possible pair partitions of $m$ do
        Form new model $m'$ based on $m$ but with $p$ and $q$ merged into a joint partition;
        Evaluate combined complexity criterion $C_c(m')$;
        if $m'$ improves over $m_{best}$ then
            $m_{best} \leftarrow m'$;
    $m \leftarrow m_{best}$;
until No improvement was found;
```

ity $O(n^2)$. Implementing a cache to avoid recomputation of MPMs, can decrease the model by one factor.

4.2 Experiments

We conducted scalability experiments on concatenated shuffled HPAIF for ECGA with tournament size 8, where minimal population sizes were determined with the bisection method [22]. cGA was omitted, as it uses an univariate model, thus is not able to solve hierarchical functions.

We searched for population sizes up to the maximal size of 409600 (initial population size of 100 doubled 12 times), for wich 19 out of 20 trial runs are successful, in the bound of a maximum $N = 1000$ generations. If such population size could be determined using the bisection method, we ran ECGA 50 times for each problem size and averaged the results from the successful runs.

Experiments considered HPAIF functions with number of subfunctions $m = \{2^2, 2^3, 2^4\}$ and basic subfunction size $k = [5 - 12]$.

In a first step we investigated how hard is the proposed HPAIF function without the linkage learning task. We run a modified ECGA that is informed about the underlaying basic structure, with linkage groups for the m subfunctions hardwired. Knowledge about the hierarchical structure was not inputted. As the search space is sampled according to the correct basic linkages, the identification and mixing of building-blocks is much easier.

Multivariate EDAs can efficiently solve hierarchical problems. Thus, as expected, the basic structure injected, informed ECGA, solved each test instance quite easily. For the hardest test case, with $k = 12$ and $m = 32$ (total problem size of 384 bits) the bisection method determined a required populations size of 22400 with which the method averaged 198240 objective function evaluations until convergence.

$[X_1, X_3]$	$[X_2]$
$P(X_1 = 0 \text{ and } X_3 = 0) = 0.0$	$P(X_2 = 0) = 0.25$
$P(X_1 = 0 \text{ and } X_3 = 1) = 0.5$	$P(X_2 = 1) = 0.75$
$P(X_1 = 1 \text{ and } X_3 = 0) = 0.5$	
$P(X_1 = 1 \text{ and } X_3 = 1) = 0.0$	

Table 2: MPM over 3 variables. X_1 and X_3 are linked together defining a joint distribution where X_1 having the same values as X_3 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.

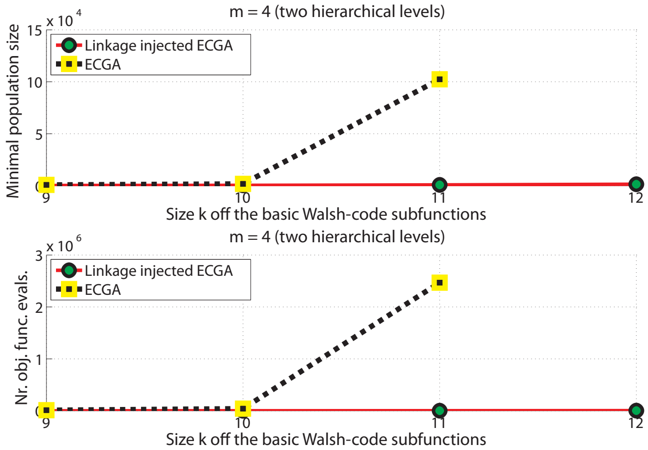

Figure 4: Scaling of the minimal population size and number of average objective function evaluation on HPAIF with *k={9,10,11,12}*, for the linkage informed ECGA and normal ECGA. Results are presented only for *m=4*, as for higher values the ECGA could not perform within the traced domains.

While the assumption of hardwired linkages is not realistic, this experiment helps in determining an absolute lower bound on the performance of ECGA. It also shows, that if linkages are detected in a timely manner, the efficient solving of HPAIF is manageable.

The scalability of ECGA regarding minimal population size on HPAIF for $k = \{5, 6, 7, 8\}$ and $m = \{4, 16, 32\}$ is depicted in Fig. 2. The scaling of the number of average objective function evaluations for the same k-s and m-s is presented in Fig 3. As a baseline for comparison, the performance of the linkage informed ECGA is also included.

Contrary to our initial expectation, for $k = \{5, 6\}$ the ECGA solved HPAIF for all hierarchical levels, with a scalability proportional to the one depicted by the informed ECGA. Without an adequate model, solving reliably HXOR with 4 hierarchical levels, 32 variables is not a trivial task. A closer look to the MPMs reported through the runs revealed that the ECGA is able to learn some of the good linkages on HPAIF. This is surprising, as it was believed that ECGA should be unable to learn the linkages between variables in the Walsh code subfunctions [3]. The explanation for this result will require a more in-depth analysis. Our first intuition is, that due to a random sampling bias, some Walsh codes based schemata are overrepresented in the selected individuals from the first generations. This invalidates the presumption of perfect pairwise orthogonality and makes possible the detection of some linkages.

For $k = 7$ the ECGA could successfully solve only the HPAIF with $m = \{2, 3\}$. For $m = 4$ the method was not able to attain reliably the global optima. While useful linkage groups were discovered in this case also, their efficient combinations was not possible. The formed MPMs were heavily biased toward few configurations which more often than not, did not contain complementary, most competing schemata. For $k = 8$ the method is only successful for $m = 4$,

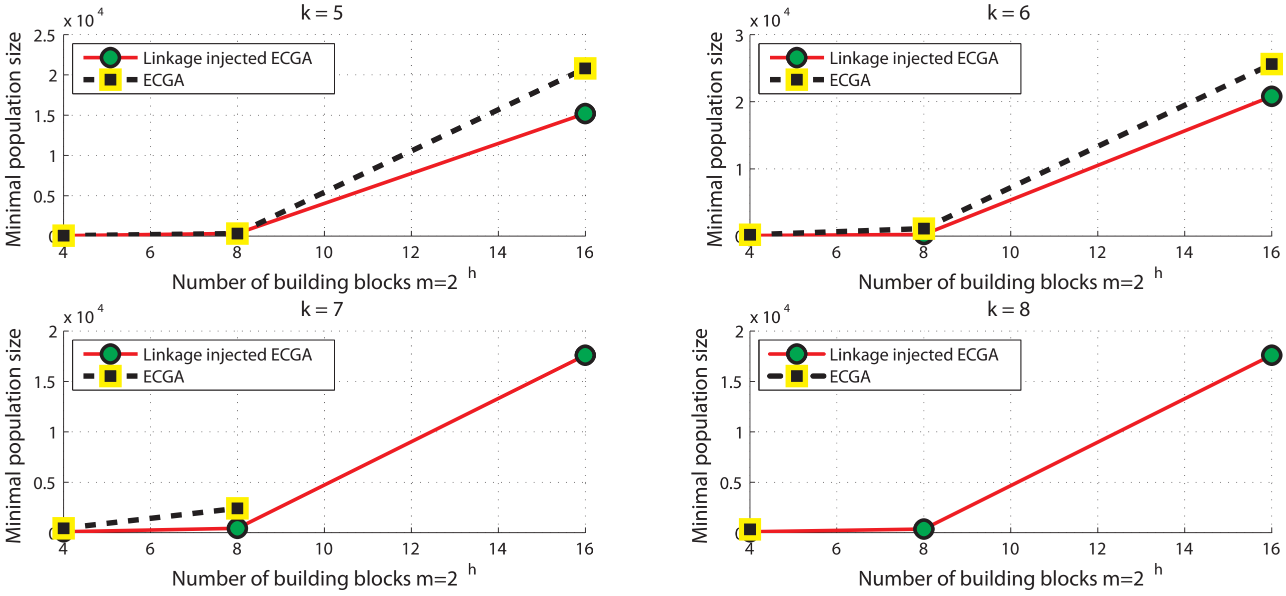

Figure 2: Minimal population sizes on HPAIF with k={$5,6,7,8$}, for the linkage informed ECGA and normal ECGA, found with the bisection.

where the building-block mixing requirements for the successful solving are low.

This trend holds for $k = \{9, 10, 11\}$, where the ECGA is only able to discover one of the optima for the smallest m. For $k = 12$ the method is not able to solve any of the test instances. Even with the minimum number of four subfunctions and with a maximum population size of 409600 the method only found a global optima twice in the 20 runs performed by the bisection procedure.

Performance of the ECGA with regard to the population size and average number of function evaluations, again using as a baseline the performance of the informed ECGA, is depicted in Fig 4. It can be observed that at $k = 11$ there is a complexity explosion, with a several orders of magnitude increase in the minimal population size and number of average number of objective function evaluations.

The difficulty of HPAIF is twofold and is controlled by the parameters k and $m = 2^h$. Bigger k-s stress a method ability to discover good building-blocks, while higher values for m require better and more performant building-block mixing and decision making. In our experiments the ECGA showed a remarkable and unexpected robustness in tackling bigger k-s. In the long run it could discover good schemata and sometimes even deduce parts of their underlaying linkage map. This is in line with the conclusions of previous experiments [4, 3], that found that it is difficult to design very hard problems for EDAs by targeting the linkage learning task.

However, ECGA showed a much higher sensitivity to the parameter m, which targets task (2) and (3) of the recombinative framework. Without a prompt and qualitative model, the ECGA failed to efficiently mix building-blocks and decide between competing schemata. The experiments showed that if the correct linkages are not detected in time, efficient mixing of subfunctions is not possible at higher levels due to allelic premature convergence.

Relying heavily just on pairwise linkage detection may heavily prolong the time needed to identify and supply raw subsolutions or building-blocks. This can fatally hinder the performance for any problem where efficient discovery of even more fit, higher order building-blocks is vital.

5. CONCLUSIONS AND FUTURE WORK

Present multivariate EDAs heavily rely on the exploitation of pairwise linkages to construct higher order models, to avoid the computationally very expensive task of searching through the whole possible search space of multivariable linkages. Historical attempts to outline scenarios for which this approach would lead to the failure of the optimization process were seldom successful.

In this paper we argue that with the current approach is not possible to efficiently solve problems that require the mixing and decision making among building-blocks that are hard to discover using the current approaches as they do not exhibit pairwise interactions. To demonstrate this, we introduced the Hierarchical Pairwise Allelic Independent function based on Walsh code subfunctions and hierarchical XOR relation and showed that ECGA can not solve it in an efficient manner.

We conclude that current model building techniques can heavily increase the "innovation time" of the methods. This is problematic for problems where mixing of subsolutions is required. Here, the takeover and convergence mechanism will operate on an incomplete set of building-blocks, leading to allelic premature convergence.

In a future effort, we will investigate how ECGA and other EDAs can discover Walsh code based linkages. Future work should also focus on finding computationally cheaper methods to asses multivariate dependency which preferably can be implemented in parallel architectures. Previous result had shown that an efficient parallel EDA could even scale up to billion bit sized problems [28].

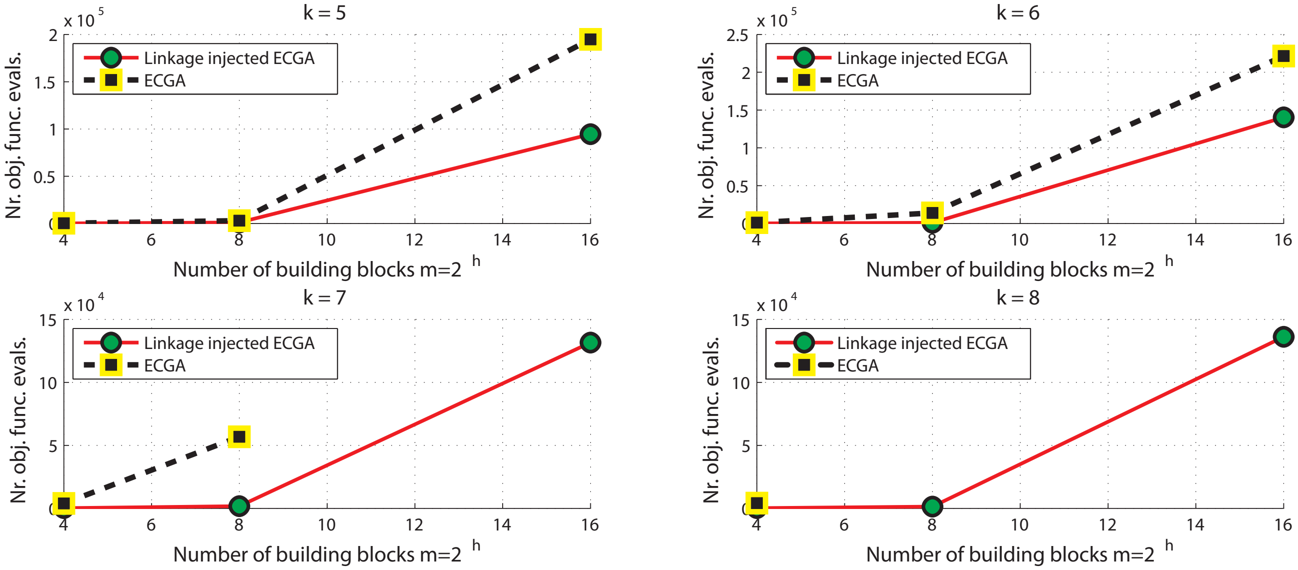

Figure 3: Number of average objective function evaluations on HPAIF with k={5,6,7,8}, for the linkage informed ECGA and normal ECGA, using population sizes determined by the bisection method.

Acknowledgments

This research is supported by Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania. We also acknowledge the support of the Sapientia Institute for Research Programs (KPI).

6. REFERENCES

[1] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Pittsburgh, PA, USA, 1994.

[2] J. S. D. Bonet, C. L. Isbell, and P. A. Viola. MIMIC: Finding optima by estimating probability densities. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems,*, volume 9, pages 424–430. MIT Press, 1996.

[3] 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 and evolutionary computation*, GECCO '09, pages 397–404, New York, NY, USA, 2009. ACM.

[4] D. J. Coffin and R. E. Smith. Why is parity hard for estimation of distribution algorithms? In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 624–624, New York, NY, USA, 2007. ACM.

[5] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 93–108, San Mateo, 1993. Morgan Kaufmann.

[6] C. Echegoyen, J. A. Lozano, R. Santana, and P. L. uaga. Exact bayesian network learning in estimation of distribution algorithms. In D. Srinivasan and L. Wang, editors, *2007 IEEE Congress on Evolutionary Computation*, pages 1051–1058, Singapore, 25-28 Sept. 2007. IEEE Computational Intelligence Society, IEEE Press.

[7] R. Etxeberria and P. Larranaga. Global optimization using Bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 151–173, 1999.

[8] S. Forrest and M. Mitchell. What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *Machine Learning*, 13, 1993.

[9] D. Goldberg. Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems*, 3(2):129–152, 1989.

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[11] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[12] G. Harik. Linkage learning via probabilistic modeling in the ECGA. Technical Report IlliGAL Report no. 99010, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Feb. 04 1999.

[13] G. Harik, F. Lobo, and D. Goldberg. The compact genetic algorithm. *Evolutionary Computation, IEEE Transactions on*, 3(4):287–297, 1999.

[14] P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer, Norwell, MA, USA, 2001.

[15] C. Lima, M. Pelikan, D. Goldberg, F. Lobo, K. Sastry, and M. Hauschild. Influence of selection and replacement strategies on linkage learning in BOA. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1083–1090. IEEE, 2007.

[16] C. Lima, M. Pelikan, K. Sastry, M. Butz, D. Goldberg, and F. Lobo. Substructural neighborhoods for local

search in the Bayesian optimization algorithm. pages 232–241. Springer, 2006.

[17] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evol. Comput.*, 13(1):1–27, 2005.

[18] H. Mühlenbein and T. Mahnig. The factorized distribution algorithm for additively decomposed functions. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 752–759, Mayflower Hotel, Washington D.C., USA, 6-9 July 1999. IEEE Press.

[19] H. Muhlenbein and G. PaaB. From recombination of genes to the estimation of distributions: I. binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving From Nature-PPSN IV (4th PPSN'96)*, volume 1141 of *Lecture Notes in Computer Science (LNCS) September 1996*, pages 178–187. Springer-Verlag, Berlin, 96.

[20] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In L. S. et al., editor, *GECCO '01:*, pages 511–518, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.

[21] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In W. B. et al., editor, *GECCO '99*, volume I, pages 525–532, Orlando, FL, 13-17 July 1999. Morgan Kaufmann Publishers, San Fransisco, CA.

[22] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. Bayesian optimization algorithm, population sizing, and time to convergence. In L. D. Whitley, D. E. Goldberg, E. Cantú-Paz, L. Spector, I. C. Parmee, and H.-G. Beyer, editors, *GECCO*, pages 275–282. Morgan Kaufmann, 2000.

[23] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Comput. Optim. Appl.*, 21:5–20, January 2002.

[24] M. Pelikan and H. Muhlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer-Verlag.

[25] E. Radetic and M. Pelikan. Spurious dependencies and EDA scalability. In *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation*, GECCO 2010, pages 303–310, New York, NY, USA, 2010. ACM.

[26] J. Rissanen. Modelling by the shortest data description. *Automatica*, 14:465–471, 1978.

[27] R. Santana, P. Larrañaga, and J. A. Lozano. Interactions and dependencies in Estimation of Distribution Agorithms. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC-2005)*, volume 2, pages 1418–1425, Edinburgh, U.K., 2005. IEEE Press.

[28] K. Sastry, D. E. Goldberg, and X. Llora. Towards billion-bit optimization via a parallel estimation of distribution algorithm. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 577–584, New York, NY, USA, 2007. ACM.

[29] H. A. Simon. *The Sciences of the Artificial.* MIT Press, Cambridge, Massachusetts, first edition, 1969.

[30] R. A. Watson, G. Hornby, and J. B. Pollack. Modeling building-block interdependency. In *PPSN V: Proc. of the 5th International Conference on Parallel Problem Solving from Nature*, pages 97–108, London, UK, 1998. Springer, LNCS.

[31] R. A. Watson and J. B. Pollack. Hierarchically consistent test problems for genetic algorithms: Summary and additional results. In S. Brave, editor, *GECCO '99: Late Breaking Papers*, pages 292–297, Orlando, Florida, USA, 13 July 1999.

[32] P. Winward and D. E. Goldberg. Fluctuating crosstalk, deterministic noise, and ga scalability. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 1361–1368, New York, NY, USA, 2006. ACM.

[33] H. Wu and J. L. Shapiro. Does overfitting affect performance in estimation of distribution algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 433–434, New York, NY, USA, 2006. ACM.

[34] T.-L. Yu and D. E. Goldberg. Conquering hierarchical difficulty by explicit chunking: substructural chromosome compression. In *GECCO 2006:*, pages 1385–1392, NY, USA, 2006. ACM Press.

[35] T.-L. Yu, D. E. Goldberg, A. Yassine, and Y.-P. Chen. Genetic algorithm design inspired by organizational theory: Pilot study of a dependency structure matrix driven genetic algorithm. In *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 1620–1621, Chicago, 12-16 July 2003. Springer-Verlag.