

The Effects of Representational Bias on Collaboration Methods in Cooperative Coevolution

R. Paul Wiegand, William C. Liles, and Kenneth A. De Jong

George Mason University, Fairfax, VA 22030, USA,
paul@tesseract.org, wliles@gmu.edu, kdejong@gmu.edu

Abstract. Cooperative coevolutionary algorithms (CCEAs) have been applied to many optimization problems with varied success. Recent empirical studies have shown that choices surrounding methods of collaboration may have a strong impact on the success of the algorithm. Moreover, certain properties of the problem landscape, such as variable interaction, greatly influence how these choices should be made. A more general view of variable interaction is one that considers epistatic linkages which span population boundaries. Such linkages can be caused by the decomposition of the actual problem, as well as by CCEA representation decisions regarding population structure. We posit that it is the *way* in which represented problem components interact, and not necessarily the existence of cross-population epistatic linkages that impacts these decisions. In order to explore this issue, we identify two different kinds of *representational bias* with respect to the population structure of the algorithm, *decompositional bias* and *linkage bias*. We provide analysis and constructive examples which help illustrate that even when the algorithm’s representation is poorly suited for the problem, the choice of how best to select collaborators can be unaffected.

1 Introduction

Coevolutionary Algorithms (CEAs) are interesting extensions to traditional Evolutionary Algorithms (EAs). While fitness in an EA is determined objectively, fitness in a CEA is determined subjectively based on how an individual interacts with other individuals. In cooperative coevolution, individuals that participate in successful interactions (*collaborations*) are rewarded while unsuccessful collaborations are punished.

In this paper, we focus on the particular class of cooperative coevolutionary algorithms (CCEAs) defined by [1, 2]. A standard approach to applying CCEAs to an optimization problem starts by trying to identify some reasonable static decomposition of the problem representation into components represented by each population. So, for example, if given a function of m variables to optimize, one might choose to put each variable in a separate CCEA population. Once a decomposition is established, the fitness of components in one population is estimated by choosing one or more collaborators from the other populations.

There have been several attempts to understand how collaborators are best chosen in this paradigm. Early work suggested that these choices were tied to the amount of epistatic interaction among the function variables [1, 3]. In a CCEA that uses an N -variable decomposition for such fitness landscapes, this leads to the notion of

cross-population epistasis, and to the simple intuition that increasing amounts of cross-population epistasis will require more complex collaboration mechanisms. Unfortunately, the issue is more complicated than this. For example, the simplest collaboration method seems to work best when applying a CCEA to a non-linearly separable, quadratic problem despite the existence of cross-population epistasis [4].

This paper extends and clarifies these issues by focusing on how optimization problems are represented in a CCEA. We identify two kinds of representational bias, *decompositional bias* and *linkage bias*, and show how these biases affect choices of collaboration method. The paper is laid out as follows. In the next section, we will briefly outline some background regarding existing analyses of coevolutionary algorithms, the cooperative coevolutionary architecture on which we will be focusing, and some of the choices surrounding collaboration in this algorithm. The third section discusses what we believe to be the important characteristics of problems, namely decomposability and epistasis. The fourth and fifth sections take the two kinds of representational bias in turn, first discussing the implications of decompositional bias with respect to collaboration, then those of linkage bias. The final section will conclude by discussing our improved understanding how problem characteristics affect choices of collaboration methodology.

2 Cooperative Coevolution

2.1 Existing Analysis of Coevolutionary Algorithms

Much of the analysis of coevolutionary algorithms has focused on their complicated dynamics. For example, considerable effort has been spent trying to understand how one can measure progress in a system where individual fitnesses are subjective in order to help identify some of the pathological behaviors exhibited by these algorithms [5–7]. Additionally, some basic theoretical work uses ideas from simple genetic algorithm theory provided by [8], and applies them to competitive coevolution [9]. That work explores the mechanics of a simple competitive coevolutionary algorithm from an evolutionary game theoretic viewpoint. [10] extended this model to analyze cooperative coevolution for the purposes of investigating their potential as optimizers.

One of the issues of considerable practical significance for coevolutionary algorithms is how to assess the fitness of an individual in one population (species) when that fitness depends in part on the individuals in other populations (species). The theoretical models typically assume “full mixing” in the sense that an individual’s fitness is determined by a complete set of “interactions” with all other species. In the simple case that each of the coevolving p populations contains i individuals, the number of “interactions” per fitness evaluation is i^{p-1} .

As a consequence, there is strong practical motivation to estimate fitness by selecting only a small subset of the possible interactions, particularly when $p \gg 2$. This process is usually accomplished by selecting a small number “partners” [3] or “collaborators” [1] from the other species whose interactions are the basis of fitness estimates. Immediate questions arise as to: 1) how many partners, 2) how to select the partners, and, in the case of multiple interactions, 3) how to combine the results of multiple interactions into a single fitness estimate. The answers to these questions are far from simple

and depend on a number of factors including 1) whether the coevolutionary model is competitive or cooperative, and 2) the degree and type of cross-population epistasis present [1, 11, 2–4, 12].

2.2 Collaboration in CCEAs

Our focus is on understanding these issues for cooperative coevolutionary models, in particular for the CCEA architecture developed in [1, 2]. In this model there is a single global fitness function for the entire system, but the search space has been decomposed into a number of independent subspaces, each of which is explored in parallel by independent EA populations. In this case, in order to evaluate the fitness of an individual in one population, one or more collaborators must be selected from the other populations in order to assemble a complete object for a global evaluation.

A variety of studies including [1, 3, 4] suggest that, if the degree of cross-population epistasis is not too strong, then selecting the current best individual from each of the other populations and performing a single global fitness evaluation is a surprisingly robust collaboration mechanism. When there is a strong degree of epistasis, a more complex strategy involving more than one collaborator from each of the other populations and a less greedy selection method for those collaborators can improve performance.

Finally, if multiple function evaluations are involved for each subcomponent, how can one combine these results to obtain a single fitness value? Here the literature is fairly consistent in recommending assigning the maximum value obtained as the fitness.

2.3 Representation in CCEAs

However, for the practitioner, there are still a number of questions to be answered including *how* to decompose a complex problem in a way that leads to an effective CCEA-based solution. This is a familiar question for standard EAs, e.g. deciding how best to represent problems (such as Traveling Salesperson problems) to achieve good EA-based solutions. For CCEAs, representations must additionally involve decompositions into separately evolving subcomponents.

The difficulty for the CCEA practitioner is that there is seldom sufficient *a priori* information to select an “optimal” representation or even one with sufficient knowledge to make informed choices about the appropriate collaboration mechanism to use. However, it is clear that any choice of representation introduces a bias in the system that can potentially have strong effects on a particular collaboration strategy.

Intuitively, CCEA representation biases are a result of the granularity of the decomposition (how many subcomponents) and the resulting epistatic interactions among the subcomponents. The focus of this paper will be on understanding these biases and their effects on collaboration mechanisms.

3 Research Methodology

In order to better control for the representational properties that we feel are important to choices of CCEA collaboration mechanisms, this paper focuses on *pseudo-boolean*

functions, that is the objective value is assessed by mapping binary strings of length n to real values, $\mathcal{F} : \{0,1\}^n \rightarrow \mathfrak{R}$. The two representational properties of interest are decomposability and epistasis.

3.1 Decomposability

For our purposes, a function is considered decomposable if it can be decomposed into a sum of some number of smaller independent functions. These functions do not necessarily have to be identical. More formally, a function \mathcal{F} is *decomposable* if there exist a set of independent functions, $\{f_1, f_2, \dots, f_m\}$ such that $\mathcal{F}(x) = \sum_{i=1}^m f_i(x_i)$. Some types of problems are rendered more tractable for optimization because of this property since optimization can be done as m independent optimizations [13].

For pseudo-boolean functions, each x_i refer to partitions of the main string, or *building blocks* of the problem. Of particular interest are m -decomposable functions, i.e., those for which an m block decomposition exists, but no finer grained partition exists.

An example of a bit-wise decomposable problem (i.e., $m = \text{string_length}$) is the classic `ONESMAX` problem in which fitness is the sum of all the 1 bits in the string. A familiar example of a problem that is not decomposable (i.e., $m = 1$) is the `LEADINGONES` problem. Both of these problems are defined formally in the following sections.

Ideally, for such problems, one would like to map each decomposable unit into a CCEA population. In general, however, the practitioner and the CCEA don't have explicit decomposability information. As a consequence, a CCEA can be mismatched in terms of the number of populations and what gets mapped into each population. We refer to this as the *decompositional bias* of a CCEA.

3.2 Epistasis

This notion of decomposability is intimately tied to the ideas of epistasis. For pseudo-boolean functions we define *epistasis* to mean non-linear interactions between bit positions in the problem [14]. So, for example, a highly epistatic pseudo-boolean function like `LEADINGONES` is 1-decomposable while the non-epistatic `ONESMAX` problem is fully decomposable.

Of interest here is the kind of epistasis that a decomposition exhibits (e.g., [12]). Epistatic interactions can be *positive* in the sense that the contribution of a piece of the representation due to its non-linear interaction with other pieces have the same kind of impact on fitness as the contribution of the piece itself; however, the *magnitude* of that impact depends on the value of the other pieces involved in the interaction. Similarly, epistatic interactions can be *negative* in the sense that the impact of the individual pieces involved have the opposite effect on fitness as the contribution of their non-linear combinations, but again the magnitude of this opposing contribution depends on the pieces involved. In addition, one can have epistatic interactions in which *neither the sign nor the magnitude* of the effect can be predicted from the individual components. As we will see, the particular form of epistasis has an important effect on collaboration mechanisms.

3.3 Experimental Framework

In the following sections we will construct several problems that exhibit these properties and we analyze them both formally and empirically. For the empirical studies we used a CCEA with the following properties. A steady state GA was used for evolving each of the populations. Each steady state GA uses a ranked-based selection method, such that a single offspring replaces the worst individual each generation Bit-flip mutation was applied at a rate of $1/r$, where r is the number of bits of individuals in a given population. Parameterized uniform crossover was applied 100% of the time to produce a single offspring in such a way that there was a 0.2 probability of swapping any given bit. As part of the reported experiments, we varied the number of populations, p , but in all cases the number of individuals in each population was 10.

During preliminary sensitivity experiments, various algorithm parameter values were used. These included generational versus steady state models, proportional versus ranked-based selection methods, different population sizes, different variational operators (and rates). Though not reported in this paper, the results were consistent with our reported findings. Our choices for the final algorithm were based on performance results.

4 The Effects of Decompositional Bias on Collaboration

Obviously, different problems have different degrees of decomposability that may or may not be reflected in the particular CCEA representation chosen. Since decomposability information is not generally available for difficult optimization problems, our focus here is on the case where there is some kind of mismatch between the CCEA representation and a problem’s “natural” decomposition. The question here is not whether such mismatches make the problem harder to solve. This will clearly happen for some problems [1, 3, 12, 4]. Instead, the question is whether adopting more complex collaboration methods can alleviate such mismatches.

4.1 Controlling Decompositional Bias Experimentally

In order to answer this question, we need to have problems in which we can explicitly control their inherent decomposability. For pseudo-boolean functions this is not difficult to do. One simply defines a function in terms of the sum of m independent subfunctions which are themselves not decomposable. A simple example is obtained by choosing a non-decomposable function of k bits and concatenating m of these k -bit blocks to form a km -bit problem the value of which is just the sum of the individual functional blocks.

More formally, let $\mathcal{F} : \{0, 1\}^n \rightarrow \mathfrak{R}$ be some objective function over a binary string of length $n = mk$ and $f : \{0, 1\}^k$ be some non-decomposable function over a binary string of length k . Then, given $x \in \{0, 1\}^n$,

$$\mathcal{F}(x) = \sum_{i=0}^{m-1} f(m_i)$$

where m_i represents the i th block of k bits in x .

From a practitioner’s point of view, barring any problem specific knowledge, the simplest way to represent pseudo-boolean functions is to break up bit strings of length n into p blocks and assign each block to a different population. Hence, a decompositional mismatch of the representation may be due to over-decomposition ($p > m$) or under-decomposition ($p < m$). If there are more populations than there are decomposition blocks, there is likely to be strong interaction between populations in the system with respect to the problem, i.e., cross-population epistasis. If $p < m$, the advantage of the parallelism of coevolutionary search is sacrificed.

4.2 Effects of Collaboration Methods

We begin to answer the question of whether problems introduced by decompositional bias can be alleviated by more complex collaboration methods by observing that, if there is no cross-population epistasis, a simple selection method for collaboration is sufficient. If the two populations represent independent pieces of the problem, then optimizing one population independently of the other will result in the complete optimization of the problem. As long as the collaborators chosen for each population member are the same, it doesn’t matter how we chose the collaborator. However, it does matter how many collaborators we choose, since picking more than one will incur more unnecessary computational cost in the way of objective function evaluations. Therefore, in the absence of cross-population epistasis, selecting the single best individual¹ from the other populations for collaboration is sufficient. In fact, one could pick this individual randomly, as long as it was the same individual for each member of the population during a given generation. The point isn’t that any partnering scheme will result in a better collaboration than another, but that since each population can essentially be optimized independently, we only need a *consistent* sample from which to establish a collaboration.

So why would a more complicated collaboration selection method be needed? Recall that how one chooses collaborators is essentially a choice about how one *samples the potential interactions* with the other population. There has to be a reason to believe that more than one sample is needed, or that sampling with a particular bias (say choosing the best) will result in a poor characterization of the relationship between the populations. Either way, some interaction between the populations is certainly needed to justify this. More than simply having such epistasis is at issue, however.

Consider the `LeadingOnes` problem, $f(x) = \sum_{i=1}^k \prod_{j=1}^i x_j$. This problem is certainly not decomposable. Further, if we aggregate m of them, we can study the effects of running a CCEA when the number of populations p is $\geq m$.

In order to study the effects of collaboration on such situations, we constructed the following experiment. Using the CCEA described in the Methodology section, we experimented with a concatenated `LeadingOnes` problem. In this particular case there were 128 bits in the total bit string of the problem, subdivided evenly into $m = 2$ blocks. A total of 6 collaboration selection methods were used. The number of collaborators chosen for a given evaluation was varied (1, 2, & 3) and two selection biases were used: s -best and s -random (without replacement). We varied the number of populations, p ,

¹ ”Best” here means the most fit individual(s) in other population(s) from previous evaluation.

but in all cases the number of individuals in each population was 10. The results for $p = 2$ through $p = 16$ in Table 1 show the average number of evaluations it took the algorithms to reach the optimum (50 trials each). Unless otherwise stated, confidence levels for all tests are 95%.

$p = 2$	# Collaborators			$p = 4$	# Collaborators		
	1	2	3		1	2	3
<i>s</i> -best	8016.7	16015.4	24654.2	<i>s</i> -best	8801.1	17602.1	26198.9
<i>s</i> -rand	8989.8	17247.7	25191.8	<i>s</i> -rand	10155.5	18757.9	28372.5

$p = 8$	# Collaborators			$p = 16$	# Collaborators		
	1	2	3		1	2	3
<i>s</i> -best	9821.52	19825.08	29018.32	<i>s</i> -best	11247.20	22350.32	33468.88
<i>s</i> -rand	12734.86	22134.34	32052.24	<i>s</i> -rand	19233.78	30089.90	41995.38

Table 1. Steady state CCEA results on the `LeadingOnes` problem. Each value represents the mean number of evaluations needed to reach the optimum out of 50 trials. From the top left corner, proceeding clockwise, the tables represent data for decompositional biases created using two, four, eight and sixteen populations.

The Tukey-Means test indicates that in all cases choosing one collaborator is clearly better. This might at first be puzzling since there is clearly cross-population epistasis present when $p > 2$. However, note that a mutation which turns some bit to 1 in an individual in the first population will always result either a neutral or positive change in fitness, regardless of the contents of the other population. The reverse is true, as well. In addition, this decomposition is also asymmetric for $p = 4$ in that the second and fourth populations will remain relatively unimportant to the first and third populations for some time during the evolution, since each `LeadingOnes` subproblem is solved left-to-right.

One observation that can be made is that by changing the number of populations, the mutation rate is effectively being increased (recall that the mutation is $1/r$, where r is the number of bits per individual in each population). Such issues may be relevant, consequently we ran all population oriented experiments in the paper (including the preceding one) using a constant $1/64$ mutation rate. The results (not reported) remain consistent with those reported here.

We can make the problem slightly more interesting by making the right-hand side of the bit string play a more important role. We will do this by scaling the `LeadingOnes` part by k and subtracting `OnesMax` from the total, i.e., $f(x) = k \sum_{i=1}^k \prod_{j=1}^i x_j - \sum_{i=1}^k x_i$. Now not only will the right side matter, but there is some tension between individual bit contributions to fitness and those of their non-linear interactions. Moreover, this tension is "one directional" in a sense. Take the string: "110000..." as an example. Flipping the fourth bit to a one will decrease the fitness slightly if the third bit remains 0, while flipping both the third and fourth bits will increase the fitness. However, the same is not true on the other side. Flipping the third bit while the fourth bit remains 0 will also increase fitness. So some of the interactions have this property of sign-dependent epistasis, while others will not. In addition, the linear effects of the bits are very muted compared to the non-linear effects due to the scaling issue.

Using the same experimental setup as before, we studied the effects of collaboration on `LeadingOnes-OnesMax`. The results for $p = 2$ through $p = 16$ in Table 2 show the average number of evaluations it took the algorithms to reach the optimum (50 trials each).

$p = 2$	# Collaborators			$p = 4$	# Collaborators		
	1	2	3		1	2	3
<i>s</i> -best	15592.9	30846.8	48259.6	<i>s</i> -best	15549.4	30974.6	45445.7
<i>s</i> -random	16000.7	31320.6	46406.9	<i>s</i> -random	16381.6	30319.4	44736.0

$p = 8$	# Collaborators			$p = 16$	# Collaborators		
	1	2	3		1	2	3
<i>s</i> -best	14014.40	28969.24	44172.40	<i>s</i> -best	14247.10	27911.04	41735.98
<i>s</i> -random	17343.58	28795.78	45252.94	<i>s</i> -random	21653.86	33975.28	47792.74

Table 2. Steady state CCEA results on the `LeadingOnes-OnesMax` problem. Each value represents the mean number of evaluations needed to reach the optimum out of 50 trials. From the top left corner, proceeding clockwise, the tables represent data for decompositional biases created using two, four, eight and sixteen populations.

In all cases there was no statistical reason to choose another collaboration method other than the single best individual from the other populations. Not only does this increased decompositional bias not alter the collaboration methodology, it appears as though this problem becomes *easier* for the CCEA to solve, not harder. This turns out to be statistically significant only for the $p = 8$ and $p = 16$ cases (not shown) where there is one or two "best" collaborators chosen.

So far, these experiments confirm what we see in practice, namely that the simple collaboration method involving just the best individuals from each population is quite robust even when there is cross-population epistasis. However, what is still not clear is when it fails. To understand that better, we focus on the various forms of cross-population epistasis.

5 The Effects of Linkage Bias on Collaboration

The decompositional bias of the previous section focused on potential mismatches between a problem's "natural" decomposition into m components and the number of CCEA populations p used. Even if $p = m$, there is still the question as to whether the CCEA breaks up the string so that each "natural" block is assigned its own population. If not, breaking up tightly linked bits can result in significant cross-population epistasis. In general, the degree to which linked bits in a block are assigned to the same population for the purposes of representation can be thought of as linkage bias.

5.1 Controlling Linkage Bias Experimentally

Again it isn't hard to construct a way of controlling this bias. We define a mask over the entire bit string which specifies to which population a given bit belongs, $\mathcal{M} \in \{1, 2, \dots, p\}^n$. Note that in the case of these mask definitions, the superscript suggests

repetition, and not an exponent. For problems like those in the previous section involving a series of m concatenated non-decomposable r -bit blocks, a mask which corresponds to the most biased linkage (i.e. is more closely aligned with the real problem) is $\mathcal{M}_s = 1^r 2^r \dots p^r$. Coming up with a mask which is highly pathological is very problem dependent, but a mask which will turn out to be commonly quite bad is $\mathcal{M}_h = (123 \dots p)^r$. Here every bit in a block is distributed to every population, resulting in the likelihood of a high degree of cross-population epistasis.

As noted earlier, any increase in the amount of cross-population epistasis is likely to make a problem more difficult to solve using a CCEA. The question at hand is whether adopting a more complex collaboration method can alleviate these difficulties. By applying different types of masks, which distribute different pieces of the blocks of the problem to different degrees, we can explore the affect that varying degrees of linkage bias have on collaboration methods.

5.2 Effects of Collaboration Methods

We begin by considering again the `LeadingOnes-OnesMax` problem, assuming $m = p = 2$. Using the $\mathcal{M}_s = 11 \dots 1 \ 22 \dots 2$ mask presents us the same problem we've already discussed, where there is no cross-population epistasis, while the mask $\mathcal{M}_h = 1212 \dots 12 \ 2121 \dots 21$ creates a situation with very strong cross-population epistasis. Using the same experimental setup as before, we studied the effects that these two masks had on the choice of collaboration methods. The results are presented in Table 3.

\mathcal{M}_s	# Collaborators			\mathcal{M}_h	# Collaborators		
	1	2	3		1	2	3
<i>s</i> -best	15592.9	30846.8	48259.6	<i>s</i> -best	15305.4	28939.1	45756.5
<i>s</i> -random	16000.7	31320.6	46406.9	<i>s</i> -random	17862.5	32802.2	47439.4

Table 3. Steady state CCEA results on the `LeadingOnes-OnesMax` problem. Each value represents the mean number of evaluations needed to reach the optimum out of 50 trials. The left table represents a linkage bias which uses the \mathcal{M}_s mask, while the right uses the \mathcal{M}_h mask.

Differences between the means for *s*-best and *s*-random groups for \mathcal{M}_h are significant for one and two collaborators, but not for three. There are no statistically significant differences between these groups (for the same number of collaborators) for the simpler linkage bias.

Once again simply distributing the epistatic linkages across the population boundaries is insufficient to require that a more complicated collaboration method be used. This may seem surprising at first, but note that, for this particular problem, introducing such masks does not change the type of cross-population epistasis, only its degree. Moreover, although not germane to our question, it is interesting to note that in this particular case it seems that increasing the mixing seems to *improve* performance versus the \mathcal{M}_s mask (this is significant for all but the 3-random case). In the case of our generational CCEA experiments, this was true to statistical significance for all groups.

What we have failed to construct so far is a case of the most difficult form of cross-population epistasis: namely, when neither the sign nor the magnitude of the

interaction can be reliably predicted. There is a simple change to the current problem that will result in cross-population epistasis of this type, namely by changing the LeadingOnes component to count only paired ones. More formally, we defined a LeadingPairedOnes-OnesMax problem given by $f(x) = 2k \sum_{i=1}^k \prod_{j=1}^{i-1} (x_j x_{j+1}) - \sum_{i=1}^k x_i$. Interestingly, this problem is so much more difficult that the optimum was frequently never found. As a consequence Table 4 represents the mean fitness values obtained after a constant 100,000 evaluations (here higher is better).

\mathcal{M}_s	# Collaborators			\mathcal{M}_h	# Collaborators		
	1	2	3		1	2	3
<i>s</i> -best	3328.9	2086.5	1590.1	<i>s</i> -best	647.6	705.6	761.0
<i>s</i> -random	3366.7	2152.0	1784.1	<i>s</i> -random	1247.3	1383.4	1469.1

Table 4. Steady state CCEA results of the LeadingPairedOnes-OnesMax problem. Each value represents the mean value obtained after 100,000 function evaluations out of 50 trials. The left table represents a linkage bias which uses the \mathcal{M}_s mask, while the right uses the \mathcal{M}_h mask.

Now we see exactly the reverse situation. Although it is clearly better to select a single collaborator when there are no cross-population epistatic linkages of this type, as soon as those linkages are spread across populations a more complex collaboration mechanism is required. In the latter case, increasing the number of collaborators does in fact result in statistically improved performance, as does picking the individual randomly rather than greedily. However, in the \mathcal{M}_s case, with the exception of the three collaborator groups, there is no statistical reason to pick randomly over picking the best.

6 Conclusions

Choosing an effective representation of a problem is a critical design decision for any EA-based problem solving. In the case of CCEAs, representation decisions involve additional choices as to how to decompose a problem effectively. Without sufficient *a priori* knowledge about a particular problem (the usual case), particular representations can introduce biases related to the degree to which they match the underlying problem characteristics. This in turn can affect the choice of collaboration mechanism to be used.

Using several well understood pseudo-boolean functions, we explored the effects of two kinds of representational bias (decomposition bias and linkage bias) on collaboration mechanism choices. We were able to show, somewhat surprisingly, that decompositional bias does not appear to dramatically affect the choice of collaboration mechanisms. For the practitioner, this means that the standard collaboration mechanism of choosing the single best individual from each of the other subpopulations is reasonably robust across decompositional biases.

Equally surprising was the fact that this same collaboration mechanism can be robust across different linkage biases as well, but not always. To understand this better, the cross-population epistatic interactions resulting from these linkage biases were studied in more detail. In some cases these interaction are positively or negatively correlated with fitness, in the sense that, though the magnitude of the fitness change depends on

the relationship between the linear and non-linear pieces of the problem, the sign does not. The standard collaboration mechanism worked fine for these cases.

However, in the case where both the sign and the magnitude of the fitness contribution are uncorrelated, the standard collaboration mechanism breaks down and a more complex mechanism is required. Intuitively, in such situations additional samples are required to obtain reasonable estimates of fitness.

Clearly, the results presented here are preliminary in nature, and a more thorough examination of these issues is needed. However, we believe these results already provide useful guidance to the CCEA practitioner. An interesting open question for EA design in general is whether this notion of different types of epistasis will also help clarify the effects of gene linkages within a genome.

References

1. M. Potter and K. De Jong. A cooperative coevolutionary approach to function optimization. In Y. Davidor and H.-P. Schwefel, editors, *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN III)*, pages 249–257. Springer-Verlag, 1994.
2. M. Potter. *The Design and Analysis of a Computational Model of Cooperative CoEvolution*. PhD thesis, George Mason University, Fairfax, Virginia, 1997.
3. L. Bull. Evolutionary computing in multi-agent environments: Partners. In Thomas Baeck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA)*, pages 370–377. Morgan Kaufmann, 1997.
4. R. Paul Wiegand, William Liles, and Kenneth De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In Spector [15], pages 1235–1242.
5. R. Watson and J. Pollack. Coevolutionary dynamics in a minimal substrate. In Spector [15], pages 702–709.
6. S. Ficici and J. Pollack. Challenges in coevolutionary learning: Arms–race dynamics, open-endedness, and mediocre stable states. In Adami et al, editor, *Proceedings of the Sixth International Conference on Artificial Life*, pages 238–247, Cambridge, MA, 1998. MIT Press.
7. D. Cliff and G. F. Miller. Tracking the red queen: Measurements of adaptive progress in coevolutionary simulations. In *Proceedings of the Third European Conference on Artificial Life*, pages 200–218. Springer-Verlag, 1995.
8. M. Vose. *The Simple Genetic Algorithm*. MIT Press, 1999.
9. S. Ficici and J. Pollack. A game-theoretic approach to the simple coevolutionary algorithm. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 467–476. Springer-Verlag, 2000.
10. R. Paul Wiegand, William Liles, and Kenneth De Jong. Analyzing cooperative coevolution with evolutionary game theory. In D. Fogel, editor, *Proceedings of CEC 2002*. IEEE Press, 2002. (To appear).
11. J. Paredis. Coevolutionary computation. *Artificial Life Journal*, 2(3), 1996.
12. L. Bull. On coevolutionary genetic algorithms. *Soft Computing*, 5:201–207, 2001.
13. R. Salomon. Performance degradation of genetic algorithms under coordinate rotation. In L. Fogel, P. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 153–161. MIT Press, 1996.
14. Yuval Davidor. Epistasis variance: A viewpoint on ga-hardness. In G. Rawlins, editor, *Foundations of Genetic Algorithms (FOGA)*, pages 23–35. Morgan Kaufmann, 1990.
15. L. Spector, editor. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001*. Morgan Kaufmann, 2001.