

Cooperative Coevolution and Univariate Estimation of Distribution Algorithms

Christopher Vo
Dept. Computer Science
George Mason University
4400 University Dr., MSN 4A5
Fairfax, VA 22030
cvo1@gmu.edu

Liviu Panait
Google, Inc.
604 Arizona Avenue
Santa Monica, CA 90401
liviu@google.com

Sean Luke
Dept. Computer Science
George Mason University
4400 University Dr., MSN 4A5
Fairfax, VA 22030
sean@cs.gmu.edu

ABSTRACT

In this paper, we discuss a curious relationship between Cooperative Coevolutionary Algorithms (CCEAs) and univariate Estimation of Distribution Algorithms (EDAs). Specifically, the distribution model for univariate EDAs is equivalent to the infinite population EGT model common in the analysis of CCEAs. This relationship may permit cross-pollination between these two disparate fields. As an example, we derive a new EDA based on a known CCEA from the literature, and provide some preliminary experimental analysis of the algorithm.

Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity; G.1.6 [Optimization]: Global Optimization

General Terms

Theory, Algorithms

Keywords

Cooperative Coevolution, Estimation of Distribution Algorithms, Evolutionary Game Theory

1. INTRODUCTION

We describe a relationship between two seemingly unrelated corners of the evolutionary computation community: Cooperative Coevolutionary Algorithms (CCEAs) and univariate Estimation of Distribution Algorithms (EDAs). These two families of algorithms were independently proposed and developed. Cooperative coevolution has emerged as a mechanism to simplify the search space by projecting it into multiple smaller subspaces, each searched by a separate population; whereas EDAs have often been proposed as competitors with “traditional” sample-based EAs, by replacing the sample population with a distribution estimate.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FOGA '09, January 9–11, 2009, Orlando, Florida, USA.
Copyright 2009 ACM 978-1-60558-414-0/09/01 ...\$5.00.

In particular, univariate Estimation of Distribution Algorithms represent the population distribution as a collection of separate per-gene distributions.

As it turns out, these algorithms have a strong but non-intuitive relationship. The per-gene distributions of univariate EDAs are not only closely related to the multiple populations of CCEAs, but such distributions are in fact *equivalent* to the infinite-sized “populations” used in a very common theoretical model for CCEAs, one based on Evolutionary Game Theory (EGT). This relationship promises to yield cross-pollination between the two fields: CCEAs have lately built a strong theoretical formulation; and EDAs have both good theory and efficient algorithms. We hope that this paper kindles discussion between the two fields.

We begin the paper by first discussing CCEAs and relevant theory. We then discuss univariate EDAs and outline one well-known univariate EDA algorithm as an example. We then formally show the relationship between the two. Based on this relationship, we offer an example of direct transfer from one area to the other: namely, we introduce a new theoretical univariate EDA formulation derived from a CCEA EGT model. We prove that the algorithm can converge to the optimum with any desired probability, though it is not efficient compared to existing EDA algorithms. We also provide an actual EDA algorithm which is an approximation of the theoretical version.

2. COOPERATIVE COEVOLUTION

Coevolutionary algorithms generally assign fitness to an individual not based on an absolute measure but rather on the interaction of that individual with other individuals in the evolutionary system. The hallmark of a coevolutionary algorithm is that the relative order of any two individuals may change depending on the presence of those other individuals in the system.

In this paper, we focus on *n-population cooperative* arrangements, popularly known as Cooperative Coevolutionary Algorithms (CCEAs) [13, 14]. In CCEAs, the solution space is broken into some n sub-solution spaces, and each sub-solution space is assigned a population. An individual is assessed by grouping it with individuals from the other populations to form a complete solution; the quality of this solution is then incorporated into the individual’s fitness. Cooperative coevolutionary algorithms can be generational or steady-state, and often take one of two forms: *serial* versus *parallel* algorithms. One common parallel generational CCEA may be described as:

ALGORITHM 1: *Parallel Generational CCEA*

```

loop
  for each population  $p \in P$  do
    for each individual  $i \in p$  do
      Evaluate( $i, p, P$ )
  for each population  $p \in P$  do
    Breed whole population  $p$ 

```

Much of cooperative coevolution research has focused on the specifics of the Evaluate(i, p, P) function. Choice of evaluation procedure in CCEAs is known to lead to pathologies. Certain of these have been studied at length using a theoretical model for CCEAs which, while somewhat different from actual CCEAs, provides insight into their dynamics. We discuss this model next.

2.1 The Evolutionary Game Theory Infinite Population Model

Analyses of cooperative (and other) coevolution will often make use of an infinite population formulation derived from evolutionary game theory (EGT). This formulation usually assumes that each population is infinite in size, but is drawn from a finite set of genotypes. Much EGT work in cooperative coevolution has focused on two populations. The first population is represented by a vector x where x_i indicates the proportion of genotype i in the population. The second population is represented by the vector y . There also exists a matrix A whose elements a_{ij} represent the reward when genotypes i (from the first population) and j (from the second population) are combined to form a joint solution.

One common EGT model breaks the evolutionary process into two parts. First, the fitness of each individual is assessed. We will use the vector u to represent the fitness of the genotypes in the first population such that each genotype i has fitness u_i . Likewise, we will use w for the second population. Wiegand [19] defined the fitness of a genotype as the average reward received when pairing it with *every* member of the other population. That is, at time t :

$$u_i^{(t)} = \sum_j a_{ij} y_j^{(t)} \quad w_j^{(t)} = \sum_i a_{ij} x_i^{(t)} \quad (1)$$

Second, we then update the genotype proportions for the next generation (time $t + 1$) using a formulation that simulates fitness-proportional selection:

$$x_i^{(t+1)} = x_i^{(t)} \left(\frac{u_i^{(t)}}{\sum_k x_k u_k^{(t)}} \right) \quad y_j^{(t+1)} = y_j^{(t)} \left(\frac{w_j^{(t)}}{\sum_k y_k w_k^{(t)}} \right) \quad (2)$$

Wiegand found that this ‘‘complete mixing’’ model could converge towards local suboptima surrounding Nash Equilibria in the joint space: if a suboptimum basin were large and broad, the system would collect at its peak rather than at another taller but narrower peak centered at a global optimum. This was largely because the fitness procedure *averaged* the performance of an individual over *all* individuals in the corresponding population, without regard to how good collaborators those corresponding individuals were. Wiegand termed this pathology *relative overgeneralization*.

Later research has shown that the system *will* converge if we change the fitness assessment procedure. One solution is to base the fitness of individuals in a population not on average collaboration but rather on the maximum per-

formance over all collaborations. Following Panait [8] we might change Equation 1 to:

$$u_i^{(t)} = \max_j a_{ij} \quad w_j^{(t)} = \max_i a_{ij} \quad (3)$$

Panait provided a proof of convergence to the optimum [8] using this in combination with tournament selection rather than fitness-proportional selection, assuming that the optimum is unique. Panait’s derivation of tournament selection (of tournament size H) transformed Equation 2 to:

$$x_i^{(t+1)} = x_i^{(t)} \frac{\left(\sum_{\forall k: u_k^{(t)} \leq u_i^{(t)}} x_k^{(t)} \right)^H - \left(\sum_{\forall k: u_k^{(t)} < u_i^{(t)}} x_k^{(t)} \right)^H}{\sum_{\forall k: u_k^{(t)} = u_i^{(t)}} x_k^{(t)}^H}$$

$$y_j^{(t+1)} = y_j^{(t)} \frac{\left(\sum_{\forall k: w_k^{(t)} \leq w_j^{(t)}} y_k^{(t)} \right)^H - \left(\sum_{\forall k: w_k^{(t)} < w_j^{(t)}} y_k^{(t)} \right)^H}{\sum_{\forall k: w_k^{(t)} = w_j^{(t)}} y_k^{(t)}^H} \quad (4)$$

This curious equation is a result of the order statistics to compute the expected maximum over a set of size H . In each subequation there are two terms raised to H each. These compute the probability that, of a tournament of size H , the winners (there may be ties) will include a genotype whose fitness is the same as genotype i . The first term gives the probability that all H tournament entrants will have a fitness less than or equal to i ’s fitness, and the second term gives the probability that all will have a fitness less than that of i . The denominators in each subequation compute the probability that the first such winner is in fact i , as opposed to other fitness-equivalent genotypes.

So far, these theoretical models are fairly divorced from real-world CCEAs: the population is infinite; there is no breeding, only selection; and the evaluation procedure involves scanning across all possible collaborators. But this situation may be improved somewhat. Panait et al. [10] provided a weakened convergence proof for a more realistic evaluation procedure: take the maximum performance when paired N times with randomly chosen collaborators. The proof shows that for any probability ϵ , there exists an N_ϵ such that given N_ϵ evaluations, the algorithm is guaranteed to achieve convergence to the global optimum with probability $1 - \epsilon$, when coupled with tournament selection.

The maximum-of- N evaluation procedure, which replaces Equations 1 or 3, is:

$$u_i^{(t)} = \sum_j a_{ij} y_j^{(t)} \frac{\left(\sum_{\forall k: a_{ik} \leq a_{ij}} y_k^{(t)} \right)^N - \left(\sum_{\forall k: a_{ik} < a_{ij}} y_k^{(t)} \right)^N}{\sum_{\forall k: a_{ik} = a_{ij}} y_k^{(t)}^N}$$

$$w_i^{(t)} = \sum_j a_{ij} x_j^{(t)} \frac{\left(\sum_{\forall k: a_{kj} \leq a_{ij}} x_k^{(t)} \right)^N - \left(\sum_{\forall k: a_{kj} < a_{ij}} x_k^{(t)} \right)^N}{\sum_{\forall k: a_{kj} = a_{ij}} x_k^{(t)}^N} \quad (5)$$

Note the similarity to Equation 4. This equation is also an order statistic to compute the maximum over N evaluations.

How large should N be? In a real scenario, N is effectively bounded by the size of the collaborating population(s). But even this upper bound is problematic: large values of N are more accurate and more likely to converge to the optimum; but may require more total number of evaluations than is realistic given the evaluation budget. Thus recent empir-

ical work [2, 9] has focused on reducing the total number of evaluations by identifying an *archive* of individuals from the collaborating population(s) which provide as good an assessment as testing with the *entire* collaborating population would provide. As it turns out, this archive size can be very small, resulting in a significant reduction in evaluations.

3. UNIVARIATE ESTIMATION OF DISTRIBUTION ALGORITHMS

Estimation of Distribution Algorithms (EDAs) replace the evolutionary computation population with a statistical distribution of an infinite population. Most such algorithms iteratively generate samples (individuals) from the distribution, test those samples, and then update the distribution so that high-fitness samples are generated more often in the future and low-fitness samples are generated less often.

An important design decision for EDAs is to select a representation for the probability distribution. An obvious problem is that the joint distribution over a typical solution space is of high dimensionality and complexity. Early on, a common approach was to break the joint distribution into separate distributions per-gene. That is, we assume an individual consists of a set of genes, and for each gene, we maintain a distribution of probabilities of the possible settings (alleles) for that gene. In the simplest case, if the individual were a Boolean vector, then each gene distribution would be represented by a single real value $[0, 1]$ indicating the probability of setting the gene to a 1 instead of 0. If the individual were instead a vector of floating-point numbers, each gene may be represented as a Gaussian distribution over the range of possible values. Common univariate EDAs include the Univariate Marginal Distribution Algorithm (UMDA) [4], the Compact Genetic Algorithm (CGA) [15], and Population-Based Incremental Learning (PBIL) [1]. To illustrate, here is the pseudocode for PBIL, one of several such EDAs:

ALGORITHM 2: *PBIL*

loop

for q from 1 to Q **do**

 Create an individual i_q by choosing an allele at random from distribution D_g for each gene g

 Evaluate(i_q)

 Select the best R individuals from among $i_1 \dots i_q$

for each gene g **do**

 Let N_g be the distribution of alleles for g among the R best individuals

 Update gene distribution $D_g \leftarrow (1 - \alpha)D_g + \alpha N_g$

By pushing the joint distribution into individual marginal distributions, univariate EDAs discard information that is normally available to a more traditional evolutionary algorithm. Such information is important to solve non-separable problems. In univariate EDAs, each distribution is being updated based solely on its performance, without consideration of the other distributions with which it is being conjoined. Non-separable problems require such consideration, as their fitness is based on the nonlinear combination of various elements. Recognizing this weakness, EDA designers have attempted to create richer distributions involving more relationships among the genes. Perhaps best known are variations of the Bayesian Optimization Algorithm (BOA) [12, 11], which attempt to use a Bayesian network to model the entire joint space in a sparse manner.

Despite these difficulties, there has been some theoretical work on convergence properties in univariate EDAs. UMDA has been shown to converge to the optimum for separable problems [6], and for non-separable problems when augmented with a simulated-annealing-like Boltzmann selection [5, 7]. A theoretical infinite-population version of UMDA has also been shown to converge to the optimum [20, 21]. Rastegar and Hariri have shown convergence to local optima for PBIL [17] and CGA [16].

4. EDAS AND THE EGT INFINITE POPULATION MODEL OF CCEAS

While CCEAs in practice may have any number $M > 1$ of populations, previous theoretical work in CCEAs has generally focused on the $M = 2$ case. We begin by extending the model to any $M > 1$.

To do this, we define X to be the set of collaborating populations $\{X_1 \dots X_M\}$. We define ${}_p x_i$ to be the proportion of genotype i of population X_p . Next, we define Y_p to be the set of all possible *tuples* of genotypes chosen from all populations other than X_p , that is, $Y_p = X_1 \times \dots \times X_{p-1} \times X_{p+1} \times \dots \times X_M$. Thus Y_p consists of all possible collaborating tuples for members of X_p . We define ${}_p y_j$ to be the proportion of *tuple* j from Y_p . Finally, A is redefined such that its elements a_{ij} define the reward obtained when genotype ${}_p x_i$ from population X_p is combined with the genotypes in tuple ${}_p y_j \in Y_p$ (we omit indicating the p , as in ${}_p a_{ij}$, because it will always be clear from context). Now we can expand Equation 5 to

$${}_p u_i^{(t)} = \sum_{j \in Y_p} {}_p y_j^{(t)} a_{ij} \frac{\left(\sum_{\substack{k \in Y_p: \\ a_{ik} \leq a_{ij}}} {}_p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{ik} < a_{ij}}} {}_p y_k^{(t)} \right)^N}{\sum_{\substack{k \in Y_p: \\ a_{ik} = a_{ij}}} {}_p y_k^{(t)}} \quad (6)$$

for each population X_p . Likewise Equation 4 expands to

$${}_p x_i^{(t+1)} = {}_p x_i^{(t)} \frac{\left(\sum_{\substack{k \in X_p: \\ {}_p u_k^{(t)} \leq {}_p u_i^{(t)}}} {}_p x_k^{(t)} \right)^H - \left(\sum_{\substack{k \in X_p: \\ {}_p u_k^{(t)} < {}_p u_i^{(t)}}} {}_p x_k^{(t)} \right)^H}{\sum_{\substack{k \in X_p: \\ {}_p u_k^{(t)} = {}_p u_i^{(t)}}} {}_p x_k^{(t)}} \quad (7)$$

In Section 2.1, we discussed a proof of an ϵ -bounds on convergence to the optimum in a two-population EGT CCEA using the maximum-of- N -collaborators evaluation procedure (Equation 3) in combination with tournament selection (Equation 4). We extended this to the more general M -population model described above. The theorem and proof for this extension are described in Section 6.

Consider this more general result. CCEAs do not operate over a joint population but rather over a set of marginal populations, each responsible for some portion of the joint solution. In the EGT infinite population model of CCEAs, these marginal populations are infinite in size — that is, they are (in the general case) $M > 1$ distributions rather than samples. Crucially, *univariate EDAs do exactly the same thing*: they break the joint solution into some $M > 1$ pieces (perhaps genes or clusters of genes), and each piece has a

marginal distribution over the possible forms that piece can take. Though we are not used to viewing EDAs’ marginal distributions as “infinite populations” in the CCEA sense, that is what they are. The EGT framework used in CCEA theory is not just an equivalent model for theoretical univariate EDAs, it *is a univariate EDA*.

This implies that univariate EDAs and “real” (as opposed to EGT) CCEAs are cousin algorithms. There are only two significant differences between them. First, CCEAs represent their marginal distributions with samples (the individuals), whereas EDAs commonly represent their marginal distributions with tables, histograms, or parameterized distributions (such as Gaussians). Second, because they have actual samples in their marginal distributions, CCEAs employ EC-style breeding operators to update those samples.¹

This connection between the two techniques may permit some cross-pollination. For example, the CCEA community has spent considerable energy to understand exactly why CCEA models exhibit pathologies. This work may prove fruitful in explaining similar issues in EDAs. Likewise, the EDA community has generated efficient algorithms which may improve on CCEA approaches. The EDA community has also moved from univariate to richer representations of the joint distribution: these might inform CCEAs as well.

This theorem suggests a new EDA algorithm, derived directly from the parallel generational CCEA algorithm (Algorithm 1), with optimal convergence properties as shown in the theorem. For each allele in each gene, the algorithm repeatedly constructs N individuals using that allele, then determines the maximum fitness among the N . In the theoretical model, this procedure is notionally iterated infinitely and the expected maximum-of- N fitness is assessed. The algorithm below is obviously an approximation as it uses only a single sample rather than the expected value. Then the distributions are adjusted to reflect tournament selection over the expected maximum allele fitnesses. We augment the theoretical algorithm with an optional PBIL-style “folding in” of the new distribution, using a parameter α . The algorithm, dubbed the Cooperative Estimation of Distribution Algorithm (CEDA), is:

ALGORITHM 3. *CEDA*

```

loop
  for each gene  $g$  do
    for each allele  $a \in g$  do
      for  $n$  from 1... $N$  do
        Construct an individual  $i$  using gene  $g$  fixed to  $a$ , and with other alleles selected at random under the remaining gene distributions.
        Evaluate( $i$ )
         $F_{g,a} \leftarrow \max$  fitness over all  $N$  individuals
      for each gene  $g$  do
        Let  $N_g$  be the distribution of alleles for  $g$  resulting from performing tournament selection of size  $H$  over mean allele fitnesses  $\bar{F}_{g,a} \forall a \in g$  (using Equation 4).
        Update gene distribution  $D_g \leftarrow (1 - \alpha)D_g + \alpha N_g$ 

```

¹We believe that standard one-population EAs may be cousins with EDAs consisting of one joint distribution over the whole space: and that such a joint-distribution EDA may be equivalent to the infinite population EA models popularized by Vose [18].

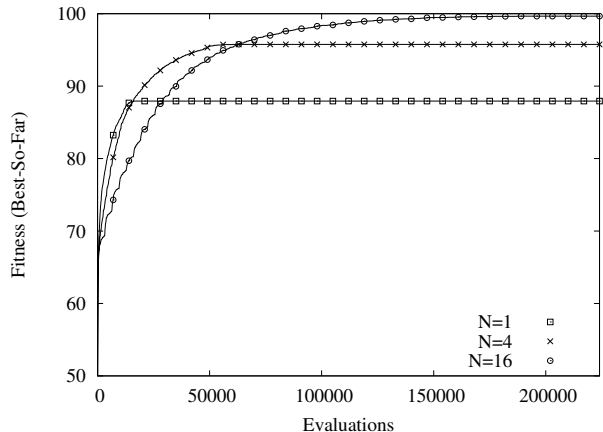


Figure 1: Mean best-so-far fitness versus number of evaluations, for N collaborators, $\alpha = 1$, on a 100-bit MaxOnes problem where $N = 1$, $N = 4$, and $N = 16$. Confidence intervals are omitted for clarity.

The algorithm is not particularly efficient: for each gene, we construct and test multiple individuals to assess that gene, but do not reuse their results to inform other gene distributions. As a result, in this form, it would be expected to require many more evaluations per round than PBIL, CGA, and UMDA—but we offer it here as an example of just how close CCEAs and univariate EDAs are.

4.1 Comparison and Results

This section presents some of the simulation results for CEDA. All experiments were averaged over 50 runs and use a tournament size of $H = 2$.

From the theorem in Section 6, we expect the probability of convergence to the global optimum for the EGT model to increase with number of collaborators. We likewise expect to see a similar effect as we increase the number of collaborators in CEDA. Figure 1 plots the convergence trajectory of CEDA with 1, 4, and 16 collaborators on MAXONES, using the definition:

DEFINITION 1. MAXONES: $\{0,1\}^n \rightarrow \mathbb{R}$ is defined as: $\text{MAXONES}(x) = \sum_i^n x_i$.

We observe that the solution quality improves as N increases: without a sufficiently large value of N , CEDA converges too rapidly, and ultimately prematurely, even on as simple a problem as MAXONES. The differences in means shown are highly statistically significant: using an unknown-variance 2-tailed t -test, all p -values are less than 10^{-15} .

We then compared CEDA with one univariate EDA (namely, PBIL) on the LEADINGONESBLOCKS problem, defined as:

DEFINITION 2. For $n \in \mathbb{R}$ and $b \in \{1, \dots, n\}$ so that $n/b \in \mathbb{N}$, LEADINGONESBLOCKS $_b$: $\{0,1\}^n \rightarrow \mathbb{R}$ is defined as: $\text{LEADINGONESBLOCKS}_b(x) = \sum_{i=1}^{n/b} \prod_{j=1}^{b-i} x_j$.

Figure 2 shows the results for the two, with $H = 2$, $N = 8$, $\alpha = 1$ for CEDA, and PBIL set to 100 samples, with a selection size of 10. CEDA clearly does not converge to the optimum fitness. (Very highly statistically significant with

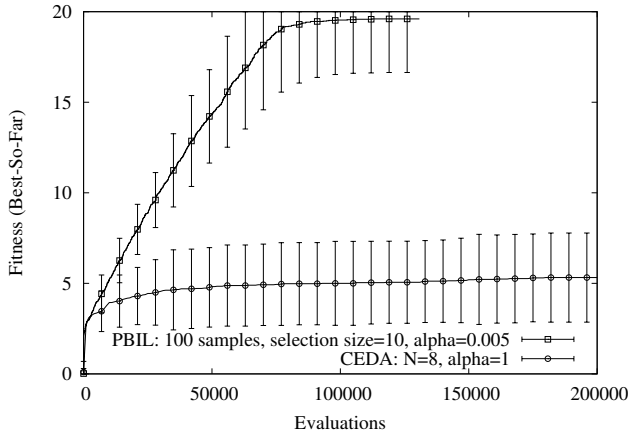


Figure 2: Best-so-far trajectory versus number of evaluations for LeadingOnesBlocks problem, 100 bits, 5-bit blocks (so that the ideal fitness is 20). PBIL (100 samples, selection size 10, $\alpha = 0.005$) is compared to CEDA ($H = 2, N = 8, \alpha = 1$). Bars show 95% confidence intervals.

$p < 10^{-65}$). Rather than dramatically increase the number of samples (N) in CEDA to achieve an optimal solution quality, we chose instead to apply a probability update rule ($\alpha < 1$) that is similar to the one used in PBIL, thus only gradually mixing in new values to the population distributions. Experiment suggests that a good value for CEDA is $\alpha = 0.05$. Figure 3 shows the results after applying $\alpha = 0.05$ to CEDA with $N = 8$. Using a smaller learning rate in this manner helps CEDA to achieve the optimal solution.

However, as the figures indicate, CEDA is still wasteful in terms of the number of evaluations. In Section 4 we noted that CEDA can be very inefficient as we construct and evaluate at least N unique individuals for *every single allele*, but do not reuse our results to inform the distributions for the other alleles. Figure 3 illustrates this problem.

The wastefulness of CEDA in terms of evaluations is bad news in practice, since we have found that CEDA quickly gets trapped in local optima on some problems when the number of collaborators is insufficient. We’ve found similar results on related test problems.

5. CONCLUSION

As it turns out, univariate EDAs and CCEAs are very similar methods. The standard univariate EDA data model is essentially identical to the CCEA infinite-population EGT model. Given this, it is possible to transfer theory and algorithms from CCEAs to EDAs and vice versa. As an example, we have extended an existing CCEA theoretical result, originally for the 2-population CCEA case, to the M -population case. This result shows that a given CCEA algorithm can be made to converge with a probability of $1 - \epsilon$ for any desired setting of ϵ regardless of the problem. We then transferred this result to the univariate EDA framework as well, demonstrating an EDA algorithm that approximates the theoretical model. However, the algorithm is very expensive in terms of evaluations, and as expected, it does not perform well compared to PBIL, one of several common univariate EDA algorithms.

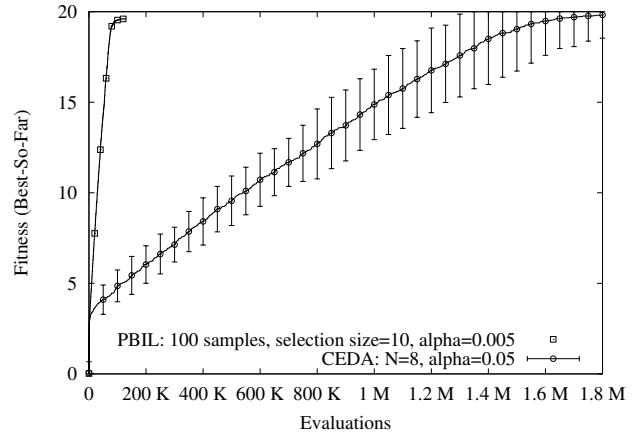


Figure 3: Best-so-far trajectory versus number of evaluations for LeadingOnesBlocks problem, 100 bits, 5-bit blocks (so that the ideal fitness is 20). PBIL (100 samples, selection size 10, $\alpha = 0.005$) is compared to CEDA ($H = 2, N = 8, \alpha = 0.05$). Bars show 95% confidence intervals (omitted on PBIL for clarity, see Figure 2 instead).

We believe this example serves as an illustration of what might be possible given the connection between these two techniques. This opens up many avenues for future work. For example, how would CCEA-style marginal “population” sample distributions compare to EDA-style marginal distributions? Could we convert algorithms such as CGA, PBIL, or UMDA to operate in the sample-distribution situation of CCEAs? Can further theory from either side inform the other? What would the equivalent of BOA be in a CCEA context? And importantly: are there other disparate methods which might actually be closely related?

6. PROOFS

As discussed in Section 2.1, it has been shown that a two-population cooperative coevolutionary EGT model, with tournament selection, and with maximum-of- M -collaborations fitness assessment, will converge to the optimum within some ϵ probability given a sufficiently large value of M . This model is the one described by Equations 4 and 5. The theorem here extends this convergence proof to an N -population cooperative coevolutionary Evolutionary Game Theory model.

Notation.

For the following proof, we apply the same notation used in Section 4. We describe it again here, for clarity.

We use X_p to denote the space of genotypes of the p -th population. For simplicity, we have $X_p = \{1, 2, 3, \dots, n_p\}$ where n_p is the number of genotypes for the p -th population. For each population p from 1 to M , and for each genotype i from 1 to n_p , we let $p x_i^{(t)}$ denote the ratio of individuals with genotype i in population p at generation t . We further define $Y_p = X_1 \times \dots \times X_{p-1} \times X_{p+1} \times \dots \times X_M$ as the joint space of all possible collaborators for an individual from population p (notice that X_p is missing).

Here we will deviate from our previous equations in our use of j . Now j will represent a *tuple* of genotypes chosen from various populations to collaborate with genotype i .

We will also extend ${}_p y_j$ to refer not to the proportion of genotype j in the second population (as was the case earlier) but rather to the proportion of collaborating tuple j in the joint collaboration space. That is, for tuple

$${}_p u_i^{(t)} = \sum_{j \in Y_p} a_{ij} \frac{{}_p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{ik} = a_{ij}}} {}_p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{ik} \leq a_{ij}}} {}_p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{ik} < a_{ij}}} {}_p y_k^{(t)} \right)^N \right) \quad (8)$$

$${}_p x_i^{(t+1)} = \frac{{}_p x_i^{(t)}}{\sum_{\substack{k \in X_p: \\ {}_p u_k^{(t)} = {}_p u_i^{(t)}}} {}_p x_k^{(t)}} \left(\left(\sum_{\substack{k \in X_p: \\ {}_p u_k^{(t)} \leq {}_p u_i^{(t)}}} {}_p x_k^{(t)} \right)^H - \left(\sum_{\substack{k \in X_p: \\ {}_p u_k^{(t)} < {}_p u_i^{(t)}}} {}_p x_k^{(t)} \right)^H \right) \quad (9)$$

LEMMA 1. Assume the populations for the EGT model are initialized at random based on a uniform distribution over all possible initial populations. Then, for any $\epsilon > 0$, there exists $\theta_\epsilon > 0$ such that

$$P\left(\min_{i=1..n_p} {}_p x_i^{(0)} \leq \theta_\epsilon\right) < \epsilon \quad (10)$$

$$P\left(\max_{i=1..n_p} {}_p x_i^{(0)} \geq 1 - \theta_\epsilon\right) < \epsilon \quad (11)$$

for all populations p from 1 to M .

PROOF. One method to sample the simplex Δ^n uniformly is described in [3] (pages 568–569): take $n - 1$ uniformly distributed numbers in $[0, 1]$, sort them, and use the differences between consecutive numbers (also, the difference between the smallest number and 0, and the difference between 1 and the largest number) as the coordinates for the point.

Let p be an arbitrary population from 1 to M . It follows that $({}_p x_i^{(0)})_{i=1..n_p}$ can be generated as the difference between $n - 1$ numbers generated uniformly in the range $[0, 1]$. It then follows that $\min_{i=1..n} {}_p x_i^{(0)}$ is the closest distance between two such numbers (and possibly including the boundaries 0 and 1).

Suppose $\gamma > 0$ is a sufficiently small number. We iterate over the $n_p - 1$ uniformly-distributed random numbers that are needed to generate an initial population $({}_p x_i^{(0)})_{i=1..n_p}$. The probability that the first number is not within γ of the boundaries 0 and 1 is $1 - 2\gamma$. The probability that the second number is not within γ of the boundaries, or of the first number, is less than or equal to $1 - 4\gamma$. In general, the probability that the k th number is not within γ of the boundaries, or of the first $k - 1$ numbers, is less than or equal to $1 - 2k\gamma$. Since the numbers are generated independently of one another, the probability that the closest pair of points (considering the boundaries) is farther apart than γ is:

$j \in Y_p$ with $j = (j_1, \dots, j_{p-1}, j_{p+1}, \dots, j_M)$, we use the notation ${}_p y_j^{(t)} = \prod_{v=1..M: v \neq p} v x_{j_v}^{(t)}$. Likewise, a_{ij} is the reward for genotype i when combined with collaborators in tuple j .

Likewise for clarity, we repeat Equations 6 and 7, which describe the formal model for CCEAs with N individuals. For each population p and each genotype $i \in p$:

$$\begin{aligned} P\left(\min_{i=1..n_p} {}_p x_i^{(0)} \leq \gamma\right) &= 1 - P\left(\min_{i=1..n_p} {}_p x_i^{(0)} > \gamma\right) \\ &= 1 - \prod_{i=1}^{n_p-1} (1 - 2i\gamma) \\ &\leq 1 - (1 - 2(n_p - 1)\gamma)^{n_p-1} \\ &\leq 1 - (1 - 2(n_{p^*} - 1)\gamma)^{n_{p^*}-1} \end{aligned}$$

where $n_{p^*} = \max_{i=1..M} n_i$. Given that

$$\lim_{\gamma \rightarrow 0} 1 - (1 - 2(n_{p^*} - 1)\gamma)^{n_{p^*}-1} = 0$$

it follows that for any $\epsilon > 0$ there exists $\theta_\epsilon > 0$ such that $P(\min_{i=1..n_p} {}_p x_i^{(0)} \leq \theta_\epsilon) < \epsilon$ for all populations $p_1..p_M$.

To prove Inequality 11, note that $\max_{i=1..n_p} {}_p x_i^{(0)} \geq 1 - \theta_\epsilon$ implies that all other ${}_p x_i$ ratios except for the maximum are smaller than θ_ϵ , which, as proven above, occurs with probability smaller than ϵ . \square

LEMMA 2. Assume the populations for the EGT model are initialized at random based on a uniform distribution over all possible initial populations. Then, for any $\epsilon > 0$, there exists $\eta_\epsilon > 0$ such that

$$\begin{aligned} P\left(\min_{p=1..M} \min_{i=1..n_p} {}_p x_i^{(0)} > \eta_\epsilon \wedge \max_{p=1..M} \max_{i=1..n_p} {}_p x_i^{(0)} < 1 - \eta_\epsilon\right) \\ \geq 1 - \epsilon \end{aligned}$$

In other words, there is an arbitrary probability that the initial populations contain reasonable values (not too close to either 0 or 1) for all proportions of genotypes.

PROOF. We apply Lemma 1 for $\frac{1 - \sqrt{1 - \epsilon}}{2}$, which is greater than 0. The specific value of η_ϵ for this proof equals the value of $\theta_{\frac{1 - \sqrt{1 - \epsilon}}{2}}$ from Lemma 1. It follows that:

$$\begin{aligned}
& P\left(\min_{p=1..M} \min_{i=1..n_p} p x_i^{(0)} > \eta_\epsilon \wedge \max_{p=1..M} \max_{i=1..n_p} p x_i^{(0)} < 1 - \eta_\epsilon\right) \\
&= \prod_{p=1..M} P\left(\min_{i=1..n_p} p x_i^{(0)} > \theta_{\frac{1-M\sqrt{1-\epsilon}}{2}} \wedge \max_{i=1..n_p} p x_i^{(0)} < 1 - \theta_{\frac{1-M\sqrt{1-\epsilon}}{2}}\right) \\
&= \prod_{p=1..M} \left(1 - P\left(\min_{i=1..n_p} p x_i^{(0)} \leq \theta_{\frac{1-M\sqrt{1-\epsilon}}{2}} \vee \max_{i=1..n_p} p x_i^{(0)} \geq 1 - \theta_{\frac{1-M\sqrt{1-\epsilon}}{2}}\right)\right) \\
&\geq \prod_{p=1..M} \left(1 - \left(P\left(\min_{i=1..n_p} p x_i^{(0)} \leq \theta_{\frac{1-M\sqrt{1-\epsilon}}{2}}\right) + P\left(\max_{i=1..n_p} p x_i^{(0)} \geq 1 - \theta_{\frac{1-M\sqrt{1-\epsilon}}{2}}\right)\right)\right) \\
&\geq \prod_{p=1..M} \left(1 - 2 \frac{1-M\sqrt{1-\epsilon}}{2}\right) \\
&= 1 - \epsilon \quad \square
\end{aligned}$$

THEOREM 1. *Given a joint reward system with a unique global optimum $a_{i_1^* i_2^* \dots i_M^*}$, for any $\epsilon > 0$ and any $H \geq 2$, there exists a value $N_\epsilon \geq 1$ such that the theoretical CCEA model in Equations 6–7 converges to the global optimum with probability greater than $(1 - \epsilon)$ for any number of collaborators N such that $N \geq N_\epsilon$.*

PROOF. We only use ϵ as a guarantee for the worst case scenario for the proportions of individuals in the initial populations. From Lemma 2, it follows that there exists $\eta_\epsilon > 0$ such that with probability at least $1 - \epsilon$, it holds that $\eta_\epsilon < p x_i^{(0)} < 1 - \eta_\epsilon$ for all genotypes i in all populations

p . In other words, with probability ϵ , the initial populations will not have any proportion of individuals that cover more than $1 - \eta_\epsilon$, nor cover less than η_ϵ of the entire population.

We will prove that there exists $N_\epsilon \geq 0$ such that the EGT model converges to the global optimum for any $N \geq N_\epsilon$ and for all initial configurations that satisfy $\eta_\epsilon < p x_{i^*}^{(0)} < 1 - \eta_\epsilon$ for all populations p . To this end, let α be the second highest element joint reward ($\alpha < a_{i^* j^*}$). It follows that $p u_i^{(t)} < \alpha$ for all $i \neq i^*$ in all populations p . This is because (by refining Equation 6):

$$\begin{aligned}
p u_i^{(t)} &= \sum_{j \in Y_p} a_{ij} \frac{p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{ik} = a_{ij}}} p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{ik} \leq a_{ij}}} p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{ik} < a_{ij}}} p y_k^{(t)} \right)^N \right) \\
p u_i^{(t)} &\leq \sum_{j \in Y_p} \alpha \frac{p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{ik} = a_{ij}}} p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{ik} \leq a_{ij}}} p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{ik} < a_{ij}}} p y_k^{(t)} \right)^N \right) \\
&\leq \alpha \sum_{j \in Y_p} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{ik} \leq a_{ij}}} p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{ik} < a_{ij}}} p y_k^{(t)} \right)^N \right) \leq \alpha
\end{aligned}$$

Next, we work on identifying a lower bound for $p u_{i^*}^{(t)}$. For simplicity, let i^* stand for i_p^* , and j^* stand for the optimal tuple of collaborators for i^* .

$$\begin{aligned}
p u_{i^*}^{(t)} &= \sum_{j \in Y_p} a_{i^* j} \frac{p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} \leq a_{i^* j}}} p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} < a_{i^* j}}} p y_k^{(t)} \right)^N \right) \\
&= a_{i^* j^*} \left(1 - \left(1 - p y_{j^*}^{(t)}\right)^N\right) + \\
&\quad \sum_{\substack{j \in Y_p: \\ j \neq j^*}} a_{i^* j} \frac{p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} \leq a_{i^* j}}} p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} < a_{i^* j}}} p y_k^{(t)} \right)^N \right)
\end{aligned}$$

We further refine the lower bound for ${}_p u_{i^*}^{(t)}$:

$$\begin{aligned}
{}_p u_{i^*}^{(t)} &= a_{i^* j^*} \left(1 - \left(1 - {}_p y_{j^*}^{(t)} \right)^N \right) + \\
&\quad \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} \geq 0}} a_{i^* j} \frac{{}_p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} {}_p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} \leq a_{i^* j}} {}_p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} < a_{i^* j}} {}_p y_k^{(t)} \right)^N \right) + \\
&\quad \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \frac{{}_p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} {}_p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} \leq a_{i^* j}} {}_p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} < a_{i^* j}} {}_p y_k^{(t)} \right)^N \right) \\
&\geq a_{i^* j^*} \left(1 - \left(1 - {}_p y_{j^*}^{(t)} \right)^N \right) + \\
&\quad \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \frac{{}_p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} {}_p y_k^{(t)}} \left(\left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} \leq a_{i^* j}} {}_p y_k^{(t)} \right)^N - \left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} < a_{i^* j}} {}_p y_k^{(t)} \right)^N \right) \\
&\geq a_{i^* j^*} \left(1 - \left(1 - {}_p y_{j^*}^{(t)} \right)^N \right) + \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \frac{{}_p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} {}_p y_k^{(t)}} \left(\sum_{\substack{k \in Y_p: \\ a_{i^* k} \leq a_{i^* j}} {}_p y_k^{(t)} \right)^N
\end{aligned}$$

Given that $\sum_{k=1}^m {}_p y_k^{(t)} = 1$, we further refine the previous inequality:

$$\begin{aligned}
{}_p u_{i^*}^{(t)} &\geq a_{i^* j^*} \left(1 - \left(1 - {}_p y_{j^*}^{(t)} \right)^N \right) + \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \frac{{}_p y_j^{(t)}}{\sum_{\substack{k \in Y_p: \\ a_{i^* k} = a_{i^* j}}} {}_p y_k^{(t)}} \left(1 - {}_p y_{j^*}^{(t)} \right)^N \\
&\geq a_{i^* j^*} \left(1 - \left(1 - {}_p y_{j^*}^{(t)} \right)^N \right) + \left(1 - {}_p y_{j^*}^{(t)} \right)^N \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \\
&= a_{i^* j^*} - \left(1 - {}_p y_{j^*}^{(t)} \right)^N \left(a_{i^* j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \right) \tag{12}
\end{aligned}$$

$$= a_{i^* j^*} - \left(1 - \prod_{\substack{r=1, \dots, M: \\ r \neq p}} r x_{j^*}^{(t)} \right)^N \left(a_{i^* j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \right) \tag{13}$$

The inequalities $\eta_\epsilon < r x_{j^*}^{(0)} < 1 - \eta_\epsilon$ hold for all initial populations r , as inferred earlier from Lemma 2. It follows from Equation 13 that:

$${}_p u_{i^*}^{(0)} \geq a_{i^* j^*} - \left(1 - \eta_\epsilon^{M-1} \right)^N \left(a_{i^* j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^* j} < 0}} a_{i^* j} \right) \tag{14}$$

However,

$$\lim_{N \rightarrow \infty} a_{i^*j^*} - \left(1 - \eta_\epsilon^{M-1}\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) = a_{i^*j^*} \quad (15)$$

Given that $a_{i^*j^*} > \alpha$, Equation 15 implies that there exists $N_p \geq 1$ such that

$$a_{i^*j^*} - \left(1 - \eta_\epsilon^{M-1}\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) > \alpha \quad (16)$$

for all $N \geq N_p$. From Equations 13 and 16, it follows that ${}_p u_{i^*}^{(0)} > \alpha$ for all $N \geq N_p$. Observe that N_p does not depend on the initial population p we considered this far.

Let $N_\epsilon = \max_{p=1..M} (N_p)$, and let $N \geq N_\epsilon$. Next, we show by induction by t (the number of iterations of the model, i.e. the number of generations) that the following inequalities hold for all populations p :

$$\begin{aligned} {}_p u_{i^*}^{(t)} &\geq a_{i^*j^*} - \left(1 - \eta_\epsilon^{M-1}\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) \\ {}_p x_{i^*}^{(t+1)} &\geq {}_p x_{i^*}^{(t)} \end{aligned}$$

At the first generation ($t = 0$), the first inequality holds (from Equation 14). For a population p , we combine this with the definition of N . It follows that ${}_p u_{i^*}^{(0)} > {}_p u_i^{(0)}$ for all $i \neq i^*$. As a consequence, ${}_p x_{i^*}^{(1)} = 1 - (1 - {}_p x_{i^*}^{(0)})^H > {}_p x_{i^*}^{(0)}$ (from Equation 6).

To prove the inductive step, it follows from Equation 13 and from the inductive hypothesis that

$$\begin{aligned} {}_p u_{i^*}^{(t+1)} &\geq a_{i^*j^*} - \left(1 - {}_p y_{j^*}^{(t+1)}\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) \\ &\geq a_{i^*j^*} - \left(1 - {}_p y_{j^*}^{(t)}\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) \\ &\dots \\ &\geq a_{i^*j^*} - \left(1 - {}_p y_{j^*}^{(0)}\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) \\ &\geq a_{i^*j^*} - \left(1 - \eta_\epsilon^M - 1\right)^N \left(a_{i^*j^*} - \sum_{\substack{j \in Y_p: \\ j \neq j^* \wedge a_{i^*j} < 0}} a_{i^*j} \right) \end{aligned}$$

Given the definitions of N and α , this also implies that ${}_p u_{i^*}^{(t+1)} > \alpha > {}_p u_i^{(t+1)}$ for all $i \neq i^*$. As a consequence, ${}_p x_{i^*}^{(t+1)} = 1 - (1 - {}_p x_{i^*}^{(t)})^H \geq {}_p x_{i^*}^{(t)}$ (from Equation 6).

Having shown that ${}_p x_{i^*}^{(t)}$ are monotonically increasing for all populations p , and given that they are all bounded between 0 and 1, it follows that they each converge to some value. Given that ${}_p u_{i^*}^{(t)} > {}_p u_i^{(t)}$ for all $i \neq i^*$ at each iteration, it follows that ${}_p x_{i^*}^{(t+1)} = 1 - (1 - {}_p x_{i^*}^{(t)})^H$ at each iteration as well. If ${}_p \bar{x}$ is the limit of the ${}_p x_{i^*}^{(t)}$ values when t goes to ∞ , then ${}_p \bar{x} = 1 - (1 - {}_p \bar{x})^H$, which implies that ${}_p \bar{x}$ is either 0 or 1. We can rule out the 0 limit because the values of ${}_p x_{i^*}^{(t)}$ are monotonically increasing and ${}_p x_{i^*}^{(0)} > \eta_\epsilon$. Thus, ${}_p x_{i^*}^{(t)}$ converges to 1 for all populations p . \square

7. REFERENCES

- [1] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994.
- [2] A. Bucci and J. Pollack. On identifying global optima in cooperative coevolution. In Hans-Georg Beyer et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*, pages 539–544. ACM, 2005.
- [3] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986.
- [4] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.
- [5] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5:215–247, 1999.
- [6] H. Mühlenbein and T. Manig. Convergence theory and applications of the factorized distribution algorithm. *Journal of Computing and Information Technology*, 7, 1999.
- [7] H. Mühlenbein and T. Manig. FDA: A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(1), 1999.
- [8] L. Panait. *The Analysis and Design of Concurrent Learning Algorithms for Cooperative Multiagent Systems*. PhD thesis, George Mason University, Fairfax, Virginia, 2006.
- [9] L. Panait, S. Luke, and J. Harrison. Archive-based cooperative coevolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 345–352. ACM, 2006.
- [10] L. Panait, K. Tuyls, and S. Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, 9(Mar):423–457, 2008.
- [11] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 511–518. Morgan Kaufmann, 2001.

- [12] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, volume I, pages 525–532, San Francisco, CA, 1999. Morgan Kaufmann, Publishers.
- [13] 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, 1994.
- [14] M. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.
- [15] G. R., Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, 1999.
- [16] R. Rastegar and A. Hariri. A step forward in studying the compact genetic algorithm. *Evolutionary Computation*, 14(3), 2006.
- [17] R. Rastegar, A. Hariri, and M. Mazoochi. A convergence proof for the population based incremental learning algorithm. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 2005.
- [18] M. D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, USA, 1998.
- [19] R. P. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, Virginia, 2004.
- [20] Q. Zhang. On the convergence of a factorized distribution algorithm with truncation selection. *Complexity*, 9(4), 2003.
- [21] Q. Zhang and H. Mühlenbein. On the convergence of a class of estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2), April 2004.