

Cooperative Coevolution and Univariate Estimation of Distribution Algorithms

[Extended Abstract]

By Anonymous

Abstract

In this paper, we discuss a curious relationship between Cooperative Coevolutionary Algorithms (CCEAs) and Univariate EDAs. Inspired by the theory of CCEAs, we also present a new EDA with theoretical convergence guarantees, and some preliminary experimental results in comparison with existing Univariate EDAs.

1 Introduction

Over the last decade, the field of estimation of distribution algorithms (EDAs) has led to variety of interesting models and tools which have several advantages over their related traditional evolutionary algorithms (EAs). In particular, EDAs possess a unique flexibility to efficiently represent interactions between variables in higher-order problems.

...

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 other individuals in the system.

The most common coevolutionary frameworks are *one-population competitive*, *two-population competitive*, and *n-population cooperative* arrangements. In the one-population competitive coevolutionary algorithm, individuals in a single population are assessed by pitting them against other individuals in the same population, often in a game (for example, evolving checker players [3]). In the two-population competitive arrangement, individuals from one population are pitted against individuals against an opposing population. Here, typically only one population contains solutions of interest to the experimenter; the second population serves as a foil to push the first population towards robust solutions (for example, a sorting networks versus sorting problems [5]).

In this paper, we focus on *n-population cooperative* arrangements, popularly known as Cooperative Coevolutionary Algorithms (CCEAs) [16, 17]. In CCEAs, the solution space is broken into some n subsolution spaces, and each subsolution 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 either generational or (less commonly) steady-state, and often take one of two forms: *serial* versus *parallel* algorithms. In a serial algorithm, each population is evaluated and updated in turn, round-robin. In the parallel algorithm, all of the populations are evaluated before any of them is bred. Here we show the two generational versions of these algorithms:

```

loop (ALGORITHM 1. Serial Generational CCEA)
  for each population  $p \in P$  do
    for each individual  $i \in p$  do
      Evaluate( $i, p, P$ )
    Breed whole population  $p$ 

```

```

loop (ALGORITHM 2. Parallel Generational CCEA)
  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 coevolutionary research has focused on the specifics of the Evaluate(i, p, P) function. Choice of evaluation procedure in CCEAs is known to lead to a variety of 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.

3 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 has a (typically) finite set of genotypes, and that all populations have an infinite number of individuals. Much EGT work in cooperative coevolution has focused on two populations; therefore, we will focus on the two-population case in this section. **The first population is represented by a vector x where x_i indicates the proportion of genotype i in the population.** Likewise, the second population is represented by the vector y . We also assume there 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 [21] 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 so that each genotype i has fitness u_i . Likewise, we will use w for the second population. Wiegand 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 :

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

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 discovered that this “complete mixing” model could converge towards local suboptima surrounding Nash Equilibria in the joint space: if a suboptima basin were large and broad, the system would collect at its peak rather than at another tall but narrow 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 a collaborator those individuals were. That is, the fittest individuals tended to be jacks-of-all-trades, doing reasonably well with the average collaborator, rather those which performed optimally when paired with the optimal collaborator (but perhaps poorly on average). Wiegand termed this pathology *relative overgeneralization*.

Later research has shown that the system *will* converge if we change the fitness assessment procedure. The intuitive solution is to base the fitness of individuals in a population not on average collaboration but rather on the maximum performance over all collaborations. Following Panait [11] 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 [11] 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:

$$x_i^{(t+1)} = x_i^{(t)} \frac{\left(\sum_{\forall k: u_k^{(t)} \leq u_i^{(t)}} x_k^{(t)} \right)^H - \left(\sum_{\forall k: u_k^{(t)} < u_i^{(t)}} x_k^{(t)} \right)^H}{\sum_{\forall k: u_k^{(t)} = u_i^{(t)}} x_k^{(t)}} \quad (4)$$

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

This curious equation is a result of order statistics. 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 remaining elements 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 can be improved somewhat. Panait, [TuylsThuyles](#), and Luke [13] have provided a weakened convergence proof for when the evaluation procedure for a genotype is to take its maximum performance when paired N times with randomly chosen collaborators, and using tournament selection (Equation 4), both common practice in real

CCEAs. The proof shows that for any probability ϵ there exists a size N which is guaranteed to achieve convergence within ϵ .

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

$$\begin{aligned}
 u_i^{(t)} &= \sum_j a_{ij} y_j^{(t)} \frac{\left(\sum_{\forall k: a_{ik} \leq a_{ij}} y_k^{(t)}\right)^N - \left(\sum_{\forall k: a_{ik} < a_{ij}} y_k^{(t)}\right)^N}{\sum_{\forall k: a_{ik} = a_{ij}} y_k^{(t)}} \\
 w_i^{(t)} &= \sum_i a_{ij} x_i^{(t)} \frac{\left(\sum_{\forall k: a_{kj} \leq a_{ij}} x_k^{(t)}\right)^N - \left(\sum_{\forall k: a_{kj} < a_{ij}} x_k^{(t)}\right)^N}{\sum_{\forall k: a_{kj} = a_{ij}} x_k^{(t)}}
 \end{aligned} \tag{5}$$

Note the similarity to Equation 4. This equation is again a result from order statistics due to use of “max”. In the first subequation, for example, the fractional term and the $y_j^{(t)}$ together indicate the probability that a given pairing $\langle i, j \rangle$ will provide the highest reward for a given individual i out of N such pairings. This is then multiplied by the reward a_{ij} and summed to compute the expected maximum reward for i when doing N pairings.

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 rapidly to the optimum; but may require more total number of evaluations than is realistic given the evaluation budget. Thus recent work [2, 12] 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 *entire* collaborating population would provide. As it turns out, this archive size can be very small, resulting in a significant reduction in evaluations.

4 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 in the future.

An important design decision for EDAs is to select a representation for the probability distribution. An obvious problem is that the distribution of an infinite population, over the entire solution space, is of high dimensionality and complexity. Early on, one common approach was to break the joint distribution into separate distributions per-allele. That is, we assume an individual consists of a set of alleles, and for each allele, we maintain a distribution of probabilities of the gene settings for that allele. In the simplest case, if the individual is a boolean vector, then each allele distribution could be represented by a single number $[0, 1]$ indicating the probability of choosing a 1 instead of a 0. If the individual is a vector of floating-point numbers, each allele may possibly be represented as a gaussian distribution over the range of possible values. Common Univariate EDAs include the Univariate Marginal Distribution Algorithm (UMDA) [7], the Compact Genetic Algorithm (CGA) [18], and Population-Based Incremental Learning (PBIL) [1]. To illustrate, here is the pseudocode for PBIL, a simple but effective univariate EDA:

loop

(ALGORITHM 3. *PBIL*)

for q from 1 to Q **do**

 Create an individual i_q by choosing a gene at random under each allele distribution.

 Evaluate(i_q)

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

for each allele distribution a **do**

 Change the distribution to reflect the distribution of genes for that allele
 among the R best individuals

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 particular 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 alleles. Perhaps best known are variations of the Bayesian Optimization Algorithm (BOA) [15, 14], 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 [9], and for non-separable problems when augmented with a simulated-annealing-like Boltzman selection [8, 10]. A theoretical infinite-population version of UMDA has also been shown to converge to the optimum [22, 23]. Rastegar and Hariri have shown convergence to local optima for PBIL [20] and CGA [19].

5 EDAs and the EGT Infinite Population Model of CCEAs

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 distributions rather than samples. We wish to point out here that, crucially, *univariate EDAs do exactly the same thing*. We are not used to viewing EDAs’ marginal distributions as “infinite populations” in the CCEA sense, but that is precisely what they are. The EGT framework that is 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). And second, because they have actual samples in their marginal distributions, CCEAs employ EC-style breeding operators to update those samples. **This is similar to the cousin-relationship between EC algorithms and EDAs** This is in the same sense as EC algorithms being cousins with EDAs consisting of one joint distribution over the whole space.

This connection between the two techniques may permit some cross-pollination. For example, the CCEA community has expended 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 existing CCEA approaches, and may transfer theory as well. The EDA community has also moved from univariate to bivariate and bayesian network representations of the joint distribution: perhaps these might inform CCEAs as well.

5.1 A Proof and an Algorithm

In Section 3 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). As a first example of the potential for cross-pollination, we have extended this two-population proof to the M -population case. We include the theorem and its proof in Section 7.

This theorem suggests a new EDA algorithm, derived directly from the parallel generational CCEA algorithm (Algorithm 2), with optimal convergence properties as shown in the theorem. The algorithm is not particularly efficient: for each allele, we construct and test multiple individuals to assess that allele, but do not reuse their results to inform other allele distributions. As a result, in its proven 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.

The algorithm is:

```

loop (ALGORITHM 4. CMLA)
  for each allele distribution  $a$  do
    for each gene value  $g \in a$  do
      for N times do
        Construct an individual  $i$  using allele  $a$  fixed to  $g$ , and with other genes
          selected at random under the remaining allele distributions.
        Evaluate( $i$ )
      for each allele distribution  $a$  do
        Change the distribution to reflect performing tournament selection of size  $H$ 
          over the genes in  $a$  (using Equation 4).

```

5.2 Comparison and Results

This section presents some of the simulation results for CMLA in terms of both fitness evaluations and generations. All experiments were averaged over 50 runs and utilize a tournament size of $H = 2$.

From the theorem in Section 7, we expect that the probability of convergence to the global optimum for the EGT model increases with the number of collaborators. We expect to see a similar effect as we increase the number of collaborators in the CMLA. Figure 1 plots the convergence trajectory of CMLA with 1, 4, and 16 collaborators on MAXONES, a well known linear function with the following definition:

Definition 1. MAXONES: $\{0, 1\}^n \rightarrow \mathbb{R}$ is defined as:

$$\text{MAXONES}(x) = \sum_i^n x_i \tag{6}$$

In figure 1, observe that the solution quality increases as N increases. However, in Section 5.1 we have noted that CMLA may be very inefficient as we construct many individuals and evaluate them for each allele, but do not reuse the results to inform the distributions for the other alleles. To illustrate this heavy cost of evaluations, Figure 2 contains exactly the same set of experiments as those in Figure 1, plotted against the number of evaluations instead of the number of generations.

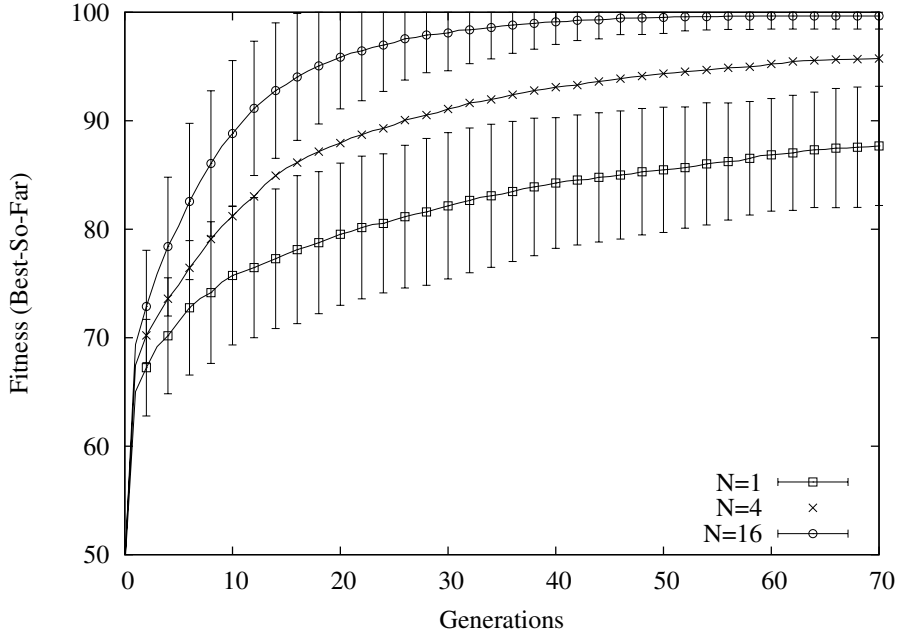


Figure 1: Best-so-far fitness versus *generations* for N collaborators on a 100-bit MAXONES problem where $N = 1$, $N = 4$, and $N = 16$. Bars show 95% confidence intervals.

The wastefulness of CMLA in terms of evaluations is bad news in practice, since we have found that CMLA quickly gets trapped in local optima on some problems when the number of collaborators is insufficient. Figure 4 shows a comparison between PBIL and CMLA ($N = 8$) on the well-known LEADINGONESBLOCKS problem [6]. The LEADINGONESBLOCKS problem counts the number of leading blocks of size b in x that have all bits set to 1. It has the following definition:

Definition 2. For $n \in \mathbb{B}$ and $b \in \{1, \dots, n\}$ so that $n/b \in \mathbb{N}$, $\text{LEADINGONESBLOCKS}_b : \{0, 1\}^n \rightarrow \mathbb{R}$ is defined as:

$$\text{LEADINGONESBLOCKS}_b(x) = \sum_{i=1}^{n/b} \prod_{j=1}^{b \cdot i} x_j \quad (7)$$

In figures 4 and 3, CMLA clearly does not converge to the optimum fitness. Rather than radically increase the number of samples in CMLA to achieve an optimal solution quality, we chose instead to apply a probability update rule that is similar to the probability update rule that is used in PBIL. This probability update rule mixes the probability values of for time t with the newly computed probability values for time $t + 1$ according to a learning rate [1]. Figures 6 and 5 show

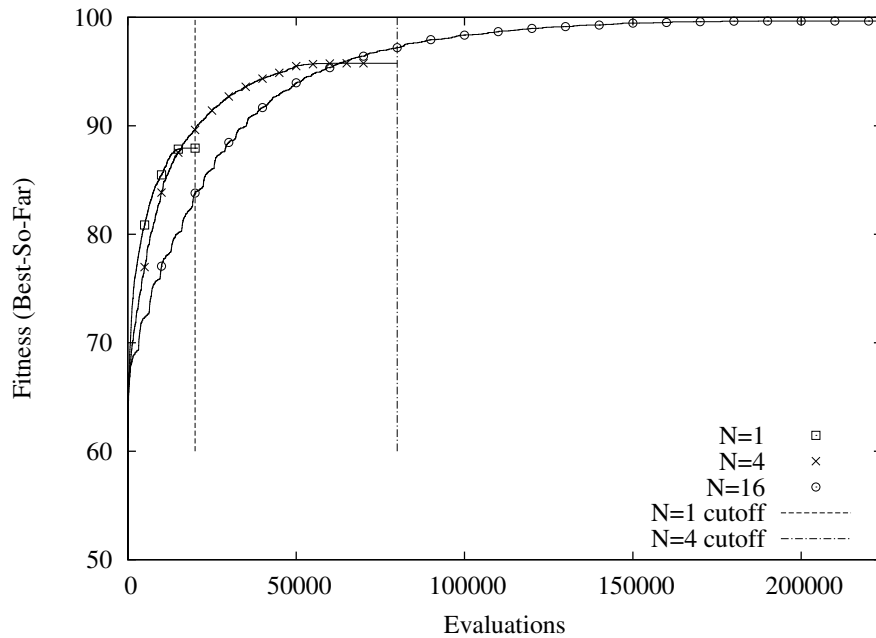


Figure 2: Best-so-far fitness versus *evaluations* for N collaborators on a 100-bit MAXONES problem where $N = 1$, $N = 4$, and $N = 16$. Bars show 95% confidence intervals, and vertical dotted lines show the evaluation budgets. A comparison of this figure to Figure 1 illustrates the heavy cost of adding more collaborators. While the solution quality increases modestly as also shown in Figure 1, the number of evaluations required per generation also increases significantly.

the results after applying a learning rate of $\alpha = 0.05$ to CMLA for $N = 8$. Using a smaller learning rate in this manner helps CMLA to achieve the optimal solution. However, as the figures indicate, CMLA is still wasteful in terms of the number of evaluations. Furthermore, while this result is interesting, the use of this update rule deviates somewhat from the formulation we have proven in Section 7 to have optimal convergence guarantees.

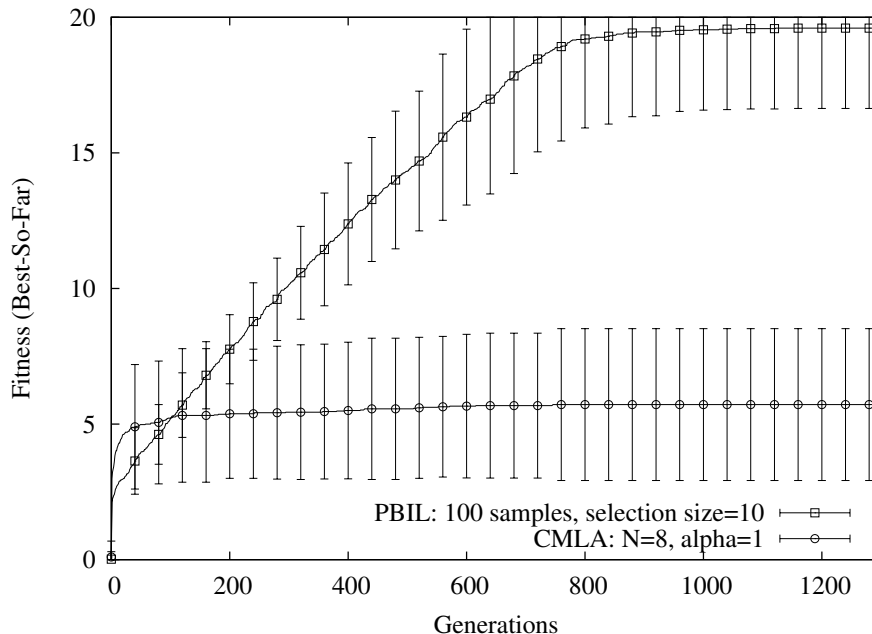


Figure 3: Best-so-far trajectory versus *generations* for LEADINGONESBLOCKS problem, 100 bits, 5-bit blocks (so that the ideal fitness is 20). For CMLA, $\alpha = 1$. Bars show 95% confidence intervals. PBIL is compared to CMLA with $N = 8$ collaborators.

6 Conclusion

A conclusion!

7 Proofs

As discussed in Section 3, 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 5 and 4. The theorem here extends this convergence proof to the N -population cooperative coevolutionary EGT model.

Notation We use (X_i) to denote the space of genotypes of the i -th population (for simplicity, we have $X_i = \{1, 2, 3, \dots, n_i\}$ where n_i is the number of genotypes for the i -th population. We further

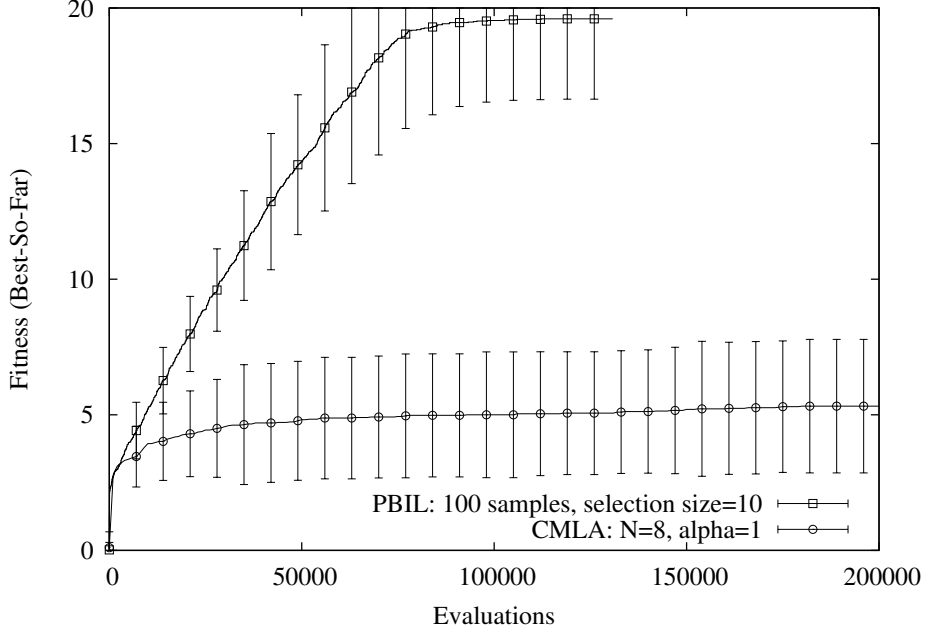


Figure 4: Best-so-far trajectory versus *evaluations* for LEADINGONESBLOCKS problem, 100 bits, 5-bit blocks (so that the ideal fitness is 20). For CMLA, $\alpha = 1$. Bars show 95% confidence intervals. PBIL is compared to CMLA with $N = 8$ collaborators.

define $X_{-i} = X_1 \times \dots \times X_{i-1} \times X_{i+1} \times \dots \times X_M$ as the joint space of all possible collaborators for an individual from population i (Notice that X_i was missing). For each population p from 1 to M , 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 .

Here we will deviate from our previous equations in our use of j . Now j will represent a *tuple* of individuals chosen from various populations to collaborate with individual i . We will also extend 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 $j \in X_{-i}$ (the i -th population is missing from j) with $j = (j_1, \dots, j_{i-1}, j_{i+1}, \dots, j_M)$, we use the notation $y_j^{(t)} = \prod_{v=1..M: v \neq p} v x_{j_v}^{(t)}$. Likewise, a_{ij} is the reward received for genotype i when combined with collaborators described by tuple j .

The formal model for CCEAs with N individuals becomes:

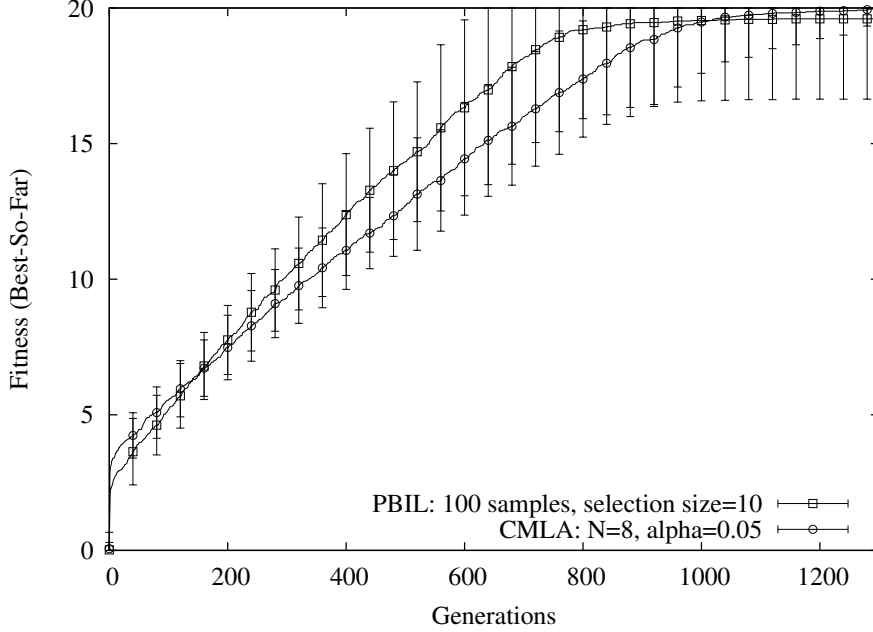


Figure 5: Best-so-far trajectory versus *generations* for LEADINGONESBLOCKS problem, 100 bits, 5-bit blocks (so that the ideal fitness is 20). For CMLA, $\alpha = 0.05$. Bars show 95% confidence intervals. PBIL is compared to CMLA with $N = 8$ collaborators.

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

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

for each population p and each genotype i in p . Note that the second equation is identical (tournament selection).

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 .

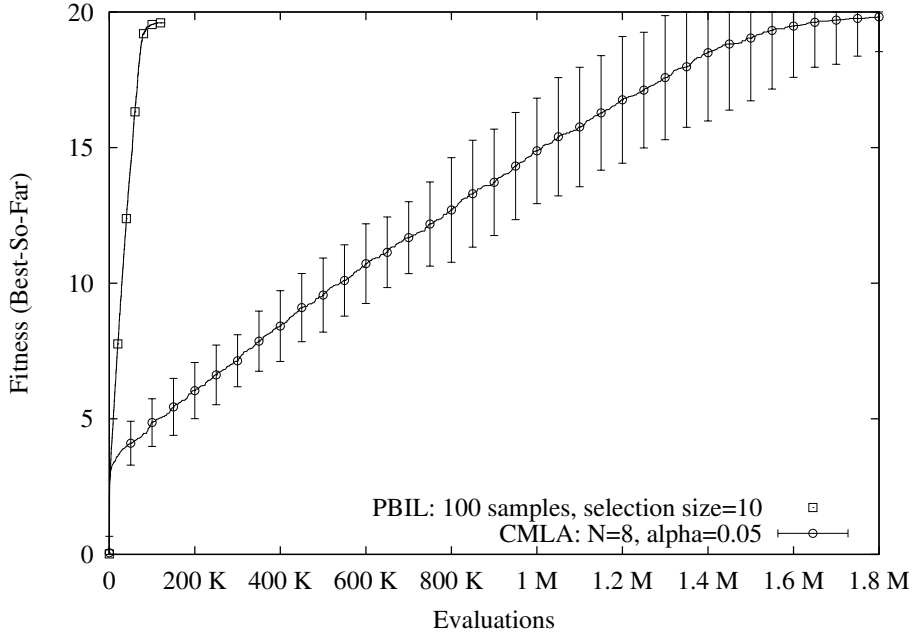


Figure 6: Best-so-far trajectory versus *evaluations* for LEADINGONESBLOCKS problem, 100 bits, 5-bit blocks (so that the ideal fitness is 20). For CMLA, $\alpha = 0.05$. Bars show 95% confidence intervals. PBIL is compared to CMLA with $N = 8$ collaborators.

Proof. One method to sample the simplex Δ^n uniformly is described in [4] (pages 568–569): take $n - 1$ uniformly distributed numbers in $[0, 1]$, then sort them, and finally 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 $\left(p x_i^{(0)}\right)_{i=1..n_p}$ can be generated as the difference between $n - 1$ numbers generated uniformly in $[0, 1]$. It follows that $\min_{i=1..n} p x_i^{(0)}$ is the closest distance between two such numbers (and possibly the boundaries 0 and 1).

Suppose $\gamma > 0$ is a small number. We iterate over the $n_p - 1$ uniformly-distributed random numbers that are needed to generate an initial population $\left(p x_i^{(0)}\right)_{i=1..n_p}$. 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 $1 - 2k\gamma$. Given that 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 equal to

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

where $n_{p^*} = \max_{i=1..M} n_i$ **Would you guys please double-check this last inequality which introduces n_{p^*} ?** 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 $P\left(\min_{i=1..n_p} p x_i^{(0)} \leq \theta_\epsilon\right) < \epsilon$ for all populations p from 1 to M .

To prove Inequality 11, consider 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 to 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*

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

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

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

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 8–9 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 populations 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 ${}_p u_i^{(t)} \leq \alpha$ for all $i \neq i^*$ in all populations p . Here's why (by refining Equation 8):

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

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^* .

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

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

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

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

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

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

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

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

However,

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

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

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

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 :

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

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, ${}_p x_{i^*}^{(1)} = 1 - \left(1 - {}_p x_{i^*}^{(0)}\right)^H > {}_p x_{i^*}^{(0)}$ (from Equation 8).

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

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

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, ${}_p x_{i^*}^{(t+1)} = 1 - \left(1 - {}_p x_{i^*}^{(t)}\right)^H \geq {}_p x_{i^*}^{(t)}$ (from Equation 8).

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 ${}_p x_{i^*}^{(t+1)} = 1 - \left(1 - {}_p x_{i^*}^{(t)}\right)^H$ at each iteration as well. If $\bar{p}x$ is the limit of the ${}_p x_{i^*}^{(t)}$ values when t goes to ∞ , then $\bar{p}x = 1 - (1 - \bar{p}x)^H$, which implies that $\bar{p}x$ 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

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] K. Chellapilla and D. B. Fogel. Evolving neural networks to play checkers without relying on expert knowledge. *IEEE-NN*, 10(6):1382, 1999.
- [4] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986.
- [5] D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Artificial Life II, SFI Studies in the Sciences of Complexity*, 10:313–324, 1991.
- [6] T. Jansen and R. P. Wiegand. The cooperative coevolutionary (1+1) ea. *Evolutionary Computation*, 12(4):405–434, 2004.
- [7] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.
- [8] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5:215–247, 1999.
- [9] H. Mühlenbein and T. Manig. Convergence theory and applications of the factorized distribution algorithm. *Journal of Computing and Information Technology*, 7, 1999.
- [10] H. Mühlenbein and T. Manig. FDA: A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(1), 1999.
- [11] L. Panait. *The Analysis and Design of Concurrent Learning Algorithms for Cooperative Multiagent Systems*. PhD thesis, George Mason University, Fairfax, Virginia, 2006.
- [12] L. Panait, S. Luke, and J. Harrison. Archive-based cooperative coevolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO-2006*, pages 345–352. ACM, 2006.
- [13] 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.
- [14] 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.
- [15] 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, Orlando, FL, 1999. Morgan Kaufmann Publishers, San Fransisco, CA.

- [16] 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.
- [17] M. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.
- [18] G. R., Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, 1999.
- [19] R. Rastegar and A. Hariri. A step forward in studying the compact genetic algorithm. *Evolutionary Computation*, 14(3), 2006.
- [20] 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.
- [21] R. P. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, Virginia, 2004.
- [22] Q. Zhang. On the convergence of a factorized distribution algorithm with truncation selection. *Complexity*, 9(4), 2003.
- [23] 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.