A Rigorous Theoretical Framework for Measuring Generalisation of Co-evolutionary Learning [a]

Xin Yao (X.Yao@cs.bham.ac.uk)
CERCIA (The Centre of Excellence for Research in Computational Intelligence and Applications)
School of Computer Science, The University of Birmingham
Edgbaston, Birmingham B15 2TT, United Kingdom
and
NICAL (Nature Inspired Computation and Applications Laboratory)
The University of Scince and Technology of China (USTC)
Hefei, Anhui 230027, China

[a] Joint work with Siang Yew Chong and Peter Tiño.

Overview of the Talk

1. Introduction
2. Theoretical Framework
3. Examples of Generalization Framework
4. Estimating Generalization in Co-evolutionary Learning
5. Concluding Remarks

Introduction: Evolutionary Learning

Very straightforward conceptually!

1. Initialise population, $X(t = 1)$.
2. Evaluate fitness of each population member.
3. Select parents from $X(t)$ based on fitness.
4. Generate offspring from parents to obtain $X(t + 1)$
5. Repeat steps (2-4) until some termination criteria are met.

Two Approaches to Evolutionary Learning

Things might get a little trickier.

Michigan Approach: Holland-style learning classifier systems (LCS), where each individual is a rule. The whole population is a complete (learning) system.

Pitt Approach: Each individual is a *complete* system.

This talk deals only with the Pitt-style evolutionary learning since it is more widely used.

Current Practice in Evolutionary Learning

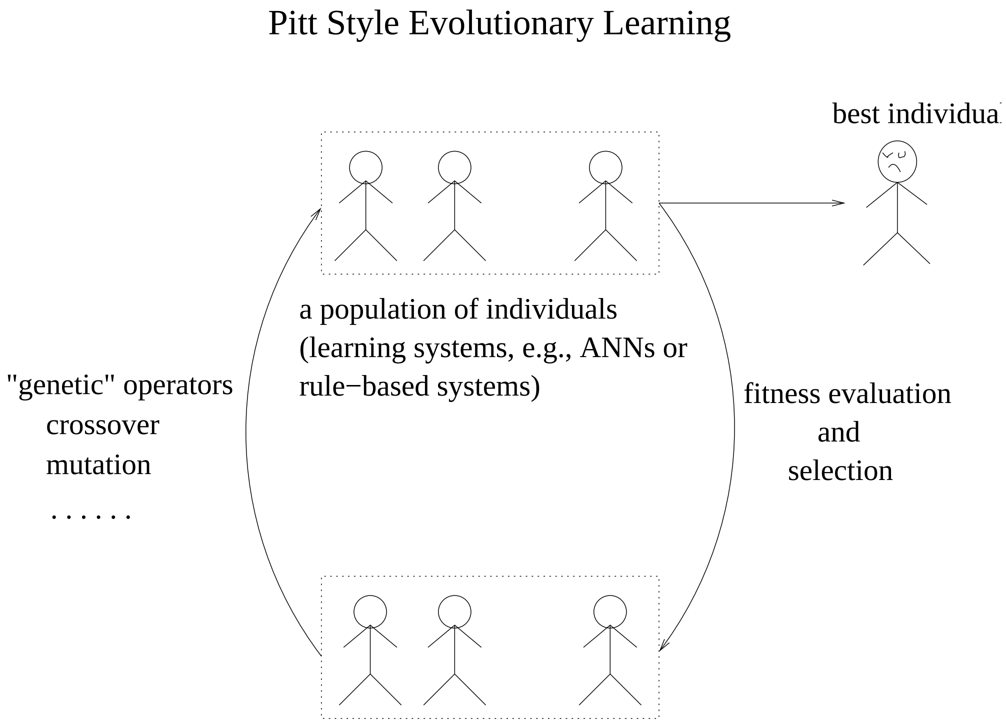

Figure 1: A general framework for Pitt style evolutionary learning.

Fitness Evaluation

1. Based on the training error.
2. Based on the training error and complexity (regularisation), e.g.,

$$\frac{1}{fitness} \propto error + \alpha * complexity$$

What If No Error Function Is Available

1. Or, we don't know how to obtain the fitness function required to evaluate the fitness of a population member, e.g.,, if we want to evolve game-playing strategies.
2. In other words, the exact teacher/target information is unavailable.

Well ... We have Co-evolutionary Learning

1. Initialise population, $X(t = 1)$.
2. *Evaluate fitness through interactions between population members.*
3. Select parents from $X(t)$ based on fitness.
4. Generate offspring from parents to obtain $X(t + 1)$
5. Repeat steps (2-4) until some termination criteria are met.

Co-evolutionary Learning In One Slide

1. There has been a huge body of literature on co-evolution, especially since Hillis's seminal work in 1991. If we google "Co-evolutionary Learning," we would get more than $200k+$ hits. There is also an IEEE CIS Task Force on Co-evolution. ...
2. Many issues have been raised and discussed: robustness, cycles, mediocre stable states, ...
3. It is a great idea, but sometimes it just does not do what you hope it would do — frustrating!
4. There is only a small body of literature on theoretical aspects of co-evolution. We need more vigourous theories to move the research forward.

A Simple Research Question

- If I invent a wonderful co-evolutionary learning algorithm and use it to co-evolve a really intelligent game-playing strategy (e.g., for chess, car racing, iterated prisoner's dilemma, or others), how do I know it would perform well against a new opponent that it has never seen before?
- Can we say anything at all about the ability (performance) of our co-evolved solutions in a new and unseen environment?
- Sounds like **generalisation** to me.

Generalisation? We Know That!

1. There have been various discussions about robustness of co-evolved solutions.
2. However, we still do not have any *quantitative* analysis of generalisation performance, e.g., an absolute quality measure for co-evolved solutions.
3. It is still very hard to compare performance between different co-evolutionary learning algorithms (when applied to a problem).

An Early Attempt

[a]

An empirical approach to estimate generalisation of co-evolved solutions:

1. Sample a large number of random test strategies.
2. Co-evolved strategies compete against these test strategies.
3. Generalization is taken to be the average performance against these test strategies.
4. Note that the number of test strategies should be significantly larger than what co-evolution can search (so that the vast majority of the test strategies are unseen by co-evolved strategies).

[a] P. Darwen and X. Yao, "On evolving robust strategies for iterated prisoner's dilemma," In Progress in Evolutionary Computation, Lecture Notes in Artificial Intelligence, Vol. 956, Springer-Verlag, Heidelberg, Germany, pp.276-292, 1995

The Need for a General Theoretical Framework

1. Although the empirical estimation can give us some information, it is unknown how accurate the estimate is to the true value.
2. What is needed is a theoretical framework that would enable us to define/compute the true generalisation and the accuracy of an estimation.
3. But how? Learn from others!

Theoretical Framework — Problem Formulation

1. In co-evolutionary learning, we consider the performance (quality) of a solution relative to other solutions.
2. This is achieved through interactions or comparisons between solutions.
3. This can be framed in the context of game-playing.

Theoretical Framework — $G_i(j)$

1. Let us have two strategies i and j.
2. A game is played between these two strategies.
3. Let $G_i(j)$ be the game outcome of strategy i playing against strategy j.
4. Similarly, let $G_j(i)$ be the outcome of strategy j playing against strategy i.
5. Strategy i is said to *solve* the test provided by strategy j if $G_i(j) \geq G_j(i)$. Strategy i *wins* against j if $G_i(j) > G_j(i)$.

True Generalization Performance

1. Given a co-evolved strategy i, let test strategies j be obtained from strategy space $\mathbf{S}$. The true generalization performance of strategy i, G_i, is:

$$G_i = E_{P_1(j)}[G_i(j)] = \int_{\mathbf{S}} G_i(j) P_1(j) \mathrm{dj}, \tag{1}$$

where G_i is the expectation of strategy i's performance against j, $G_i(j)$, w.r.t. distribution $P_1(j)$ over strategy space $\mathbf{S}$.

2. A simplified form:

$$G_i = \frac{1}{M} \sum_{j}^{M} G_i(j), \tag{2}$$

which is simply its average performance against all strategies j.

Estimating Generalization Performance

1. In reality, it is very hard or even impossible to compute the true generalisation performance using either Equation 1 or Equation 2.
2. We have to reply on estimation using a random sample of N test strategies j, S_N.
3. The estimated generalization performance of strategy i is given by:

$$\hat{G}_i(S_N) = \frac{1}{N} \sum_{j \in S_N} G_i(j), \tag{3}$$

   where S_N is the sample of N test strategies randomly drawn from $\mathbf{S}$ (we'll use notation $\hat{G}_i$ for $\hat{G}_i(S_N)$).

How Good Is the Estimation

1. We want to know how accurate the estimate $\hat{G}_i$ is compared to G_i, i.e., how small $|\hat{G}_i - G_i|$ is.
2. We don't know G_i, so we can't compute $|\hat{G}_i - G_i|$. So frustrating!
3. Fortunately, we can make a statistical claim as to how confident we are that $|\hat{G}_i - G_i| \leq \epsilon$.

Chebyshev's Theorem

Chebyshev's Theorem[a]: *Consider a random variable U distributed according to the probability density $p(u)$. Given a positive number $a > 0$, we can bound the probability that $U \leq -a$ or $U > a$, i.e., the probability that U falls outside $[-a, +a]$, by*

$$P(|U| \geq a) \leq \frac{E[U^2]}{a^2},$$

where $E[U^2]$ is the mean of the new random variable $V = U^2$ with respect to p.

[a] B. V. Gnedenko and G. V. Gnedenko, *Theory of Probability*, Taylor & Francis, 1998

Bound for Estimation Accuracy

Applying Chebyshev's Theorem, we derive the following:

$$P(|\hat{G}_i - G_i| \geq \epsilon) \leq \frac{\sigma_i^2}{N \cdot \epsilon^2} \tag{4}$$

More Usable Bound

1. In general, for the random variable $G_i(j)$ distributed over the interval $[G_{\text{MIN}}, G_{\text{MAX}}]$, $\sigma_{\text{MAX}} = (G_{\text{MAX}} + G_{\text{MIN}})/2$[a].

2. With this, we obtain the following lemma:

   Lemma 0.1 *For a strategy i, let $\hat{G}_i$ be the estimated generalization performance with respect to N random test strategies and G_i be the true generalization performance. Consider the absolute difference $|\hat{G}_i - G_i|$, which is a random variable with distribution P_N taken on a compact interval $[G_{\text{MIN}}, G_{\text{MAX}}]$ of length $R = G_{\text{MAX}} - G_{\text{MIN}}$. Then, for any positive number $\epsilon > 0$:*

$$P_N(|\hat{G}_i - G_i| \geq \epsilon) \leq \frac{R^2}{4N \cdot \epsilon^2}. \tag{5}$$

[a] H. I. Jacobson, "The maximum variance of restricted unimodal distributions," *The Annals of Mathematical Statistics*, vol. 40, no. 5, pp. 1746-1752, 1969

Generality of the Generalisation Framework

The framework is extremely general.

1. The framework is *independent* of the complexity of the game, since it is independent of the size of the strategy space, and independent of the strategy distribution in the strategy space.
2. Both G_{MAX} and G_{MIN} are often known *a priori* as they are defined by the game, which means that we can always obtain an upper bound using (5).
3. The framework is *independent* of learning algorithms since the bound holds for any strategy in the strategy space.

Estimation Accuracy: An Illustration

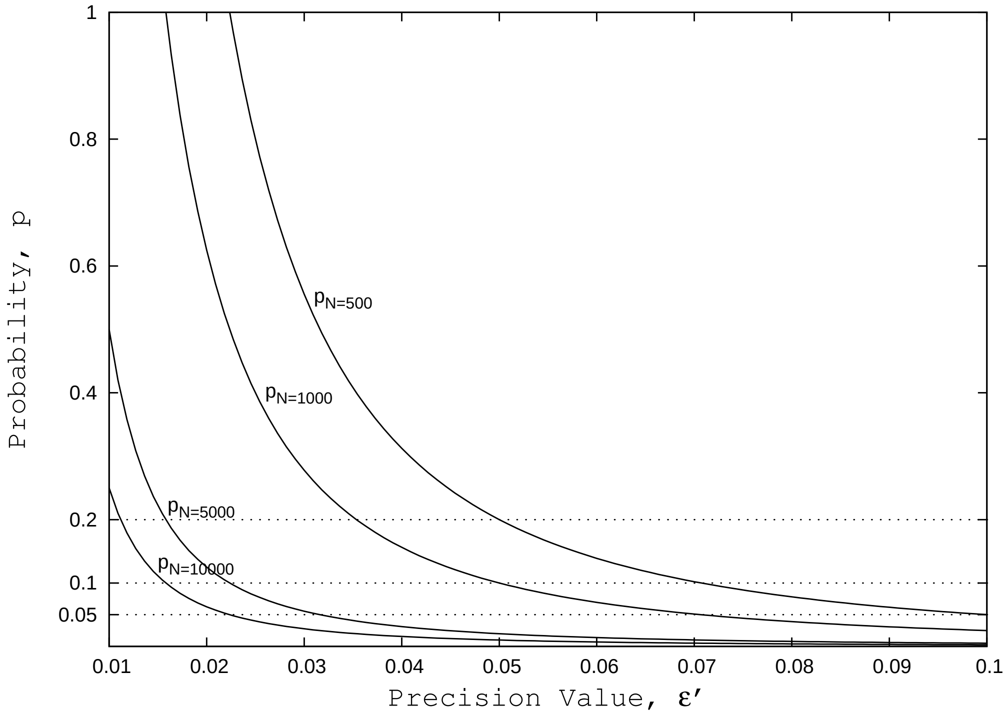

Figure 2: Equation 5 can be simplified to $P(|D_N|' \geq \epsilon') \leq \frac{1}{4N \cdot \epsilon'^2}$, where $\epsilon' = \epsilon/R$ and $|D_N|' = |\hat{G}_i - G_i|/R$. This figure shows the relationship between P_N for different N and precision ϵ' in $[0.01, 0.1]$.

Can We Obtain Tighter Bounds?

1. Chebyshev's bound is not very tight.
2. Furthermore, in Equation 4, we bound σ_i^2 using R^2 to obtain Equation 5 (assuming the worst-case).
3. It is possible to find a tighter upper bound for σ_i^2.

Recall What We Did

Consider a random variable X with the underlying distribution P_X over a real interval $[a, b]$, i.e., $X \in [a, b]$. For N realizations $x_1, x_2, ..., x_N$, the empirical mean is $\hat{E}_{P_X}[X] = \hat{\mu}_N = \frac{1}{N}\sum_{j=1}^{N} x_j$ while the true mean is $E_{P_X}[X]$. The true variance is $\sigma^2 = E_{P_X}\left[\left(X - E_{P_X}[X]\right)^2\right]$. Applying Chebyshev's Theorem gives us:

$$P(|\hat{\mu}_N - \mu| \geq \epsilon) \leq \frac{\sigma^2}{N \cdot \epsilon^2} \leq \frac{R_X^2}{4N \cdot \epsilon^2},$$

where the maximum variance for a random variable over $[a, b]$ is $\sigma^2_{\mathrm{MAX}} = R_X^2/4$ with $R_X = b - a$.

If we knew $E_{P_X}[X]$

we could obtain an unbiased estimate of σ^2:

$$\hat{\sigma}_N^2 = \frac{1}{N}\sum_{j=1}^{N}(x_j - \mu)^2 = \frac{1}{N}\sum_{j=1}^{N} y_j,$$

where y_j, $j = 1, ..., N$, are realizations of a new random variable $Y = (X - E_{P_X}[X])^2$. $\hat{\sigma}_N^2$ is the empirical mean of Y based on a sample $\{y_j\}_{j=1}^N$ and $E_{P_Y}[Y] = E_{P_X}\left[(X - E_{P_X}[X])^2\right] = \sigma^2$ is the true mean of Y. We can apply Chebyshev's Theorem for Y to obtain:

$$P(|\hat{\sigma}_N^2 - \sigma^2| \geq \delta) \leq \frac{\mathrm{Var}_{\mathrm{P_Y}}[\mathrm{Y}]}{N \cdot \delta^2} \leq \frac{R_X^4}{4N \cdot \delta^2},$$

where $\mathrm{Var_{MAX}[Y]} \leq \frac{\mathrm{R_Y^2}}{4} = \frac{\mathrm{(b-a)^4}}{4} = \frac{\mathrm{R_X^4}}{4}$ given that the range for Y is $R_Y = y_{\max} - y_{\min} = (b-a)^2 - 0 = (b-a)^2 = R_X^2$.

But We Don't Know $E_{P_X}[X]$

1. we need to examine the correction required for the fact that $\hat{\mu}$ rather than $\mu = E_{P_X}[X]$ is used to calculate the variance $\hat{\sigma}^2_N$. (Note that the worst-case is when $\hat{\sigma}^2_N$ is underestimated.)
2. We know that with probability at least $c_1 = 1 - R^2_X/(4N\epsilon^2)$, $|\hat{\mu} - \mu| \leq \epsilon$. With probability c_1, we have $(x_j - \hat{\mu}_N - \epsilon)^2 \geq (x_j - \mu)^2$. So, with probability c_1, we can be sure that:

$$\hat{\sigma}^2_{N,U} = \frac{1}{N}\sum_{j=1}^{N}(x_j - \hat{\mu}_N - \epsilon)^2 \geq \hat{\sigma}^2_N.$$

3. With probability $c_2 = 1 - R^4_X/(4N\delta^2)$, we have the upper bound $\sigma^2 \leq \hat{\sigma}^2_N + \delta$.

Obtaining a Tighter Bound

1. In order to combine the probabilities in a simple way, we must make the events independent, i.e., we need two sets of N-tuples of observations. The first set is used to estimate $\hat{\mu}_{1,N} = \frac{1}{N}\sum_{j=1}^{N} x_{1,j}$. The second set is used to estimate $\hat{\mu}_{2,N} = \frac{1}{N}\sum_{j=1}^{N} x_{2,j}$ and $\hat{\sigma}^2_{N,U} = \frac{1}{N}\sum_{j=1}^{N}(x_{2,j} - \hat{\mu}_{2,N} - \epsilon)^2$.
   Then, with probability:
   $$c_1 \cdot c_2 = \left(1 - \frac{R_X^2}{4N \cdot \epsilon^2}\right) \cdot \left(1 - \frac{R_X^4}{4N \cdot \delta^2}\right),$$
   we have $\sigma^2 \leq \hat{\sigma}^2_{N,U} + \delta$ since we know that $\sigma^2 \leq \hat{\sigma}^2_N + \delta$ and $\hat{\sigma}^2_N \leq \hat{\sigma}^2_{N,U}$.
2. In other words, with probability at least $c_1 c_2$ the following inequality holds:
   $$P(|\hat{\mu}_{1,N} - \mu| \geq \epsilon) \leq \frac{\hat{\sigma}^2_{N,U} + \delta}{N \cdot \epsilon^2}.$$
   However, for this inequality to be true, we require that $\sigma^2 \leq \hat{\sigma}^2_{N,U} + \delta < \sigma^2_{\mathrm{MAX}} = R_X^2/4$.

A Tighter Bound on Estimation Accuracy

Lemma 0.2 *For a strategy i, consider two independent non-overlapping sets of N test strategies: T_1 and T_2, where $T_1 \cap T_2 = \emptyset$ and $|T_1| = |T_2| = N$. The first set is used to estimate the generalization performance $\hat{G}_i(T_1) = \frac{1}{N}\sum_{j \in T_1} G_i(j)$. The second set is used to estimate the variance $\hat{\sigma}^2_{N,U} = \frac{1}{N}\sum_{j \in T_2}\left(G_i(j) - \hat{G}_i(T_2) - \epsilon\right)^2$, for some positive number $\epsilon > 0$, where $\hat{G}_i(T_2) = \frac{1}{N}\sum_{j \in T_2} G_i(j)$. Then, for $\delta > 0$ with probability at least $c_1 c_2 = (1 - R^2/(4N\epsilon^2))(1 - R^4/(4N\delta^2))$, the following inequality holds:*

$$P_N(|\hat{G}_i(T_1) - G_i| \geq \epsilon) \leq \frac{\hat{\sigma}^2_{N,U} + \delta}{N \cdot \epsilon^2}. \quad (6)$$

Some Examples — Iterated Prisoner's Dilemma

	Cooperate	Defect
Cooperate	R, R	S, T
Defect	T, S	P, P

Figure 3: The payoff matrix for the two-player IPD with two choices. The values S, P, R, T must satisfy $T > R > P > S$ and $R > (S + T)/2$. Each of the two players gets to choose to cooperate or defect in a single round. The payoff for a player depends on the choice made by the player and the opponent's.

More Complex Games — Multiple Levels of Cooperation

	-1	$-\frac{1}{3}$	$+\frac{1}{3}$	$+1$
-1	1	$2\frac{1}{3}$	$3\frac{2}{3}$	5
$-\frac{1}{3}$	$\frac{2}{3}$	2	$3\frac{1}{3}$	$4\frac{2}{3}$
$+\frac{1}{3}$	$\frac{1}{3}$	$1\frac{2}{3}$	3	$4\frac{1}{3}$
$+1$	0	$1\frac{1}{3}$	$2\frac{2}{3}$	4

Figure 4: The payoff matrix for the two-player IPD with four choices.

Different Definitions of Generalization Performance for IPD

1. Let $g(i, j)$ be the average payoff per move to strategy i when it plays an IPD game with strategy j.

2. Generalization performance in terms of the number of wins based on individual game outcomes:

$$G_{\mathrm{W}}(i,j) = \begin{cases} C_{\mathrm{WIN}} & for \qquad g(i,j) > g(j,i), \\ C_{\mathrm{LOSE}} & otherwise, \end{cases}$$

   where $C_{\mathrm{WIN}} > C_{\mathrm{LOSE}}$.

3. Generalisation performance in terms of average payoff:

$$G_{\mathrm{A}}(i,j) = g(i,j)$$

Computational Studies: Experimental Setup

1. Consider *three*-choice IPD since we can compute the true generalization performance, G_i (assuming uniform distribution of test strategies in the space for simplicity).
2. For all our experiments, we obtained $\hat{G}_i$ and G_i for 4000 strategies (which were randomly sampled).
3. The proportion of the 4000 strategies where $|D_N|' > \epsilon'$ was determined. This empirical value is then compared with Chebyshev's bound for $P(|D_N|' \geq \epsilon')$.
4. Experiments were repeated using 50 different samples of random test strategies S_N (Note: no strategy is obtained more than once).

Computational Studies: Observations

1. The empirical value for $P(|D_N|' > \epsilon')$ is less than the theoretical value given by Chebyshev's bound.
2. The empirical distribution of $G(i, j) - G(i)$ for the two definitions of generalization performance is quite similar for different sizes of S_N.
   (a) The estimation from using a smaller S_N is more similar compared to the estimation from using larger S_Ns.
   (b) The estimations are stable in terms of varying sample sizes starting from a small S_N.

Computational Studies: Results I

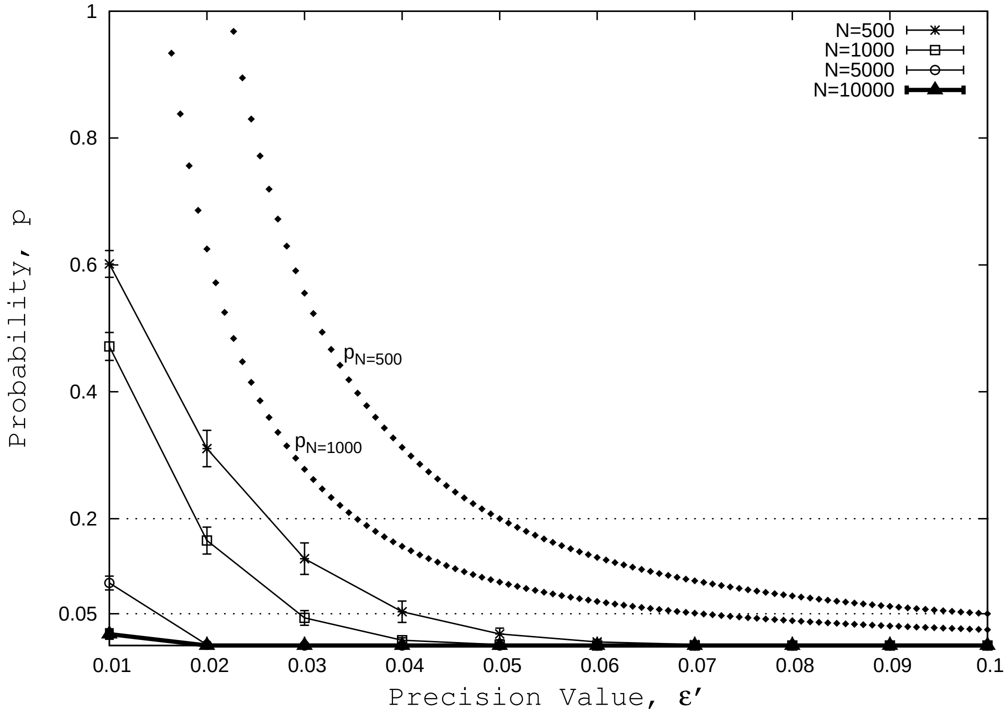

Figure 5: Values of $P(|D_N|' > \epsilon')$ for ϵ' in $[0.01, 0.1]$ and different sizes of S_N used to compute $\hat{G}_{\mathrm{W}}(i)$. Each curve is obtained by averaging over results from 50 independent samples S_N (error bars representing 95% confidence interval). P_N gives the curves for the Chebyshev's bounds.

Computational Studies: Results II

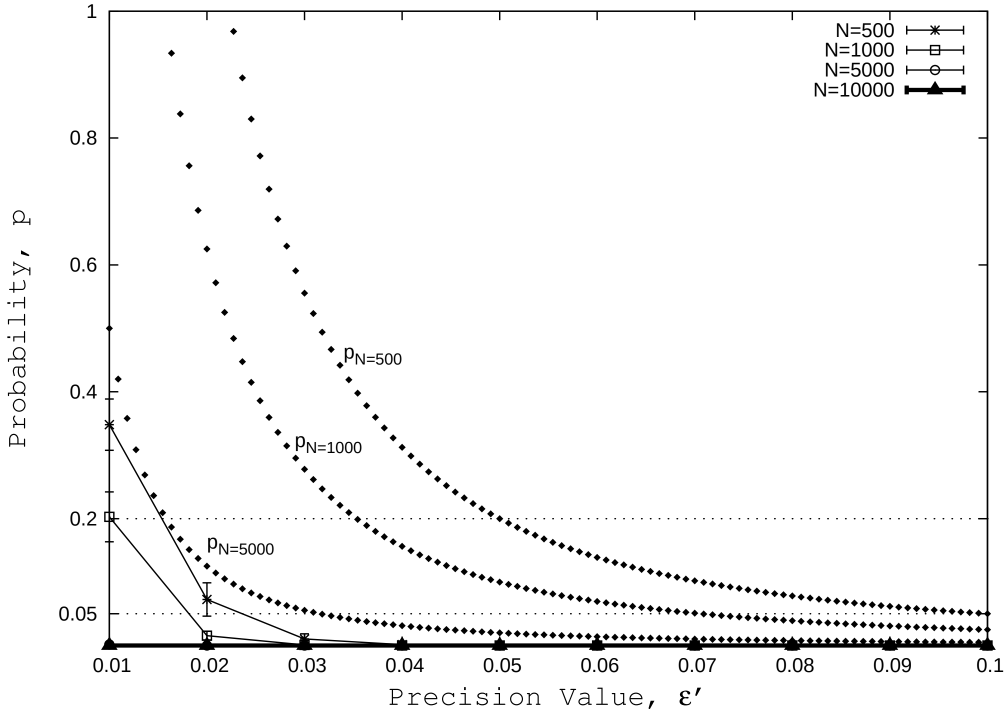

Figure 6: Values of $P(|D_N|' > \epsilon')$ for ϵ' in $[0.01, 0.1]$ and different sizes of S_N used to compute $\hat{G}_{\mathrm{A}}(i)$. Each curve is obtained by averaging over results from 50 independent samples S_N (error bars representing 95% confidence interval). P_N gives the curves for the Chebyshev's bounds.

Computational Studies: Results III

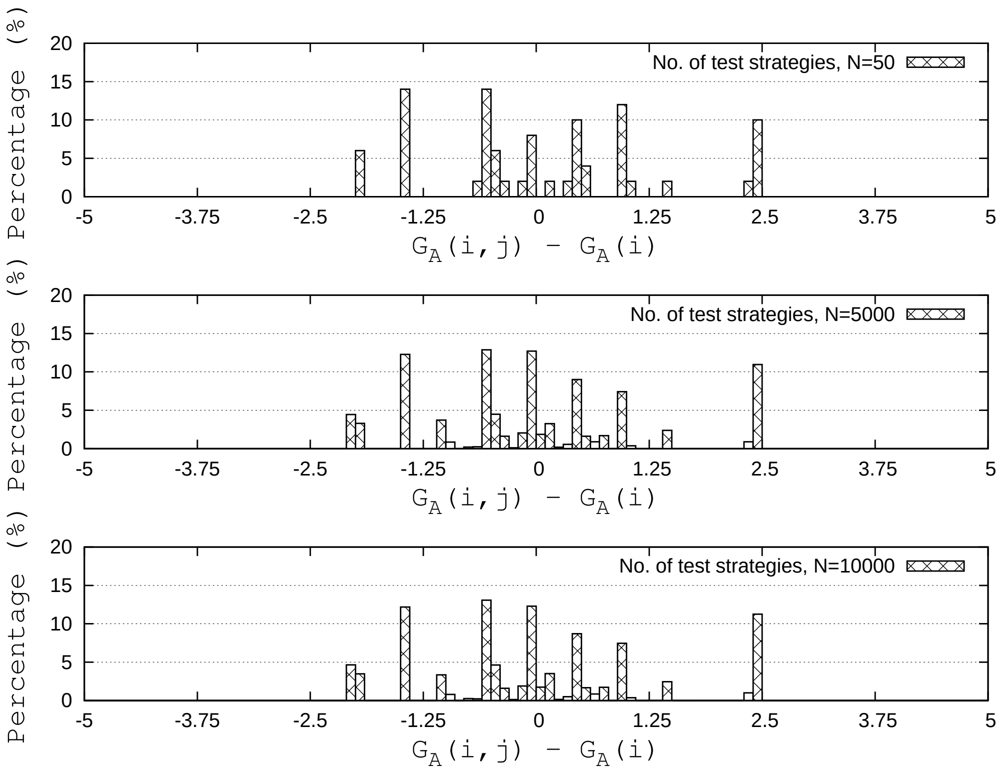

Figure 7: Empirical distributions of $G_{\mathrm{A}}(i,j) - G_{\mathrm{A}}(i)$ for $N = 50, 5000, 10000$.

Computational Studies on Tighter Bounds

1. Use similar experimental setup as before.
2. Still use *three*-choice IPD since we can compute the true variance of $G_i(j)$, σ_i^2.

Computational Studies on Tighter Bounds: Results I

Table 1: Comparison of $\hat{\sigma}_N^2$ for various N with the true value σ^2 for 8 strategies for $G_{\mathrm{W}}(i)$ where $\sigma^2 \in [0, 2500]$.

N	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
10	2500	2400	2500	2400	2400	2100	0	2400
50	2464	2496	2244	1476	2496	1924	1204	2356
100	2304	2484	2059	1344	2464	1924	1275	2436
500	2436	2500	2224	1539	2491	1895	1317	2477
1000	2419	2498	2131	1470	2484	1875	1218	2482
5000	2432	2492	2108	1481	2490	1849	1341	2476
10000	2434	2497	2118	1515	2493	1816	1321	2473
TRUE	**2436**	**2496**	**2126**	**1475**	**2491**	**1816**	**1293**	**2477**

Computational Studies on Tighter Bounds: Results II

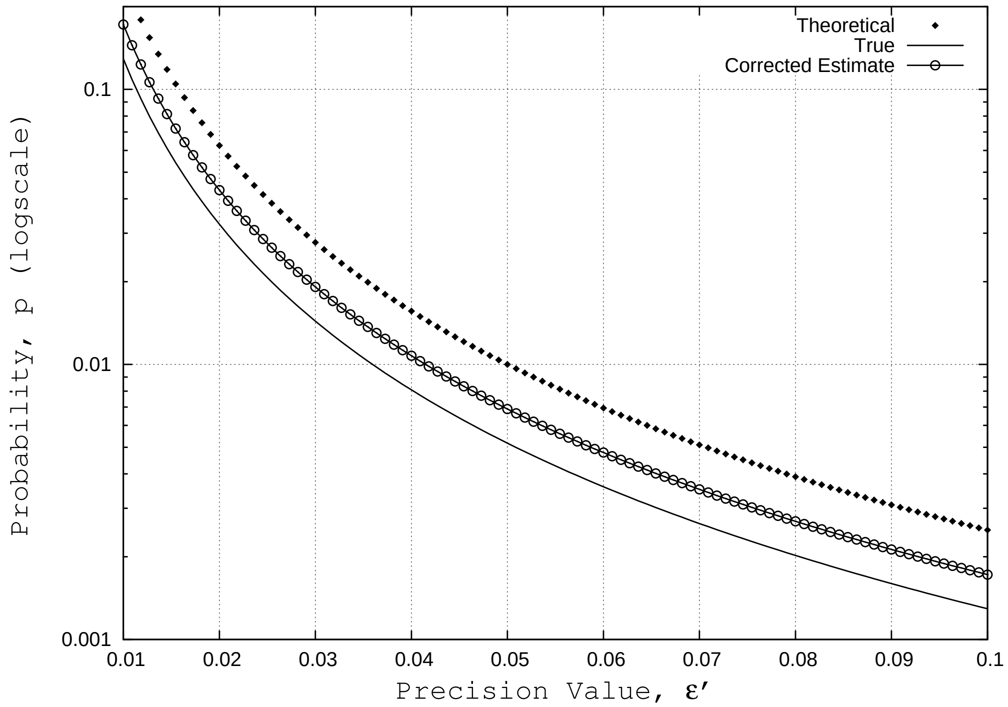

Figure 8: The curve "Theoretical" gives the original Chebyshev's bound when σ^2_{MAX} is used. The curve "True" gives the actual bound when σ^2 is used. The curve "Corrected Estimate" gives the tighter Chebyshev's bound when $\hat{\sigma}^2_{N,U} + \delta$ is used. All curves were obtained for $N = 10000$.

Generalization Performance of Co-evolutionary Learning

Classical co-evolutionary learning (CCL):

1. $t = 1$: Initialize POPSIZE/2 parent strategies, $P_i, i = 1, 2, ...,$ POPSIZE/2, randomly.
2. Generate POPSIZE/2 offspring, $O_i, i = 1, 2, ...,$ POPSIZE/2, from POPSIZE/2 parents using variation.
3. All pairs of strategies compete, including the pair where a strategy plays itself (i.e., round-robin tournament). For POPSIZE strategies, every strategy competes a total of POPSIZE games.
4. Select the best POPSIZE/2 strategies based on total payoffs of all games played. Increment generation step, $t = t + 1$.
5. Step 2 to 4 are repeated until a termination criterion (i.e., a fixed number of generations) is met.

Strategy Representation

	$+1$	-1
$+1$	m_{11}	m_{12}
-1	m_{21}	m_{22}

Figure 9: The direct look-up table representation for memory-one classical IPD strategies. m_{fm} used to represent the first move directly when there is no prior history played yet.

Performance Measures

1. Best,

$$\mathrm{Best}(\mathrm{G}_{\mathrm{SPOP_U}}) = \hat{\mathrm{G}}_{\mathrm{spop_1}}. \tag{7}$$

2. Average,

$$\mathrm{Avg}(\mathrm{G}_{\mathrm{SPOP_U}}) = \frac{1}{\mathrm{U}} \left(\sum_{\mathrm{l}}^{\mathrm{U}} \hat{\mathrm{G}}_{\mathrm{spop_l}} \right). \tag{8}$$

3. Ensemble,

$$\mathrm{Ens}(\mathrm{G}_{\mathrm{SPOP_U}}) = \frac{1}{\mathrm{N}} \left(\sum_{\mathrm{j} \in \mathrm{S_N}} \min\Big(\sum_{\mathrm{l}}^{\mathrm{U}} \mathrm{G}_{\mathrm{spop_l}}(\mathrm{j}), \mathrm{G}_{\mathrm{MAX}} \Big) \right). \tag{9}$$

Two-Choice IPD

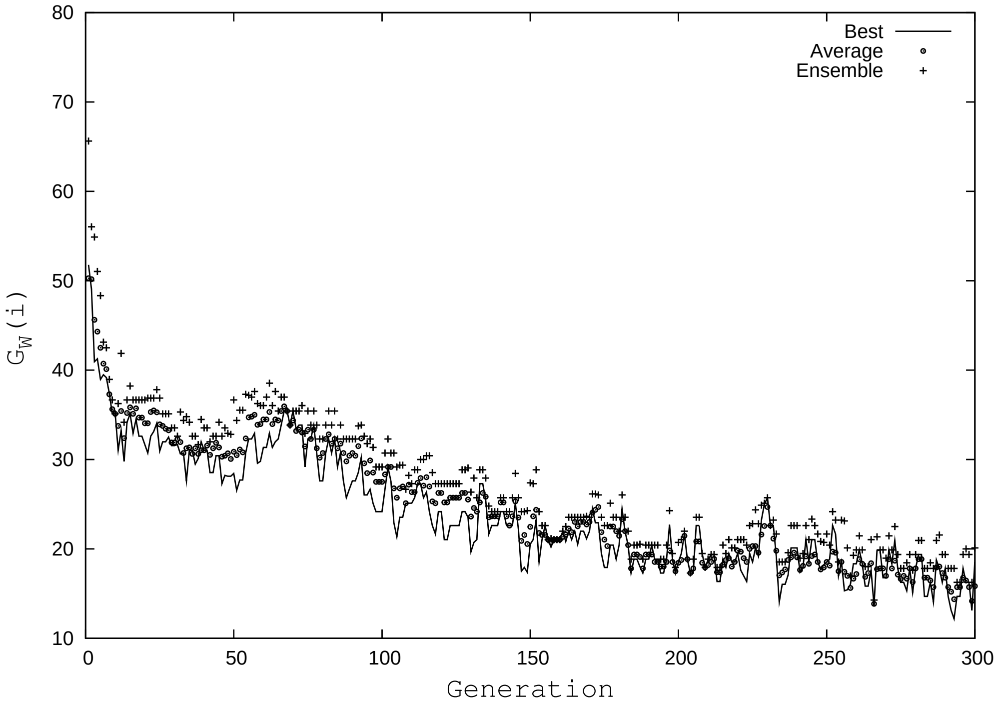

Figure 10: Generalization performance of CCL defined by $G_{\mathrm{W}}(i)$. All graphs are averaged over measurements from 30 independent runs.

Three-Choice IPD

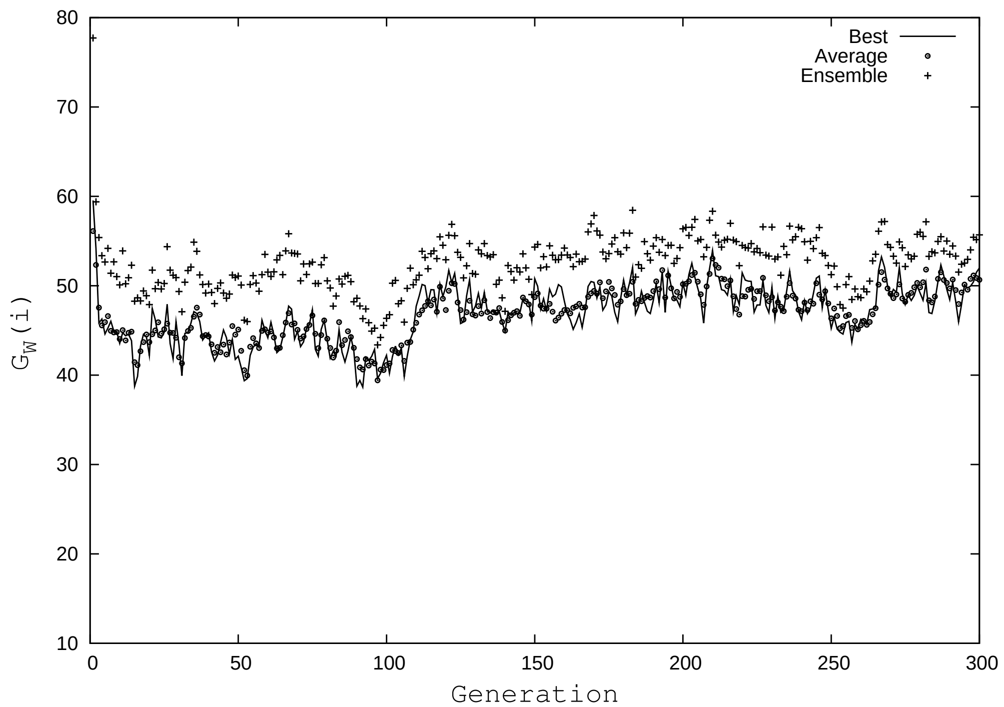

Figure 11: Generalization performance of CCL defined by $G_{\mathrm{W}}(i)$. All graphs are averaged over measurements from 30 independent runs.

Some Observations

1. The poor generalization performance for the simpler *two*-choice IPD is due to the population overspecializing to naive cooperators.
2. Proportion of AllC for the n-choice IPD is given by $n^{(n^2-n)}/n^{(n^2+1)} = 1/n^{(n+1)}$.
3. The empirical value is similar to the proportion. For the *two*-choice IPD, around 12% (true value: 12.5%). For the *three*-choice IPD, around 0.2% (true value: around 1%).
4. Simple co-evolutionary learning does not guarantee a solution that generalizes well even for simple problems, e.g., for *two*-choice IPD, co-evolution having searched more strategies than the total number of unique strategies in the strategy space (i.e., 32) .
5. Selection does not necessarily bias towards increasing generalization performance.

Counteracting Overspecialization by Encouraging Diversity

[a]

1. In a speciation approach through implicit fitness sharing, the shared fitness for each strategy i in the population is obtained by repeating the following procedure C times:
   (a) From the population, select a sample of σ opponent strategies.
   (b) Find the opponent strategy in sample σ that achieves the highest score (or the largest winning margin) against i.
   (c) The best opponent strategy in σ receives a score. In the case of a tie, the score is shared equally.
2. Co-evolutionary learning of *four*-choice IPD game was studied.

[a] P. Darwen and X. Yao, "Speciation as automatic categorical modularization," IEEE Transactions on Evolutionary Computation, Vol. 1, No. 2, pp.101-108, 1997.

Generalisation Performance for the *Four*-choice IPD

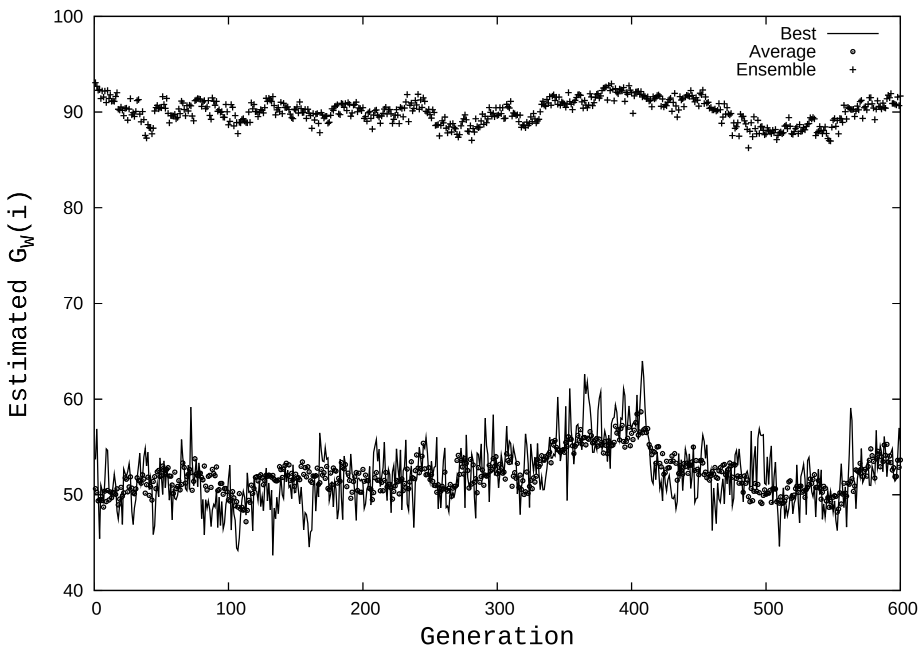

Figure 12: $\hat{G}_{\mathrm{W}}(i)$ of CCL with speciation across the evolutionary process with "Best", "Average", and "Ensemble" measurements, for the *four*-choice IPD. All graphs are averaged over 30 independent runs.

Biased vs. Unbiased Samples

1. So far, measurements are made with respect to an unbiased sample, e.g., consider uniform distribution of test strategies in the space.
2. However, for some problems, one may be more interested with a biased sample.
3. For actual games, the majority of strategies in the space may be poor or mediocre. Rather than consider each strategy having equal chances of being met in some tournament, we may be more interested in a small number of strong-performing strategies.
4. The question now is how we can sample these strategies. We can't sample strong-performing strategies with higher probabilities because we don't usually have a metric strategy space.

An Approximate Solution: Partial Enumerative Search

1. Randomly sample a large number of test strategies and choose the best performing strategies as our biased test sample.
2. Since the sample is much larger than what co-evolution can possibly search, we hope that some of these test strategies are good (beats a lot of other strategies) and unseen (co-evolved strategies never met them).

However,

1. A sample of obtained from a single partial enumerative search may not be sufficiently diversed as these strategies might be behaviorally similar (i.e., having the same weakness that can be exploited).
2. We address this problem by repeating the procedure several times, i.e., multiple partial enumerative search.

Multiple Partial Enumerative Search

1. $r = 1$: Sample PS strategies, $Q_i, i = 1, 2, ..., \text{PS}$, randomly.
2. Every strategy competes with all other strategies in the sample, including itself (i.e., each strategy competes a total of PS games).
3. Detect the strategy index $s \in \{1, 2, ..., \text{PS}\}$ so that Q_s is yielding the highest total payoff. Let $Q^{(r)} := Q_s$. $r = r + 1$.
4. Repeat steps one to three PE times to obtain PE-sized biased sample of test strategies, $Q^{(r)}, r = 1, 2, ..., \text{PE}$.

Four-choice IPD Results

1. We obtain 20 test strategies for the biased test sample from 20 partial enumerative search of 10^6 random test strategies.
2. We use 10^6 test strategies for the unbiased test sample.
3. In general, for the IPD game, generalization performance w.r.t. an unbiased test sample is higher in comparison to that w.r.t. a biased test sample.

With Unbiased Test Sample

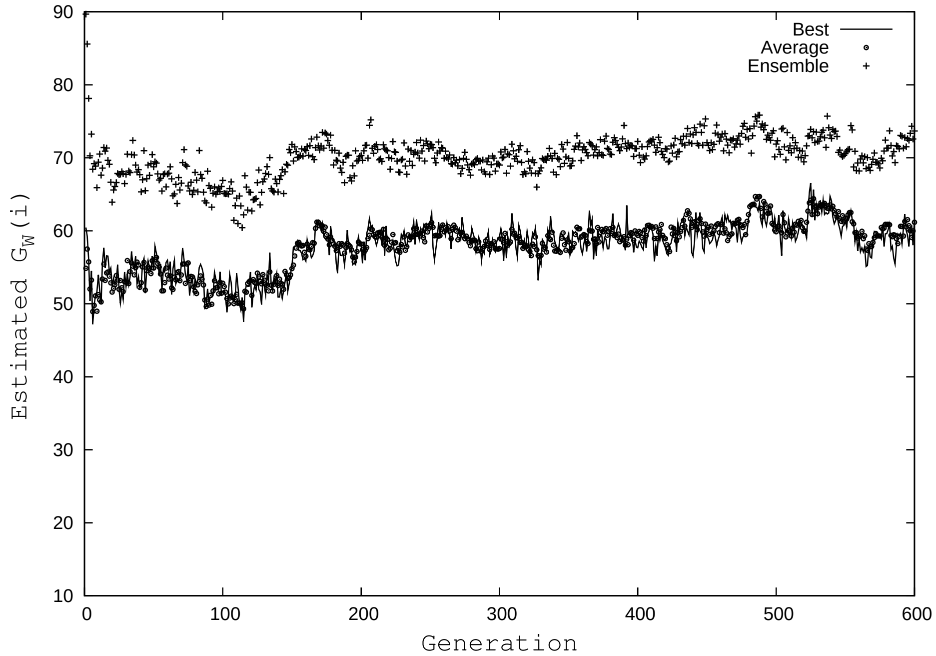

Figure 13: $\hat{G}_{\mathrm{W}}(i)$ of CCL across the evolutionary process with "Best", "Average", and "Ensemble" measurements, for the *four*-choice IPD. All graphs are averaged over 30 independent runs.

With Biased Test Sample

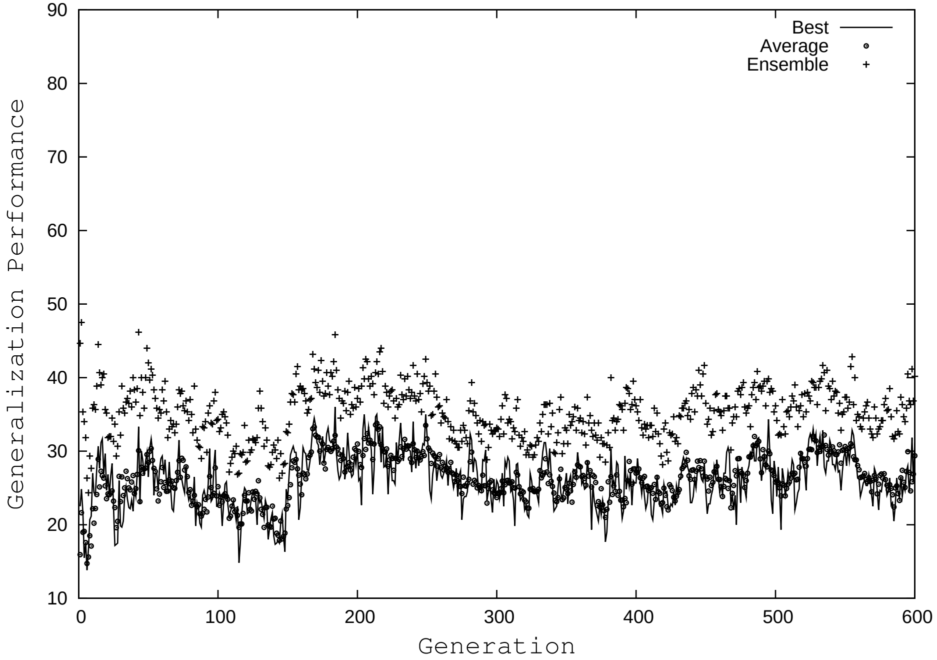

Figure 14: $\hat{G}^{\mathrm{B}}_{\mathrm{W}}(i)$ of CCL across the evolutionary process with "Best", "Average", and "Ensemble" measurements, for the *four*-choice IPD. All graphs are averaged over 30 independent runs.

Conclusions

1. We have presented a theoretical framework for measuring generalisation performance rigorously in co-evolutionary learning. For the first time, quantitative analysis of generalisation performance of any co-evolutionary learning system can be performed.
2. The framework is extremely general and independent of any games, distributions and algorithms.
3. The theoretical framework can be applied to concrete games and algorithms, and estimate the generalisation performances.
4. Empirical results show that a small sample is usually good enough in estimating the generalisation performance.
5. More details:

   S. Y. Chong, P. Tino and X. Yao, "Measuring Generalization Performance in Co-evolutionary Learning," *IEEE Transactions on Evolutionary Computation*, **accepted in August 2007.**