Game Playing and Coevolution

Siang Yew Chong

School of Computer Science
University of Nottingham Malaysia

Marie Skłodowska-Curie Fellow
School of Computer Science
University of Birmingham, UK

March 2019

[1] Joint Work with Peter Tino and Xin Yao

Preliminaries

1. The study of **games**, like Artificial Intelligence, is wide.
2. They range from **games of entertainment** to more serious ones such as those used to **model systems** involving situations of **strategic decision making** in various settings (biology, politics, economics, etc.).

Preliminaries

1. Some **categorization** of games through actions (moves).
   - 1.1 **Two players** (Chess) vs. **Multi-players** (Poker).
   - 1.2 **Deterministic** (Chess) vs. **Non-deterministic** (Poker), i.e., element of chance affecting actions.
   - 1.3 **Perfect Information** (Chess) vs. **Imperfect Information** (Poker), i.e., whether there is knowledge of actions by opponent.
   - 1.4 **Zero-sum** (Chess) vs. **Nonzero-sum** (Prisoner's Dilemma), i.e., on the game-play outcome.

Motivation of this Lecture

1. Broadly, **coevolutionary systems** – *populations of agents whose behavior changes in response to their interaction outcomes* – used to model real-world strategic decision-making systems.
2. Main aims:
   - 2.1 understanding mechanisms of coevolutionary processes, and
   - 2.2 developing robust, efficient, and flexible algorithms based on these mechanisms to solve difficult problems (e.g., not possible with classical methods).

Real-world Problem Solving With EC - A Simple Example[1]

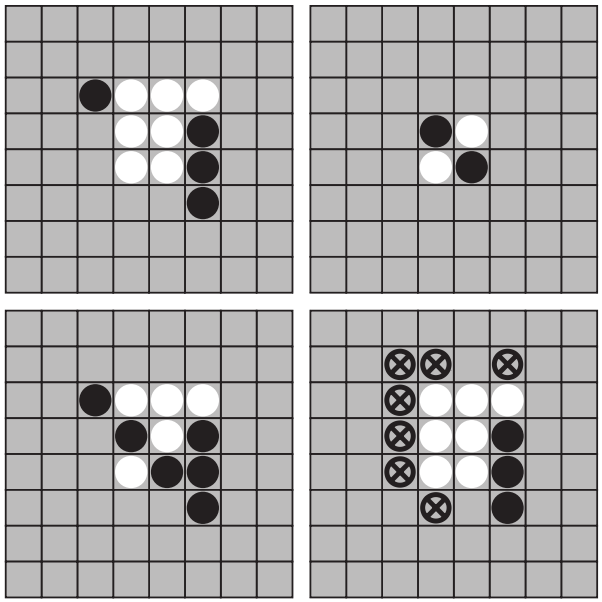


Figure: The 2-player board game Othello.

[1] SY Chong et al, *CEC2003*.

Intelligent Strategic Decision-Making[2]

1. **Problem solving** tasks may involve some forms of **decision-making** processes, e.g., given some inputs (Othello board states), provide the required outputs (legal moves).
2. One is interested with an **intelligent** solution for strategic decision-making (game strategy) that can:
   - 2.1 **predict** future outcomes (moves the opponent is likely to make),
   - 2.2 take appropriate **decisions** (legal moves) given certain goals (win the Othello game), and
   - 2.3 **adapt** its behavior to meet those goals over a range of environment (opponents).

[2] K. Chellapilla and D. B. Fogel, "Evolution, Neural Networks, Games, and Intelligence," *Proceedings of the IEEE*, Vol. 87, No. 9, pp. 1471-1496, Sep. 1999.

Game Solutions I

1. A standard **minimax** approach – a recursive search algorithm on the game-tree of board states from alternating moves between the player and opponent – choose the best move for each turn.
2. Board state **evaluation functions** are (typically) hand-crafted with a deep-ply search for mid-games + opening and end game databases.
3. Most board games are considered solved – Othello, chess and Go (almost).

Game Solutions II

1. An alternative **evolutionary** approach – more complex/involved evaluation functions + much shallower game-tree search (if any).
2. Board state **evaluation functions** (game strategies) have to be discovered (learned).

Complex Representation for Strategic Decision-Making

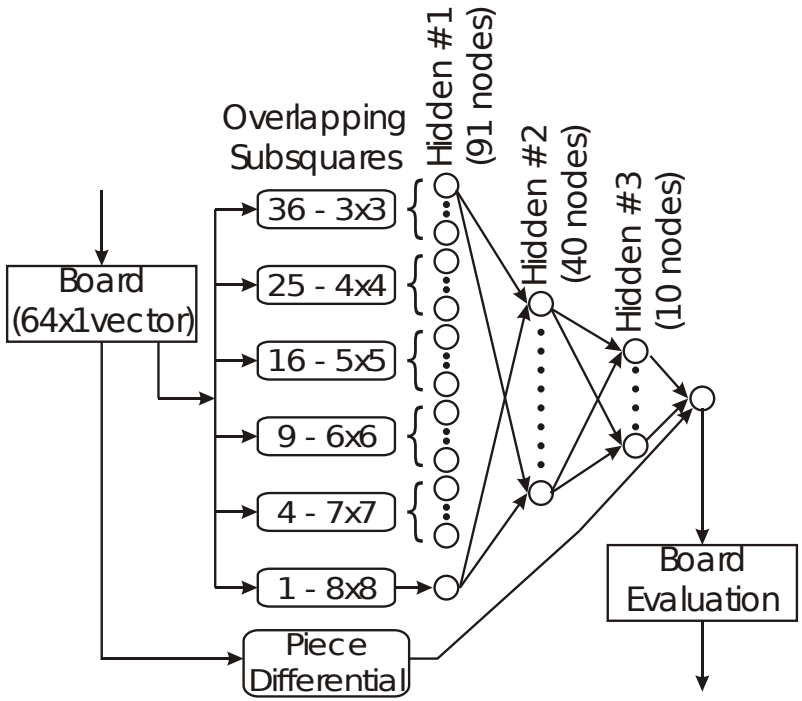


Figure: Neural-network-based **parameterized representation** for board games with **nonlinear** spatial features of the board (SY Chong et al, *IEEE TEVC*, 2005).

Coevolutionary Learning

1. Problem solving becomes a search for the best strategy – **optimizing** representation parameters (e.g., neural connections).
2. This optimization process reformulated as a **learning** task. Coevolutionary learning provides an attractive framework based on natural evolutionary processes of:
   - 2.1 **variation** to generate new solutions, and
   - 2.2 **selection** to test solutions.

Fitness Evaluations

1. Fitness evaluations based on training error estimated from a teacher-target sample (game strategies) that is evolving as well – **coevolutionary systems**.
2. Coevolutionary approach – little (if any) preprogrammed knowledge:
   - 2.1 fitness evaluations based on **interactions between competing solutions** are modeled as **game-play**,
   - 2.2 **arms-race** dynamics – survival of the fittest – process of **adaptation**.

Coevolutionary Learning Framework

A typical coevolutionary system involves a **generate-and-test procedure**

1. Initialize population, $X(t = 1)$.
2. Evaluate fitness through interactions (game-plays) between population members (agents).
3. Select parents from $X(t)$ based on fitness.
4. Generate offspring from parents to obtain $X(t + 1)$ (e.g. random variations on ANN weights).
5. Repeat steps (2-4) until some termination criteria are met.

Note: The iteration count t is termed **generation**.

Coevolutionary Algorithm

procedure $\mathrm{CEA}(X, \mu, \lambda, t_{\mathrm{stop}})$
  $t := 0$
  Initialize population in X
  $X = \{x\}_1^\mu \cup \{x\}_{\mu+1}^\lambda, x \in \mathbb{R}^d$
  while $t \leq t_{\mathrm{stop}}$ **do**
    Game plays for X
    Compute scores $\{s\}_1^\lambda$ for X
    Sort X according to decreasing score sequence $(s_\lambda, s_{\lambda-1}, \ldots)$
    Select next generation's parent population
    $\{\tilde{x}\}_1^\mu := s(X)$ ▷ selection operator s, e.g. $\{\tilde{x}\}_1^\mu = \{x\}_1^\mu$
    Generate next generation's offspring population
    $\tilde{x}_j = v(\tilde{x}_k), j = \mu + 1, \ldots, \lambda, \ k = 1, \ldots, \mu$ ▷ variation operator v
    $X := \tilde{X}$.
  end while
end procedure

ENN-based Othello players vs Positional Players

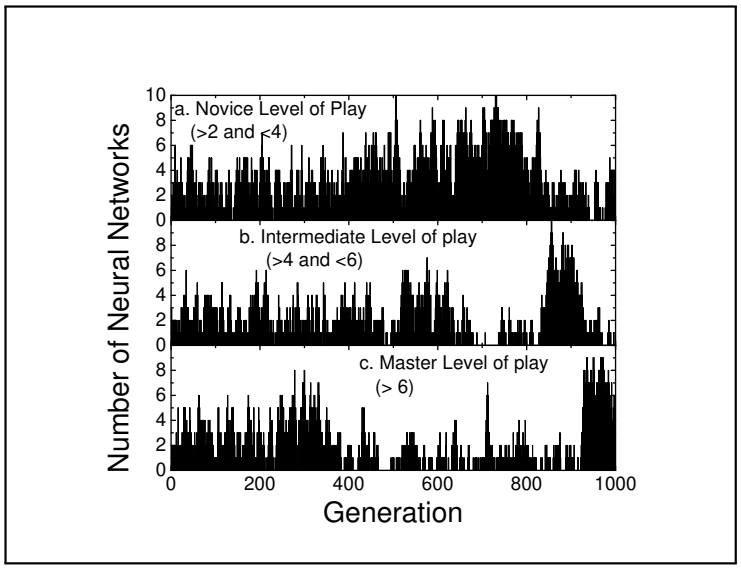


Figure: More evolved players defeating positional players with deeper minimax search.

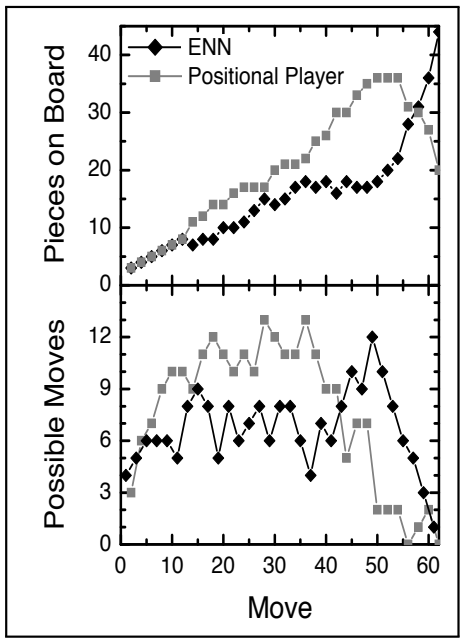


Figure: ENN-based Othello player exhibiting mobility strategies.

Coevolutionary Computation and Applications

1. Application as a general form of generate-and-test search methodology:
   - 1.1 population-based, stochastic search algorithms where adaptation is the central process of the search mechanism,
   - 1.2 search problems in the context of optimization, learning and others.
2. Another application is where coevolutionary computation is used for **simulation** and **modeling** purposes.

Real-World Problems: Constrains of Resources[3]

1. The Tragedy of the Commons:
   - 1.1 Consider and open pasture **free** for all farmers.
   - 1.2 A **rational** farmer would maximize his **personal** gain, e.g., add more cows for grazing.
   - 1.3 The **tragedy** is that if every farmer seeks to maximize his gain, they would **collectively** destroy the pasture due to overgrazing.

[3] G. Hardin, "The Tragedy of the Commons," *Science*, Vol. 162, pp. 1243-1248, Dec. 1968.

What are the Solutions?protect[4]

1. Axelrod and many others earlier studied through **simulation** and **modeling** of this problem–**Prisoner's Dilemma**–to understand the **conditions** that allow mutual cooperation to occur within a group of selfish individuals.

   ...an elegant embodiment of the tension between individual rationality (reflected in the incentive of both sides to be selfish) and group rationality (reflected in the higher payoff to both sides for mutual cooperation).

[4] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1984.

Prisoner's Dilemma

	Cooperate	*Defect*
Cooperate	R	S
Defect	T	P

Figure: The payoff matrix for the two-player IPD with two choices. The values S, P, R, T must satisfy **constraints** (i) $T > R > P > S$ and (ii) $R > (S + T)/2$. 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.

Why use coevolution to Simulate and Model?[5]

1. coevolutionary computation framework:
   1.1 Emphasizes the **learning** of behaviors through an **adaptation** process based on strategic interactions between competing solutions.
   1.2 Simulates and models the specific conditions that lead to the learning outcomes of certain behaviors.
2. As an example, the literature has shown that cooperative behaviors can be learned, i.e., mechanism of **direct reciprocity** modeled as repeated encounters encourages the learning of **cooperative** behaviors through **coevolution** of IPD strategies.

[5] S. Y. Chong, J. Humble, G. Kendall, J. Li, and X. Yao, "Learning IPD strategies through coevolution," in *The Iterated Prisoners' Dilemma: 20 Years On*. Singapore: World Scientific Press, 2007.

Understanding Complex Real-World Interactions

1. However, complex real-world interactions rarely involve just two simple choices.
2. How about having more choices with intermediate cooperation levels where strategies can subtly exploit opponents?
3. With coevolutionary learning, the model can be extended to simulate interactions involving more choices.

More Complex Games - More Choices

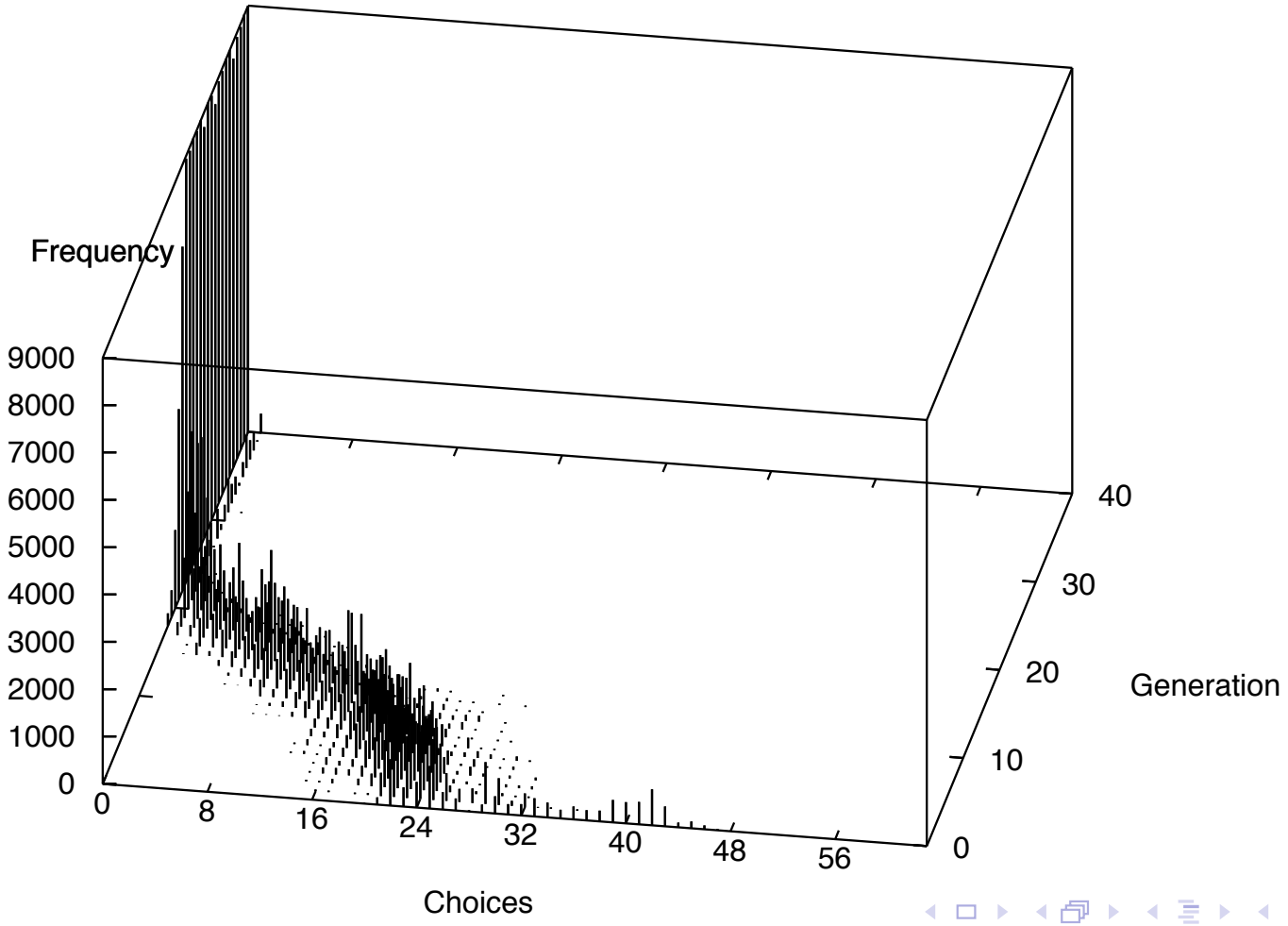
	-1	$-\frac{1}{3}$	$+\frac{1}{3}$	$+1$
-1	1	$2\frac{1}{3}$	$3\frac{2}{3}$	5
$-\frac{1}{3}$	$\frac{2}{3}$	2	$3\frac{1}{3}$	$4\frac{2}{3}$
$+\frac{1}{3}$	$\frac{1}{3}$	$1\frac{2}{3}$	3	$4\frac{1}{3}$
$+1$	0	$1\frac{1}{3}$	$2\frac{2}{3}$	4

Figure: The payoff matrix for the two-player IPD with four choices. The payoffs can be obtained through a linear interpolation, $p_A = 2.5 - 0.5c_A + 2c_B, \quad -1 \leq c_A, c_B \leq 1$, with $+1$ representing full cooperation, and -1 representing full defection.

Coevolutionary Learning of Complex IPD

1. Consider a large class of **deterministic** and **reactive** IPD strategies (for a wide range of complex behaviors at a level of abstraction we can analyze).
2. We use neural networks for strategy representation (for its scalability).
3. Coevolutionary learning model:
   - 3.1 Initialize $N/2$ strategies.
   - 3.2 Generate $N/2$ offspring from $N/2$ parents to obtain the population.
   - 3.3 Each strategy's fitness is assigned as the average payoffs of IPD games (e.g., full-mixing).
   - 3.4 Select $N/2$ parents based on fitness values for the next generation.
   - 3.5 Repeat steps (b-d) for T_g generations.

Strategies Only Learning Defection Play



More Choices Lead to Defection Outcomes

1. Literature has shown co-evolving cooperative strategies is difficult and requires certain conditions to be met:
   - 1.1 Harrald and Fogel (1996): minimum **complexity** in representation (e.g., neural networks).
   - 1.2 Darwen and Yao (2001): **behavioral diversity** is needed, not genetic diversity.
   - 1.3 Chong and Yao (2005): behavioral diversity is dependent on **representation**[6].

[6] S. Y. Chong and X. Yao, "Behavioral Diversity, Choices, and Noise in the Iterated Prisoner's Dilemma," *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 6, pp. 540-551, Dec. 2005.

Why More Choices Lead to Defection?[7]

1. With coevolutionary learning, strategies learn behaviors through an adaptation process, which depends on the average payoff.
2. With more choices, it becomes increasingly difficult to resolve the intention of an **intermediate play**, i.e., is it a **subtle exploitation** or a signal to engender **further cooperation**?
3. More incentives to adapt behaviors to exploit partners (play lower cooperation levels and obtain higher average payoffs) when strategies cannot to resolve the intention of opponents in the short-term (through the average payoff).

[7] S. Y. Chong and X. Yao, "Multiple Choices and Reputation in Multiagent Interactions," *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 6, pp. 689-711, Dec. 2007.

One Fundamental Research Question

1. Suppose I have a coevolutionary algorithm to learn game strategies, how do I know how well the co-evolved strategy performs against **new** and **unseen** opponents?
2. Suppose I have a coevolutionary model to simulate an interaction, what can I say about the outcome of strategic behaviors, i.e., are the behaviors **robust**?
3. In other words, how well co-evolved strategies **generalize**?
4. Answering this question is the key to perform rigorous analysis, e.g., whether solutions obtained through coevolution meet the required objectives.

A Need for a Theoretical Framework

1. For many difficult problems, we only have **test cases** as means to **measure performance** (opponents in games) of solutions.
2. Early attempts used empirical estimates with random samples of test cases. But what do they really tell us, e.g., how accurate are the estimates?
3. Strong motivation for a theoretical framework:
   - 3.1 to establish the notion of performance, and
   - 3.2 perform **rigorous quantitative analysis** in coevolutionary computation.

Generalization Performance

1. Consider a **game** and a set $\mathcal{S}$ of M **pure strategies** $\{1, 2, 3, \ldots, M\}$.
2. The **game outcome** of strategy i against j is given by $G_i(j)$.
3. Different definitions of $G_i(j)$ indicate different **quality measures**. One typical example is **win-lose**:

$$G_{\mathrm{W}}(i,j) = \begin{cases} G_{\mathrm{MAX}} & \textit{for} \quad g(i,j) > g(j,i), \\ G_{\mathrm{MIN}} & \textit{for} \quad \textit{otherwise}, \end{cases}$$

where $G_{\mathrm{MAX}} > G_{\mathrm{MIN}}$, and $g(i,j)$ is the payoff to i in a game against j.

Generalization Performance

1. Selection of individual test strategies represented by a random variable J taking on values $j \in \mathcal{S}$ with probability $P_{\mathcal{S}}(j)$.
2. **True generalization performance** of strategy i, G_i:

$$G_i = E_{P_1(j)}[G_i(j)] = \sum_{j=1}^{M} P_{\mathcal{S}}(j) G_i(j) \tag{1}$$

where G_i is the mean of the random variable $G_i(j)$.

Estimated Generalization Performance

1. In practice, we **estimate** generalization performance G_i through a random sample S_N of N test strategies drawn i.i.d. from $\mathcal{S}$ with probability $P_{\mathcal{S}}$

$$\hat{G}_i(S_N) = \frac{1}{N} \sum_{j \in S_N} G_i(j). \tag{2}$$

2. Note 1: we are unable or cannot calculate G_i.
3. Note 2: we also cannot calculate the error $\left|\hat{G}_i - G_i\right|$.

How Accurate is your Estimate?

1. So, we make a **statistical claim** as to the **confidence** (**probability**) with the accuracy (**precision**) of the estimate for a **sample size** N using **Chebyshev's bounds**.
2. From Chebyshev's Theorem, we obtain

$$P_N(|\hat{G}_i - G_i| \geq \epsilon) \leq \frac{\sigma_i^2}{N \cdot \epsilon^2} \leq \frac{R^2}{4N \cdot \epsilon^2}. \tag{3}$$

   for any positive $\epsilon > 0$.
3. Since random variable $G_i(J)$ varies within finite interval $[G_{\text{MIN}}, G_{\text{MAX}}]$ of size R, its variance is upper-bounded by $\sigma^2_{\text{MAX}} = R^2/4$.

Chebyshev's Bound

1. Chebyshev's bounds can be restated into another more useful form. Let $ID_N| = |\hat{G}_i - G_i|$, $ID_N|' = ID_N|/R$, $\epsilon' = \epsilon/R$. Then:

$$P(|D(N)|' \leq \epsilon') \geq 1 - \frac{1}{4N \cdot \epsilon'^2}. \quad (4)$$

2. Note: The analysis can be extended to games with probabilistic strategies as well since D_N is a random variable.

Relationship in the Chebyshev's Bound

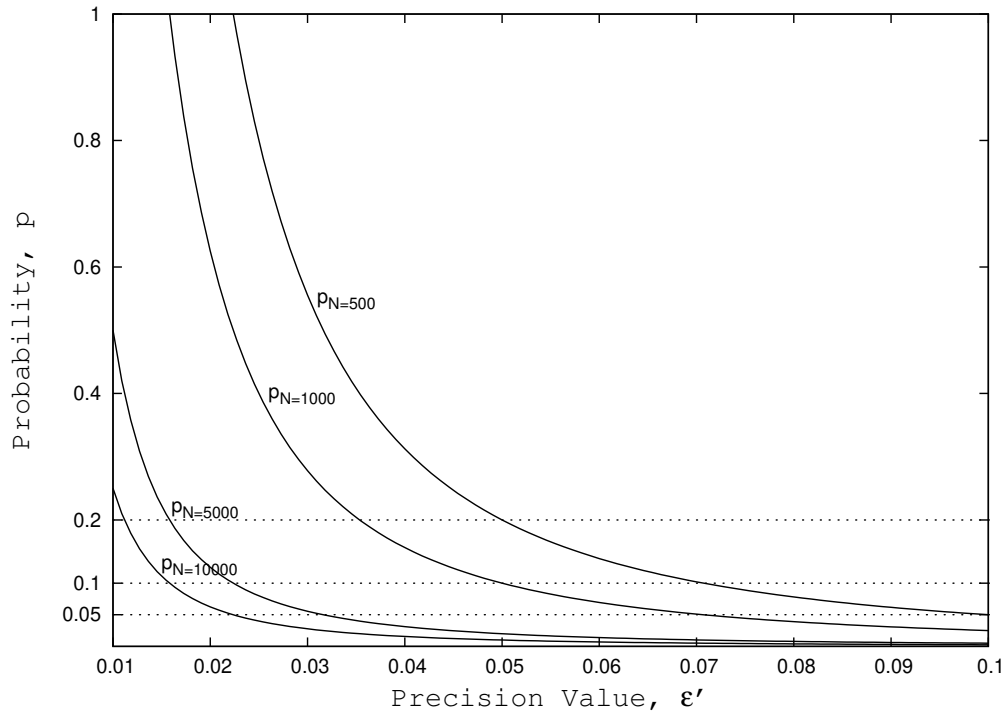


Figure: Example: given $N = 5000$, one can claim with confidence $100(1 - 0.2)\%$ that $|D(N)|' < 0.02$.

How Useful is the Framework?

1. **Chebyshev's bounds** are:
   - 1.1 **Independent** of the **complexity of the game** (e.g. only sample size N matters).
   - 1.2 **Independent** of the **learning algorithms** (**distribution-free**, e.g., can be used to estimate the performance for any strategy in any game).

Estimates Smaller than Theoretical Values

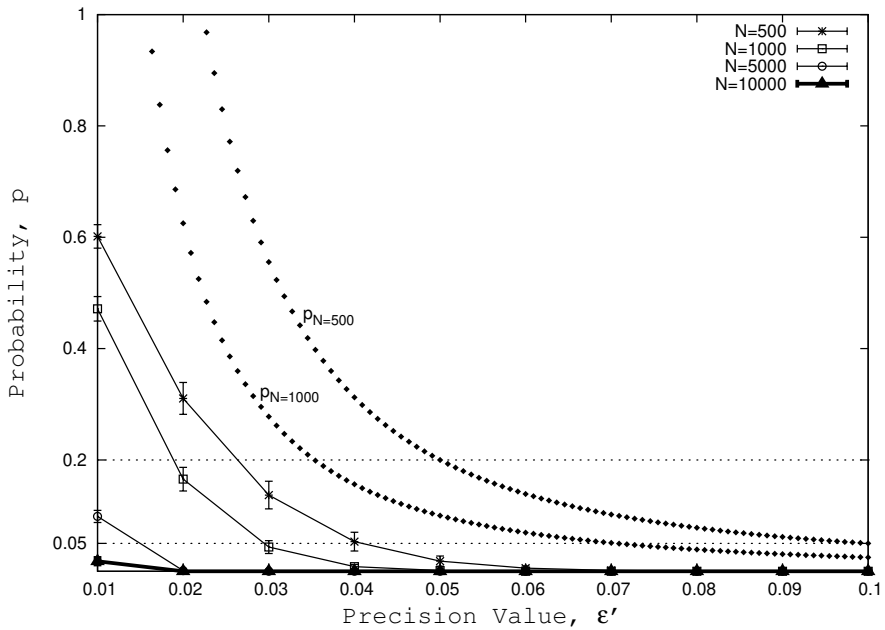


Figure: Chebyshev's bounds vs empirical estimates for $P(|D_N|' > \epsilon'), \epsilon' \in [0.01, 0.1]$ for different sample sizes S_N. Note that Chebyshev's is an **upper-bound**, i.e., $\sigma_i^2 \leq R^2/4$.

Estimations Stable in terms of Varying S_N Sizes

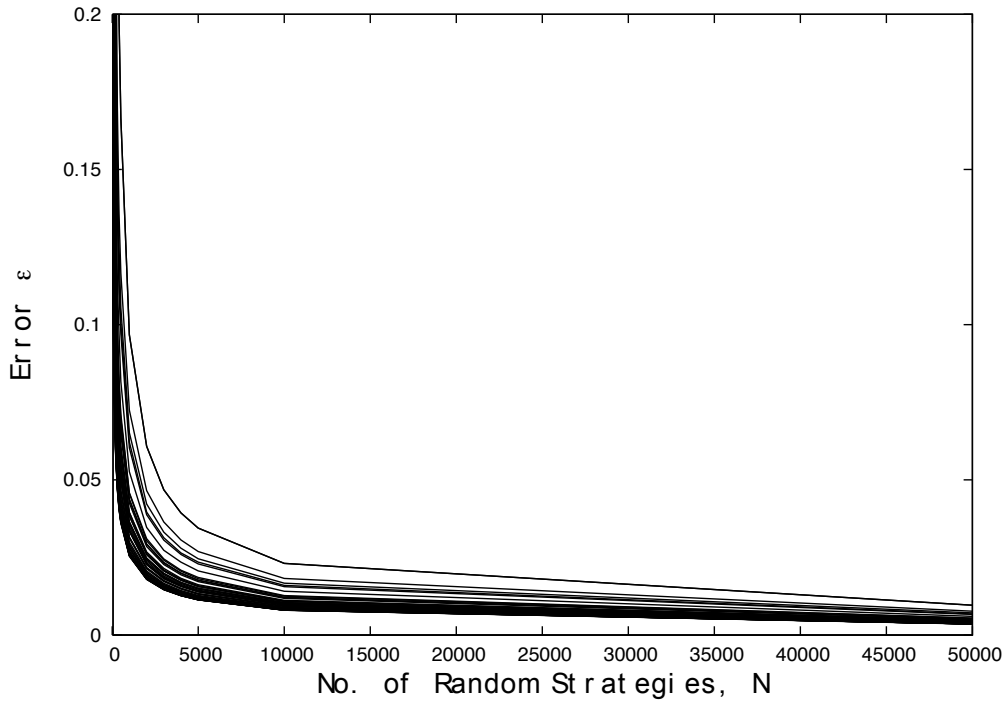


Figure: Empirical results for 50 strategies i. There is a **trade-off** (around $N = 2000$) between improving estimates and computational cost is roughly the **same for most of the strategies**.

Why Estimations Stable?

1. Note 1: Game outcome is a random variable $G_i(J)$ with finite mean and variance.
2. Note 2: S_N is drawn i.i.d. to compute **generalization estimates** $\hat{G}_i(S_N)$ (that takes the sum $G_i(1) + \cdots + G_i(N)$).
3. Note 3: $\hat{G}_i(S_N)$ is **realization** of a random variable.
4. By **Central Limit Theorem**, $\hat{G}_i(S_N)$ is **Gaussian-distributed** for **large enough** N.

Convergence Speed of $\hat{G}_i(S_N)$ to a Gaussian

1. **Berry-Esseen Theorem**[8]: Cummulative distribution function (cdf) F_i of $Z_i(\mathcal{S}_N)$ converges (pointwise) to the cdf Φ of the standard normal distribution $N(0,1)$: For any $x \in \mathbb{R}$,

$$|F_i(x) - \Phi(x)| \leq \frac{0.7975}{\sqrt{N}} \frac{\rho_i}{\sigma_i^3}. \tag{5}$$

2. Note: The term on RHS is bounded away from zero but is **dominated** by $1/\sqrt{N}$ given **finite** moments (ρ_i and σ_i).

[8] EB Manoukian, *Modern Concepts and Theorems of Mathematical Statistics*, Springer-Verlag, 1986.

Generalization Performance in Coevolutionary Learning

Classical coevolutionary learning (CCL):

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

Strategy Representation

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

Figure: The direct look-up table representation provides one-to-one mapping (strategy representation and behavior) for **deterministic and reactive memory-one** PD strategies. m_{fm} used to represent the first move directly when there is no prior history played yet.

What Measurements can be made?

1. Best,

$$\mathrm{Best}(G_{\mathrm{SPOP}_U}) = \hat{G}_{\mathrm{spop}_1}. \tag{6}$$

2. Average,

$$\mathrm{Avg}(G_{\mathrm{SPOP}_U}) = \frac{1}{U}\left(\sum_{I}^{U} \hat{G}_{\mathrm{spop}_I}\right). \tag{7}$$

3. Ensemble,

$$\mathrm{Ens}(G_{\mathrm{SPOP}_U}) = \frac{1}{N}\left(\sum_{j \in S_N} \min\left(\sum_{I}^{U} G_{\mathrm{spop}_I}(j), G_{\mathrm{MAX}}\right)\right). \tag{8}$$

Two-Choice IPD

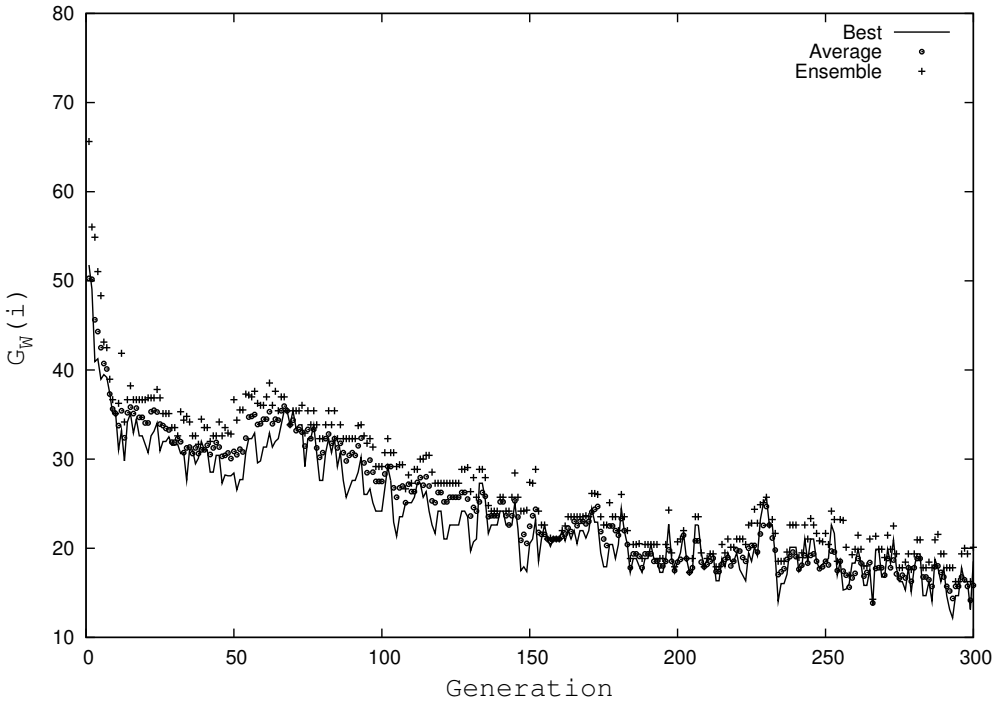


Figure: Generalization performance of CCL defined by $G_W(i)$. All graphs are averaged from measurements over 30 independent runs.

Some Observations

1. The **poor** generalization performance $G_{\mathrm{W}}(i)$ for the simpler *two*-choice IPD is due to the population overspecializing to **naive cooperators**.
2. E.g., representation of ALLC will have $m_{\mathrm{fm}} = +1$ and:

$$\begin{pmatrix} +1 & +1 \\ * & * \end{pmatrix}$$

   where $*$ can be $+1$ or -1 in the *two*-choice IPD.
3. Proportion of ALLC in search space is $n^{(n^2-n)}/n^{(n^2+1)} = 1/n^{(n+1)}$.
4. For *two*-choice IPD, we have **higher proportion** of ALLC. It is **more likely** (and observed) for CCL to search for ALLC.

Coevolutionary Learning with Speciation

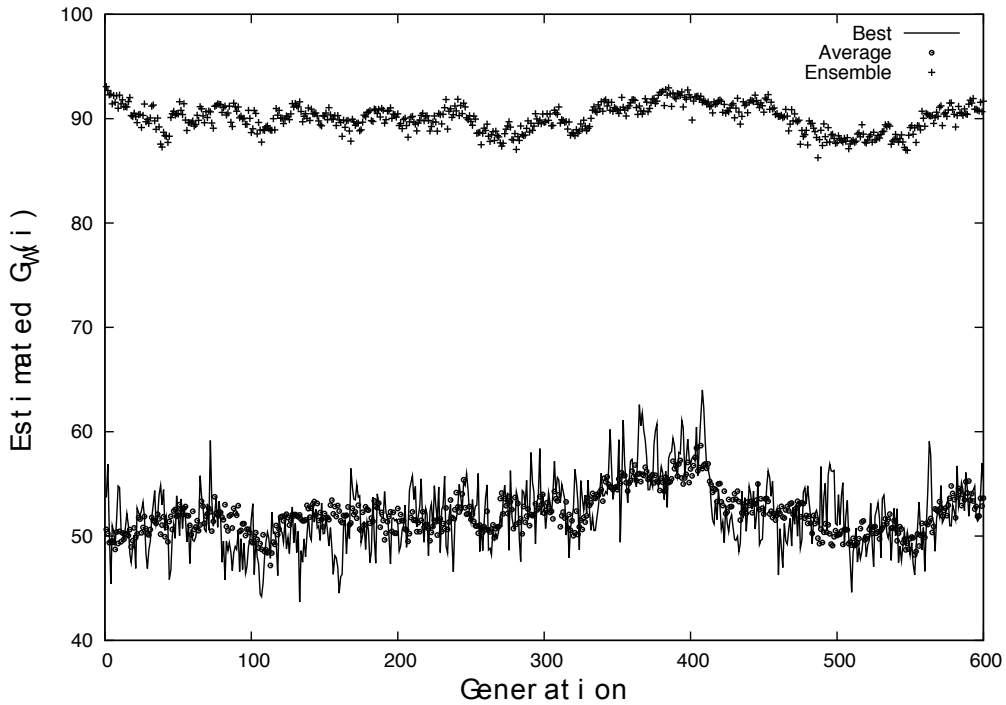


Figure: For the *four*-choice IPD, ensemble of individual speciated strategies covers (defeats) a significantly larger number of opponents, i.e., generalize well.

References and Additional Reading

1. S. Y. Chong and X. Yao, "Behavioral Diversity, Choices, and Noise in the Iterated Prisoner's Dilemma," IEEE Transactions on Evolutionary Computation, Vol. 9, No. 6, pp. 540-551, Dec. 2005.
2. S. Y. Chong, P. Tino, and X. Yao, "Measuring Generalization Performance in Co-evolutionary Learning," IEEE Transactions on Evolutionary Computation, Vol. 12, No. 4, pp. 479-505, Aug. 2008.