

# The Role of Behavioral Diversity and Difficulty of Opponents in Coevolving Game-Playing Agents

Marcin Szubert, Wojciech Jaśkowski, Paweł Liskowski, and Krzysztof Krawiec

Institute of Computing Science, Poznan University of Technology, Poznań, Poland  
{mszubert,wjaskowski,pliskowski,kkrawiec}@cs.put.poznan.pl

[Preprint – this paper has been accepted for presentation at *EvoApplications 2015*]

**Abstract** Generalization performance of learning agents depends on the training experience to which they have been exposed. In game-playing domains, that experience is determined by the opponents faced during learning. This analytical study investigates two characteristics of opponents in competitive coevolutionary learning: behavioral diversity and difficulty (performance against other players). To assess diversity, we propose a generic intra-game behavioral distance measure, that could be adopted to other sequential decision problems. We monitor both characteristics in two-population coevolutionary learning of Othello strategies, attempting to explain their relationship with the generalization performance achieved by the evolved solutions. The main observation is the existence of a non-obvious trade-off between difficulty and diversity, with the latter being essential for obtaining high generalization performance.

**Keywords:** behavioral diversity, diversity maintenance, test difficulty, competitive coevolution, generalization performance, games, Othello.

## 1 Introduction

Competitive coevolution has been regarded as an appealing alternative to conventional evolutionary algorithms in domains where the objective function is difficult to define or expensive to compute. Instead of relying on a static fitness function, in coevolution individuals are evaluated on the basis of interactions with each other [1]. This approach is particularly suitable to games where it is natural to assess the skills of a given game-playing agent by inspecting how it performs against a set of opponents.

One of the main motivations behind using coevolution in games is the belief that it is able to encourage *arms race* between competing individuals and thus provide a pedagogical series of increasingly complex challenges [2]. However, even if coevolution succeeds in providing progressively more difficult opponents, it can still suffer from undesired phenomena known as coevolutionary pathologies [3,4]. For instance, recent works on Othello demonstrate that in the long-term perspective coevolution tends to overspecialize on beating the strong players while forgetting how to deal with the weaker ones, so that in the end the resulting strategies do not generalize well [5]. Consequently, in terms of generalization performance, coevolution has been found less effective than simple evolution with fitness evaluated against a set of random opponents [6].

In this paper, we attempt to improve the generalization performance of Othello-playing agents by promoting the diversity among the coevolving opponents. In contrast to many genotypic diversity maintenance techniques studied in the past [7,8,9], here we focus on the *behavioral* characteristics of individuals. For this purpose, we define a simple behavioral distance measure, applicable not only to Othello but to any sequential decision making problem. We hypothesize that by promoting both difficulty and behavioral diversity in the population of tests (opponents), coevolution can outperform the evolutionary algorithm with random sampling of opponents. To verify this thesis, we investigate the interplay between the behavioral diversity and difficulty of coevolving opponents as well as the relationships of these two issues with the generalization performance of evolved solutions.

## 2 Diversity Maintenance Techniques

In analogy to the key role played by diversity of living organisms in the theory of natural selection, maintaining diversity of candidate solutions in a population has been long perceived as crucial for the effectiveness of evolutionary algorithms [7]. Lack of diversity has been linked to major problems, including slow progress and premature convergence to suboptimal regions of the search space. In response, numerous diversity maintenance techniques have been proposed to sustain exploration of the search space.

The most popular approaches to diversity maintenance include *crowding* and *fitness sharing* [9], which both modify the selection process by promoting the individuals that are most different from the rest of population. For instance, fitness sharing consists in dividing fitness values by the *niche count* to demote the individuals which are similar to each other (i.e., occupy the same niche). Techniques like these do not directly manipulate genotypes and are thus often referred to as *implicit*; alternatively, diversity can be stimulated *explicitly* by simply increasing the strength of a mutation operator.

In the context of competitive coevolution, deficit of diversity can be linked with frequently reported pathologies such as overspecialization, mediocre stable states [3] or disengagement (loss of fitness gradient [4]). For instance, a population of opponents that has converged to a single difficult opponent may present a too demanding challenge for the candidate solutions and so disengage from them. Besides the conventional fitness sharing, a number of diversity maintenance techniques tailored for coevolution have been employed to remedy such situations. Examples include competitive fitness sharing [10] and reduced parasite virulence [11]. The impact of these techniques on the performance of a single-population coevolution was investigated by Chong *et al.* [12].

Most of diversity maintenance techniques rely on measuring some form of distance between the evolving individuals. Typically, the distance measure is defined in the original search space, i.e., at the level of genotypes. For instance, for solutions represented as vectors of real parameters, the Euclidean distance may be used to assess their similarity.

However, if the mapping between genotypes and phenotypes/behaviors is complex, which is often the case in nontrivial problems like games, individuals that are apart according to a genotypic distance measure can exhibit very similar behaviors. And vice versa: a minute modification of the genotype can fundamentally alter individual’s behavior. For instance, consider the popular WPC strategy representation studied in this paper (cf. Section 4.1). Scaling of the entire genotype (weight matrix) has no effect on strategy behavior whatsoever, because it does not change the ordering of evaluations of board states (cf. Eq. 2). On the other hand, a small modification of a weight associated with, e.g., one of the central board locations, can change the way a strategy plays the opening of a game and so diametrically change its performance.

Promoting genotypic diversity does not guarantee thus diverse behaviors. Since it is the behavior that matters in the end, recent works [13,14] promote diversity by measuring distance directly in the space of behaviors. Although assessing behavioral distance typically requires defining a task-specific measure, generic measures for the entire class of sequential decision problems have been also proposed, based on, e.g., normalized compression distance [15].

In coevolutionary algorithms, competitive fitness sharing [10] can be seen as a step towards behavior-based diversity maintenance, as it measures the distance between individuals with respect to the results of their interactions with individuals in population. For instance, in the context of game-playing, it will promote the strategies winning with the opponents that few other strategies can beat.

In this paper, we consider two-population coevolutionary algorithm and, in contrast to most of the past works, focus on the behavioral diversity in the population of *tests* (opponents) and its relationship with the generalization performance in the population of solutions. To that aim, we devise a novel intra-game behavioral distance measure that compares players with respect to the decisions they make in particular game states. To the best of our knowledge, the only work that employs a comparable behavioral distance in competitive coevolution is the recent paper by Gomes *et al.* [16]. The authors define a task-specific distance measure for the predator-prey pursuit problem and apply it to promote behavioral novelty in both coevolving populations. In this context, the measure proposed in the next section has the advantage of being problem-independent.

### 3 Measuring Distance between Game-Playing Agents

In order to maintain behavioral diversity, we propose a measure of behavioral distance between two game-playing agents. This measure relies on comparing actions the given agents would make in each of a predefined set of game states.

#### 3.1 Behavior Characterization Vector

In sequential decision making problems [17], an agent can be identified with a policy  $\pi : S \rightarrow A$  that, for each possible state of the environment  $s \in S$  produces an action  $a \in A$ . Assuming that the environment has the Markov

property, current state of the environment provides enough information to take an action, i.e., the history of previous states and actions is irrelevant. Therefore, the behavior of an agent can be fully characterized by independently considering the actions it would take in every possible state of the environment. Assuming a finite space of  $m$  states, the behavior of an agent  $t$  equipped with policy  $\pi_t$  can be thus expressed as a vector:

$$\beta_t = \langle \pi_t(s_1), \pi_t(s_2), \dots, \pi_t(s_m) \rangle,$$

which is referred to as a *behavior characterization vector* [16,13]. We denote an agent by ‘ $t$ ’ to emphasize that in this paper we are ultimately interested in the behaviors of *tests* in a two-population coevolutionary algorithm.

$\beta$  captures the complete account of agent’s behavioral characteristics: nothing more can be said about its behavior, because all possible states have been taken into account. In practice however,  $m$  is often prohibitively large ( $\approx 10^{28}$  for the game of Othello considered in this paper), so a technically realizable behavioral analysis needs to sacrifice the completeness by relying on a reasonably sized subset  $S' \subseteq S$  of representative states. How informative a particular implementation of  $\beta$  is, depends on the actual choice of  $S'$ . In board games like Othello, an interaction episode (game) between players always starts from the same initial state. Certain states are thus much more likely to be visited than others, and this observation will motivate the particular choice of  $S'$  we describe in the experimental section.

### 3.2 Mean Behavioral Distance

We employ the behavior characterization vector as a means to measure the diversity in a population of agents – game strategies. Given the behavior characterization vectors  $\beta_t$  and  $\beta_{t'}$  of two game-playing agents (strategies)  $t$  and  $t'$ , determined on the same subset of  $m$  distinct states, we define the *behavioral distance* between them as:

$$d(t, t') = D_{Hamming}(\beta_t, \beta_{t'}) = \sum_{i=1}^m \delta(\beta_t[i], \beta_{t'}[i]), \quad (1)$$

where  $\delta$  is the Kronecker delta. We resort to the Hamming distance, because definitions of actions depend on the problem of consideration. Without referring to problem-specific knowledge, it is impossible to judge how similar two actions are; the only statement that can be made for certain is whether they are identical or not. This is particularly true for Othello where actions are discrete and refer to different board positions. Obviously, in domains with continuous actions (e.g.,  $A = \mathbb{R}$ ), other ways of comparing individuals would apply (and, as a matter of fact, would be unavoidable, as two continuous actions are almost always distinct).

With  $d$  as a means for pairwise comparison of agents’ behaviors, we define the internal diversity of any nonempty set of agents  $T$  as the average behavioral

distance between a pair agents in  $T$ , i.e.,

$$d(T) = \frac{2}{|T|(|T| - 1)} \sum_{t, t' \in T, t \neq t'} d(t, t').$$

By definition,  $d(T) \in [0, m]$ .

## 4 Experimental Setup

In the following, we detail key elements which constitute the conceptual framework of this study: the definition of the game along with representation of its strategies, the algorithms that learn to play the game, and the performance measures used to assess quality and diversity of the obtained game-playing agents.

### 4.1 Othello and WPC representation

**Othello** Othello is a deterministic, perfect information, zero-sum board game played by two players on an  $8 \times 8$  board. It involves black and white pieces. At the beginning of the game, each player has two pieces placed diagonally in the center of the board. The players take turns by placing one new piece on an empty board field. The black player moves first. A move is legal if the newly placed piece makes one or more of the opponent’s pieces enclosed from both ends of a horizontal, vertical or diagonal segment. The enclosed pieces are then changed to the opposite colors. The game ends when neither player has a legal move. A player who has then more pieces on the board wins, or if both players have the same number of pieces, the game ends in a draw.

**Strategy representation** Our agents are represented by position-weighted piece counter (WPC), which is arguably the simplest state evaluation function for Othello [18]. WPC assigns a weight  $w_i$  to board location  $i$  and uses scalar product to calculate the utility  $f$  of a board state  $\mathbf{b}$ :

$$f(\mathbf{b}) = \sum_{i=1}^{8 \times 8} w_i b_i, \quad (2)$$

where  $b_i$  is 0, +1 or  $-1$  for, respectively, an empty location, black piece, or white piece. The game-playing agents interpret  $f(\mathbf{b})$  in a complementary manner: the black player prefers the moves leading towards the states with higher values, whereas the lower values are favored by the white player.

We employ WPC as a state evaluator in a 1-ply setup, i.e., given a board state, a game-playing agent generates all legal moves and applies  $f$  to the resulting states. The state gauged as the most favorable determines the move to be made, while ties are resolved randomly.

## 4.2 Generalization Performance

Our objective is to find game-playing agents that maximize the *expected utility* [19]. We approximate the expected utility by playing a number of games against random opponents. A random opponent is a player drawn from the solution space, i.e., the space of all admissible WPCs, by which we mean WPCs with weights from the interval  $[-10, 10]$ .

The score awarded for a single game is either 0, 0.5 or 1 for lose, draw or win, respectively. For symmetry, we employ double games, where both agents play one game as black and the other as white. The performance measure is the average score in 25 000 double games against random opponents.

Since this measure tests how an individual generalizes over the space of all possible players, it is also referred to as *generalization performance* [6].

## 4.3 Learning Algorithms

Two evolutionary algorithms are employed to learn WPC weights. Both are driven by the interactions that take place between game-playing agents. Each evolutionary run consists of 500 generations, in each of them 5 000 games (2 500 double games) are played, which adds up to the total effort of 2 500 000 games per run.

**Coevolutionary Learning** The first algorithm is a two-population competitive coevolutionary learning (CEL) [2], which maintains individuals separated into two populations: i) candidate solutions, and ii) tests. Tests act as opponents that challenge the candidate solutions. The fitness of a candidate solution is defined as the average result of interactions (i.e., double games) with all tests in the second population, while calculating the fitness of an opponent involves behavioral fitness sharing detailed in Section 4.4.

Both populations employ the  $(\mu, \lambda)$ -evolutionary strategy [20], where  $\mu = 25$ ,  $\lambda = 50$ . Initially, they both contain  $\lambda$  randomly generated individuals — real vectors of WPC weights drawn from  $[-0.2, 0.2]$ . The mutation operator perturbs all the weights using additive noise. The WPC weight  $w'_i$  of an offspring is obtained by adding a small random value to the corresponding weight  $w_i$  of the parent:

$$w'_i = w_i + \delta \cdot \mathcal{U}(-1, 1), \quad (3)$$

where  $\mathcal{U}(-1, 1)$  is a real value drawn uniformly from the range  $[-1, 1]$  and  $\delta$  is the mutation strength. Weights resulting from mutation are clamped to the interval  $[-10, 10]$  effectively making the value equal to the respective bound, e.g., if  $|w'_i| > 10$ , we set  $w'_i := 10$ . Consequently, the search space of strategies is a  $[-10, 10]^{64}$  hypercube. For the population of candidate solutions we use  $\delta_s = 0.1$ , while for tests we consider multiple values  $\delta_t = \{0.1, 1.0, 5.0, 10.0\}$ .

**Random Sampling Evolutionary Learning** We compare CEL with the random sampling evolutionary learning (RSEL) [6]. RSEL maintains a single population of candidate solutions bred in the same way as in CEL and differs from it only in fitness evaluation. In RSEL, the fitness of each candidate solution is computed as an average result of interactions with a sample  $T$  of random opponents, which is drawn once per generation. For fair comparison with CEL, we set  $|T| = \lambda = 50$ .

Notice that the fitness employed in RSEL is an unbiased estimator of the generalization performance of the game-playing agent (cf. Section 4.2). RSEL has been found to surpass both one- and two-population coevolution on generalization performance for 1-ply Othello [6,5].

#### 4.4 Diversity Maintenance Techniques

We equip CEL with two mechanisms that can promote diversity in the population of opponents: 1) increasing strength  $\delta_t$  of the mutation operator, which can explicitly stimulate genotypic diversity, and 2) *Behavioral Fitness Sharing* (BFS) that implicitly promotes behavioral diversity (see Section 2).

BFS augments fitness sharing with behavioral distance between the opponents in the population of tests. The fitness  $f'_i$  of an opponent  $i$  is given by:

$$f'_i = \frac{f_i}{\sum_j \max(0, 1 - \frac{d_{ij}}{\sigma})}, \quad (4)$$

where  $f_i$  is the conventional fitness defined as an average interaction outcome between the opponent  $i$  and the coevolving candidate solutions,  $d_{ij}$  is the behavioral distance between opponent  $i$  and opponent  $j$  (Eq. 1), and  $\sigma$  is the niche radius. Individuals whose distance to each other is lower than  $\sigma$  share the fitness. We use  $\sigma = \{0, 10, 20, 30, 40, 50, 60, 70\}$  and employ BFS in combination with the uniform mutation operator (Eq. 3) and mutation strength  $\delta_t = \{0.1, 1.0, 5.0, 10.0\}$ . Note that by setting  $\sigma$  to 0 we completely turn off fitness sharing.

Computing  $d_{ij}$  involves inspecting the behavior of both strategies on a number of states (cf. Section 3.1). For this purpose, we use a set of all  $m = 71$  distinct Othello states reachable in the first 4 moves of the game.

#### 4.5 Opponent Population Measures

Apart from the generalization performance of candidate solutions, we probe the populations of opponents in CEL with additional ‘instruments’.

The first of them is the **behavioral diversity** in the population of opponents, expressed as the mean behavioral distance (Eq. 3.2). For two-population coevolution, this measure changes over time, since the population of opponents evolves. RSEL, in contrast, draws the opponent uniformly from the WPC space. Assuming that random WPCs are equally likely to make every admissible move in each of the 71 considered initial states, we can calculate the expected behavioral distance between any two opponents  $t$  and  $t'$  used by RSEL analytically:

$$\mathbb{E}[d(t, t')] = \mathbb{E}\left[\sum_{i=1}^m \delta(\beta_t[i], \beta_{t'}[i])\right] = \sum_{i=1}^m \mathbb{E}[\delta(\beta_t[i], \beta_{t'}[i])] = m\left(1 - \frac{1}{bf}\right) \approx 53.25,$$

where  $m = 71$  and  $bf$  is the branching factor (i.e., the average number of possible actions from each state), which at the early stages of the game of Othello is approximately equal to 4.

The second indicator of interest is **opponent difficulty**. In recent years, a significant amount of work has been devoted to study the population of opponents (a.k.a. tests or parasites), which play the role of *evaluation set* in two-population coevolution [6,21,5]. Noteworthy, it has been found that the effectiveness of the evaluation set (or population of tests) depends on its difficulty [22]. That is why, we employ it as another gauge in this study.

Since the game of Othello is symmetric, meaning that the roles of candidate solutions and tests are interchangeable, the difficulty of an opponent (test) boils down to its generalization performance (see Section 4.2). In other words, the better the opponent (generalization performance) the more difficult it is to beat (opponent difficulty). Technically, by opponents' difficulty we mean the average generalization performance of the players in the second co-evolving population.

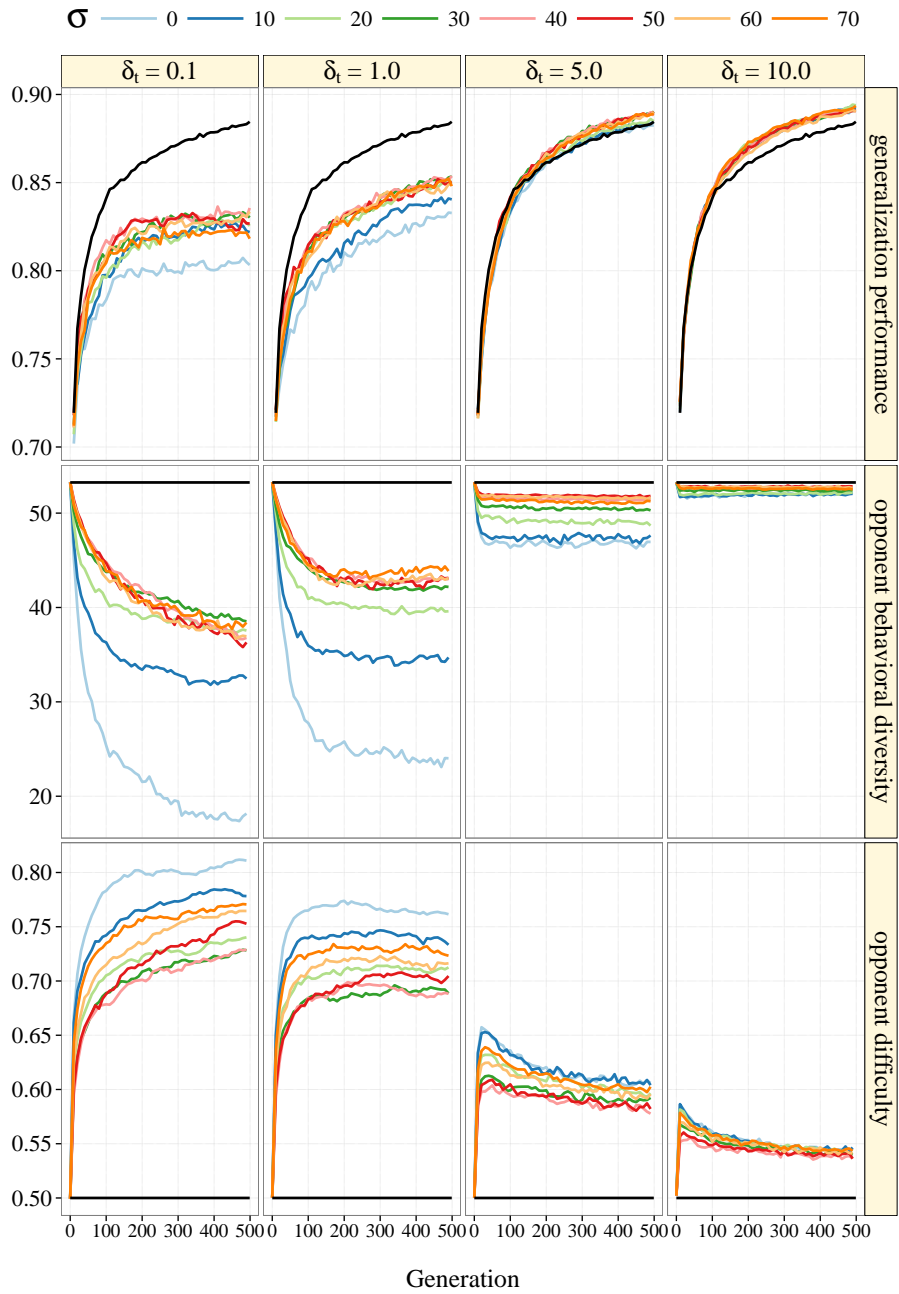
## 5 Results

We performed 100 experimental runs of the CEL algorithm for each combination of  $\delta_t$  and  $\sigma$  parameters and 100 runs of the RSEL algorithm. For each generation, we measured the generalization performance of the individual with the highest fitness in the population of solutions. Additionally, in the case of CEL, we assessed also the behavioral diversity and difficulty in the population of opponents.

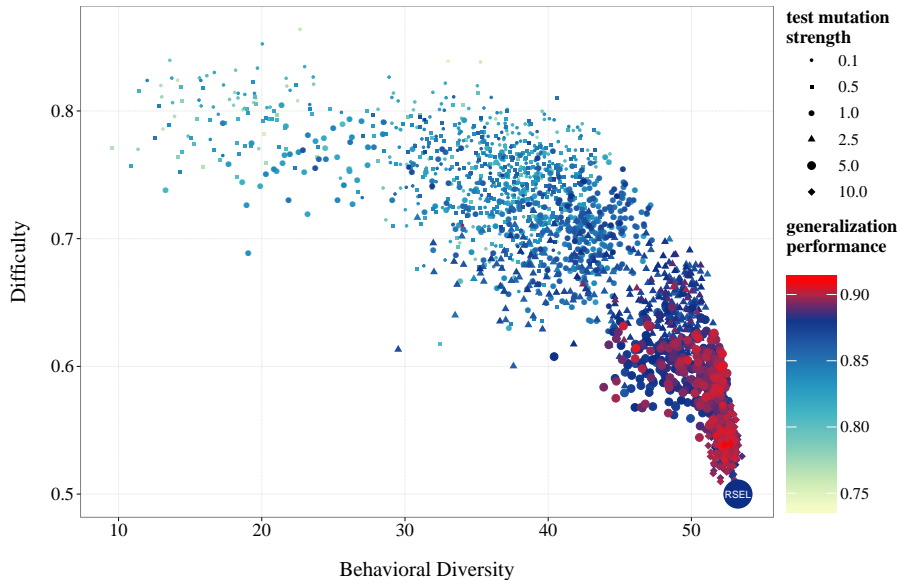
The averaged results of all experiments are illustrated in Figure 1. The figure visualizes the impact of applying particular diversity maintenance techniques with different parameters on the generalization performance of solutions as well as on the behavioral diversity and difficulty of the coevolving opponents. The first observation is that both diversity preserving mechanisms result in improving the generalization performance. In particular, applying sufficiently strong mutation can lead to outperforming the RSEL algorithm (illustrated by the black curve in the plots). Fitness sharing alone does not attain that performance level, but it clearly benefits from moderate niche count (while low values of this parameter are clearly less advantageous).

Besides generalization performance, the figure confirms also that both considered diversity preservation methods succeed in increasing the behavioral diversity in the population of opponents. Finally, the last row of plots demonstrates the significant effect of diversity maintenance techniques on the difficulty of the coevolved opponents. One general observation is that the influence of behavioral fitness sharing technique and its niche radius parameter ( $\sigma$ ) is much larger in the case of moderate mutation strength ( $\delta_t$ ). If the mutation is strong, the population becomes explicitly diversified even without fitness sharing.





**Figure 1.** The generalization performance of solutions vs. behavioral diversity and difficulty of opponents as a function of the number of generations, for different parameters of diversity maintenance techniques. The black series correspond to the RSEL algorithm.



**Figure 2.** The relationship between difficulty and diversity of the coevolving opponents and their impact on the generalization performance of solutions. Each point corresponds to the mean difficulty and diversity in the final generation of tests coevolved in a single evolutionary run. Shapes of points illustrate the strength of mutation operator operating in the population of tests. Colors reflect the generalization performance obtained by the corresponding population of solutions.

To better visualize the relationship between the behavioral diversity and the difficulty of opponents, we prepared a scatter plot shown in Figure 2. Altogether, there are 3200 points in the plot, each of which corresponds to the final population of a single run of coevolutionary algorithm (each of 32 combinations of  $(\sigma, \delta_t)$  parameters times 100 experimental runs). Additionally, an artificial point was added to the plot to illustrate the mean performance obtained with the RSEL algorithm. The coordinates of this point correspond to the expected values of difficulty and diversity in a random sample of opponents.

The figure evidences a trade-off between behavioral diversity and difficulty. The employed algorithms were unable to provide final populations of opponents that were both difficult and diverse at the same time (albeit it is not certain if such sets of opponents *exist* in the first place). Using small mutation strength lead to evolving populations of difficult opponents that behave very similarly to each other. Increasing the mutation strength allows to obtain richer repertoire of opponent behaviors but their mean difficulty simultaneously decreases. Importantly, the highest generalization performance was obtained by coevolutionary algorithms that maintained very diversified population of opponents that were on average slightly, albeit consistently, more difficult than the random ones.

## 6 Conclusions

Encouraging the behavioral diversity, rather than the genotypic one, has been recently successfully applied to improve the performance of evolved robot controllers [13,14]. In this study we have adopted this idea to coevolution of Othello-playing agents, in order to verify whether learning against opponents that are both challenging (difficult) and behaviorally diverse can improve the generalization performance of developed strategies. We expected that the difficulty of opponents would arise naturally due to coevolutionary arms races. To maintain behavioral diversity, we have employed a novel measure of distance between game-playing agents which was integrated with the conventional fitness sharing method. Apart from that we also stimulated diversity explicitly by increasing the genotypic variation among opponents.

Although both methods of sustaining diversity in the population of opponents succeeded in improving behavioral diversity and resulted in higher generalization performance of solutions, they also lead to reduced difficulty of opponents. Due to this trade-off, diverse and simultaneously difficult opponent populations did not emerge, and we were not able to fully verify the initial hypothesis. While failing to provide such populations, behavioral diversity turns out to be a much stronger determinant of performance than difficulty. Though in part anticipated, the extent of this disproportion is rather striking: note that the populations of opponents that secure the best performance of host strategies are only slightly more difficult than the random ones (difficulty 0.6 or less). On the other hand, the high performance of RSEL is consistent with this observation.

## Acknowledgments

This work has been supported by the Polish Ministry of Science and Higher Education, grant No. 09/91/DSMK/0568. W. Jaśkowski has been supported by the Polish National Science Centre grant no. DEC-2013/09/D/ST6/03932.

## References

1. Popovici, E., Bucci, A., Wiegand, R.P., de Jong, E.D.: Coevolutionary Principles. In Rozenberg, G., Bäck, T., Kok, J.N., eds.: Handbook of Natural Computing. Springer (2012) 987–1033
2. Nolfi, S., Floreano, D.: Coevolving Predator and Prey Robots: Do "Arms Races" Arise in Artificial Evolution? *Artificial Life* 4(4) (1998) 311–335
3. Ficici, S.G., Pollack, J.B.: Challenges in Coevolutionary Learning: Arms-race Dynamics, Open-endedness, and Medicocre Stable States. In: Proceedings of the Sixth International Conference on Artificial Life. ALIFE, Cambridge, MA, USA, MIT Press (1998) 238–247
4. Watson, R.A., Pollack, J.B.: Coevolutionary Dynamics in a Minimal Substrate. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), Morgan Kaufmann (2001) 702–709

5. Jaśkowski, W., Liskowski, P., Szubert, M.G., Krawiec, K.: Improving Coevolution by Random Sampling. In: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference. GECCO '13, New York, NY, USA, ACM (2013) 1141–1148
6. Chong, S.Y., Tino, P., Ku, D.C., Yao, X.: Improving Generalization Performance in Co-Evolutionary Learning. *IEEE Transactions on Evolutionary Computation* **16**(1) (2012) 70–85
7. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms and Their Application, Hillsdale, NJ, USA, L. Erlbaum Associates Inc. (1987) 41–49
8. Mahfoud, S.W.: Niching Methods for Genetic Algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL (1995)
9. Sareni, B., Krahenbuhl, L.: Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation* **2**(3) (1998) 97–106
10. Rosin, C.D., Belew, R.K.: New Methods for Competitive Coevolution. *Evolutionary Computation* **5**(1) (1997) 1–29
11. Cartlidge, J., Bullock, S.: Combating coevolutionary disengagement by reducing parasite virulence. *Evolutionary Computation* **12**(2) (2004) 193–222
12. Chong, S.Y., Tino, P., Yao, X.: Relationship Between Generalization and Diversity in Coevolutionary Learning. *IEEE Transactions on Computational Intelligence and AI in Games* **1**(3) (2009) 214–232
13. Lehman, J., Stanley, K.O.: Abandoning Objectives: Evolution through the Search for Novelty Alone. *Evolutionary Computation* **19**(2) (2011) 189–223
14. Mouret, J.B., Doncieux, S.: Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation* **20**(1) (2012) 91–133
15. Gomez, F.J.: Sustaining diversity using behavioral information distance. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. GECCO '09, New York, NY, USA, ACM (2009) 113–120
16. Gomes, J.C., Mariano, P., Christensen, A.L.: Novelty search in competitive coevolution. In: Proceedings of the 13th International Conference on Parallel Problem Solving from Nature - PPSN XIII, Ljubljana, Slovenia (2014) 233–242
17. Moriarty, D.E., Schultz, A.C., Grefenstette, J.J.: Evolutionary Algorithms for Reinforcement Learning. *Journal of Artificial Intelligence Research* **11** (1999) 241–276
18. Lucas, S.M., Runarsson, T.P.: Temporal Difference Learning Versus Co-Evolution for Acquiring Othello Position Evaluation. In Louis, S.J., Kendall, G., eds.: Proceedings of the 2006 IEEE Symposium on Computational Intelligence and Games, CIG 2006, IEEE (2006) 52–59
19. de Jong, E.D.: The MaxSolve Algorithm for Coevolution. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation. GECCO '05, New York, NY, USA, ACM (2005) 483–489
20. Beyer, H.G., Schwefel, H.P.: Evolution strategies—a comprehensive introduction. *Natural computing* **1**(1) (2002) 3–52
21. de Jong, E.D., Pollack, J.B.: Ideal Evaluation from Coevolution. *Evolutionary Computation* **12**(2) (2004) 159–192
22. Szubert, M.G., Jaśkowski, W., Liskowski, P., Krawiec, K.: Shaping Fitness Function for Evolutionary Learning of Game Strategies. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation. GECCO '13, New York, NY, USA, ACM (2013) 1149–1156