

How game complexity affects the playing behavior of synthetic agents

Chairi Kiourt¹, Dimitris Kalles¹, and Panagiotis Kanellopoulos²

¹School of Science and Technology, Hellenic Open University, Patras, Greece, chairik, kalles@eap.gr

²CTI Diophantus and University of Patras, Rion, Greece. kanellop@ceid.upatras.gr

Abstract—Agent based simulation of social organizations, via the investigation of agents’ training and learning tactics and strategies, has been inspired by the ability of humans to learn from social environments which are rich in agents, interactions and partial or hidden information. Such richness is a source of complexity that an effective learner has to be able to navigate. This paper focuses on the investigation of the impact of the environmental complexity on the game playing-and-learning behavior of synthetic agents. We demonstrate our approach using two independent turn-based zero-sum games as the basis of forming social events which are characterized both by competition and cooperation. The paper’s key highlight is that as the complexity of a social environment changes, an effective player has to adapt its learning and playing profile to maintain a given performance profile.

I. INTRODUCTION

Turn-based zero-sum games are most popular when it comes to studying social environments and multi-agent systems [1]–[3]. For a game agent, the social environment is represented by a game with all its agents, components and entities, such as rules, pay-offs and penalties, amongst others [2], [4], [5], while learning in a game is said to occur when an agent changes a strategy or a tactic in response to new information [5]–[8]. Social simulation involves artificial agents with different characteristics (synthetic agents), which interact with other agents, possibly employing a mix of cooperative and competitive attitudes, towards the investigation of social learning phenomena [4], [5], [9].

The mimicking of human playing behaviors by synthetic agents is a realistic method for simulating game-play social events [5], where the social environment (games) as well as the other agents (opponents) [10], [11] are among the key factors which affect the playing behavior of the agents.

The solvability of board games is being investigated for over 25 years [12]–[15]. Several studies focusing on board game complexity have shown that board games vary from low to high complexity levels [13]–[15], which are mainly based on the game configuration and the state space of the game, with more complex games having larger rule set and more detailed game mechanics. In general, solvability is related to the *state-space complexity* and *game-tree complexity* of games [14], [15]. The *state-space complexity* is defined as the number of legal game positions obtainable from the initial position of the game. The *game-tree complexity* is defined as the number of leaf nodes in the solution search tree of the initial position

of the game. In our investigation, we adopted the *state-space complexity* approach, which is the most-known and widely used [13]–[15].

The complexity of a large set of well-known games has been calculated [14], [15] at various levels, but their usability in multi-agent systems as regards the impact on the agents’ learning/playing progress is still a flourishing research field.

In this article, we study the game complexity impact on the learning/training progress of synthetic agents, as well as on their playing behaviors, by adopting two different board games. Different playing behaviors [5] are adopted for the agents’ playing and learning progress. We experiment with varying complexity levels of *Connect-4* (a medium complexity game) and *RLGame* (an adaptable complexity game). These two different games cover an important range of diverse social environments, as we are able to experiment at multiple complexity levels, as determined by a legality-based model for calculating state-space complexity. Our experiments indicate that synthetic agents mimic quite well some human-like playing behaviors in board games. Additionally, we demonstrate that key learning parameters, such as exploitation-vs-exploration trade-off, learning backup and discount rates, and speed of learning are important elements for developing human-like playing behaviors for strategy board games. Furthermore, we highlight that, as the complexity of a social environment changes, the playing behavior (essentially, the learning parameters set-up) of a synthetic agent has to adapt to maintain a given performance profile.

II. BACKGROUND KNOWLEDGE

In this section, we describe the games adopted for the experimental sessions, the structure of the synthetic agents’ learning mechanisms and the development of the social environments.

Connect-4 is a relatively recent game, fairly similar to *tic-tac-toe*, but uses a 6×7 board with gravity. Both agents have 21 identical ‘coins’, and each agent may only place its coins in the lowest available slot in a selected column (essentially, by inserting a coin at the free top of the column and allowing it to “fall”). The goal of the game is to connect four of one’s own coins of the same color next to each other vertically, horizontally or diagonally before the opponent reaches that goal. If all of both agents’ coins are placed and no agent has achieved this goal, the game is a draw. *Connect-4* is a turn-based game and each agent has exactly one move per turn.

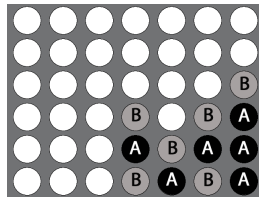
Fig. 1: A *Connect-4* game in which player B wins.

TABLE I: A description of game configurations

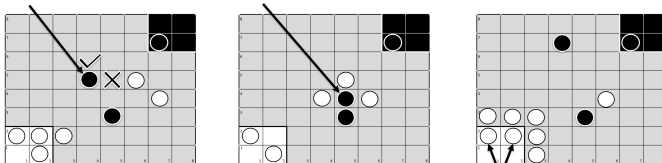
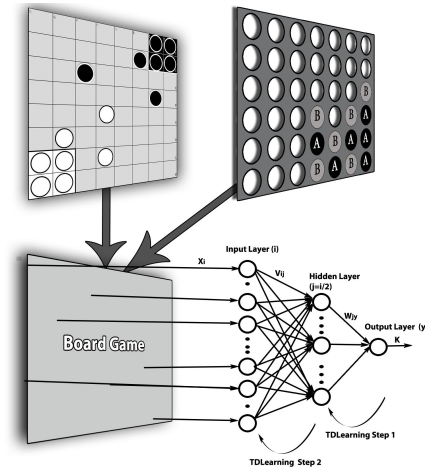
Board size (n)	5, 6, 7, 8, 9, 10
Base size (α)	2, 3, 4
Number of pawns (β)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

It has a medium state space complexity of 4.5×10^{12} board positions [16]. Fig. 1 depicts an example of the *Connect-4* game, in which agent B wins the game.

RLGame is a board game [17] involving two agents and their pawns, played on a square board. Two $\alpha \times \alpha$ square bases are on opposite board corners; these are initially populated by β pawns for each agent, with the white agent starting from the lower left base and the black agent starting from the upper right one. The possible configurations of the *RLGame* are presented in Table I. The goal for each agent is to move a pawn into the opponent's base or to force all opponent pawns out of the board (it is the player not the pawn who acts as an agent, in our scenario). The base is considered as a single square, therefore a pawn can move out of the base to any adjacent free square. Agents take turns and pawns move one at a time, with the white agent moving first. A pawn can move vertically or horizontally to an adjacent free square, provided that the maximum distance from its base is not decreased (so, backward moves are not allowed). A pawn that cannot move is lost (more than one pawn may be lost in one move). An agent also loses by running out of pawns.

The implementation of some of the most important rules is depicted in Fig.2. In the leftmost board the pawn indicated by the arrow demonstrates a legal ("tick") and an illegal ("cross") move, the illegal move being due to the rule that does not allow decreasing the distance from the home (black) base. The rightmost boards demonstrate the loss of pawns, with arrows showing pawn casualties. A "trapped" pawn, either in the middle of the board or when there is no free square next to its base, automatically draws away from the game.

For our study, in both games, each agent is an autonomous system that acts according to its characteristics and knowledge. The learning mechanism used (Fig. 3) is based on reinforcement learning, by approximating the value function with a neural network [2], [4], as already documented in similar studies [18], [19]. Each autonomous (back propagation) neural

Fig. 2: Example of *RLGame* rules into action.Fig. 3: Learning mechanism of *RLGame* and *Connect-4*

network [20] is trained after each player makes a move. The board positions for the next possible move are used as input-layer nodes, along with flags regarding the overall board coverage. The hidden layer consists of half as many hidden nodes. A single node in the output layer denotes the extent of the expectation to win when one starts from a specific game-board configuration and then makes a specific move. After each move the values of the neural network are updated through the *temporal difference learning method*, which is a combination of Monte Carlo and dynamic programming [20]. As a result, collective training is accomplished by putting an agent against other agents so that knowledge (experience) is accumulated.

For both games, the agent's goal is to learn an optimal strategy that will maximize the expected sum of rewards within a specific amount of time, determining which action should be taken next, given the current state of the environment. The strategy to select between moves is ϵ -Greedy (ϵ), with ϵ denoting the probability to select the best move (exploitation), according to present knowledge, and $1 - \epsilon$ denoting a random move (exploration) [21]. The learning mechanism is associated with two additional learning parameters, Gamma (γ) and Lambda (λ). A risky or a conservative agent behavior is determined by the parameter, which specifies the learning strategy of the agent and determines the values of future payoffs, with values in $[0, 1]$; effectively, *large values are associated with long-term strategies*. The speed and quality of agent learning is associated with λ , which is the learning rate of the neural network, also in $[0, 1]$. *Small values of λ can result in slow, smooth learning; large values could lead to accelerated, unstable learning*. These properties are what we, henceforth, term as "characteristic values" for the playing agents.

RLGame and *Connect-4*, in their tournament versions [5], both fit the description of an autonomous organization and of a social environment, as defined by Ferber et al. [1], [4]. Depending on the number of agents, social categories can be split into sub-categories of *micro-social environments*, *environments composed of agent groups* and *global societies*, which are the next level of the cooperation and competition

extremes of social organizations [2], [4].

On one hand, *RLGame* was chosen because it is a fairly complex game for studying learning mechanism and developing new algorithms, because all the pawns of the game have the same playing attributes. It is not as complicated as *Chess*, where almost all pieces have their own playing attributes, or *Go*, which would make it difficult to study in detail the new learning algorithms. Furthermore, its complexity scales with the number of pawns and board dimensions, which allows for fewer non-linear phenomena that are endemic in games like *Chess*, *Go*, or *Othello* (for example, knight movement in *Chess* or column color inversion in *Othello*, are both instances of such phenomena). We view this as a key facilitator in our quest for opponent modelling (but acknowledge the importance and the interestingness of non-linear aspects of game play). On the other hand, *Connect-4* was chosen due to its low complexity. With these two different games, we believe that we cover a quite important range of diverse environments, as we can accommodate several levels of complexity in *RLGame* and pretty low complexity in *Connect-4*.

III. GAME COMPLEXITY

Combinatorial game theory provides several ways to measure the game complexity of two-person zero-sum games with perfect information [13], [14], such as: *state-space complexity*, *game tree size*, *decision complexity*, *game-tree complexity* and *computational complexity*. In this study, we use the *state-space complexity* approach, which is the most known and widely used [13]–[15]. Nowadays, dozens of games are solved by many different algorithms [14], [15].

Connect-4 is one of the first turn-based zero-sum games solved by computer [12]. It has a medium state space complexity of 4.5×10^{12} board positions in 6×7 board size [16]. Tromp [22] presented some game theoretical values of *Connect-4* on medium board sizes up to $width + height = 15$, some of which are presented in Table II [23].

TABLE II: *Connect-4*, game configurations associated to their state-space sizes.

Height, Width (size)	State space complexity	β (coins per player)
8,2	1.33×10^4	8
8,3	8.42×10^6	12
8,4	1.10×10^9	16
7,4	1.35×10^8	14
7,5	1.42×10^{10}	17.5
6,5	1.04×10^9	15
6,6	6.92×10^{10}	18
6,7	4.53×10^{12}	21
5,5	6.98×10^7	12.5
5,6	2.82×10^9	15
5,7	1.13×10^{10}	17

The complexity of the *RLGame* depends mainly on the value of parameters n , α , and β . The number of the various positions that might occur is bounded from above by:

$$\sum_{i=1}^{\beta} \sum_{j=1}^{\beta} \binom{n^2 - 2\alpha^2}{i+j} \binom{i+j}{i} (1+2(\beta-i))(1+2(\beta-j)). \quad (1)$$

The first (leftmost) term denotes the number of ways to place $i+j$ pawns in the playing field (on the board but outside the bases) and the second term denotes the number of ways to partition these $i+j$ pawns into i white and j black pawns. The two rightmost terms intend to capture, for each given configuration of i white and j black pawns in the playing field, the additional number of positions that may occur because each player might have pawns in its own base and no pawn in the enemy base (there are $\beta - i$ such configurations for the white player) or a single pawn in the enemy base and possibly some pawns in its own base (again, there are $\beta - i$ such configurations for the white player).

Naturally, the above formula overestimates the number of possible states since it also includes illegal states, so we devised a simple simulation with the following steps to derive a better estimate.

- Given n , α , and β , we examine all valid configuration profiles (n, α, β, i, j) where i, j denote the number of white and black pawns in the playing field.
- We generated 1000 random positions per valid profile and tested whether some of them contained dead pawns (e.g., pawns with no legal moves).
- For each configuration profile, we multiplied the fraction of such “legit” positions (we did not check whether a position without dead pawns can actually arise in a real game) with $[1 + 2(\beta - i)][1 + 2(\beta - j)]$ to take into account the $\beta - i$ (respectively, $\beta - j$) white (respectively, black) pawns that are not in the playing field for each configuration profile.
- We summed the number of “legit” positions over all configuration profiles for the given values of n , α and β and calculated the ratio of “apparently legit states” provided by this simulation over the “theoretical estimation” provided by the formula.

Table I reviews the (n, α, β) configurations we used; since bases should be at least one square apart in any given board, we eventually end-up with fewer valid (n, α) combinations (shown alongside the results in Table III). Additionally, for valid configurations we demand that $0 < i \leq \beta$ and $0 < j \leq \beta$.

TABLE III: *RLGame*, games’ extreme configurations associated to their state-space sizes. We state the theoretical upper bound and the ratio of “legit” positions that arose in simulations.

Board, Base (size)	$\beta = 1$		$\beta = 10$	
	formula	ratio	formula	ratio
5,2	3.83×10^2	.991	1.11×10^{10}	.127
6,2	9.33×10^2	.997	1.50×10^{14}	.088
7,2	1.89×10^3	.999	6.93×10^{17}	.177
7,3	1.12×10^3	.994	1.37×10^{15}	.113
8,2	3.43×10^3	.998	7.21×10^{20}	.373
8,3	2.36×10^3	1.	9.10×10^{18}	.254
9,2	5.70×10^3	.996	2.40×10^{23}	.562
9,3	4.29×10^3	.997	9.64×10^{21}	.486
9,4	2.66×10^3	1.	3.72×10^{19}	.315
10,2	8.93×10^3	1.	3.50×10^{25}	.712
10,3	7.14×10^3	1.	2.96×10^{24}	.645
10,4	4.97×10^3	.998	5.12×10^{22}	.530

We only report the state-space size for the extreme cases of

$\beta = 1$ and $\beta = 10$ for each (n, α) configuration used, since we observed that the approximation ratios strictly decrease with increasing values of β (thus creating more room for pawn interdependencies which lead to illegal moves). These results are shown in Table III and confirm that, even for relatively small dimensions, state space complexity is well over 10^{10} .

IV. EXPERIMENTAL SESSIONS

In order to study the game complexity effect in synthetic agents' learning/training process as well as in their playing behaviors, in multi-agent social environments, three independent tournament sessions (experiments) with the same pre-configurations were designed and run for both *RLGame* and *Connect-4*; for simplicity, we will name these tournament sessions as $RL - R(x \times y)$ for *RLGame* and $C4 - R(x \times y)$ for *Connect-4*, where $(x \times y)$ presents the game configuration. Table IV presents the game configurations selected for the tournament sessions (experiments). We chose three different game configurations for each game, in order to study three different complexity level of each game. We remark that in the following we only compare agents playing the same game; we never compare an agent from *RLGame* to an agent from *Connect-4*.

According to the scenario of these tournaments sessions, we initiated 64 agents in a *Round Robin* tournament with 10 games per match. All agents had different characteristic value configurations for ϵ, γ and λ , with values ranging from 0.6 to 0.9, with an increment step of 0.1. Four different values for each characteristic value ($\epsilon, \gamma, \lambda$), implies $4^3 = 64$ agents with different playing behaviors (different characteristic values). Each agent played 63 matches against different agents, resulting in a total number of $\binom{64}{1} \times 10 = 200,160$ games, for each tournament session. All tournament sessions were identical in terms of agent configurations and flow of execution.

The ranges of the characteristic values ($\epsilon, \gamma, \lambda$) are selected, because of their association with the playing behaviors of the agent [5]. For example, if we had an agent that exploits 5% of its knowledge (ϵ), then it almost always learn something new and would only rarely demonstrate what it learned [20], [24]. Also, if we set $\lambda = 0.05$, the agent would learn very slow, which is not effective in case the opponent opts to play head-on attack (one pawn moving directly to the opponent base for *RLGame*), as an agent with a low λ may be less interested to learn a more structured strategy by using many pawns that may defend its base or to force opponent pawns out of the board. Wiering et al. [25] suggested that λ values larger than 0.6 perform best. The discount rate parameter, γ , as reported by Sutton and Barton [20], tilts the agent towards being myopic and only concerned with maximizing immediate rewards when $\gamma = 0$, while it allows the agent to become more farsighted and take future rewards into account more strongly when $\gamma = 1$. For this reason, on one hand, by setting the γ values roughly to 0.6, we may say that the agent adopts short term strategies (risky), on the other hand, by setting the γ values to 0.9 we represent the agents with long term strategies (conservative agents). With the characteristic values $\epsilon, \gamma, \lambda$ ranging between 0.6 and 0.9, we kept a balance.

Based on the agents' characteristic values ($\epsilon, \gamma, \lambda$) and their performance, we developed a set of playing behavior descriptors [5], see Table V.

The first three descriptors are composed from the characteristic values derived from previous experiments [5]. Those three descriptors define the characteristics limits, which determine playing behaviors depending on their preferred strategies. Simply put, every descriptor may represent a synthetic agent's playing behavior in the experimental social environment. An example of synthetic agent's playing behavior is that a 'Knowledge Exploiter' (high ϵ value) and 'Conservative' (high γ value) and 'Fast, Unstable Learner' (high λ values) agent tends to be 'Bad playing' (high r value), which we do not consider positive for a game-playing agent.

The agents are rated by using the *ReSkill* tool [26]. All the last ratings of tournament sessions are converted to rankings (r), in order to compare more effectively the experiments by using statistical methods, such as the *Spearman's rank correlation coefficient* (ρ) [27], which measures the statistical dependence between two variables, and is specifically efficient at capturing the monotonic (non-linear, in general) correlation on ranks and the *Kendall rank correlation coefficient* (τ) [28], which measures the ordinal association between two measured quantities, both considered as adequate statistical measures to compare ranking lists quantitatively [29]. As known, the range of both coefficients falls within $[-1, 1]$, with high negative values representing strong negative correlation, low absolute values representing small or no correlation and high positive values representing strong positive correlation. Table VI shows a *Spearman's* and *Kendall's* correlation coefficients distance heat-map, for the tournament sessions introduced in Table IV. The top value of each cell shows the ρ correlation coefficient while the bottom value of each cell the correlation coefficient. Darker gray cells indicate a high correlation between two tournament sessions (agent rankings), while lighter gray cells indicate a strong negative correlation. Table VI also represents an indicative correlation between the state-space complexities of the social environments.

In order to verify the tournament sessions' correlations, we applied a *k-means* clustering for all tournament sessions and we developed the heat-maps of Fig. 4. We set the number of the *k-means* clusters fixed to 3 (C1, C2 and C3), to build three clusters based on the agents' performance (by using the agents' rankings from each tournament sessions). Also, we set the re-runs of the *k-means* algorithm to 100 and the *maximal iterations* within each algorithm run to 300. Due to the number of the agents (64) and the number of the tournament sessions (3 for each game), the *k-means* configuration was good enough to show the best correlation between the agents' performances associated to the tournament sessions. We tested the *k-means* algorithm with larger number of *re-runs* and *maximal iteration* but there was no difference in the result. Fig. 4 presents three rows for each game (one for each tournament session) and 64 columns (one for each agent). The columns are separated in three clusters for each game. Each cluster (C1, C2 and C3) depicts the association of the agents, based on their rankings in the three tournament sessions. Each agent (rows in the graphs) is composed from three colored cells, where each

TABLE IV: Selected game configurations for the tournament sessions (experiments).

<i>Connect-4</i>			<i>RLGame</i>		
Experiment (Tournament) name	Size (Height, Width)	State space complexity	Experiment (Tournament) name	Size (Board, Base)	State space complexity ($\beta = 10$)
$C4 - R(8 \times 3)$	8,3	8.42×10^6	$RL - R(5 \times 2)$	5,2	1.11×10^{10}
$C4 - R(7 \times 4)$	7,4	1.35×10^8	$RL - R(7 \times 2)$	7,2	6.93×10^{17}
$C4 - R(6 \times 7)$	6,7	4.53×10^{12}	$RL - R(10 \times 2)$	10,2	3.50×10^{25}

TABLE V: Agents playing behavior descriptors based on their characteristic values and their performance

Characteristic Values	Key parameters (Playing behavior descriptors)
$0.6 \leq \epsilon \leq 0.9$	Exploration, exploitation tradeoff (knowledge explorer to exploiter)
$0.6 \leq \gamma \leq 0.9$	Learning back-up and discount rates (risky to conservative, short to long term strategies)
$0.6 \leq \lambda \leq 0.9$	Speed & stability of learning (slow smooth to fast and unstable learning)
$1 \leq r \leq 64$	Agents' rankings, performance (good playing to bad playing agents)

cell depicts the performance of the agent in the corresponding tournament session. The colored bars, from light grey to dark grey, at the right of each graph, depict the ranking positions. In example, each dark gray cell depicts a bad playing agent in the corresponding tournament (row), the darkest cell of the C3 cluster, tournament session $C4 - R(8 \times 3)$, shows the worst playing agent of that experimental session, which was ranked in the 64th position in the last round of the tournament. The correlation between the agents of each cluster (C1, C2 and C3), of each game, is depicted by a tree graph (dendrogram) in the top of each cluster. Each row (tournament session) and column (agents) are clustered by leaf ordering. As leaves we mean the lines (leaf of the dendrogram) that show the correlation between two variables (agents or tournament sessions). For example, the leaves: $C4 - R(8 \times 3)$ and $C4 - R(7 \times 4)$ are higher related (rows of *Connect-4* game), than the leaf $C4 - R(6 \times 7)$, which differs more than the two other leaves. This can be confirmed if one checks the color shades of the cells (agent) in the three tournaments (three cells in a row). If an agent has similar color shades in the three cells, it means that the agent performs the same in the three tournament sessions of the game. For example, the top performer agent of C1 cluster in *Connect-4* game is *Agt_48*, each cell of each tournament session has intense light gray color.

Fig. 5 depicts the spatial allocation of each cluster, resulting from the *k-means* clustering (Fig. 4), associated to the average number of the agents' characteristic values (ϵ - γ - λ), respectively for each game (*Connect-4* and *RLGame*). The shapes in the graphs in Fig. 5 indicate the state-space complexity of the different tournament sessions of each game. The circles represent the high state-space complexities, triangles represent the medium state-space complexities and the squares represent the low state-space complexities respectively for each game. The colors of the shapes represent the C1, C2 and C3 clusters. In example, the black square in the left graph depicts the C3 cluster (bad playing agents) of the *Connect-4*'s lowest state space complexity, in a special allocation of the characteristic

values (ϵ - γ - λ). This means that the bad playing agents of the *Connect-4*'s lowest complexity, seem to have low ϵ -greedy ($\epsilon \approx 0.68$), high lambda ($\lambda \approx 0.85$) and medium gamma ($\gamma \approx 0.72$). If we associate these characteristic values with the playing behaviour descriptors of Table V, we can say that a bad playing agent in a low complexity environment of the *Connect-4* game, seems to be an "exploiter", a "fast, unstable learner", which takes into account "medium-term strategies".

V. DISCUSSION

The correlation coefficient analysis that compared all the tournament sessions of both games (Table VI) shows a high correlation coefficient between the three tournament sessions of *RLGame*. The correlation coefficient between two experiments (two different tournament sessions) presents the similarity or the differentiation of the agents' performances (agents with the same playing profile) in the studied experimental state spaces. *Connect-4*'s tournament sessions show a quite good correlation between the two lower complexity state-spaces ($C4 - R(8 \times 3)$ and $C4 - R(7 \times 4)$), while the correlation of the higher complexity state space compared to the two lower complexity state-spaces of the *Connect-4* appears to be neutral, with about 0 correlation coefficient ($C4 - R(8 \times 3)$ and $C4 - R(7 \times 4)$ correlation compared to $C4 - R(6 \times 7)$). An important highlight is, that while the complexity of the *Connect-4* increases, the negative correlation between the *Connect-4*'s and *RLGame*'s tournament sessions decreases (third column and last three rows of Table 6). For example, the correlation between $C4 - R(8 \times 3)$ and all *RLGame* tournament sessions show an average $\rho \approx -0.268$ and $\tau \approx -0.177$, while the correlation between the $C4 - R(6 \times 7)$ and all *RLGame* tournament sessions, shows an average $\rho \approx -0.191$ and $\tau \approx -0.128$, which is an increase of 4% for ρ and 3% for correlations. This highlights that as the complexity level of *Connect-4* increases (referring to the $C4 - R(6 \times 7)$ variant), stronger positive correlation with all the tournament session of *RLGame* is observed, as both ρ and τ values increase from negative to 0. Generally in *RLGame*, agents with similar playing profiles behave in the same way as the state complexity of *RLGame* changes, while this is not the case for agents in *Connect-4*. We had originally reported that we attributed the differences in performance of agents of the same set-up to the different complexity of the *Connect-4* and *RLGame* games. This is further strengthened by the finding that a *Connect-4* variant of higher complexity is closer to *RLGame*.

The *k-means* clustering shows a higher correlation between the *RLGame* tournament sessions than the corresponding *Connect-4*'s tournament sessions, which is depicted by the heat-maps of Fig. 4 and supports the results of correlation

TABLE VI: Spearman's and Kendall's correlation coefficients comparison of each tournament session, presented as a distance heat-map, where high distances are presented with light gray and smaller distances with darker gray

	$C4 - R(8 \times 3)$ 8.42×10^6	$C4 - R(7 \times 4)$ 1.35×10^8	$C4 - R(6 \times 7)$ 4.53×10^{12}	$RL - R(5 \times 2)$ 1.11×10^{10}	$RL - R(7 \times 2)$ 6.93×10^{17}	$RL - R(10 \times 2)$ 3.50×10^{25}
$C4 - R(8 \times 3)$ 8.42×10^6	1	0.340	-0.043	-0.340	-0.274	-0.192
$C4 - R(7 \times 4)$ 1.35×10^8	0.340	1	0.090	-0.477	-0.5	-0.518
$C4 - R(6 \times 7)$ 4.53×10^{12}	-0.043	0.090	1	-0.179	-0.167	-0.229
$RL - R(5 \times 2)$ 1.11×10^{10}	-0.340	-0.477	-0.179	1	0.673	0.720
$RL - R(7 \times 2)$ 6.93×10^{17}	-0.274	-0.500	-0.167	0.673	1	0.740
$RL - R(10 \times 2)$ 3.50×10^{25}	-0.192	-0.518	-0.229	0.720	0.740	1
	-0.134	-0.362	-0.138	0.519	0.561	

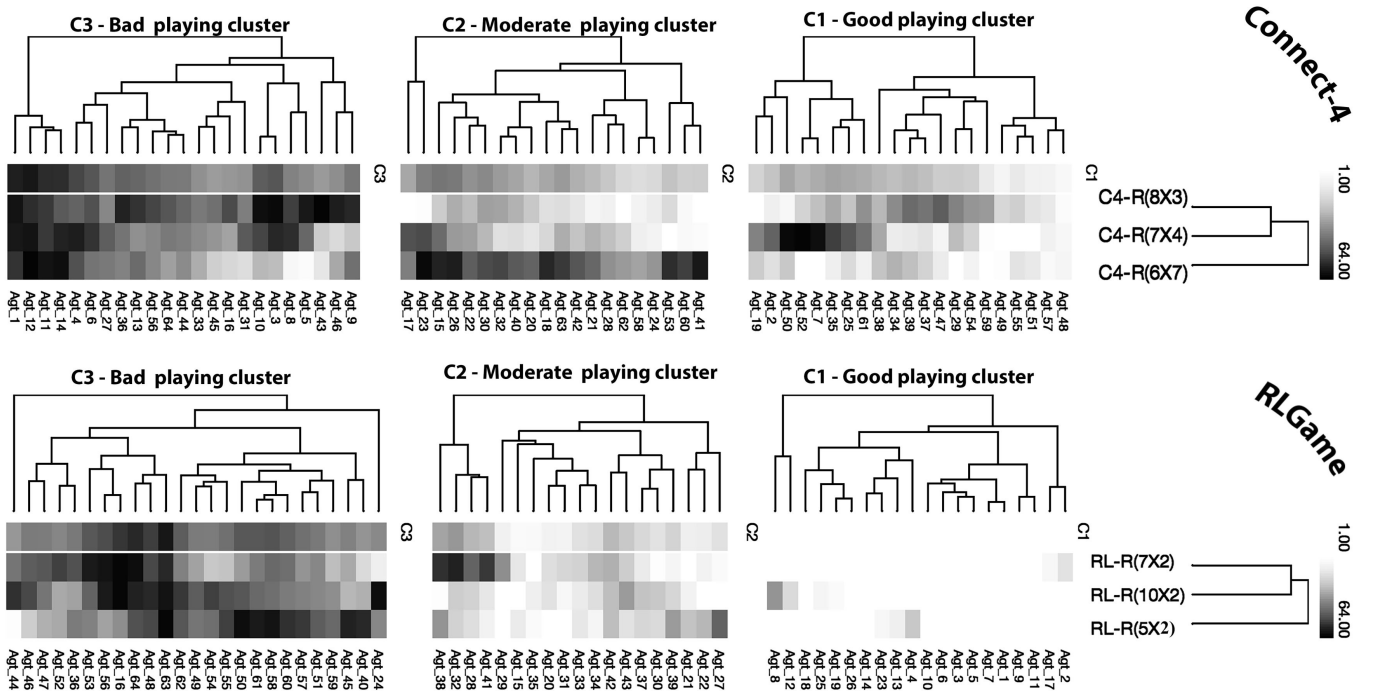


Fig. 4: K-means clustering for each tournament session of both games and a dendrogram representing the correlation of agents and tournament sessions

coefficient analysis. The color shades (heat-maps) of the *RLGame* tournament sessions are more evenly allocated compared to the heat-maps of the *Connect-4* tournament sessions. The single most uneven color allocation of the *Connect-4*'s heat-maps appears in the C2 cluster, where one mostly finds moderate playing agents and highlights that almost all agents of this cluster played better in the two lower levels the *Connect-4* state-space complexity variants.

The special allocations of the clusters C1, C2 and C3 (for both games), associated with the characteristic values (ϵ - γ - λ) and the performance of the cluster, highlight an estimation of the synthetic agents' playing behaviors of each cluster, as shown in Fig. 5. For example, the good playing agents of the two lowest state-space complexities configurations of *Connect-4* game (C1 clusters of $C4 - R(8 \times 3)$ and $C4 - R(7 \times 4)$), tend to have high ϵ -greedy ($\epsilon \approx 0.81 \Rightarrow$ knowledge exploiters), medium lambda ($\lambda \approx 0.76 \Rightarrow$ medium speed learner) and small gamma ($\gamma \approx 0.69 \Rightarrow$ risky (short term strategy selection)). The two graphs of Fig. 5 highlight

important differences in the agents' performance and playing behaviors based on the games and their complexity variations, such as:

- Good playing agents tend to be exploiters (high ϵ value) in *Connect-4*, in contrast to *RLGame*, where good playing agents tend to be explorers (low ϵ value), which is reasonable since *RLGame* is much more complex than *Connect-4* and the good playing agents respond to the environment, thus shifting towards becoming knowledge explorers.
- Bad playing agents are associated with low ϵ values in *Connect-4* and high ϵ values in *RLGame*, which is exactly the opposite to the good playing agents in both games.
- Moderate playing agents are scattered in both graphs (both games) and their playing behaviors is not clear.

It is clear that the performance of the agents depends on the game and on its complexity level. Due to the higher complexity level of the *RLGame*, the good playing agents need to be more sophisticated (more knowledge explorers, slow and smooth

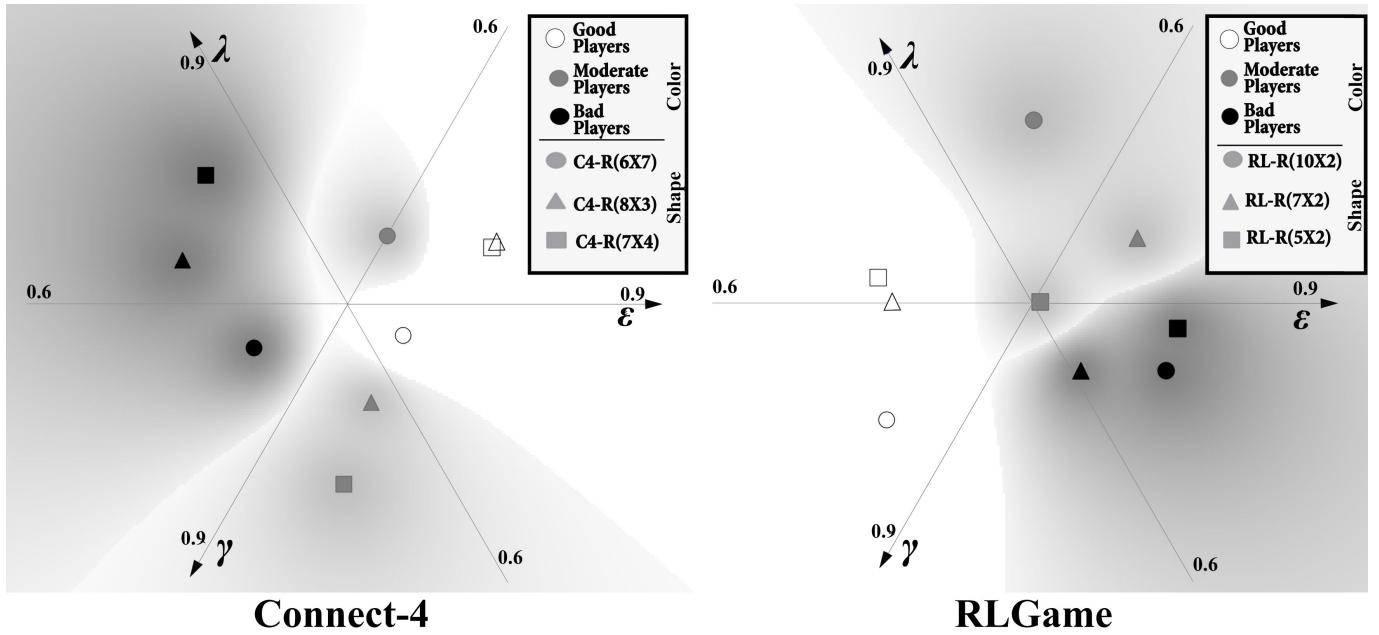


Fig. 5: Spatial allocation of the cluster (C1, C2 and C3), associated to the characteristic values $(\epsilon-\gamma-\lambda)$, of both games (*Connect-4* and *RLGame*).

learners and focusing on longer term strategies), which is not surprising if one aims at a more realistic simulation of playing behavior.

Each good playing agents' cluster changes its characteristic values $(\epsilon-\gamma-\lambda)$, only by slight shifting (as in Fig. 5), as the complexity of the game increases. By observing the C1 clusters of the two lower complexity tournament sessions of both games ($C4-R(8 \times 3)$ and $C4-R(7 \times 4)$ for *Connect-4*, $RL-R(5 \times 2)$ and $RL-R(7 \times 2)$ for *RLGame*), we highlight that they have similar playing characteristic $(\epsilon-\gamma-\lambda)$ values (the white triangles and squares are allocated to almost the same part, respectively, of each graph in Fig. 5). The C1 cluster (white circle in left graph of Fig. 5) of *Connect-4*'s highest complexity tournament session ($C4-R(6 \times 7)$) shows a slight shifting in comparison to the C1 clusters of the lower complexity tournament sessions ($C4-R(8 \times 3)$ and $C4-R(7 \times 4)$). We observe a shifting of about -12% for ϵ , -2% for λ and +8% for γ .

The C1 cluster (white circle in right graph of Fig. 5) of the *RLGame*'s highest complexity tournament session ($RL-R(10 \times 2)$), shows a similar slight shifting, in comparison to the C1 clusters of the lower complexity tournament sessions ($RL-R(5 \times 2)$ and $RL-R(7 \times 2)$). A shifting of about -2% for ϵ , -7% for λ and +3% for γ is observed.

Such shifting of the ϵ , γ and λ values indicates that as the complexity of the environment increases (environments of *Connect-4* and *RLGame*), good playing agents tend to become more sophisticated (*more knowledge explorers, more slow and smoother learners and focused in longer-term strategies*). The largest shifting appears in the C2 clusters (moderate playing agents) of both games' all complexity levels, which indicates that the moderate playing agents are hard to classify based on their characteristic values $(\epsilon-\gamma-\lambda)$. The C3 clusters of the *Connect-4* seem to be more affected by low ϵ values, while the C3 clusters of the *RLGame* seem to be more affected by

high ϵ values.

VI. CONCLUSION AND FUTURE DIRECTIONS

Based on the outcomes of the experimental tournament sessions, which spanned three different complexity levels for each game, *Connect-4* and *RLGame*, where we used the same agents' playing profile setups (same characteristic values $\epsilon-\gamma-\lambda$), we highlighted that an agents' playing profile does not readily lead to a comparable performance when the complexity of the environment (game) changes.

If an agent focuses on a specific performance level, in environments of varying complexity, its playing profile (characteristic values $\epsilon-\gamma-\lambda$) has to be re-adapted along specific directions based on the environment complexity. Our findings suggest that, as complexity increases (from *Connect-4* to *RLGame* and from a low-complexity *RLGame* variant to a higher complexity one), an agent stands a better chance of maintaining its performance profile (as indicated by its ranking), by decreasing its ϵ and λ values and increasing its γ one (though, of course, the exact change ratios may be too elusive to define). For this reason, we state that the re-adaptation of the agents' characteristic values depends on the game and its complexity but, broadly speaking, we note that as the complexity of the environment increases, good playing agents have to be more sophisticated: increasing their knowledge exploration bias (lower ϵ values), becoming slower and smoother learners (lower λ values) and focusing on longer term strategies (higher γ values). These findings are corroborated by the experimental sessions of both games, *Connect-4* and *RLGame* and it appears that an agent with a given $\epsilon-\gamma-\lambda$ profile cannot expect to maintain its performance profile if the environment changes with respect to the underlying complexity. Experimenting with a *Connect-4* variant of large $n \times m$ dimensions and maybe extending *Connect-4* to *Connect-k* could eventually shift the association with *RLGame* to larger

positive values, thus further strengthening the validity of our findings.

The experimental results of this paper highlight that synthetic agents are important elements of the simulation of realistic social environments and that just a handful of characteristic values (ϵ - γ - λ), namely, the exploitation-vs-exploration trade-off, learning backup and discount rates, and speed of learning, can synthesize a diverse population of agents with starkly different learning and playing behaviors.

An apparently promising and interesting investigation direction concerns the synthetic agents' application to other games (better known ones) and other complexity levels, such as checkers, chess etc., to investigate the learning progress of the synthetic agents' and the adjustability of their playing behaviors in diverse social environments. Additionally, as we highlighted that a synthetic agent's playing behavior may have to change in response to a change in the environment's complexity, this raises the generic question of how to modify one's characteristic values (ϵ - γ - λ) based on an assessment of the surrounding environment. Such an assessment could be based either on the complexity of the environment or on the level of the opponent but both approaches involve making an estimation based on limited information (for example, a limited number of games against some opponents should be able to help an agent to gauge whether it operates in a complex or simple environment or where its opponents might be situated in terms of their values in the ϵ - γ - λ parameters). Thus, adapting oneself based on incomplete and possibly partially accurate information is a huge challenge.

REFERENCES

- [1] Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: Proceedings of the 4th International Workshop on Agent-Oriented Software Engineering (AOSE), pp. 214–230 (2003)
- [2] Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press (2008)
- [3] Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley Publishing, 2nd ed. (2009)
- [4] Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley Longman Publishing (1999)
- [5] Kiourt, C., Kalles, D.: Synthetic learning agents in game-playing social environments. *Adaptive Behavior* 24:6, 411–427 (2016)
- [6] Marom, Y., Maistros, G., Hayes, G.: Experiments with a social learning model. *Adaptive Behavior* 9:3–4, 209–240 (2001)
- [7] Al-Khateeb, B., Kendall, G.: Introducing a round robin tournament into evolutionary individual and social learning checkers. In: Proceedings of the Developments in E-systems Engineering (DeSE), pp. 294–299 (2011)
- [8] Caballero, A., Botía, J., Gómez-Skarmeta, A.: Using cognitive agents in social simulations. *Engineering Applications of Artificial Intelligence* 24, 1098–1109 (2011)
- [9] Gilbert, N., Troitzsch, K. G.: Simulation for the Social Scientist. Open University Press (2005)
- [10] Kiourt, C., Kalles, D.: Using opponent models to train inexperienced synthetic agents in social environments. In: Proceedings of the 2016 IEEE Conference on Computational Intelligence and Games (CIG), pp. 1–4 (2016)
- [11] Kiourt, C., Kalles, D.: Learning in multi agent social environments with opponent models. In: Proceedings of the 13th European Conference on Multi-Agent Systems (EUMAS), pp. 137–144 (2016)
- [12] Allis, L. V.: Knowledge-based approach of connect four: The game is over, white to move wins. Master's thesis, Vrije Universiteit (1988)
- [13] Allis, L. V.: Searching for Solutions in Games and Artificial Intelligence. PhD thesis, Maastricht University (1994)
- [14] van den Herik, H., Uiterwijk, J. W., van Rijswijck, J.: Games solved: Now and in the future. *Artificial Intelligence* 134:1, 277–311 (2002)
- [15] Heule, M., Rothkrantz, L.: Solving games. *Science of Computer Programming*: 67:1, 105 – 124 (2007)
- [16] Edelkamp, S., Kissmann, P.: Symbolic classification of general two-player games. In: Proceedings of the 31st Annual German Conference on AI (KI), pp. 185–192 (2008)
- [17] Kalles, D., Kanellopoulos, P.: On verifying game designs and playing strategies using reinforcement learning. In: Proceedings of the 2001 ACM Symposium on Applied Computing (SAC), pp. 6–11 (2001)
- [18] Tesauro, G.: Practical issues in temporal difference learning. *Machine Learning* 8, 257–277 (1992)
- [19] Tesauro, G.: Temporal difference learning and td-gammon. *Communications of the ACM* 38, 58–68 (1995)
- [20] Sutton, R. S., Barto, A. G.: Introduction to Reinforcement Learning. MIT Press (1998)
- [21] March, J. G.: Exploration and exploitation in organizational learning. *Organization Science* 2, 71–87 (1991)
- [22] Tromp, J.: Solving connect-4 on medium board sizes. *ICGA Journal* 31:1, 110–112 (2008)
- [23] Tromp, J.: John's connect four playground. <https://tromp.github.io/c4/c4.html>. Accessed: 2017-10-27.
- [24] Sutton, R. S.: Learning to predict by the methods of temporal differences. *Machine Learning* 3, 9–44 (1988)
- [25] Wiering, M. A., Patist, J. P., Mannen, H.: Learning to play board games using temporal difference methods. Technical Report: UU-CS-2005-048 (2005)
- [26] Kiourt, C., Pavlidis, G., Kalles, D.: Reskill: Relative skill-level calculation system. In: Proceedings of the 9th Hellenic Conference on Artificial Intelligence (SETN), pp. 39:1–39:4 (2016)
- [27] Spearman, C.: The proof and measurement of association between two things. *American Journal Psychology* 15:1, 70–101 (1904)
- [28] Kendall, M. A new measure of rank correlation. *Biometrika* 30:1-2, 81–93 (1936)
- [29] Langville, N., Meyer, C.: Who's #1?: The Science of Rating and Ranking. Princeton University Press (2012)