A Generic Approach for Generating Interesting Interactive Pac-Man Opponents

Georgios N. Yannakakis
Centre for Intelligent Systems
and their Applications
The University of Edinburgh
AT, Crichton Street, EH8 9LE
g.yannakakis@sms.ed.ac.uk

John Hallam
Mærsk Mc-Kinney Møller Institute
for Production Technology
University of Southern Denmark
Campusvej 55, DK-5230, Odense M
john@mip.sdu.dk

Abstract- This paper follows on from our previous work focused on formulating an efficient generic measure of user's satisfaction ('interest') when playing predator/prey games. Viewing the game from the predators' (i.e. opponents') perspective, a robust on-line neuro-evolution learning mechanism has been presented capable of increasing — independently of the initial behavior and playing strategy — the well known Pac-Man game's interest as well as keeping that interest at high levels while the game is being played. This mechanism has also demonstrated high adaptability to changing *Pac-Man* playing strategies in a relatively simple playing stage. In the work presented here, we attempt to test the on-line learning mechanism over more complex stages and to explore the relation between the interest measure and the topology of the stage. Results show that the interest measure proposed is independent of the stage's complexity and topology, which demonstrates the approach's generality for this game.

1 Introduction

Over the last 25 years there have been major steps forward in computer games' graphics technology: from abstract 2D designs to complex realistic virtual worlds combined with advanced physics engines; from simple shape character representations to advanced human-like characters. Meanwhile, artificial intelligence (AI) techniques (e.g. machine learning) in computer games are nowadays still in their very early stages, since computer games continue to use simple rule-based finite and fuzzy state machines for nearly all their AI needs [1], [2]. These statements are supported by the fact that we still meet newly released games with the same 20-year old concept in brand new graphics engines.

From another viewpoint, the explosion of multi-player on-line gaming over the last years indicates the increasing human need for more intelligent opponents. This fact also reveals that interactive opponents can generate interesting games, or else increase the perceived satisfaction of the player. Moreover, machine learning techniques are able to produce characters with intelligent capabilities useful to any game's concept. Therefore, conceptually, the absolute necessity of artificial intelligence techniques and particularly machine learning and on-line interaction in game development stems from the human need for playing against intelligent opponents. These techniques will create the illusion of intelligence up to the level that is demanded by humans [3]. Unfortunately, instead of designing intelligent opponents to play against, game developers mainly concentrate and invest in the graphical presentation of the game. We believe that players' demand for more interesting games will pressure towards an 'AI revolution' in computer games in the years to come.

Predator/prey games is a very popular category of computer games and among its best representatives is the classical Pac-Man released by Namco (Japan) in 1980. Even though Pac-Man's basic concept — the player's (*PacMan's*) goal is to eat all the pellets appearing in a maze-shaped stage while avoiding being killed by four opponent characters named '*Ghosts*'— and graphics are very simple, the game still keeps players interested after so many years, and its basic ideas are still found in many newly released games. There are some examples, in the Pac-Man domain literature, of researchers attempting to teach a controller to drive *Pac-Man* in order to acquire as many pellets as possible and to avoid being eaten by *Ghosts* [4].

On the other hand, there are many researchers who use predator/prey domains in order to obtain efficient emergent teamwork of either homogeneous or heterogeneous groups of predators. For example, Luke and Spector [5], among others, have designed an environment similar to the Pac-Man game (the Serengeti world) in order to examine different breeding strategies and coordination mechanisms for the predators. Finally, there are examples of work in which both the predators' and the prey's strategies are co-evolved in continuous or grid-based environments [6], [7].

Recently, there have been attempts to mimic human behavior off-line, from samples of human playing, in a specific virtual environment. In [8], among others, human-like opponent behaviors are emerged through supervised learning techniques in *Quake*. Even though complex opponent behaviors emerge, there is no further analysis of whether these behaviors contribute to the satisfaction of the player (i.e. interest of game). In other words, researchers hypothesize — by looking at the vast number of multi-player on-line games played daily on the web — that by generating human-like opponents they enable the player to gain more satisfaction from the game. This hypothesis might be true up to a point; however, since there is no explicit notion of interest defined, there is no evidence that a specific opponent behavior generates more or less interesting games. Such a hypothesis is the core of Iida's work on board games. He proposed a general metric of entertainment for variants of chess games depending on average game length and possible moves [9].

Similar to [5], we view Pac-Man from the *Ghosts'* perspective and we attempt to off-line emerge effective teamwork hunting behaviors based on evolutionary computation techniques, applied to homogeneous neural controlled [10] *Ghosts*. However, playing a prey/predator computer game like Pac-Man against optimal hunters cannot be interesting because of the fact that you are consistently and effectively killed. To this end, we believe that the interest of any computer game is directly related to the interest generated by the opponents' behavior rather than to the graphics or even the player's behavior. Thus, when 'interesting game' is mentioned we mainly refer to interesting opponents to play against.

In [11], we introduced an efficient generic measure of interest of predator/prey games. We also presented a robust on-line (i.e. while the game is played) neuro-evolution learning approach capable of increasing — independently of the initial behavior and *PacMan*'s playing strategy — the game's interest as well as keeping that interest at high levels while the game is being played. This mechanism demonstrated high robustness and adaptability to changing types of *PacMan* player (i.e. playing strategies) in a relatively simple playing stage. In the work presented here, we attempt to test the on-line learning mechanism over more complex stages and furthermore to explore the relation between the interest measure and the topology of the stage. Results show that the interest measure introduced in [11] is independent of the stage's design which demonstrates the approach's generality for this game.

The arcade version of Pac-Man uses a handful of very simple rules and scripted sequences of actions combined with some random decision-making to make the *Ghosts'* behavior less predictable. The game's interest decreases at the point where *Ghosts* are too fast to beat [12]. In our Pac-Man version we require *Ghosts* to keep learning and constantly adapting to the player's strategy instead of being opponents with fixed strategies. In addition, we explore learning procedures that achieve good real-time performance (i.e. low computational effort while playing).

2 The Pac-Man World

The computer game test-bed studied is a modified version of the original Pac-Man computer game released by Namco. The player's (*PacMan's*) goal is to eat all the pellets appearing in a maze-shaped stage while avoiding being killed by the four *Ghosts*. The game is over when either all pellets in the stage are eaten by *PacMan* or *Ghosts* manage to kill *PacMan*. In that case, the game restarts from the same initial positions for all five characters. Compared to commercial versions of the game a number of features (e.g. power-pills) are omitted for simplicity; these features do not qualitatively alter the nature of 'interesting' in games of low interest.

As stressed before, the Pac-Man game is investigated from the viewpoint of *Ghosts* and more specifically how *Ghosts'* emergent adaptive behaviors can contribute to the interest of the game. Pac-Man — as a computer game domain for emerging adaptive behaviors — is a two-dimensional, multi-agent, grid-motion, predator/prey game. The game field (i.e. stage) consists of corridors and walls. Both the stage's dimensions and its maze structure are predefined. For the experiments presented in this paper we use a 19×29 grid maze-stage where corridors are 1 grid-cell wide. The snapshot of the Pac-Man game illustrated in Figure 1 constitutes one of the four different stages used for our experiments. Information about the selected stages' design and the criteria for their selection are presented in Section 2.1.

The characters visualized in the Pac-Man game (as illustrated in Figure 1) are a white circle that represents *PacMan* and 4 ghost-like characters representing the *Ghosts*. Additionally, there are black squares that represent the pellets and dark grey blocks of walls.

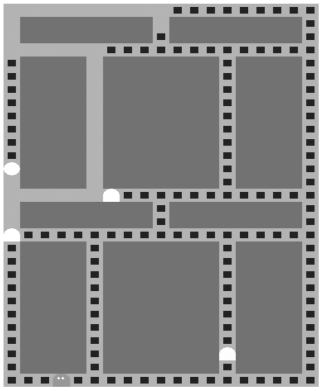

Figure 1: Snapshot of the Pac-Man game

PacMan moves at double the *Ghosts'* speed and since there are no dead ends, it is impossible for a single *Ghost* to complete the task of killing it. Since *PacMan* moves faster than a *Ghost*, the only effective way to kill *PacMan* is for a group of *Ghosts* to hunt cooperatively. It is worth mentioning that one of *Ghosts'* properties is permeability. In other words, two or more *Ghosts* can simultaneously occupy the same cell of the game grid.

The simulation procedure of the Pac-Man game is as follows. *PacMan* and *Ghosts* are placed in the game field (initial positions) so that there is a suitably large distance between them. Then, the following occur at each simulation step:

1. Both *PacMan* and *Ghosts* gather information from their environment.

2. *PacMan* and *Ghosts* take a movement decision every simulation step and every second simulation step respectively. (That is how *PacMan* achieves double the *Ghost's* speed.)

3. If the game is over (i.e. all pellets are eaten, *PacMan* is killed, or the simulation step is greater than a predetermined large number), then a new game starts from the same initial positions.

4. Statistical data such as number of pellets eaten, simulation steps to kill *PacMan* as well as the total *Ghosts'* visits to each cell of the game grid are recorded.

2.1 Stages

As previously mentioned, in this paper we attempt to test the on-line learning mechanism's ability to generate interesting games (as presented in [11]) over more complex stages and, furthermore, over stages of different topology.

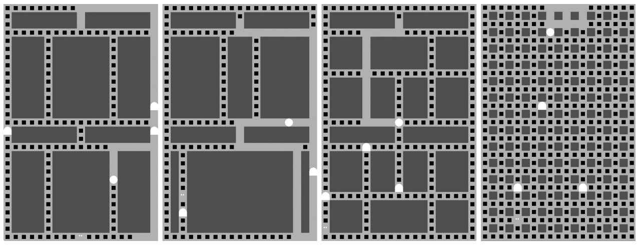

Figure 2: The 4 different stages of the game. Increasing complexity from left to right: Easy (A and B), Normal and Hard.

2.1.1 Complexity

In order to distinguish between stages of different complexity, we require an appropriate measure to quantify this feature of the stage. This measure is

$$C = 1/E\{L\} \tag{1}$$

where C is the complexity measure and $E\{L\}$ is the average corridor length of the stage.

According to (1), complexity is inversely proportional to the average corridor length of the stage. That is, the longer the average corridor length, the easier for the *Ghosts* to block *PacMan* and, therefore, the less complex the stage.

Figure 2 illustrates the four different stages used for the experiments presented here. Complexity measure values for the Easy A, Easy B, Normal and Hard stages are 0.16, 0.16, 0.22 and 0.98 respectively. Easy A stage is the test-bed used in [11]. Furthermore, given that a) blocks of walls should be included b) corridors should be 1 grid-square wide and c) dead ends should be absent, Hard stage is the most complex Pac-Man stage for the *Ghosts* to play.

2.1.2 Topology

Stages of the same complexity, measured by (1), can differ in topology (i.e. layout of blocks on the stage). Thus, in the case of Easy A and Easy B (see Figure 2), stages have the same complexity value but are topologically different.

The choice of these four stages is made so as to examine the on-line learning approach's ability to emerge interesting opponents in stages of different complexity or equally complex stages of different topology. Results presented in Section 6 show that the mechanism's efficiency is independent of both the stage complexity and stage topology and, furthermore, illustrate the approach's generality for the game.

2.2 PacMan

Both the difficulty and, to a lesser degree, the interest of the game are directly affected by the intelligence of the *PacMan* player. We chose three fixed *Ghost*-avoidance and pellet-eating strategies for the *PacMan* player, differing in complexity and effectiveness. Each strategy is based on decision making applying a cost or probability approximation to the player's 4 neighbor cells (i.e. up, down, left and right). Even though the initial positions are constant, the non-deterministic motion of *PacMan* provides lots of diversity within games.

- Cost-Based (CB) *PacMan*: The CB *PacMan* moves towards its neighbor cell of minimal cost. Cell costs are assigned as follows: $c_p = 0$, $c_e = 10$, $c_{ng} = 50$, $c_g = 100$, where c_p: cost of a cell with a pellet (pellet cell); c_e: cost of an empty cell; c_g: cost of a cell occupied by a *Ghost* (*Ghost* cell); c_{ng}: cost of a *Ghost's* 4 neighbor cells. Wall cells are not assigned any cost and are ignored by *PacMan*. In case of equal minimal neighbor cell costs (e.g. two neighbor cells with pellets), the CB *PacMan* makes a random decision with equal probabilities among these cells. In other words, the CB *PacMan* moves towards a cost minimization path that produces effective *Ghost*-avoidance and (to a lesser degree) pellet-eating behaviors but only in the local neighbor cell area.
- Rule-Based (RB) *PacMan*: The RB *PacMan* is a CB *PacMan* plus an additional rule for more effective and global pellet-eating behavior. This rule can be described as follows. If all *PacMan's* neighbor cells are empty ($c = 10$), then the probability of moving towards each one of the available directions (i.e. not towards wall cells) is inversely proportional to the distance (measured in grid-cells) to the closest pellet on that direction.
- Advanced (ADV) *PacMan*: The ADV *PacMan* checks in every non-occluded direction for *Ghosts*. If there is at least one *Ghost* in sight, then the probability of moving towards each one of the available directions is directly proportional to the distance to a *Ghost* in that direction. If there is no *Ghost* in sight, then the ADV *PacMan* behaves like a RB *PacMan*. The ADV moving strategy is expected to produce a more global *Ghost*-avoidance behavior built upon the RB *PacMan's* good pellet-eating strategy.

2.3 Neural Controlled Ghosts

A multi-layered fully connected feedforward neural controller, where the sigmoid function is employed at each neuron, manages the *Ghosts'* motion. Using their sensors, *Ghosts* inspect the environment from their own point of view and decide their next action. Each *Ghost*'s perceived input consists of the relative coordinates of *PacMan* and the closest *Ghost*. We deliberately exclude from consideration any global sensing, e.g. information about the dispersion of the *Ghosts* as a whole, because we are interested specifically in the minimal sensing scenario. The neural network's output is a four-dimensional vector with respective values from 0 to 1 that represents the *Ghost*'s four movement options (up, down, left and right respectively). Each *Ghost* moves towards the available — unobstructed by walls — direction represented by the highest output value. Available movements include the *Ghost*'s previous cell position.

2.4 Fixed Strategy Ghosts

Apart from the neural controlled *Ghosts*, three additional fixed non-evolving strategies have been tested for controlling the *Ghost's* motion. These strategies are used as baseline behaviors for comparison with any neural controller emerged behavior.

- Random (R): *Ghosts* that randomly decide their next available movement. Available movements have equal probabilities to be picked.
- Followers (F): *Ghosts* designed to follow *PacMan* constantly. Their strategy is based on moving so as to reduce the greatest of their relative distances ($\Delta_{x,P}, \Delta_{y,P}$) from *PacMan*.
- Near-Optimal (O): A *Ghost* strategy designed to produce attractive forces between *Ghosts* and *PacMan* as well as repulsive forces among the *Ghosts*. For each *Ghost* X and Y values are calculated as follows.

$$\begin{aligned} X &= \text{sign}[\Delta_{x,P}]h(\Delta_{x,P}, L_x, 0.25) \\ &- \text{sign}[\Delta_{x,C}]h(\Delta_{x,C} - 1, L_x, 10) \qquad (2) \\ Y &= \text{sign}[\Delta_{y,P}]h(\Delta_{y,P}, L_y, 0.25) \\ &- \text{sign}[\Delta_{y,C}]h(\Delta_{y,C} - 1, L_y, 10) \qquad (3) \end{aligned}$$

where sign$[z]$=$z/|z|$ and $h(z, z_m, p) = [1 - (|z|/z_m)]^p$. X and Y values represent the axis on which the near-optimal *Ghost* will move. Hence, the axis is picked from the maximum of $|X|$ and $|Y|$ whereas, the direction is decided from this value's sign. That is, if $|X| > |Y|$, then go right if sign$[X] > 0$ or go left if sign$[X] < 0$; if $|Y| > |X|$, then go up if sign$[Y] > 0$ or go down if sign$[Y] < 0$.

3 Interesting Behavior

In order to find an objective (as possible) measure of interest in the Pac-Man computer game we first need to define the criteria that make a game interesting. Then, second, we need to quantify and combine all these criteria in a mathematical formula — as introduced in [11]. The game should then be tested by human players to have this formulation of interest cross-validated against the interest the game produces in real conditions. This last part of our investigation constitutes a crucial phase of future work and it is discussed in Section 7.

To simplify this procedure we will ignore the graphics' and the sound effects' contributions to the interest of the game and we will concentrate on the opponents' behaviors. That is because, we believe, the computer-guided opponent character contributes the vast majority of features that make a computer game interesting.

By being as objective and generic as possible, we believe that the criteria that collectively define interest on the Pac-Man game are as follows.

1. *When the game is neither too hard nor too easy*. In other words, the game is interesting when *Ghosts* manage to kill *PacMan* sometimes but not always. In that sense, optimal behaviors are not interesting behaviors and *vice versa*.

2. *When there is diversity in opponents' behavior over the games*. That is, when *Ghosts* are able to find different ways of hunting and killing *PacMan* in each game so that their strategy is less predictable.

3. *When opponents' behavior is aggressive rather than static*. That is, *Ghosts* that move towards killing *PacMan* but meanwhile, move constantly all over the game field instead of simply following it. This behavior gives player the impression of an intelligent strategic *Ghosts'* plan which increases the game interest.

In order to estimate and quantify each of the aforementioned criteria of the game's interest, we let the examined group of *Ghosts* play the game N times and we record the simulation steps t_k taken to kill *PacMan* as well as the total number of *Ghosts'* visits v_{ik} at each cell i of the grid game field for each game k. Each game is played for a sufficiently large evaluation period of t_{max} simulation steps which corresponds to the minimum simulation period required by the RB *PacMan* (best pellet-eater) to clear the stage of pellets — in the experiments presented here t_{max} is 300 for the Easy stage, 320 for the Normal stage and 466 for the Hard stage.

Given these, the quantifications of the Pac-Man game's three interest criteria can be presented as follows.

1. According to the first criterion, an estimate of how interesting the behavior is, is given by T in (4).

$$T = [1 - (E\{t_k\}/\max\{t_k\})]^{p_1} \qquad (4)$$

where $E\{t_k\}$ is the average number of simulation steps taken to kill *PacMan* over the N games; $\max\{t_k\}$ is the maximum t_k over the N games; p_1 is a weighting parameter (for the experiments presented here $p_1 = 0.5$);

The T estimate of interest demonstrates that the greater the difference between the average number of steps taken to kill *PacMan* and the maximum number of steps taken to kill *PacMan*, the higher the interest of the game. Given (4), both poor-killing ('*too easy*') and near-optimal ('*too hard*') behaviors get low interest estimate values (i.e. $E\{t_k\} \simeq \max\{t_k\}$).

2. The interest estimate for the second criterion is given by S in (5).

$$S = (\sigma/\sigma_{max})^{p_2} \qquad (5)$$

where

$$\sigma_{max} = \frac{1}{2}\sqrt{\frac{N}{(N-1)}}(t_{max} - t_{min}) \qquad (6)$$

and σ is the standard deviation of t_k over the N games; σ_{max} is an estimate of the maximum value of σ; p_2 is a weighting parameter (for the experiments presented here $p_2 = 1$); t_{min} is the minimum number of simulation steps required for the fixed strategy Near-Optimal *Ghosts* to kill *PacMan* ($t_{min} \leq t_k$). In this paper, t_{min} is 33 simulation steps for the Easy stage; 35 for the Normal stage and 63 for the Hard stage.

The S estimate of interest demonstrates that the greater the standard deviation of the steps taken to kill *PacMan* over N games, the higher the interest of the behavior. Therefore, by using (5) we promote *Ghosts* that produce high diversity in the time taken to kill *PacMan*.

3. A good measure for quantifying the third interest criterion is through entropy of the *Ghosts'* cell visits in a game, which quantifies the completeness and uniformity with which the *Ghosts* cover the stage. Hence, for each game, the cell visits' entropy is calculated and normalized into $[0, 1]$ via (7).

$$H_n = \left[-\frac{1}{logV_n}\sum_i \frac{v_{in}}{V_n} log\left(\frac{v_{in}}{V_n}\right)\right]^{p_3} \qquad (7)$$

where V_n is the total number of visits of all visited cells (i.e. $V_n = \sum_i v_{in}$) and p_3 is a weighting parameter (for the experiments presented here $p_3 = 4$).

Given the normalized entropy values H_n for all N games, the interest estimate for the third criterion can be represented by their average value $E\{H_n\}$ over the N games. This implies that the higher the average entropy value, the more interesting the game becomes.

All three criteria are combined linearly (8)

$$I = \frac{\gamma T + \delta S + \epsilon E\{H_n\}}{\gamma + \delta + \epsilon} \qquad (8)$$

where I is the interest value of the Pac-Man game; γ, δ and ϵ are criterion weight parameters (for the experiments presented here $\gamma = 1, \delta = 2, \epsilon = 3$).

The measure of the Pac-Man game's interest introduced in (8) can be effectively applied to any predator/prey computer game because it is based on generic features of this category of games. These features include the time required to kill the prey as well as the predators' entropy throughout the game field. We therefore believe that (8) — or a similar measure of the same concepts — constitutes a generic interest approximation of predator/prey computer games (see also [13] for a successful application on a dissimilar prey/predator game). Moreover, given the two first interest criteria previously defined, the approach's generality is expanded to all computer games. Indeed, no player likes any computer game that is too hard or too easy to play and, furthermore, any player would like diversity throughout the play of any game. The third interest criterion is applicable to games where spatial diversity is important which, apart from prey/predator games, may also include action, strategy and team sports games according to the computer game genre classification of Laird and van Lent [14].

4 Off-line learning

We use an off-line evolutionary learning approach in order to produce some 'good' (i.e. in terms of performance) initial behaviors. An additional aim of this algorithm is to emerge dissimilar behaviors of high fitness — varying from blocking to aggressive (see Section 6) — offering diverse seeds for the on-line learning mechanism in its attempt to generate emergent *Ghost* behaviors that make the game interesting.

The neural networks that determine the behavior of the *Ghosts* are themselves evolved with the evolving process limited to the connection weights of the neural network. Each *Ghost* has a genome that encodes the connection weights of its neural network. A population of 80 neural networks (*Ghosts*) is initialized randomly with initial uniformly distributed random connection weights that lie within [-5, 5]. Then, at each generation:

- Every *Ghost* in the population is cloned 4 times. These 4 clones are placed in the Pac-Man game field and play ten games of t_{max} simulation steps each. The outcome of these games is to ascertain the time taken to kill *PacMan* t_k for each game.
- Each *Ghost* is evaluated via (9) for each game and its fitness value is given by $E\{f\}$ over the N_t games.

  $$f = [1 - (t_k/t_{max})]^{\frac{1}{4}} \qquad (9)$$

  By the use of (9) we promote *Ghost* behaviors capable of achieving high performance on killing *PacMan*.
- A pure elitism selection method is used where only the 10% fittest solutions are able to breed and, therefore, determine the members of the intermediate population. Each parent clones an equal number of offspring in order to replace the non-picked solutions from elitism.
- Mutation occurs in each gene (connection weight) of each offspring's genome with a small probability p_m (e.g. 0.02). A uniform random distribution is used again to define the mutated value of the connection weight.

The algorithm is terminated when a predetermined number of generations g is completed (e.g. $g = 1000$) and the fittest *Ghost's* connection weights are saved.

5 On-line learning (OLL)

This learning approach is based on the idea of *Ghosts* that learn while they are playing against *PacMan*. In other words, *Ghosts* that are reactive to any player's behavior and learn from its strategy instead of being the predictable and, therefore, uninteresting characters that exist in all versions of this game today. Furthermore, this approach's additional objective is to keep the game's interest at high levels as long as it is being played. This mechanism is first introduced in [15] for an abstract prey-predator game called "Dead-End" and in [11] for the Pac-Man game. In this paper, we give a short description of OLL.

Beginning from any initial group of homogeneous off-line trained (OLT) *Ghosts*, OLL attempts to transform them into a group of heterogeneous *Ghosts* that are interesting to play against as follows. An OLT *Ghost* is cloned 4 times and its clones are placed in the Pac-Man game field to play against a selected *PacMan* type of player in a selected stage. Then, at each generation:

Step 1: Each *Ghost* is evaluated every t simulation steps via (10), while the game is played — $t = 50$ simulations steps in this paper.

$$f' = \sum_{i=1}^{t/2} \left\{ d_{P,2i} - d_{P,(2i-1)} \right\} \tag{10}$$

where $d_{P,i}$ is the distance between the *Ghost* and *PacMan* at the i simulation step. This fitness function promotes *Ghosts* that move towards *PacMan* within an evaluation period of t simulation steps.

Step 2: A pure elitism selection method is used where only the fittest solution is able to breed. The fittest parent clones an offspring with a probability p_c that is inversely proportional to the normalized cell visits' entropy (i.e. $p_c = 1 - H_n$) given by (7). In other words, the higher the cell visits' entropy of the *Ghosts*, the lower the probability of breeding new solutions. If there is no cloning, then go back to Step 1, else continue to Step 3.

Step 3: Mutation occurs in each gene (connection weight) of each offspring's genome with a small probability p_m (e.g. 0.02). A gaussian random distribution is used to define the mutated value of the connection weight. The mutated value is obtained from (11).

$$w_m = \mathcal{N}(w, 1 - H_n) \tag{11}$$

where w_m is the mutated connection weight value and w is the connection weight value to be mutated. The gaussian mutation, presented in (11), suggests that the higher the normalized entropy of a group of *Ghosts*, the smaller the variance of the gaussian distribution and therefore, the less disruptive the mutation process as well as the finer the precision of the GA.

Step 4: The cloned offspring is evaluated briefly via (10) in off-line mode, that is, by replacing the worst-fit member of the population and playing an off-line (i.e. no visualization of the actions) short game of t simulation steps. The fitness values of the mutated offspring and the worst-fit *Ghost* are compared and the better one is kept for the next generation. This pre-evaluation procedure for the mutated offspring attempts to minimize the probability of group behavior disruption by low-performance mutants. The fact that each mutant's behavior is not tested in a single-agent environment but within a group of heterogeneous *Ghosts* helps more towards this direction. If the worst-fit *Ghost* is replaced, then the mutated offspring takes its position in the game field as well.

The algorithm is terminated when a predetermined number of games has been played or a game of high interest (e.g. $I \geq 0.7$) is found.

We mainly use short simulation periods ($t = 50$) in order to evaluate *Ghosts* in OLL aiming to the acceleration of the on-line evolutionary process. The same period is used for the evaluation of mutated offspring; this is based on two primary objectives: 1) to apply a fair comparison between the mutated offspring and the least-fit *Ghost* (i.e. same evaluation period) and 2) to avoid undesired high computational effort in on-line mode (i.e. while playing). However, the evaluation function (10) constitutes an approximation of the examined *Ghost*'s overall performance for large simulation periods. Keeping the right balance between computational effort and performance approximation is one of the key features of this approach. In the experiments presented here, we use minimal evaluation periods capable of achieving good estimation of the *Ghosts'* performance.

6 Results

Off-line trained (OLT) emergent solutions are the OLL mechanisms' initial points in the search for more interesting games. OLT obtained behaviors are classified into the following categories:

- Blocking (B): These are OLT *Ghosts* that tend to wait for *PacMan* to enter into a specific area that is easy for them to block and kill. Their average normalized cell visit's entropy value $E\{H_n\}$ lies between 0.55 and 0.65
- Aggressive (A): These are OLT *Ghosts* that tend to follow *PacMan* all over the stage in order to kill it ($E\{H_n\} \geq 0.65$).
- Hybrid (H): These are OLT *Ghosts* that tend to behave as a Blocking-Aggressive hybrid which proves to be ineffective at killing *PacMan* ($E\{H_n\} < 0.55$).

6.1 OLL experiment

In order to portray the OLL impact on player's entertainment, the following experiment is conducted. a) Pick nine different emerged *Ghosts'* behaviors produced from off-line learning experiments — Blocking (B), Aggressive (A) and Hybrid (H) behaviors emerged by playing against each of 3 *PacMan* types — for each one of the three stages; b) starting from each OLT behavior, apply the OLL mechanism by playing against the same type of *PacMan* player and in the same stage the *Ghosts* have been trained in off-line. Initial behaviors for the Easy B stage are OLT behaviors emerged from the Easy A stage. This experiment intends to demonstrate the effect of the topology of a stage in the interest of the game; c) calculate the interest of the game every 100 games during each OLL attempt.

Interest is calculated by letting the *Ghosts* play 100 non-evolution games in the same stage against the *PacMan* type they were playing against during OLL. In order to minimize the non-deterministic effect of the *PacMan*'s strategy on the *Ghost*'s performance and interest values as well as to draw a clear picture of these averages' distribution, we apply the following bootstrapping procedure. Using a uniform random distribution we pick 10 different 50-tuples out of the 100 above-mentioned games. These 10 samples of data, of 50 games each, are used to determine the games' average as well as confidence interval values of interest. The outcome of the OLL experiment is presented Figure 3 and Figure 4.

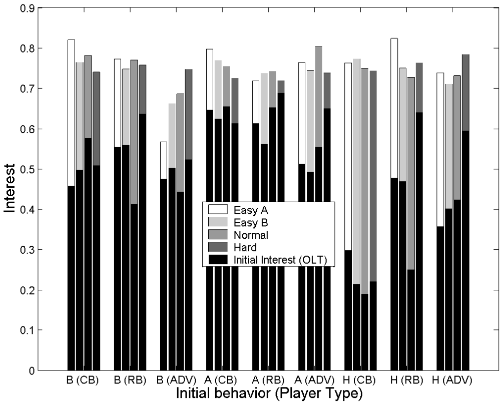

Figure 3: On-line learning effect on the interest of the game. Best interest values achieved from on-line learning on *Ghosts* trained off-line (B, A, H). Experiment Parameters: $t = 50$ simulation steps, $p_m = 0.02$, 5-hidden neurons controller.

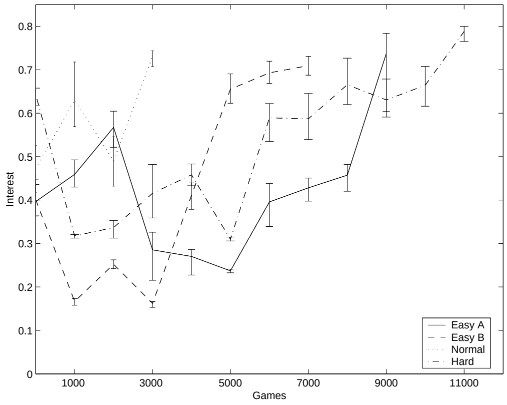

Figure 4: On-line learning effect on interest of ADV Hybrid initial behavior in all four stages. For reasons of computational effort, the OLL procedure is terminated when a game of high interest ($I \geq 0.7$) is found.

			Play Against		
		Stage	CB	RB	ADV
Fixed Behaviors	R	Easy A	0.5862	0.6054	0.5201
		Easy B	0.5831	0.5607	0.4604
		Normal	0.5468	0.5865	0.5231
		Hard	0.3907	0.3906	0.3884
	F	Easy A	0.7846	0.7756	0.7759
		Easy B	0.7072	0.6958	0.6822
		Normal	0.7848	0.8016	0.7727
		Hard	0.7727	0.7548	0.7627
	O	Easy A	0.6836	0.7198	0.6783
		Easy B	0.6491	0.6725	0.6337
		Normal	0.7297	0.7490	0.6855
		Hard	0.6922	0.7113	0.4927

Table 1: Fixed strategy *Ghosts'* (R, F, O) interest values. Values are obtained by averaging 10 samples of 50 games each.

Since there are 3 types of players, 3 initial OLT behaviors and 4 stages, the total number of different OLL experiments is 36. These experiments illustrate the overall picture of the mechanism's effectiveness over the complexity and the topology of the stage as well as the *PacMan* type and the initial behavior (see Figure 3). Due to space considerations we present only 4 (see Figure 4) out of the 36 experiments in detail here, where the evolution of interest over the OLL games (starting from the hybrid behavior emerged by playing against the ADV *PacMan* player) on each stage is illustrated.

As seen from Figure 4, the OLL mechanism manages to find ways of increasing the interest of the game regardless the stage complexity or topology. It is clear that the OLL approach constitutes a robust mechanism that, starting from suboptimal OLT *Ghosts*, manages to emerge interesting games (i.e. interesting *Ghosts*) in all 36 cases. It is worth mentioning that in 15 out of 36 different OLL attempts the best interest value is greater than the respective Follower's value (see Table 1). Furthermore, in nearly all cases, the interest measure is kept at the same level independently of stage complexity or — in the case of Easy A and B stages — stage topology. Given the confidence intervals (± 0.05 maximum, ± 0.03 on average) of the best interest values, it is revealed that the emergent interest is not significantly different from stage to stage.

However, a number in the scale of 10^3 constitutes an unrealistic number of games for a human player to play. On that basis, it is very unlikely for a human to play so many games in order to notice the game's interest increasing. The reason for the OLL process being that slow is a matter of keeping the right balance between the process' speed and its 'smoothness' (by 'smoothness' we define the interest's magnitude of change over the games). A solution to this problem is to consider the initial long period of disruption as an off-line learning procedure and start playing as soon as the game's interest is increased.

7 Conclusion & Discussion

Predator strategies in prey/predator computer games are still nowadays based on simple rules which make the game rather predictable and, therefore, uninteresting (by the time the player gains more experience and playing skills). A computer game becomes interesting primarily when there is an on-line interaction between the player and its opponents who demonstrate interesting behaviors.

Given some objective criteria for defining interest in predator/prey games, in [11] we introduced a generic method for measuring interest in such games. We saw that by using the proposed on-line learning mechanism, maximization of the individual simple distance measure (see (10)) coincides with maximization of the game's interest. Apart from being fairly robust, the proposed mechanism demonstrates high adaptability to changing types of player (i.e. playing strategies).

Moreover, in this paper, we showed that interesting games can be emerged independently of initial opponent behavior, playing strategy, stage complexity and stage topology. Independence from these four factors portrays the mechanism's generality and provides more evidence that such a mechanism will be able to produce interesting interactive opponents (i.e. games) against even the most complex human playing strategy.

As already mentioned, an important future step of this research is to discover whether the interest value computed by (8) for a game correlates with human judgement of interest. Preliminary results from a survey based on on-line questionnaires with a statistically significant sample of human subjects show that human players' notions of interest of the Pac-Man game correlate highly with the proposed measure of interest. More comprehensively, subjects are asked to determine the most interesting of several pairs of games, while their opponents are selected so as to produce significantly different interest values. Subsequently, a statistical analysis is carried out which is based on the correlation between observed human judgement of interest of these games and their respective interest values. Obtained results reveal that the interest metric (8) is consistent with the judgement of human players and will be part of a technical paper to be published shortly.

Bibliography

[1] Steven Woodcock. Game AI: The State of the Industry 2000-2001: It's not Just Art, It's Engineering. Game Developer magazine, August 2001.

[2] S. Cass. Mind games. *IEEE Spectrum*, pages 40–44, 2002.

[3] Alex J. Champandard. *AI Game Development*. New Riders Publishing, 2004.

[4] J. Koza. *Genetic Programming: On the programming of computers by means of natural selection*. MIT Press, 1992.

[5] Sean Luke and Lee Spector. Evolving teamwork and coordination with genetic programming. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 150–156, Stanford University, CA, USA, 1996. MIT Press.

[6] Thomas Haynes and Sandip Sen. Evolving behavioral strategies in predators and prey. In Sandip Sen, editor, *IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems*, pages 32–37, Montreal, Quebec, Canada, 1995. Morgan Kaufmann.

[7] G. Miller and D. Cliff. Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In D. Cliff, P. Husbands, J.-A. Meyer, and S. W. Wilson, editors, *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB-94)*, pages 411–420, Cambridge, MA, 1994. MIT Press.

[8] Christian Thurau, Christian Bauckhage, and Gerhard Sagerer. Learning human-like Movement Behavior for Computer Games. In S. Schaal, A. Ijspeert, A. Billard, Sethu Vijayakumar, J. Hallam, and J.-A. Meyer, editors, *From Animals to Animats 8: Proceedings of the 8^{th} International Conference on Simulation of Adaptive Behavior*, pages 315–323, 2004. The MIT Press.

[9] Hiroyuki Iida, N. Takeshita, and J. Yoshimura. A metric for entertainment of boardgames: its implication for evolution of chess variants. In R. Nakatsu and J. Hoshino, editors, *IWEC2002 Proceedings*, pages 65–72. Kluwer, 2003.

[10] X. Yao. Evolving artificial neural networks. In *Proceedings of the IEEE*, volume 87, pages 1423–1447, 1999.

[11] Georgios N. Yannakakis and John Hallam. Evolving Opponents for Interesting Interactive Computer Games. In S. Schaal, A. Ijspeert, A. Billard, Sethu Vijayakumar, J. Hallam, and J.-A. Meyer, editors, *From Animals to Animats 8: Proceedings of the 8^{th} International Conference on Simulation of Adaptive Behavior*, pages 499–508, 2004. The MIT Press.

[12] Steve Rabin. *AI Game Programming Wisdom*. Charles River Media, Inc, 2002.

[13] Georgios N. Yannakakis and John Hallam. Interactive Opponents Generate Interesting Games. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, pages 240–247, 2004.

[14] John E. Laird and Michael van Lent. Human-level AI's killer application: Interactive computer games. In *National Conference on Artificial Intelligence (AAAI)*, 2000.

[15] Georgios N. Yannakakis, John Levine, and John Hallam. An Evolutionary Approach for Interactive Computer Games. In *Proceedings of the Congress on Evolutionary Computation (CEC-04)*, pages 986–993, June 2004.