Predicting Dominance Rankings for Score-based Games

Spyridon Samothrakis Diego Perez Philipp Rohlfshagen Simon M. Lucas

***Abstract*—Game competitions may involve different player roles and be score-based rather than win/loss based. This raises the issue of how best to draw opponents for matches in ongoing competitions, and how best to rank the players in each role. An example is the Ms Pac-Man vs Ghosts Competition which requires competitors to develop software controllers to take charge of the game's protagonists: participants may develop software controllers for either or both Ms Pac-Man and the team of four ghosts. In this paper we compare two ranking schemes for win-loss games, Bayes Elo and Glicko. We convert the game into one of win/loss ("dominance") by matching controllers of identical type against the same opponent in a series of pair-wise comparisons. This implicitly creates a "solution concept" as to what a constitutes a good player. We analyse how many games are needed under two popular ranking algorithms, Glicko and Bayes Elo, before one can infer the strength of the players, according to our proposed solution concept, without performing an exhaustive evaluation. We show that Glicko should be the method of choice for online score-based game competitions.**

I. Introduction

Games provide excellent test-beds in which to develop, test and compare novel techniques in computational intelligence (CI). In the past, board games have served this purpose, with famous examples including Chess, Checkers and Othello. More recently, video games have attracted the attention of researchers both as a test-bed for and an application of CI methods. Video games typically offer a more visceral challenge compared to the cerebral appeal of board games, but are equally interesting from a machine intelligence point of view: arcade games such as Ms Pac-Man have been developed to be engaging, and the variety of human and computer-based opponents provides a robust way to test the efficacy of new algorithms. The popularity of many games also makes them a useful tool in education to convey complex subject matters in an interactive and entertaining fashion.

A vital aid is game competitions, as they allow researchers to test and evaluate their algorithms easily and under the exact same conditions. Competitions are also useful to attract a fresh cohort of students, researchers and game enthusiasts to the area. Within the IEEE Computational Intelligence Society, the Games Technical Committee has nurtured many interesting competitions that have leveraged existing video games or reasonably faithful implementations of them. In a similar fashion, the Game Intelligence Group at the University of Essex has organised numerous game competitions in recent years, attracting an ever-increasing number of participants from academia as well as the private sector. The most popular of these competitions is the Ms Pac-Man vs Ghosts Competition [29] which has been running since 2011 with two iterations each year. The competitions requires competitors to create software controllers for either (or both) Ms Pac-Man and the ghosts that interface directly with the game.

Competitors may submit and re-submit entries at any time prior to the deadline. Previously, all submissions would compete with one another in a round-robin tournament to establish the best controllers: Ms Pac-Man controllers attempt to maximise the score of the game while the ghosts strive to minimise the score; entries were ranked according to their total average score. As the competition grew in size the round-robin format became increasingly time-consuming and eventually infeasible. The score-based evaluation exhibited some unwanted artefacts that occasionally would favour unwanted behaviours. For instance, it was possible for a controller to rank high by playing extremely well against a (small) subset of opponents while performing only averagely against the rest.

Luckily, a wealth of alternative rating and ranking schemes exist: the more sophisticated of which compute a skill rating for each player that is updated whenever a new game outcome (win, loss or draw) has been established. The rating of a player changes in proportion to the skill of its opponent. Unfortunately these rating systems are not directly applicable to the Ms Pac-Man vs Ghosts Competition as Ms Pac-Man is a score-based game. Furthermore, the controllers involved in each comparison are heterogeneous: while Ms Pac-Man competes against the ghosts in each game played, it is the comparison of controllers of the same types that establishes the rankings. Furthermore, the game and objective is substantially different for Ms Pac-Man compared to the ghosts. It is nevertheless possible to take these factors into consideration and to use a rating scheme such as Glicko [15] for this competition. To the best of our knowledge, this is the first application of win/loss schemes to an asymmetric score-based game. We compare Bayes Elo and Glicko, two popular ranking algorithms in the context of online game competitions.

The remainder of this paper is structured as follows: Section II introduces the game of Ms Pac-Man and the Ms Pac-Man vs Ghosts Competition, which forms the bulk of our data. This is followed in Section III by a brief review of ranking schemes and other gaming competitions. These two sections form the background information of this paper. In Section IV we introduce the methodological choices made in this paper. An experimental comparison is subsequently presented in Section V, followed by conclusions and a discussion of prospects for future work in Section VI.

II. Ms Pac-Man vs Ghosts Competition

The Game Intelligence Group at the University of Essex has been running game competitions since 2007, including the Ms

Spyridon Samothrakis, Diego Perez and Simon M. Lucas (School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom; email: {ssamot, dperez,sml}@essex.ac.uk); Philipp Rohlfshagen (Schneider Electric, Adelaide 5000, Australia; email: philipp.rohlfshagen@schneider-electric.com)

Pac-Man Screen Capture Competition [23], the Ms Pac-Man vs Ghosts Competition [29] and, most recently, the Physical Travelling Salesman Competition [26]. These competitions take place once or twice a year and coincide with major IEEE CIS conferences, namely CEC, CIG and WCCI. This paper focusses on the Ms Pac-Man vs Ghosts competition which is based on the classical arcade game Ms Pac-Man.

A. Ms Pac-Man

One of the earliest commercially successful games is Pac-Man, an arcade game developed by Toru Iwatani and released in 1980 by Namco. The best known variant of the game is Ms Pac-Man, released in 1982, which introduced a female character, new maze designs and several gameplay changes. The player takes control of Ms Pac-Man using a 4-way joystick (compass points plus neutral) and needs to navigate her across a series of mazes. Screenshots of our implementation of the game are shown in Figure 1. The four mazes are played in fixed order: whenever a maze is cleared (i.e., all pills have been eaten), the game moves on to the next maze until the game is over. Each maze contains a different layout, with pills and power pills placed at specific locations. Each pill eaten scores 10 points, each power pill is worth 50 points. Ms Pac-Man starts the game with three lives; an additional life is awarded at 10,000 points. At the start of each level, the ghosts start in the lair in the middle of the maze and, after some idle time, enter the maze in their pursuit of Ms Pac-Man. Whenever a ghost has contact with Ms Pac-Man, she loses a life and the game terminates when no lives remain. However, there are also four power pills in each maze which, when eaten, turn the ghosts edible for a short period of time, allowing Ms Pac-Man to chase and eat them instead. The first ghost eaten awards 200 points and this reward doubles with each ghost eaten in succession.

From an AI perspective, the most significant difference between Ms Pac-Man and the original game is the design of the ghost team, which is no longer deterministic. The popularity of Pac-Man led to numerous strategy guides that taught gamers specific patterns of game-play that maximise the game's score (e.g., [8]). As pointed out by Mott, these patterns are not only important in mastering Pac-Man but their mere existence is one of the game's weaknesses: "Lacking any particularly inspiring AI, Pac-Man's pursuers race around the maze, following predictable paths, meaning players can effectively beat the game through memory and timing rather than inventive reactions." [25]. Ms Pac-Man not only introduced variety by having multiple mazes but the element of randomness in the ghosts' behaviour meant that gamers would need to improvise at times rather than following strict and predictable patterns. These changes make the game more engaging, and while the maximum possible score for Pac-Man was achieved in 1999, new high-scores for Ms Pac-Man are still being set: a new record of 921,360 points was set by Abdner Ashman in 2006 [6]f.

B. Ms Pac-Man vs Ghosts Competition

The Ms Pac-Man vs Ghosts Competition [29] has been open for submissions for four iterations, during which numerous

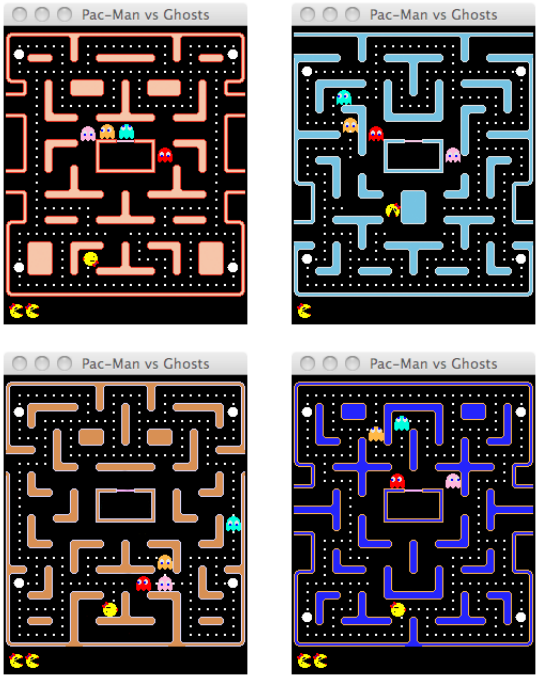


Fig. 1: Screen captures of the different levels (left-to-right, top-to-bottom, levels 1-4) of Ms Pac-Man: Ms Pac-Man (yellow) consumes pills for points (small white dots) while the ghosts (red: Blinky, pink: Pinky, green: Inky and brown: Sue) attempt to eat her; the large white dots in the corners are the power pills.

changes have been made to the scoring system in an attempt to identify the best software controllers. The competition requires competitors to submit software controllers for Ms Pac-Man and/or the ghosts. These controllers interface directly with the purpose-built game engine which is open source and written entirely in Java. Although care has been taken to implement the game as faithfully as possible, it differs from the original in some aspects. For instance, there are no bonus fruits, the speed of all characters is constant, unless the ghosts are edible, and the tunnels are shorter than in the original game. There are no restrictions regarding the techniques used but controllers have only 40 milli-seconds per game step to compute a move. Ghosts are not allowed to reverse under their own control, but at any time step a random reversal event may occur with a small probability: this forces each ghost to immediately reverse. Such events may allow Ms Pac-Man a lucky escape from even the trickiest of situations.

Prior to the deadline, competitors may submit and re-submit their entries as many times as they like. Live rankings and replays of the games played are displayed on the competition's website (www.pacman-vs-ghosts.net), allowing contestants to improve their submissions over time. Starter packages are provided to allow contestants to enter the rankings immediately following registrations. In the first three iterations of the competition, all entries submitted competed with one another in a full round-robin tournament to establish the best controllers: Ms Pac-Man controllers attempt to maximise the

score of the game while the ghosts strive to minimise the score.

C. Limiting the Duration of Games

Unlike most board games Ms Pac-Man does not naturally converge towards a terminal state and games may, at least in principle, last forever. It is thus necessary to limit the length of each game for the purpose of the competition. However, it is important to preserve the essence of the game as much as possible to prevent controllers exploiting competition-specific rules rather than performing generally well in the game. For the first three iterations of the competition the following rules were imposed: each game lasted a maximum of 16 levels and each level was limited to 3000 time steps. These rules ensured ghosts could not spoil the game by unanimously protecting the final pills of the level. Whenever the time limit of a level was reached, the game moved on to the next level and Ms Pac-Man was awarded half the points associated with the remaining pills (this was to encourage more aggressive behaviour from the ghosts). These rules limited a single game to a maximum of $(((3000 \times 40) \times 16)/1000)/60 = 32$ minutes.

Unfortunately, these rules had some unwanted consequences: Ms Pac-Man controllers would, for instance, stay away from remaining pills since the risk of capture would not be justified given the automatic award of points when the level's time limit was reached. Following the many suggestions from the competition's online forum, the rules were changed for the current iteration of the competition:

- Total duration of a game is limited to 24000 time steps.
- Maximum time per level is 4000 time steps.
- No points are awarded for remaining pills/power pills.
- The edible / lair time reductions reset at level 6.
- Each life remaining at the end of the game is awarded 800 points.

These changes imply a maximum level duration of 16 minutes and encourage Ms Pac-Man controllers to clear mazes and ghosts to pursue and eat Ms Pac-Man more actively. It will always be possible for controllers to exploit the rules of the competition (rather than trying to achieve the best possible performance in the game), but recent activity in the currently ongoing competition seems to imply that the new set of rules are fulfilling their intended purpose.

D. Randomness of Ms Pac-Man

Unlike many classical board games (e.g., Chess, Go), Ms Pac-Man has several sources of randomness that affect the outcome of each game. This uncertainty is caused by the following 3 attributes and events:

- Global reversal events: ghosts are reversed with probability 0.015 at every time step. Controllers have no control over this.
- Real-time element: the game advances every 40 milliseconds but time is measured as system time meaning that background processes of the OS (or the Java garbage collector) may affect the actual CPU time a controller has available.

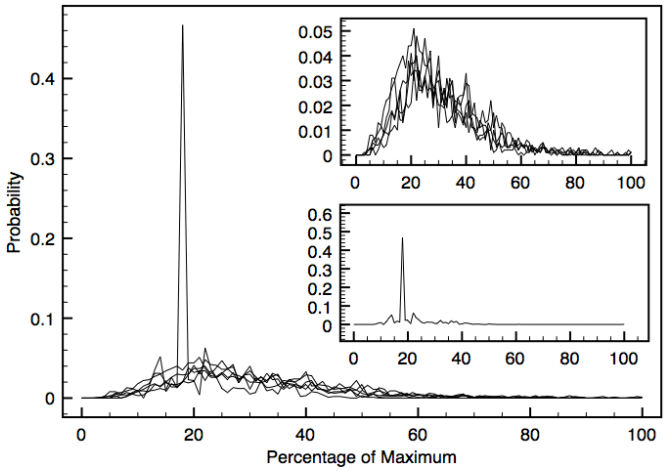


Fig. 2: Distribution of scores for the default controllers with variable degree of randomness. The bottom inlay shows the frequencies for the deterministic controllers; the top inlay shows the frequencies for the controllers with $q \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.

- The controllers themselves may contain elements of randomness to prevent their behaviour from being predictable.

Although only the first of these points explicitly creates stochasticity, all three need to be considered in the evaluation of controllers. In the first two iterations of the competition, each pairing in the round-robin tournament would play a total of 10 games (5 games being used during the submission phase to provide feedback more quickly) and the average was used to establish a representative score. In the third iteration, only 5 games were used due to the increased number of participants. It remains debatable whether elements of change (luck) should play a role in these competitions (as they do in human competitions). Nevertheless, it is vital to have a sound understanding of how the stochasticity of the game affects the outcome and hence the ratings.

Figure 2 shows the distribution of scores, as percentage of the maximum score achieved, for the default controllers included with the software with variable degrees of randomness: each controllers computes its action(s) as normal but, with some probability q, a random action is returned instead. The probabilities tested are $q \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. In the case of Ms Pac-Man, random decisions are made only at junctions and in the case of the ghosts, the probability is applied to each ghost individually and independently. From the data it is evident that if both controllers are entirely deterministic, the scores are heavily clustered around the same value (almost 50% of all games played). Once a slight degree of randomness is evident in both controllers however, the variance of the scores is much wider although some clustering remains evident. Interestingly, there appears little difference in the distribution of scores given different degrees of randomness in each controller.

III. Rankings and Competitions

A. Ranking Schemes

Humans have always been interested in ways to measure and compare their performances to establish who is best at a particular activity. The first Olympic Games, for instance, were carried out in 776 BC. Nowadays competitions are carried out for almost any discipline one can compete in, including sports, games or mental challenges and some competitions, such as the football world-cup, attract a vast number of spectators. Game competitions are generally not as popular as sports although there are trends in some countries (e.g., South Korea) where such competitions have the capacity to fill stadia. Every competition must face the issue of how to evaluate and rank competitors and often rules are required to account for many different aspects such as variations in conditions, the ability to cheat and, of course, the value of entertainment. This section reviews a range of popular formats that may be used to evaluate and rank competitors.

The overall format that is suitable for a competition depends on many factors, including the duration of the tournament (and the game), the mode of evaluation (e.g., win/loss/draw versus score-based), whether competitors have the chance to re-enter once they have lost (e.g., buy-ins in poker) and so on. The first set of schemes reviewed below are commonly used for one-off competitions, whereas the latter half are used more commonly for ongoing competitions (e.g., leagues).

One of the simplest yet very popular formats is *single elimination* (or knock-out), which consists of a succession of rounds where the winner of a single match progresses to the next round while the loser is eliminated. A slight modification of this is the *double elimination* tournament, where a player has to lose two matches to be disqualified. This is usually achieved by maintaining two different sets or brackets (Upper or Winner, and Lower or Loser brackets). Each one of these brackets behaves like a single elimination tournament with the exception that every time a player from the Upper bracket loses, she/he is sent to the lower bracket. If a contestant in the lower bracket loses, she/he is eliminated from the tournament. Both brackets merge at the end and the final of the tournament is played by the last two remaining players.

Another commonly used pairing system is the *round-robin tournament*: all competitors are paired against one another for one or more matches allowing each an equal opportunity to display their strength (although ordering might impact performances). The biggest issue with this approach is that it scales poorly and a large number of competitors may take prohibitively long to evaluate. An alternative in this case may be the *Swiss tournament*. In this system, after each match, each player is awarded points for winning, drawing or losing. All contestants are paired up to a pre-defined number of rounds and they always play against opponents of similar score. Finally, a tournament scheme similar to the round-robin scheme is the *Scheveningen* system which may be used for teams of players: each member of the each team is paired with each member of the other and the team with the most wins becomes the winner of the tournament.

Round-robin tournaments may be used easily for one-off evaluations and it is straightforward to extend them for ongoing evaluations by using *leagues*: competitors are assigned to different leagues, usually according to skill, each of which constitutes a single round-robin tournament. After each tournament is finished the best competitors are promoted while the worst are relegated. Entirely new entries may be placed into the bottom league and can subsequently move their way up.

While leagues implicitly assign a skill to each competitor (i.e., the league they are in) there are numerous explicit approaches that assign actual skill ratings to competitors. One such approach is the *Elo Rating* system [13], proposed by Arpad Elo in 1959 where each player is assigned a numerical estimate of his playing skill. This value is updated during the tournament depending on the skill of the opponents and the outcome of the game. This mechanism suffers from the initial uncertainty of a player's skill and the order in which opponents are paired. The *Bayes Elo* rating system [11] tries to overcome the problem of uncertainty as to the skill of a player by using a Bayesian approach. This system is based on choosing likelihood distributions over the Elo ratings and computes a player's probability of win or loss using the differences between the Elo ratings of the players to compete. These ranking schemes are explored further in Section IV-E.

Similarly, *Glicko* and *Glicko-2* were developed by Mark E. Glickman as an improvement of Elo. The objective was to take into account the uncertainty associated with a player's skill rating. Furthermore, Glicko includes a mechanism that increases the uncertainty over time if a player has been inactive. This ranking system is described further in Section IV-F. *Glicko-2* is an extension of Glicko that adds a volatility rating to measure the degree of expected fluctuation in the rating of the player. This value will be high when the player has a high variability in the outcomes of the games played and low otherwise. Although this value does not affect the strength of the player, it is used to update the player's skill rating.

TrueSkill is another Bayesian skill rating system that tries to tackle the problem of uncertainty on the player's skill prediction. It was developed by Microsoft Research [20] and is used to rank players for XBox Live (the online community for Microsoft XBox gamers). This technique is based on a generalisation of the Elo Rating system where the uncertainty in the player's skill is predicted by a Bayesian network. It is particularly suited for multiplayer games where multiple participants are part of the same team (which wins or loses as a group) but where the players need to be ranked individually.

There has been another approach [19], [27], [16] for evaluating player strength by comparing moves between real players and a referee/evaluation player. Let us assume that we have available some evaluation heuristic (possibly augmented by further search) that can give us a value (known in Reinforcement Learning as the Q-Value) that describes the quality of each action at each state. One can find the absolute mean difference of the Q-Value played by a player and the and the actual best move proposed by the heuristic (i.e., the referee player) and use this as a measure of player quality. There is obviously a problem here as to how we get good (i.e., close to optimal) Q-Values in the first place. Ideally, an equilibrium

player would be optimal, however, as indicated by Guid et. al. [18], [17] if one is to compare a large number of moves, a heuristic player/referee that can play an optimal move at any state with probability higher than random is enough. This approach make much better use of available game plays, thus an interesting avenue of future work would be to apply this method to Ms Pac-Man and other asymmetric score based games. This however, is outside the scope of this paper and can be the focus of feature research.

Finally, there are approaches that include a time varying aspect on the quality of the players, i.e., they assume players improve over time [12], [2]. Since the experiments done in this paper involve static players, these approaches are again outside the scope of this paper. This paper focuses on the two most popular non-proprietary ranking schemes, Bayes Elo and Glicko.

B. Rankings in Game Competitions

Game competitions are an important resource for researchers and developers to test and compare their algorithms. A wide array of different competitions have been available over the last decade. Table I summarises some of these competitions: the columns indicate the name of the competition, the year of the competition's first edition, the type of game ("S" denoting that the game is stochastic in nature), the number of players per match ("HT" implies availability of heterogeneous types of player) and the average number of participants (considering all editions of the competition). Finally, the last two columns show how the winner of a match is decided and the policy or ranking scheme used to proclaim a competition winner.

Any game competition has a particular goal in mind when it comes to correctly favouring appropriate entries. To achieve this, it is crucial to devise an appropriate ranking system that favours entries with the desired qualities and minimises the possibility of erroneously placing well due to auxiliary effects such as the exploitation of competition specific rules. It is thus valuable to continuously review such schemes in existing competitions.

Game competitions differ from one another in many aspects, including the type of game used, their goal, the amount of players taking part in a match and the way these are rated and ranked. In the case, of single-player games the participants develop bots that perform better than others independently and players are usually ranked according to the score of the game, possibly modified. For instance, the score obtained in the Mario AI competition [32] is obtained by a linear function that weighs the time taken to complete the level, enemies killed and items collected; while in The PTSP competition [26] the score corresponds to the number of waypoints visited and the time taken to do so. Other competitions, like the Simulated Car Racing championship [22] (where the first phase is single-player), are ranked based on the time taken by the player to complete one or more laps in a given circuit.

Multiplayer games, on the other hand, are typically more involved. The optimal strategy to play the game may depend on the opponent and hence it is vital to play a number of different opponents to obtain a true indication of performance. Such pairings may depend on numerous attributes including the type of game, duration of a match and the number of participants. The simplest pairing mechanism is the *round-robin tournament* which is used by several competitions including the ToroidWars or the first editions of the Ms Pac-Man versus Ghosts competition. Other contests, such as Robocup [24], employ a *knock-out tournament* that mimics the system of international football tournaments: a first stage is established where the competitors are divided into several groups and where a round-robin tournament is played. Some participants subsequently progress to the knock-out phase where the teams play each other to reach the final. Other competitions, such as Ludo and the 2011 edition of Starcraft also employed this ranking scheme.

Computer Go Competitions [9] have been running since 1984 and usually have a large number of competitors. Here, the organisers decided to use a *Swiss tournament* scheme, as playing a round-robin tournament is infeasible owing to the elevated number of entries. Another competition with a large number of entries is the Google AI Challenge, which is probably the most famous Game AI competition in the world. The amount of participants in this competition is between two and three orders of magnitude higher than any of the other competitions and hence it is infeasible to pair all competitors to establish a ranking. To optimise the tournament, the organisers have altered their ranking systems for the different editions of the competition from *Elo* to *TrueSkill*.

There are additional factors (i.e., in addition to the number of participants) that may affect the choice of ranking scheme: the duration of each game played, as well as the stochasticity of the game. The former requires fewer games to be played if time is an issue while the latter requires multiple games to be played to obtain a more reliable outcome. Games like Mario AI, Robosoccer or Tron (the first Google AI Challenge) have a finite length by default, while other games need to artificially impose a termination condition. The game used in the 2011 Google AI Challenge, Ants, could potentially run forever and the organisers introduced several cut-off rules in order to reduce the time needed to decide the winner of a match. These rules range from a maximum number of turns of play to analysis of the agents to determine when a player can no longer win the match. Likewise, the stochasticity of the game needs to be accounted for: the Starcraft [33] competition, for instance, follows a knock-out phase where five games are played and the bot that wins three of the games progresses to the next round.

Finally, special attention must also be paid to the game's player types and whether competitors are able to choose amongst these freely. One interesting example is the Starcraft competition where there are three different races that the participant may choose from. Each of these has different abilities and strategies that must be chosen accordingly. The rankings of this competition do not distinguish between player types, so it is difficult to know how this choice impacts on the player's performance. An even more stringent example of heterogeneous player types is the Ms Pac-Man versus Ghosts competition [28] where the player can choose between playing

TABLE I: Game artificial intelligence competitions.

Competition	Since	Game type	Players	Average participants	Match winner	Ranking scheme (competition winner)
Computer Go	1984	Board	2	85.17	Win/Loss	Swiss system tournament.
Robocup (simulated soccer)	1997	Sports - Real Time (RT)	2	18.8	Win/Loss	Group stage followed by knock-out phase (FIFA World Cup Scheme).
Pac-Man Screen Capture	2007	Arcade RT (S)	1	7.8	By points	Winner by highest score, considering the best result of each player after N matches.
BotPrize Turing Test	2008	First Person Shooter (FPS)	2	4.75	Judges accuracy	Judges play and tag enemies as human or bot. The bot most tagged as human wins.
Simulated Car Racing	2008	Racing RT	1,2+	7.8	Fastest driver	One-car qualification lap, best times to finals, where best racers run on M circuits, with F1 score system.
ToroidWars	2009	Arcade RT	2+	-	Survival	Full round-robin.
Mario AI (Gameplay)	2009	Platformer RT	1	12	By points	Winner by highest score, calculated as average of N plays in different levels.
Mario AI (Level generation)	2010	Platformer RT	1	6	Player choice	Levels presented 1vs1 to players who choose the best by taste. The winner is the most preferred one.
Mario AI (Learning)	2010	Platformer RT	1	4	By points	10.000 training cycles per map, one run to evaluate. Winner by highest score, sum of N levels played.
Starcraft	2010	Real Time Strategy (RTS)	2 (HT)	26	Survival	Knock-out tournament, best of 5 matches qualifies to next round (2011). Full round-robin (2012).
Google AI Challenge	2010	Arcade RT	2,2,2+	6250	Survival	Elo rating system (2010) and TrueSkill (2011).
Car Setup Optimization	2010	Racing RT	1	5	Fastest driver	Learning warm-up. Best result after N laps in M circuits. Winner by sum of F1 points per track.
Ludo	2010	Board (S)	4	23	Win/Loss	Group stage plus knock-out phase. 10K games, evaluation on last 2. Separated round-robin league.
Pac-Man vs. Ghost	2011	Arcade RT (S)	2 (HT)	39.6	By points	Full round-robin. Glicko rating system (September 2012).
Demolition Derby	2011	Racing RT	2+	3.5	Survival	Round-robin 1vs1 qualification. Best N drivers fight together in a single track to determine winner.
PTSP	2012	Navigation RT	1	28	By points	Sorted by average of best 3 out of 5 runs per map. Final sum of F1 points on N maps decides winner.
Mario AI (Turing test)	2012	Platformer RT	1	5	Judges accuracy	Bots presented 1vs1 to judges, who choose which one is most human. Bot most voted as human wins.

as Ms Pac-Man or as the ghosts and the election of character changes the nature and objective of the game. For this reason three separate rankings are required: for Ms Pac-Man, the ghosts and a combined track for players with two controllers.

IV. Methodology

A. Results of WCCI 2012

The third iteration of the competition (WCCI 2012) was the largest to date and featured a total of 80 competitors (4 of the controllers submitted failed to execute) and a total of 118 controllers: 63 Ms Pac-Man controllers and 55 ghosts controllers. Competitors were able to submit controllers as many times as they liked prior to the submission deadline and to monitor their controller's performance via the live rankings. Once the submission deadline had passed, all scores were reset and the final ranking was established using a full round-robin tournament matching every Ms Pac-Man controller against each ghost team to play 5 games each. This led to a total of $63 * 55 * 5 = 17325$ games being played (each Ms Pac-Man controller played a total of 275 games, each ghost team a total of 315 games).

B. Converting Win/Loss ranking schemes to score-based games

The majority of rating systems, including Elo, Glicko, Glicko-2 and True Skill, have been designed for symmetric win-loss games. The Ms Pac-Man competition is not only score-based but the game differs substantially for Ms Pac-Man and ghosts controllers, thus creating information asymmetry. To convert the game into a format applicable to the major rating systems the following procedure (for Ms Pac-Man controllers) is adopted

1) One ghost controller and two Ms Pac-Man controllers are selected.
2) Each Ms Pac-Man controller plays a game against the ghost controller.
3) The Ms Pac-Man controller with the highest score is given a win, the other one a loss. If both controllers have the same score, the game is designated as a tie.

The following example illustrates how this process works: let us assume we have chosen four competitors, $g1$, $g2$, $p1$ and $p2$. $g1$ and $g2$ are two ghost teams, while $p1$ and $p2$ are pacman players. We can obtain 4 outcomes from 4 games played:

$$s1 = g1 \ \mathbf{v} \ p1$$
$$s2 = g2 \ \mathbf{v} \ p1$$
$$s3 = g1 \ \mathbf{v} \ p2$$
$$s4 = g2 \ \mathbf{v} \ p2$$

, where $s1, s2, s3, s4$ are scores between the above players/agents. We can then match $g1$ and $g2$ as follows:

$s1 < s2$ is a win for $g1$, loss for $g2$
$s3 < s4$ is a win for $g1$, loss for $g2$
$s1 > s2$ is a win for $g2$, loss for $g1$
$s3 > s4$ is a win for $g2$, loss for $g1$

The two Pac-Man players may be compared in a similar fashion:

$s1 > s3$ is a win for $p1$, loss for $p2$
$s2 > s4$ is a win for $p1$, loss for $p2$
$s1 < s3$ is a win for $p2$, loss for $p1$
$s2 < s4$ is a win for $p2$, loss for $p1$

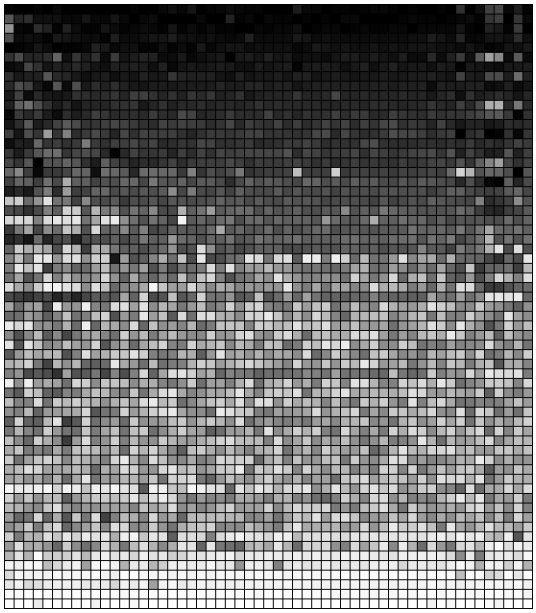
A tied score results in a draw.

C. Full Round Robin Dominance Tournament

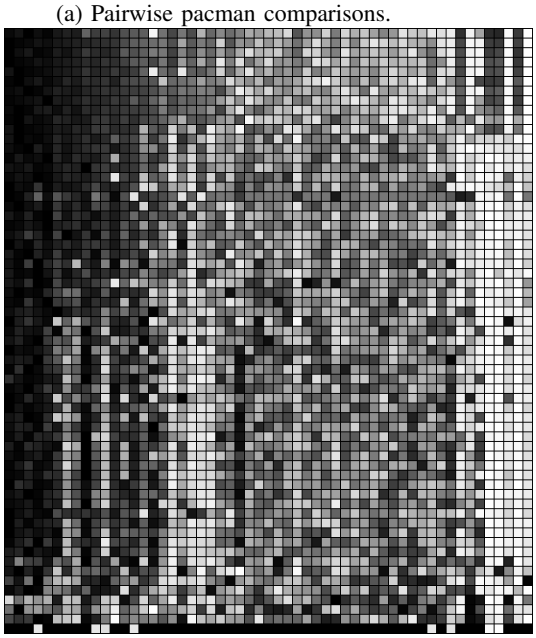
It is interesting to look at how well the controllers dominated one another and this is visualised in Figure 3: Figure 3a shows the data for the Ms Pac-Man controllers (read left-to-right) and Figure 3b the data for the ghost team controllers. For each controller of the same type, we compare its score against all other controllers of the same type against each possible opponent. The darker the colour in Figure 3 the higher the percentage of controllers it dominated (row vs column controller). Both axis represent Ms Pac-Man (for Figure 3a) or Ghost controllers (for Figure 3b). So for example in each cell in Figure 3a what is plotted is the percentage of ghost controllers (with 0 being white and 1 being black) that the row player has better scores vs the column player. In the case of the Ms Pac-Man controllers, the expected trend of better controllers dominating all others, across all types of opponents, is evident.

There is some noticeable noise, caused both by the stochasticity of the data as well as intransitivities [1]. The data for the ghosts is much more noisy but reveals some interesting points: first, it is clear that there are some ghost teams that perform especially well against particular opponents as evident by light or dark vertical lines in the graph. Furthermore, intransitivities are highlighted much more strongly. Finally, the data complements the visualisation of the (normalised) scores and explains why there is so much variation in the performance of the Ms Pac-Man controllers against the weakest ghost teams: the dominance data shows that the ghost teams ranked 48, 51, 52 and 54 seem to have performed especially well against the top 11 Ms Pac-Man controllers. Unfortunately, this performance is not rewarded in the round-robin style tournament as it does not matter who a controller dominates (i.e., how good the opponent is), but only how strongly. We can also use this data to establish a ranking based on the average degree of domination.

[1] intransitivities are technically not noise, but give the appearance of "salt and pepper" noise on the plots



(a) Pairwise pacman comparisons.



(b) Pairwise ghost comparisons.

Fig. 3: Visualisation of pairwise comparisons to establish dominance based on the scores of the WCCI 2012 iteration of the competition. X and Y axes denote controllers while the intensity of the score denotes the strength of controller row vs controller column

D. Solution Concepts and Intransitivities

The previous section highlighted some of the issues associated with the round-robin style tournament used in the competition. Some of these issues are technical (e.g., the time it takes to compute the rankings) whereas the others relate to what is required from a controller to place high in the rankings (e.g., exploitation of specific opponent types). The latter issues in particular beg the question as to what constitutes the goal (objective) of a game competition in the first place. The goal of

Ms Pac-Man is, of course, to maximise the game's score (and likewise, minimise the score if one plays as the ghosts). In the original game this requires a player to clear maze after maze while eating as many ghosts (and bonus fruits) as possible. The necessity to limit the duration of each game, however, as well as the format of the competition, might favour controllers that are not actually pursuing these objectives yet rank highly (e.g., those that exploit the time limit and do not clear the maze). Further effort is thus required to ensure the rankings align with the intention of the competition: to find the best controller in playing Ms Pac-Man.

Solution concepts may be used to establish notions of optimality that comply with multi-agent settings, where a clear objective function is not available (e.g., an already existing top player). Solution concepts were introduced in co-evolution [14], [30], cloned from Game Theory, to establish the goals individual populations need to aim for to drive the overall search. Having the highest score in multi-player games with real-valued rewards means very little, as it could have been against horrible players or really good players. Thus, in order to avoid intransitivities, one has to have some score concept that defines one's ability to act vs others. One popular such solution concept is termed min-max and defines that a good player is one that cannot be beaten. Ranking players on this concept however is hard, as it requires one to search for the worst opponent against a player, something not readily available. Hence the need for ranking solutions like Elo and Bayes Elo. Note that these solutions do not take into account intransitive players, assuming that such cases are pathological and not commonly occurring in real life players.

Without addressing the problem of solutions concepts (or by having solution concepts that allow for more than one "best") one might come across scenarios where player A beats player B, player B beats player C, but player C beats player A. This kind of Rock-Paper-Scissor dynamics is not that common in human players; it is often the case however that it occurs in players emerging from some Artificial Intelligence method - for example see Samothrakis et. al. [30] for a discussion of this problem in Othello. The problem is also apparent in elite computer poker [4] players, but absent in other types of elite players, e.g., Computer Go [5]. Overall, all players have the potential for intransitivities, but players of games of partial observability are more prone to this, due to the fact that once all suboptimal (also called "dominated") strategies are removed, the remaining ones can have intransitivities due to the very definition of the game [7]. In such cases, the optimal strategy is to mix over non-dominated strategies (something known as a mixed or behavioural strategy). This is not the case in games of full observability like Chess, Go, Backgammon or Othello (a result known for a long time, due to Zermelo's theorem [31]), so any intransitivity in fully observable games can arise mainly due to player inconsistencies/weaknesses. Formally, Ms. Pacman is a "Markov Game", duo to the fact that agents act simultaneously, and thus much closer to games of partial observability.

The work described in this paper makes use of tournaments as the "ground truth" (in the sense of machine learning). That is, the solution concept employed in this paper tries to determine if an agent performs better than all other players of the same type against as many opponents as possible. This is by no means the only solution concept (for example, heavily exploiting weak players might seem preferable, or using the min-max solution concept), but it is assumed for this work that this solution has the closest resemblance to what humans intuitively consider a good ranking. We are going to be comparing different methods against this ground truth.

E. Elo and Bayes Elo Ratings

The Elo system was originally developed for chess players [13]: it computes an approximate statistical model of game outcomes that determines the likelihood that player A beats player B. Elo ratings are among the first ratings of skill with probabilistic underpinnings: two opposing players with equal ratings are expected to win an equal number of games played (i.e., 50%). A player's rating is updated throughout the competition depending on both the outcome of the and the opponent's skill: beating stronger opponents awards more points than beating weaker ones, for instance. It is important to notice that in the first rounds of the competition the skills of the players are considered to be provisional, as not enough games have been played to indicate a trustworthy rating.

The system models player skill variation using a normal distribution (although today a logistic distribution is used more widely). This is the basis of the logistic Bradley-Terry (BT) model of paired comparisons [10]. A key assumption of the model is that the expected preference depends only on the difference in strength between two players. Hunter [21] demonstrated how a minorization-maximization (MM) algorithm could be used to solve the straight Bradley-Terry model, plus many of its generalisations, with very little effort needed to tailor it to the situation at hand.

Similar to Glicko (see next section), the attraction of the Bradley-Terry model is that it enables a player to be rated on the basis of a relatively small number of games. For game leagues with a large number of players, *potentially* reliable ratings can be based on a tiny fraction of the number of games that would be involved in a full round robin league. Using Hunter's MM algorithm, the model can be fitted to a set of game data in a small number of iterations from an arbitrary starting point, and will converge to the best possible fit. The algorithm naturally accounts for the relative strengths of the players involved in each comparison, and beating a strong player is worth more than beating a weak player. In this paper we used Heungsub Lee's implementation of Glicko [3].

The *Bayes Elo* rating system [11] tries to overcome the problem of uncertainty as to the skill of a player by using a Bayesian approach. This system is based on choosing likelihood distributions over the Elo ratings and computes a player's probability of win or loss using the differences between the Elo ratings of the players to compete. Unlike conventional Elo rating estimations, Bayes Elo estimates are, by default, independent of the ordering of the game results, but do take into account the player order (i.e., which player is first and which second). In this paper we use Coulom's implementation of Hunter's EM algorithm in his BayesElo [1] tool.

F. Glicko Ratings

Glicko rankings have been developed by Mark E. Glickman [15] to address some of the shortcomings of Elo ratings. More specifically, Glicko is a more general approach that improves on the reliability of the players' Elo ratings by taking into account the timeline of past plays: the rating of a player that has competed regularly in recent times should be more reliable than the rating of a player who has been inactive for an extended period of time (Glicko includes a mechanism that increases the uncertainty over time if a player has been inactive). Subsequently, the ratings of these players should be updated accordingly. This can be achieved by associating a measure of reliability with each rating: in addition to the player's rating r, each player also has a rating deviation δ, which is the standard deviation of r that measures the uncertainty in a rating; the higher this deviation the less uncertain we can be about a player's rating.

The value of r is changed whenever the player has competed in a game. Likewise, the value of δ is decreased. However, the value of δ is increased over time if the player is inactive to express the growing uncertainty over the value of r. The values for δ have a direct impact in the change of r given the outcome of a game, and the ratings of the two players are no longer changed symmetrically as is done with Elo. The existence of a deviation makes it more natural to express a player's skill as an interval and not a scalar: the true skill lies with probability 0.95 within $[r - 2\delta,\ r + 2\delta]$.

The update cycle of Glicko is based on rating periods: a rating period is a set of games played after which all the players' ratings are updated. First, the rating and rating deviation for all players needs to be determined. Unrated players start with $r = 1500$ and $\delta = 350$. Rated players have their rating deviation updated at each cycle. For a mathematical depiction of the algorithm please see [15].

V. Experiments

The results of the WCCI 2012 Competition (a full round robin tournament) were processed and three basic questions were formulated that are all based on the fact that a player is evaluated from only a proportion of games from the ones required to do a full round robin dominance tournament (due to processing/time constraints). These questions are:

1) Mean performance
2) Worst case performance
3) Outside top 10

The purpose of each question is as follows: mean performance indicates just that, the average rank-mistakes for each player, i.e., how many positions on average each method got wrong. The other two questions are posed because the mean might not be the best indicator of overall performance; a method that makes a single massive mistake is worse than a method that makes many minor ranking mistakes. The last question, more specifically, is posed because it is common in many tournaments to use approximate methods for initial rankings and to use cut-off point in the ranks (e.g., 10 best), on which full round robin experiments can be performed.

For each measurement, to be presented below, a number of games ranging from 1 to 80 were sampled, both BayesElo and Glicko algorithms were executed and a number of measurements were taken. This sampling was performed 45 times, in order to give the results statistical significance. BayesElo was run with the following properties:

```
advantage 0 % advantage of playing first,
            % 0 in our case, no advantage
exactdist % compute intervals
          % assuming exact opponent Elos
drawelo 0 % draw Elo, no bias in our case
prior 0 % number of virtual draws
        % 0 in our case, we have no priors
```

The above properties were chosen so as to nullify the effect of priors in Bayes Elo that assume certain advantage for the first player. Glicko was run with the following parameters:

```
MU = 1500 % Initial Mean
SIGMA = 350 % Initial Standard deviation
```

All games were converted to win/loss using dominance scores (as in the previous section).

A. Mean Performance

The first measurement taken was the mean rank error ("deviation") from the round-robin dominance ranking as per Equation 1.

$$m_0 = \left(\sum_0^n |\hat{r_i} - r_i| \right) / n \tag{1}$$

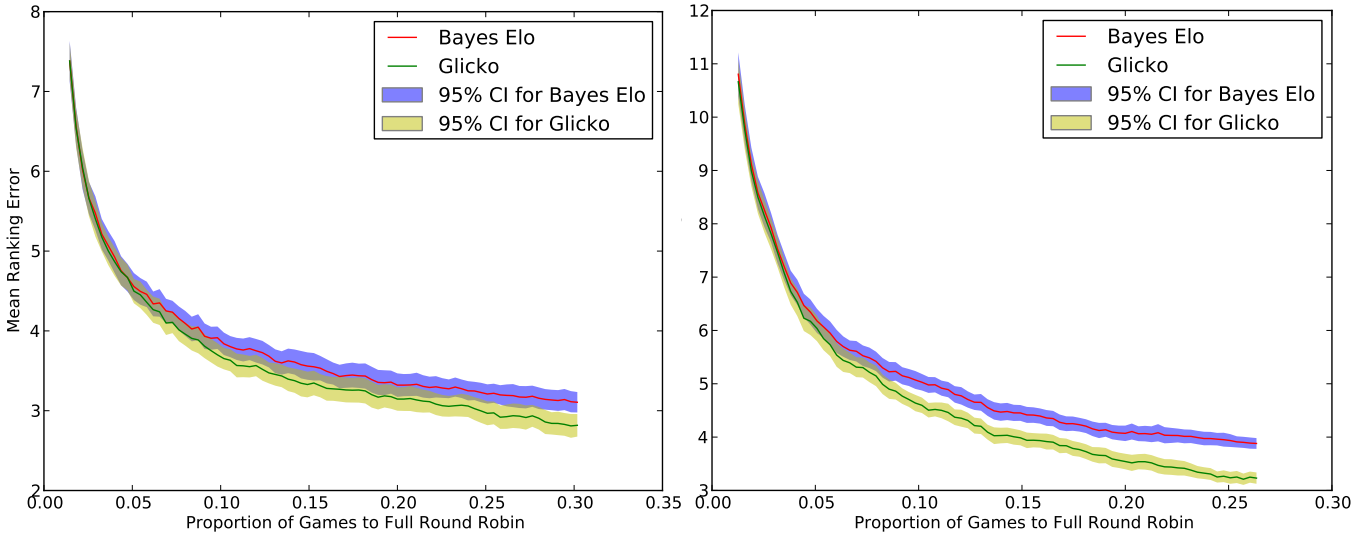
In this equation, n is the number of players in the rankings, while r_i is the ranking position of the player i. Thus, m_0 signifies the average change in rank. Figure 4 plots this for both Ms Pac-Man and the ghosts. The shaded area in this (and all subsequent graphs) is the 95% Confidence Interval (or $p = 0.05$). If the shades do not overlap, statistical significance can be observed, although one can get this result with fewer games using t-tests. For example, Ms Pac-Man scores become significant with $p = 0.031$ at around 20% of the round-robin games played. Also note that, on average, after 25% of the games have been played, the average rank error is 3. Glicko outperforms BayesElo in both player categories.

B. Worst Case Performance

The second measurement taken involves the worst position mistake made by the algorithm. More formally, this is defined as per Equation 2.

$$m_1 = \max \left(\sum_0^n |\hat{r_i} - r_i| \right) \tag{2}$$

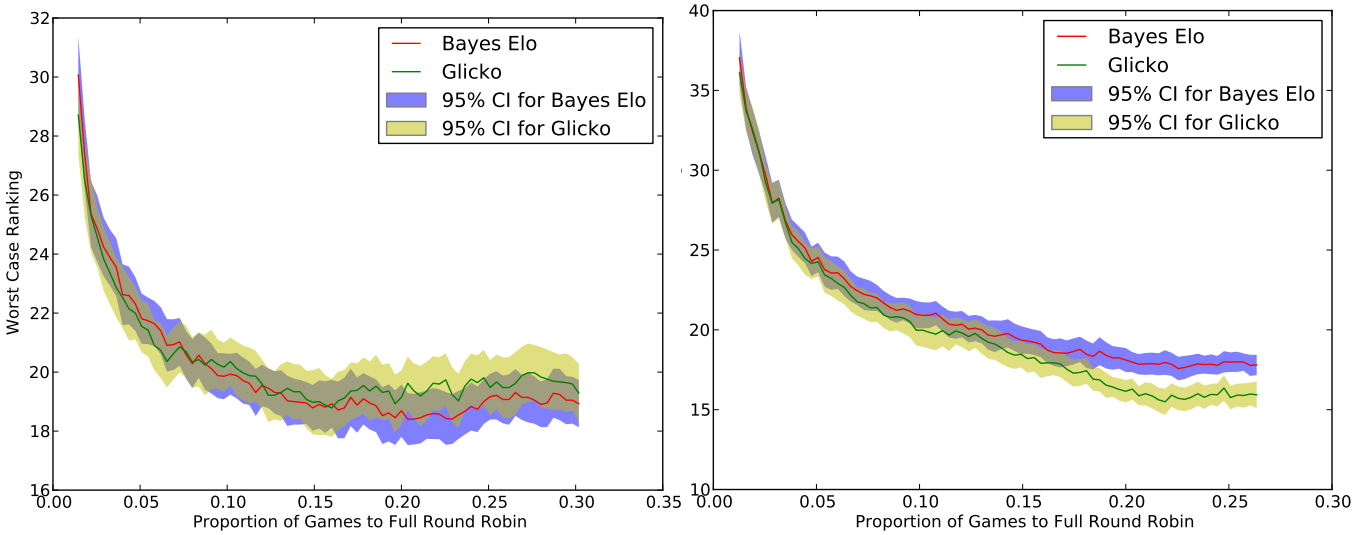
The measurement m_1 is important because even a single mistake that misplaces the rank of a player by a large margin can prove detrimental to the rankings. As can be seen in Figure 5, both ranking algorithms make noticeable mistakes, beyond what would be acceptable in competitive tournament play (in the range of $[15, 20]$). It is also worth noting that even



(a) Ms Pac-Man Mean Score

(b) Ghost Mean Score

Fig. 4: Mean Ranking Error (m_0) for both player types. Smaller scores are better



(a) Ms Pac-Man Worst Case Ranking

(b) Ghost Worst Case Ranking

Fig. 5: Worst Case Ranking Error (m_1) for both player types. Smaller scores are better.

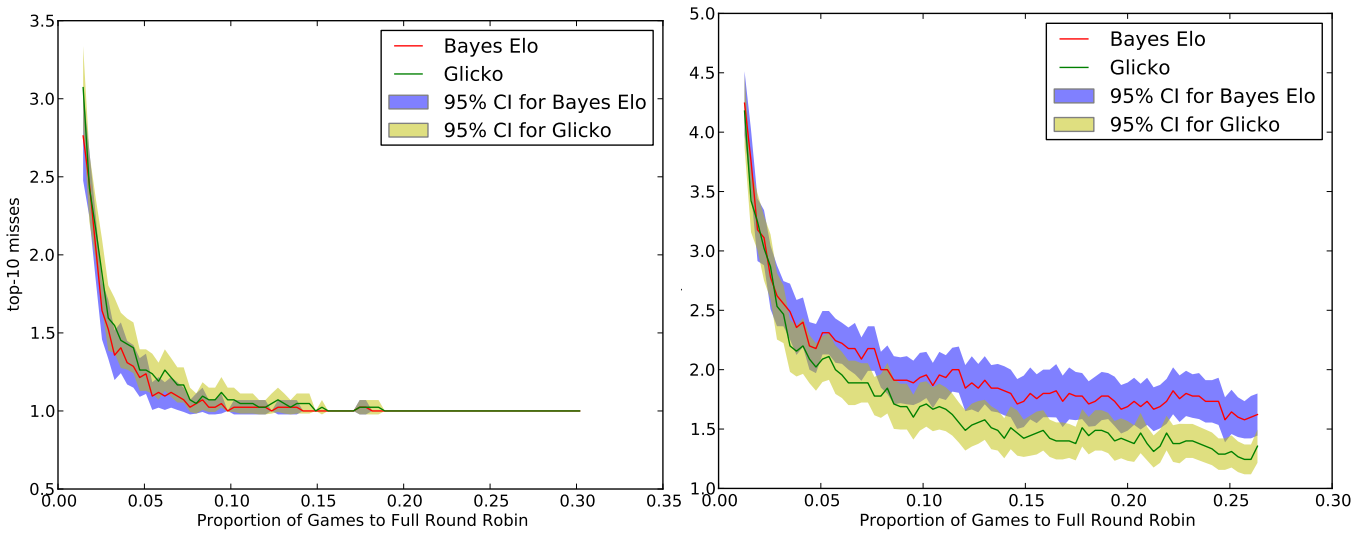
though the number of games increases, in the case of Ms Pac-Man players there is no significance in the difference between either algorithm (see Figure 5a)). Ranking also gets worse past a certain level, probably due to both algorithms getting confused by intransitivities. Again, in both cases Glicko is superior, albeit with statistical significance only in the case of the ghost teams.

C. Outside Top 10

The final measurement taken in this research is somewhat more involved and concerns the set difference of the top 10 ranked elements. A common scenario in a lot of competitions is that all participants are paired in a random (or quasi random) fashion and once the top-x (here $x = 10$) have been identified, a full round robin tournament is played among all these players. This line of thinking is also motivated by the previous experiment. Since there are big rank change errors in these predictions, it makes sense to predict the top-10 and do full round robin these predicted 10. More formally, measurement m_2 is defined in Equation 3.

$$m_2 = |\{r_0 ... r_9\} \setminus \{\hat{r_0} ... \hat{r_9}\}| \tag{3}$$

As can be seen in Figure 6b, both algorithms fail to reduce the number of mistakes in the top 10 down to zero. In the case of Ms Pac-Man, both algorithms get to the point of failing to get just one candidate (both do achieve this at position 11 - not shown in the figures) - which seems to stem mostly from intransitivities in the games played. This is obviously not satisfactory in the case of ghosts either. There is very little to be done about intransitivities, which are not taken into account by most ranking algorithms. However, in the case of ghosts,



(a) Ms Pac-Man Top 10 outside top 10

(b) Ghost Top 10 outside top 10

Fig. 6: Top 10 players not in the top-10 (m_1) for both player types. Smaller scores are better.

even with almost 80 games played (the right end of the plot), we can still get runs with two or three players that belong in the top-10, and yet are not there.

VI. Conclusion

Games provide excellent test-beds in which to develop, test and compare novel techniques in computational intelligence. Video games have recently become the focus of the academic community and numerous game competitions have become vital tools that allow researchers to test, evaluate and compare their algorithms. The competition considered in this paper is the Ms Pac-Man vs Ghosts competition that is based on the highly popular arcade game from the 80's, Ms Pac-Man. Ms Pac-Man poses numerous challenges as it is an open-ended asymmetric score-based game and several changes needed to be made to the game to limit the duration of individual games. These changes, however, may be exploited by the entries to the competition and care has to be taken that the rankings established during the competition reflect the controller's skill at playing the game rather than 'playing the competition'. Furthermore, reliable ratings may require a prohibitively large amount of games to be played. These issues may be addressed by using a suitable mechanism for selecting and evaluating opponents with the goal of establishing reliable ratings that reflect the true skill of the controllers, based on only a subset of possible games played. The rating schemes investigated in the paper are Bayes Elo and Glicko.

It is evident from the experimental results reported upon in this paper that Glicko outperforms Bayes Elo in almost all settings. Both algorithms have shown relatively good performance and are fit for purpose, but Glicko performs more reliably and should be the algorithm of choice for online rankings (at least in situations similar to the one described in this paper). None of the algorithms seems to break down completely in our scenarios and none of them seems to have any distinct pathologies, apart from failure to deal effectively with intransitivities. Dealing with intransitive players, we think, can safely be ignored for most tournaments, though we do think its a good idea to perform a full round robin among top contenders, as it minimises the risk of intransitive solutions becoming champions.

There are several aspects that warrant future research in this area. The two most important are as follows: the first is adding matchmaking capabilities to Glicko. This could have a profound impact on the quality of the rankings (as they are now done at random). The second is fine-tuning Glicko parameters for each specific game. One can envision a background optimisation process running continuously, trying to figure out which games should be played next. Again, the impact of this could be profound. Matchmaking is important in online games; it is not acceptable for, nor accepted by human players to play massively beneath or above their skill level and the ability to tackle ranking problems efficiently can have profound effects on the success of online games.

Acknowledgements

This work was partially supported by EPSRC grant EP/H048588/1 "UCT for Games and Beyond". We would like to thank all participants of the Ms Pac-Man vs Ghosts competitions, many of whom contributed actively towards making the competitions popular.

References

[1] Bayesian Elo. http://remi.coulom.free.fr/Bayesian-Elo/. Accessed: 2014-05-01.
[2] Chess Metrics. http://www.chessmetrics.com/cm/. Accessed: 2014-05-01.
[3] Glicko Python Implementation. https://github.com/sublee/glicko. Accessed: 2014-05-01.
[4] The Annual Computer Poker Competition. http://www.computerpokercompetition.org. Accessed: 2014-05-01.
[5] The Annual Computer Poker Competition. http://www.weddslist.com/kgs/past/S13.2/. Accessed: 2014-05-01.
[6] Twin Galaxies - Historical Archive. www.twingalaxies.com. Accessed: 2014-05-01.

[7] *Theory of Games and Economic Behavior*. Princeton University Press, 3 edition, May 1944.

[8] Anonymous. *Playing Ms Pac-Man to Win*. Video Game Books, inc; Simon and Schuster, 1982.

[9] Bruno Bouzy and Tristan Cazenave. Computer Go: an AI Oriented Survey. *Artif. Intell.*, 132(1):39–103, 2001.

[10] R.A. Bradley and M.E. Terry. Rank analysis of incomplete block designs. I. The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

[11] Rémi Coulom. Computing "Elo Ratings" of Move Patterns in the Game of Go. *ICGA Journal*, 30(4), 2007.

[12] Remi Coulom. Whole-history rating: A bayesian rating system for players of time-varying strength. In *Computers and games*, pages 113–124. Springer, 2008.

[13] A. E. Elo. *The rating of chessplayers, past and present*. Arco Publishing, New York, 1978.

[14] S. G. Ficici. *Solution Concepts in Coevolutionary Algorithms*. PhD thesis, Brandeis University, 2004.

[15] M.E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

[16] Matej Guid and Ivan Bratko. Computer Analysis of World Chess Champions. *ICGA journal*, 29(2):65–73, 2006.

[17] Matej Guid and Ivan Bratko. Using heuristic-search based engines for estimating human skill at chess. *ICGA Journal*, 34(2):71–81, 2011.

[18] Matej Guid, Aritz Pérez, and Ivan Bratko. How trustworthy is Craftys analysis of world chess champions. *ICGA journal*, 31(3):131–144, 2008.

[19] Guy Haworth, Ken Regan, and Giuseppe Di Fatta. Performance and prediction: Bayesian modelling of fallible choice in chess. In *Advances in Computer Games*, pages 99–110. Springer, 2010.

[20] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: A bayesian skill rating system. *Neural Information Processing Systems*, 19:569–576, 2007.

[21] David R. Hunter. MM Algorithms for Generalized Bradley-Terry Models. *The Annals of Statistics*, 32:384–406, 2004.

[22] D. Loiacono, P. Lanzi, J. Togelius, E. Onieva, D. Pelta, M. Butz, T. Lönneker, L. Cardamone, D. Perez, Y. Sáez, M. Preuss, and J. Quadflieg. The 2009 simulated car racing championship. *IEEE Transactions on Computational Intelligence and Games*, 3(2):131–147, 2010.

[23] Simon M. Lucas. Ms pac-man competition. *ACM SIGEVOlution Newsletter*, pages 37–38, 2008.

[24] Johannes Meyer, Paul Schnitzspan, Stefan Kohlbrecher, Karen Petersen, Mykhaylo Andriluka, Oliver Schwahn, Uwe Klingauf, Stefan Roth, Bernt Schiele, and Oskar Von Stryk. RoboCup 2010: Robot Soccer World Cup XIV. *Lecture Notes in Computer Science*, 11:432, 2011.

[25] T. Mott. *1001 Video Games You Must Play Before You Die*. Cassell Illustrated, 2010.

[26] D. Perez, P. Rohlfshagen, and S. Lucas. The Physical Travelling Salesman Problem: WCCI 2012 Competition. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2012.

[27] Kenneth Wingate Regan and Guy McCrossan Haworth. Intrinsic chess ratings. In *AAAI*, 2011.

[28] P. Rohlfshagen and S. M. Lucas. Ms Pac-Man versus ghost team CEC 2011 competition. In *Proceedings of IEEE Congress on Evolutionary Computation*, page to appear, 2011.

[29] P. Rohlfshagen and S.M. Lucas. Ms pac-man versus ghost team cec 2011 competition. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 70–77. IEEE, 2011.

[30] S. Samothrakis, S. Lucas, T.P. Runarsson, and D. Robles. Coevolving Game-Playing Agents: Measuring Performance and Intransitivities. *Evolutionary Computation, IEEE Transactions on*, 17(2):213–226, 2013.

[31] Ulrich Schwalbe and Paul Walker. Zermelo and the early history of game theory. *Games and economic behavior*, 34(1):123–137, 2001.

[32] J. Togelius, M. Preuss, N. Beume, S. Wessing, J. Hagelback, and G. Yannakakis. Multiobjective exploration of the StarCraft map space. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, 2010.

[33] B. G. Weber, M. Mateas, and A. Jhala. Building Human-Level AI for Real-Time Strategy Games. In *Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems*, 2011.

Spyridon Samothrakis is currently a Senior Research Officer at the University of Essex. He holds a B.Sc. from the University of Sheffield (Computer Science), an M.Sc. from the University of Sussex (Intelligent Systems) and a Ph.D (Computer Science) from the University of Essex. His interests include game theory, machine learning, evolutionary algorithms and consciousness.

Diego Perez is currently pursuing a Ph.D. in Artificial Intelligence applied to games at the University of Essex (UK). He has published in the domain of Game AI, with research interests on Reinforcement Learning and Evolutionary Computation. He has organized several Game AI competitions, as the Physical Travelling Salesman Problem and the General Video Game AI competitions, both held in IEEE conferences. He also has programming experience in the videogames industry with titles published for game consoles and PC.

Philipp Rohlfshagen received a B.Sc. in Computer Science and Artificial Intelligence from the University of Sussex, UK, in 2003, winning the prize for best undergraduate final year project. He received the M.Sc. in Natural Computation in 2004 and a Ph.D. in Evolutionary Computation in 2007, both from the University of Birmingham, UK. Philipp completed a series of post doctoral positions in evolutionary computation and games and is now a Principal Scientist working for SolveIT Software in Adelaide, Australia.

Simon Lucas (SMIEEE) is a professor of Computer Science at the University of Essex (UK) where he leads the Game Intelligence Group. His main research interests are games, evolutionary computation, and machine learning, and he has published widely in these fields with over 130 peer-reviewed papers. He is the inventor of the scanning n-tuple classifier, and is the founding Editor-in-Chief of the IEEE Transactions on Computational Intelligence and AI in Games.

List of Figures

List of Tables