UNIVERSITY OF GOTHENBURG

Reliability of ranking using champion challenge format in artificial intelligence tournament

Bachelor of Science Thesis in Software Engineering and Management

MAI PHUÖNG VAN

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2016

Reliability of ranking using champion challenge format in artificial intelligence tournament

MAI PHUÖNG VAN,

Examiner: JAN-PHILIPP STEGHÖFER

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2016

Reliability of ranking using champion challenge format in artificial intelligence tournament

Mai Phuong Van
University of Gothenburg
Software Engineering and Management
gusmaiva@student.gu.se
Supervisor: Michal Palka

Abstract

Working together with a Gothenburg-based company, Software Skills [8], a prototype of a multiplayer AI tournament format called champion challenge using multiplayer Elo rating [12, 6] was made. This experiment is to study the reliability of the prototype as well as comparing it to Software Skills's old tournament format. Even though there was not enough evidence proving the reliability of the new tournament format, there are some interesting results in comparison to the old system.

1 Introduction

Developing artificial intelligence (AI) has always been an interesting topic in the software field. There exists many AI tournaments which AI players are developed to play a certain game [9, 4, 3, 5, 1], and each of them uses different tournament format and ranking system [17] to determine the winner. Software Skills (SWS) also hosts AI tournaments using their programming tournament platform *Honeypot* [10]. As the tournaments aim to find the skilled programmers, the final ranking of each tournament's goal is to reflect how skilled the contestants are.

At present, their tournament ranking is decided by the relative performance of the AI players written by contestants compared to pre-written AI players provided by SWS. This ranking system is referred to as *relative performance* in this study. One downside of this system is that the pre-written AI players have to be rather a good in order to fairly evaluate contestants' AI players and to program such good players takes a lot of time and effort. Therefore, SWS wants to develop another format where AI players written by contestants will fight against each other and the ranking is determined using a multi-player Elo rating system [12, 6]. The tournament will be run more than one round where the first round is a complete random matching. From the second round, the players will play against players that are adjacent to them in the current ranking. This format will be referred to as *champion challenge* in this study as the players will try to get to the top and fight against the champion.

This research consists of evaluating the reliability of the champion challenge format in ranking the best AI scripts. It compares the results of both the relative performance and champion challenge format. This is done by implementing a prototype into the Honeypot platform. The research result will help to determine if the new champion challenge format is the better AI tournament format, and that will be beneficial for software companies in finding and recruiting skilled programmers.

Section 2 introduces the problem's background as well as popular tournament formats and ranking system. It also lists some existing AI tournaments and what tournament format and ranking system they use. Section 3 introduces the research questions and hypotheses. Section 4 covers the implementation of the champion challenge format. Section 5 outlines research methodology used to gain data relevant to the research questions. This includes an experiment and statistical data analysis. Section 6 presents the results gained and how the results answer the research questions. It also includes discussion of the results, limitations and further work. Section 7 concludes and gives a summary of this paper.

2 Background and related work

Software skills is a consulting company which helps other companies recruiting for skilled developers and other IT experts [8]. SWS uses various types of tests in order to help companies finding the most suitable candidates. One of their most famous tests to search for talented developers is through contests using an user-friendly platform for creating and hosting programming tournaments, *Honeypot* [10].

Honeypot is a platform where different AI programming contests can be created and hosted on the SWS website. To participate in the tournament, contestants have to go to Honeypot contest page where they can write and execute code in different programming languages. When the contestants submit their AI, it will be evaluated and ranked.

2.1 Existing tournament formats and ranking systems

To find the most suitable tournament format for the multiplayer tournament in Honeypot, we surveyed the most common tournament formats and ranking systems which are presented here.

2.1.1 Round-robin tournament

A round-robin tournament is defined as a tournament in which "players or teams engage in a game that cannot end in a tie and in which every player plays each other exactly once" [13] and the winner is determined by counting points [17]. This means the more contestants participate, the more matches there are. Therefore, the round-robin tournament is the most robust, but also most time-consuming format [17].

2.1.2 Swiss-system tournament

A Swiss-system tournament is a non-elimination format where a player plays versus a few other players in a determined number of games [11]. Like the round-robin tournament, the players should only play versus each other once. However, in the Swiss-system tournament, the players play versus opponents with a similar rating [18].

The champion challenge format is inspired by this Swiss-system tournament. However, this champion challenge format deals with four players in a game and wants to run as many games as possible so it is possible for a player to play against the same player more than once.

2.1.3 Binary elimination tournament

In binary elimination tournaments, games are divided into stages. Each stage is a set of pairwise matches and the winners proceed to the next stage while the losers get eliminated [15]. After each stage, only half of the contestants remain, which makes this tournament format economical, but the limitation of this format is selection probability [17], meaning the pairing may not be completely fair. A loser of a pair of strong players may perform better than the winner of a pair of weak players.

2.1.4 Elo rating system

The Elo rating system is the best known chess rating system developed by Apard Elo [12]. Nowadays, it is not only used in chess but also in other sport and computer games. The Elo rating system calculates the relative skill level of a pair of contestants by comparing the predicted outcome to the actual outcome of the game [16].

The probability Player$_i$ winning in a match-up versus Player$_j$ is calculated as:

$$E(R_i, R_j) = \frac{1}{1 + 10^{(R_i - R_j/400)}}$$

Where R$_i$ and R$_j$ is the Elo rating of the player. The larger the rating number, the higher rank the player. Then, after the game against the jth player, the score of ith player gains is 1 if ith player wins, 0.5 for draw and 0 for lost. This score is noted as S$_{i,j}$. Then ith player's new ranking is calculated as

$$R_{inew} = R_i + K(S_{i,j} - E(R_i, R_j))$$

Where K is a constant that affect the emphasis of the difference between the actual score and the expected score [16]. The key is in the part $(S_{i,j} - E(R_i, R_j))$ which which entails that when the contestant wins, the smaller the predicted probability of winning, the higher the player's rating raise, and when the contestant loses, the higher the probability of winning, the larger the rating drop.

2.1.5 Judge Diplomacy Player Ratings

Judge Diplomacy Player Rating (JPDR) is based on Tony Nichols and George Heintzelman's Elo Inspired Diplomacy Rating System [6, 2]. While the Elo rating system is designed for two-player games, it can be used for games that have more than two players. Each player begins with a rating of 1000 and this represents an average payer. The change of each player's rating depends on four factors: number of raw points the player won in the game (S), expected points (X), the player's past experience (E), and the value of the game (V).

S is the raw score of the game. If there are M players, total points is M and the points will be distributed to the players accordingly [6].

X is calculated based on how strong the player is and how strong the player is among the opponents. The strength of the player is calculated as $e^{R/500}$ as R is the player's old rank. The ratio between the player's strength and sum of all players in the game's strength indicates how strong the player is among the opponents. Then the expected score

will be calculated as:

$$X = M \times \frac{e^{R/500}}{\sum_{i=1}^{M} e^{R_i/500}}$$

where R_i is the old rank of all players in the game and M is the total points distributed in game [6].

V is $7.5 \times A \times P \times R$, where A is the adjustment due to the variant, P is the adjustment due to press, and R is the adjustment due to the ratio of "fully rated players" in the game [6].

E is the player's experience, depends on how much games the player has played, G. E is calculated as $E = 1 + \frac{40}{10+G}$.If G is 0, E will be 5.0 and will decrease as more games are played [6].

The final movement of a player's rating is:

$$\Delta = V \times E \times (S - X)$$

[6] And the new ranking is:

$$R_{new} = R_{old} + \Delta$$

2.2 Existing AI tournaments

Table 1 presents some existing AI tournaments and tournament format as well as ranking system that each tournament uses. Most of the tournaments use round-robin format, while the ranking methods are split between Elo rating system and simple based on win counts.

2.3 Liar's Dice game characteristic

The game that is used to compare the two tournament formats is *Liar's Dice* . Liar's Dice is a non-deterministic game where players roll their dice every turn. Players can only see their own dice, while they try to guess the total amount of dice on the board with a certain value and bid for it. Players will lose one die if the guess was wrong, and the winner is the last one with dice left [7]. For the relative performance format tournament, a contestant plays against 3 default AI players. Multiple matches will be run and the contestant will be ranked according to the total points they score. For the new champion challenge, the final ranking will be decided by a multi-player Elo rating system.

For both tournament formats, one game has 4 players and the final scores are determined by how good the players are relatively to each other. In this case, they are decided by how long a player survive in the game. As the focus of this research is the champion challenge tournament format, the scoring algorithm that we explore is follows this format, where 100 points are distributed to the players after each game.

Before exploring the characteristic of Liar's Dice game with 4 players, we can take a look at the more simple 1 versus 1 game characteristic. For non-deterministic 2-player games, all possible pairing can be put in a table, for example as shown in Table 2. For each pair, there is probability of one player winning.

Table 2. Possible pairs and example probability of a player winning against other player

	A	B	C	...
A	0.5	0.7	0.4	...
B	0.3	0.5	0.2	...
C	0.6	0.7	0.5	...
...	...	...	....	...

The probability is just an example. Though it follows 2 rules: the probability is bigger or equal than 0 and smaller or equal than 1, and probability of A winning B and B winning A has total of 1.

This table is also applicable for some deterministic games which have many different moves so that they behave like a non-deterministic game. For example, Chess is a deterministic game. If two deterministic programs play, then the result will always be the same (or, rather, there will be two results depending on which player starts). If two people play then it is reasonable to expect a different result each time.

For Liar's Dice game, it is more complicated than that. If the tournaments has X number of players, the number of possible grouping is combination of X choose 4 ($^XC_4$). For each combination, the players' scores can be described as:

$$\begin{aligned} M = &\{a, b, c, d \mid a, b, c, d \in N \wedge \\ &0 \leq a, b, c, d \leq 100 \wedge a + b + c + d = 100\} \end{aligned}$$

where a, b, c, d are each players' score. The probability of point distribution of every pairs still follow the same rules:the probability is bigger or equal than 0 and smaller or equal than 1, and probability of A winning B and B winning A has total of 1. The probability function can be describe as:

$$f : M \to [0, 1] \text{ and } \sum_{m \in M} f(m) = 1$$

For each group of players, f is different, hence different point distribution. Non-deterministic factors of the games

Table 1. Existing AI tournaments

Tournament name	Played game	Used method(s)
SSCAIT (Student Starcraft AI tournament) [9]	Starcraft game: two or more (in this case is only two) players oppose each other on a predetermined map. Each player constructs and controls buildings and units. The player win when no enemy building remains. [14] The input of the game only depends on the players and player's action lead to predictable outcomes (i.e. the outcome of build are new building), hence make the game deterministic.	For student division: Round-robin format is used. The total point of an AI player,is the percentage of its winning all matches. [9]
		Mixed division: Binary elimination is used. AI players have to win 1 game in round 1 to proceed. In round 2 and semi-final, AI players have to win 2 games, and final, the AI player has to win 3 games to win. [9]
GoMo cup [4]	Gomoku game (five in a row): deterministic game with two players. The players place their own sign on a board and win by creating a line of five or more signs	Yearly tournament: Round-robin format is used. For 1 match, the winner will get 3 points while loser get none. If the game is draw, each player will get 1 point [4]
		Overall ranking: AI players are updated and participate in many yearly tournaments. The overall ranking is determined using Elo rating system. [4]
IEEE CIG Starcraft AI competition [5]	Same with SSCAIT tournament	Round-robin format. Winner is determined by the percentage of the AI winning all matches [5]
The AI Games [1]	Tournaments with different games are held. The games are either deterministic or non-deterministic. However, all games are 1 versus 1 games.	Many 1 versus 1 games are held everyday. All of the AI games' leaderboard are determined by automated Elo rating system, adapted to for each tournament. [1]
FTG (Fighting game) AI Competition [3]	1 versus 1 fighting game. The game is deterministic as the player's action lead to predictable outcome (i.e. hitting other player lead to that other player loose health point)	Uses round robin format. The AI player is ranked according to their number of winning rounds. [3]

also affect f. Therefore, f and M are also different in different game.

M can also apply for multiplayer games where there is one winner who gets point and losers get none, in that case we have:

$$M = \{a, b, c, d \mid a, b, c, d \in N \wedge 0 \leq a, b, c, d \leq 1 \wedge a + b + c + d = 1\}$$

and probability function f now describe the probability of one player winning in that group of players.

We can also generalize M for games with different number of players other than 4:

$$M = \{a_1, a_2, ..., a_n \mid a_1, a_2, ..., a_n \in N \wedge 0 \leq a_1, a_2, ..., a_n \leq Z \wedge a_1 + a_2 + ... + a_n = Z\}$$

where n is number of players and Z is total points. The probability function stays the same. This means the scoring and ranking of champion challenge format is not only applicable for only Liar's Dice game but also many other non-deterministic games. However, the ranking depends on multiplayer Elo rating which is calculated using point distribution. As mentioned, point distribution can be affected by number of players, different games. This means the ranking might behave differently if the same experiment is carried out with a different game or a different set of players.

3 Research Questions and Hypotheses

The research questions will focus on the reliability of champion challenge format standing alone and in comparison to relative performance format.

- RQ1 - How reliable is the ranking result of champion challenge format AI programming tournament using multi-player Elo rating?
  - RQ1.1 - To what extent does the order of contestants playing affect the final ranking result?
  - RQ1.2 - To what extend does the non-deterministic factor of the game affect final rating?
  - RQ1.3 - To what extend does the amount of rounds played affect final ranking result?
- RQ2 - How reliable is the ranking result of champion challenge format AI programming tournament comparing to relative performance tournament?
  - RQ2.1 - How efficient is the AI program written by highest ranking contestant in champion challenge format comparing to relative performance format?

The main hypothesis is that champion challenge tournament gives an accurate ranking of contestants in term of programming skill, and the ranking is more accurate than relative performance tournament. For the first research question, the hypotheses reflects the sub-questions RQ1.1 and RQ1.2:

- H_{0Turns} - The final ranking results are different when the order of contestants playing changes.
- H_{1Turns} - The final ranking results are the same when the order of contestants playing changes.
- $H_{0Non-deterministic}$ - The final ratings have a huge difference when one AI program play against itself.
- $H_{1Non-deterministic}$ - The final ratings are similar when one AI program play against itself.
- $H_{0Stability}$ - The final ranking results is as stable no matter how many rounds a tournaments has
- $H_{1Stability}$ - The final ranking results is more stable the more rounds a tournament has.

The hypotheses for the second research questions are:

- $H_{0Efficiency}$ - The highest rank contestant in champion challenge format does not score higher points than relative performance format.
- $H_{1Efficiency}$ - The highest rank contestant in champion challenge format scores higher points than relative performance format.

4 Champion challenge format implementation

To make the research possible, the new champion challenge tournament format had to be implemented in Honeypot system. The Liar's dice challenge also needed to be modified to suit the new tournament format. The architecture of Honeypot [10] is presented in Figure 1. The change is done in Code Execution and Code Generation components.

The implementation of the champion challenge format involves changes in Python and JavaScript code of the Honeypot application. In Python files, changes were made to start multiple processes to listen and execute contestants' code. The processes also have a new argument telling whether the AI code is from the contestant sitting in front of the screen writing code or their opponents AI from submitted code. The argument will help the communicator code to not print out any system log from the opponents' code.

The changes in Python code also include changes in the implementation of a challenge, Liar's Dice. In the *relative performance* format, 3 AI players was written inside the implementation of the challenge by SWS. The new implementation took away the pre-written AIs and instead calls the processes that execute opponents' codes. A weak pre-written AI was kept in case there are not enough players in a game. The end score will be calculated by diving 100 points to the contestants according to how many rounds they survive in the game. Figure 2 shows that the player is playing against 2 opponents and 1 pre-written AI as there was only 2 saved solution in the database.

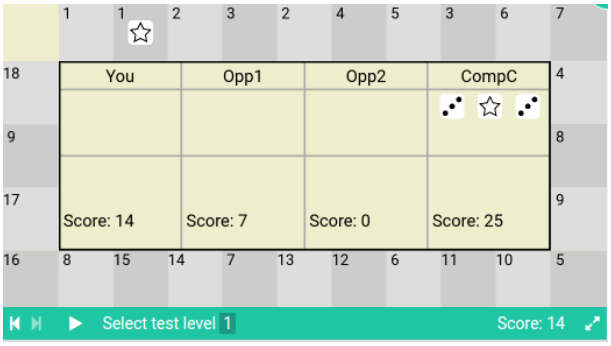

Figure 2. Game interface when a contestant play against best AIs

The tournament is run by executing a new Javascript file. To run a tournament, all submissions will be retrieved from

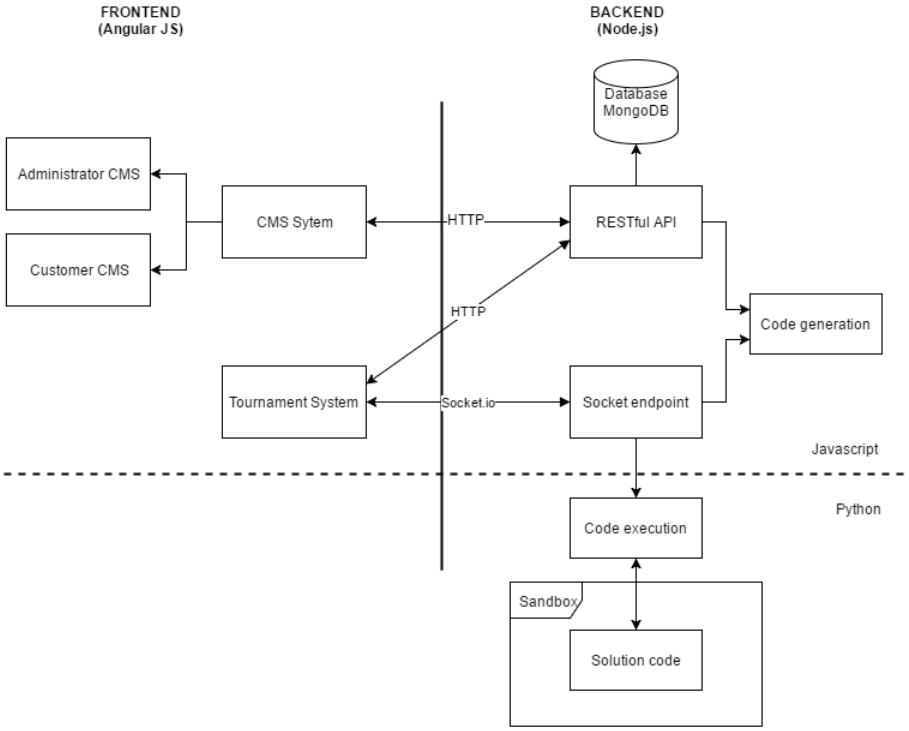

Figure 1. Architecture of Honeypot system [10]

the database and sorted by their rating. Since the Liar's dice game is for 4 players in a game, submissions then will be grouped in group of 4 and arguments to the python backend will be generated and sent. The python backend will execute the code and send back the games result. A simple interface is made to display the rating and how the rating changes (See figure 3). The players with black highlight has their rating raised after their most recent game.

The application of rating in Honeypot is based on a simplified version of the JDPR system, which itself is based on the Elo rating system. In the remainder of the paper we will refer to the rating system used for champion challenge format as the multiplayer Elo rating. V and E variable was kept constant and total points will be 100 instead of the amount of players. This system is chosen over traditional Elo rating was because the Elo rating is for a 1 versus 1 game only. In this case, a rating system can calculate rating for games that have more than 2 players is needed.

5 Research Methodology

The first step in data collection is to gather as many AI programs to run different tournaments. Relative performance format tournaments have already been held on SWS's website and permission was granted from SWS to use the submitted AI programs for the data collection step

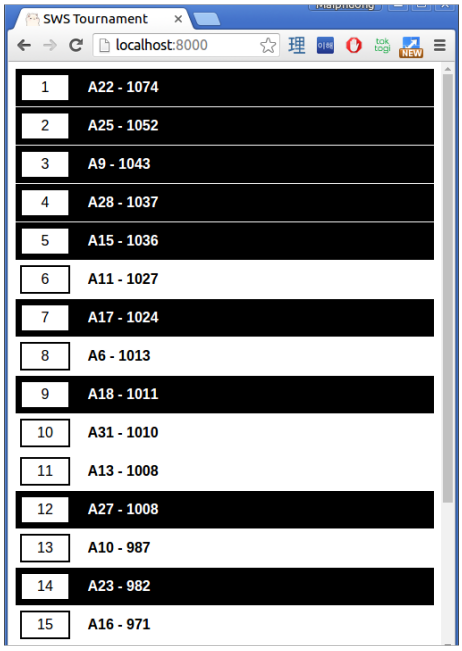

Figure 3. Simple interface of the tournament format

of the research.

The AI programs taken are the ones that play Liar's dice game. Experiments with Liar's Dice tournaments with new champion challenge format will be run to collect the necessary data.

By using the submitted AI programs, the ranks of the players in relative performance are already calculated and sorted. All applicant names will be coded according to their old rank in relative performance format. Their name will be shown in this research as A1,A2,...

To answer RQ1.1, tournaments with the new champion challenge format will be held using different start-up groups of AI players. Twenty 10-round tournaments will be executed. The players will be randomly grouped together. The final rating and ranking will be

The average rating that the AI programs get will be calculated and plotted on a graph with the error bars of standard deviation. This way, the spread of the rating and its reliability can be analysed.

To examine how non-deterministic factor of the game affect final ranking (RQ1.2) and the stability of the champion challenge ranking, two experiments will be held. For both of the experiments, tournaments will be run with the same start-up groups of AI players. Tournament will be repeated twenty times with both 10 and 50-round tournaments.

The final results of each tournament will be documented. For the last tournament with 50 rounds, the result after every 5 rounds will also be documented. Using the results recorded after every 5 rounds of 50 rounds, a line graph can be drawn. The graph can show if the score gets more stable after more rounds or not. This graph will add some visual presentation of how the score and rank change during a tournament.

The average rating and ranking for tournaments with 10 rounds and 50 rounds will be calculated and the data will be plotted the same as in RQ1.1. The analysis of the graph can show how random factor could affect the result of the game as well as if the results are more stable after 50 rounds than 10 rounds.

Comparing 10-round tournaments with both random and same start-up groups will also help the conclusion for RQ1.1

Since the data sample is quite small, bootstrapping will be used for the players' rank to measure the confidence intervals to estimate how accurate the rating is. Using sample and apply() function in R, 1000 random ranking samples were created for each player in each experiment. Each sample will be a table of rank of the players for 5 tournaments.

For each sample, mean absolute deviation of every players in every sample is calculated. This is done by using function aad() with R. This mean absolute deviation will help to evaluate how spread out the ranking is. To do the evaluation, R function quantile() is used to show the confidence interval of the mean absolute deviation of the bootstrap sample. Also, with the analysed statistic, a distribution bell plot will be made.

To answer RQ2, the AI program of the player with highest score in relative performance format was compared with the ones with highest rank in the champion challenge format. The goal of the comparison is to check if the best AI program in champion challenge format is better than the best AI in relative performance format.

If the top player of the two tournament format was different, the AI code will be compared using some metrics such as complexity, lines of code, how a move is decided. This may reveal how one AI program is effective when fighting against pre-written AI but failed at winning against other competitors' AI.

Since the applicants are coded by their old ranking, their new trend of score in the new champion challenge format will also be observed. Any AI programs that have significant difference of rank in the two format will be further examine.

6 Results and Discussion

6.1 Results

The below section describe the results from the experiments. They are divided according to the main research questions.

6.1.1 How reliable is the ranking result of champion challenge format AI programming tournament using multi-player Elo rating?

There were some problem in compiling AI programs. Therefore, only 20 out of 34 AI programs were used in the experiment. The name of the AI players that could complied can be found in Figure 4, 5, and 6.

Figure 4, 5, and 6 showed the average ranking after 20 tournaments of 10 rounds with random start-up groups, 10 rounds with same start-up groups and 50 rounds with same start-up groups. The x-axis presents each player sorted by their old rank in relative performance format start from the highest rank and y-axis presents the rank. The error bars represents standard deviation of the recorded rank. The position of average rank in all three plots look quite similar.

The plots shows that the standard deviation for both types of 10-round tournaments are quite large.

However, the standard deviation for 50-rounds tournaments is a bit smaller than the other two. Except player A28, the players had higher rank in relative performance format has rather higher rank in the champion challenge format.

Figure 7, 8, and 9 accordingly represents the average rating of each player after 20 tournaments of 10 rounds

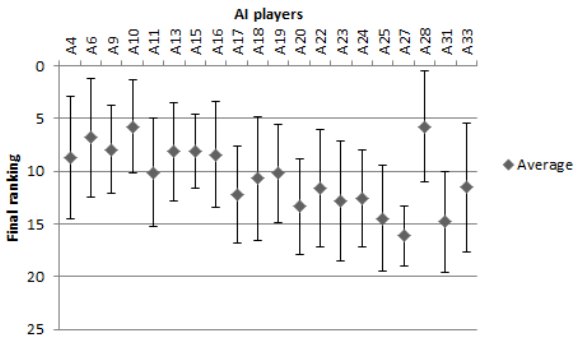

Figure 4. Final ranking after 10-round tournaments with random start-up groups

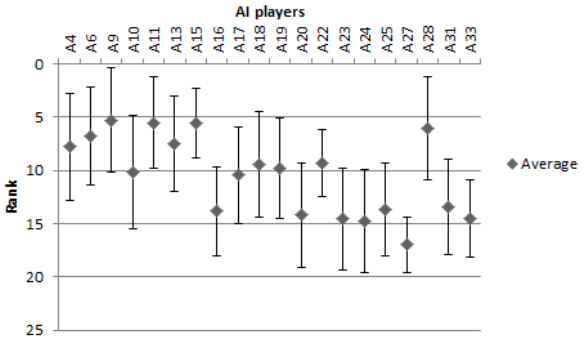

Figure 5. Final ranking after 10-round tournaments with same start-up groups

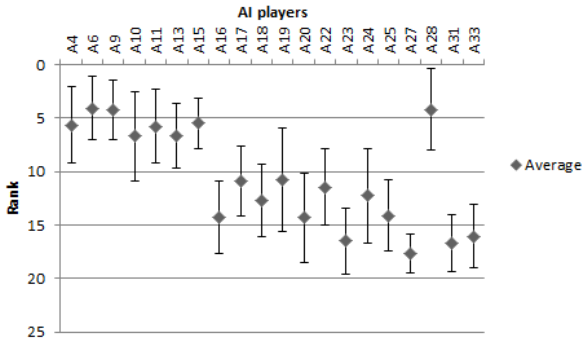

Figure 6. Final ranking after 50-round tournaments

with random start-up groups, 10 rounds with same start-up groups and 50 rounds with same start-up groups. The x-axis presents each player sorted by their old rank in relative performance format and Y-axis presents the final rating. The error bars represents standard deviation of the rating.

The standard deviation varies for different players in different type of tournament.

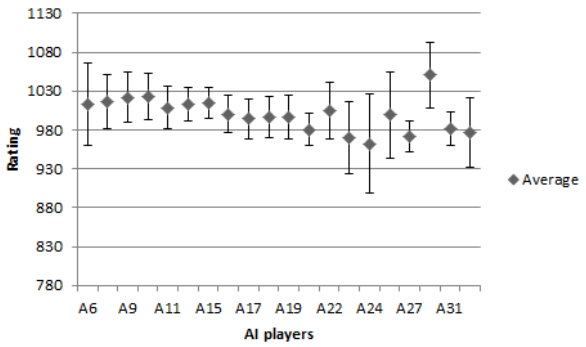

Figure 7. Final rating after 10-round tournaments with different start-up groups

However, for both types of 10-round tournaments, the average ratings of the players are quite similar as the points are almost in a straight line. Since the ratings are close and the ranking is decided by the rating, it can be the reason for the large standard deviation of the rank.

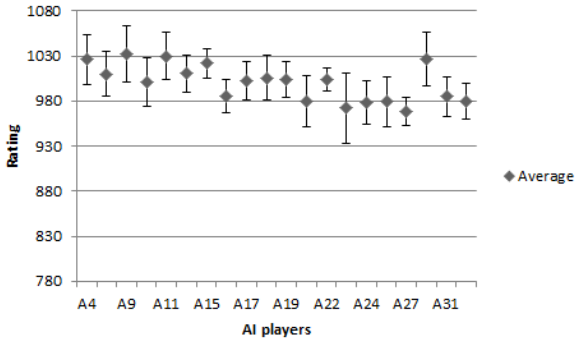

Figure 8. Final rating after 10-round tournaments with same start-up groups

Meanwhile, the average ratings of the players for 50-round tournament varies more, which can explain the small standard deviation for the average rank.

Figure 10 shows how much the rating differs during a 50-round tournament. At the beginning, every AI player starts with a rating of 1000 points. The more rounds the players play, the more spread-out the data is. Some players have quite a stable rating the longer the games go. Though, some players do not have stable rating as the tournament goes on.

The most important results in evaluating the reliability of the champion challenge format is the statistical analysis of bootstrap samples of the collected ranking data.

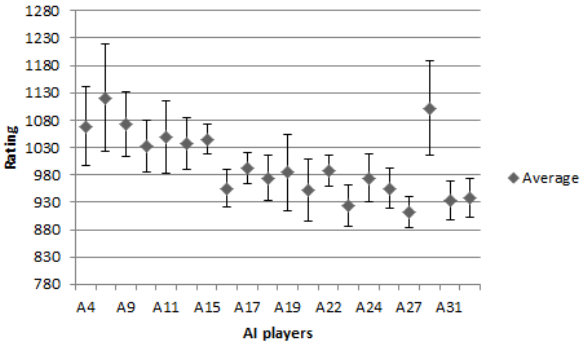

Figure 9. Final rating after 50-round tournaments

Table 3 and Figure 11 shows the distribution and 90% confidence interval of mean absolute deviation (MAD) of bootstrap samples of final ranking from 10-round tournaments with random start-up groups. The distribution curve is quite flat, the confidence interval has range of 5.36 ranks out of 20 ranks.

Table 3. Confidence interval of bootstrap samples of 10-round tournaments with random start-up groups

Quantile	5.0%	50%	95%
MAD of ranking bootstrap samples	0.72	3.20	6.08

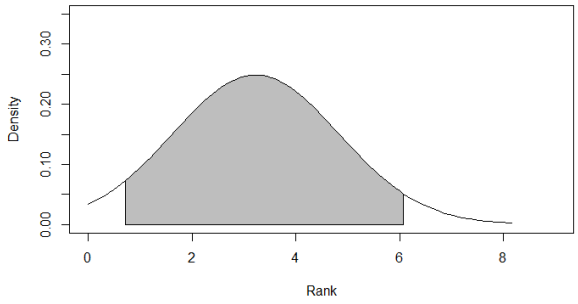

Figure 11. Distribution of mean absolute deviation of bootstrap samples of 10-round tournaments with random start-up groups

Table 4 and Figure 12 shows the distribution and 90% confidence interval of mean absolute deviation (MAD) of bootstrap samples of final ranking from 10-round tournaments with the same start-up groups. Figure 12 is quite similar to Figure 11 plot. The bell curve is quite flat and the range of 90% confidence interval is about 4.89 rank. The range of confidence interval of 10-round tournament with same start-up groups is smaller than same one with random start-up groups.

Table 4. Confidence interval of bootstrap samples of 10-round tournaments with same start-up groups

Quantile	5.0%	50%	95%
MAD of ranking bootstrap samples	0.48	2.88	5.76

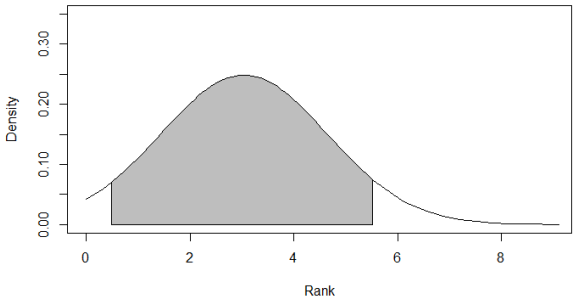

Figure 12. Distribution of mean absolute deviation of bootstrap samples of 10-round tournaments with same start-up groups

Table 5 and Figure 13 shows the distribution and 90% confidence interval of mean absolute deviation (MAD) of bootstrap samples of final ranking from 50-round tournaments. The 90% confidence interval range is 4.32 ranks. This is not so much smaller than the 10-round tournament with same start up groups. However, the distribution bell curve is much steeper than the other distributions plots.

Table 5. Confidence interval of bootstrap samples of 50-round tournament

Quantile	5.0%	50%	95%
MAD of ranking bootstrap samples	0.32	1.76	4.64

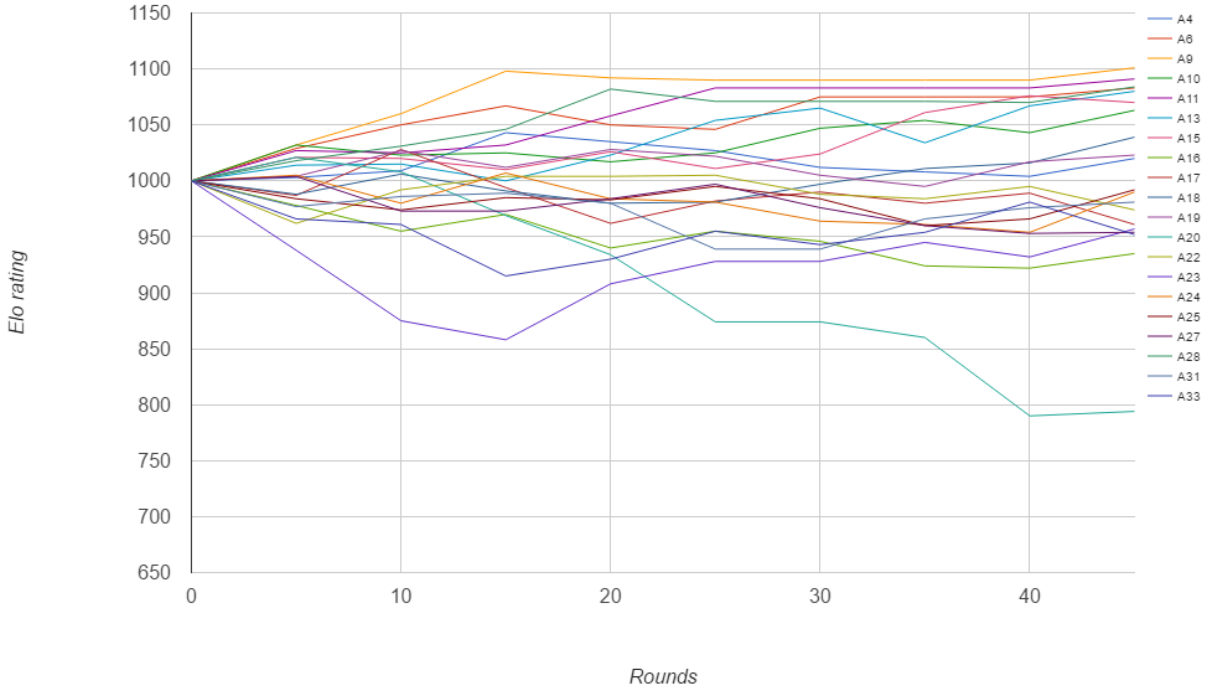

Figure 10. Rating movement in a 50-round tournament

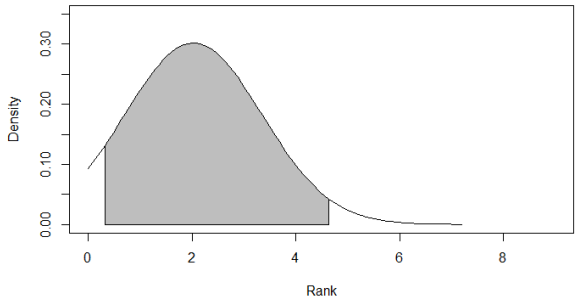

Figure 13. Distribution of mean absolute deviation of bootstrap samples of 50-round tournaments

6.1.2 RQ2.1 - How efficient is the scripts written by highest ranking contestant in champion challenge format comparing to relative performance format?

Since not all the AI players complied and worked for the experiment, it is difficult to assume a player that have good rating in the champion challenge format can be compared to the best player in relative performance format. Moreover, the recorded ranking in the experiments are not stable to pin-point the best player in champion challenge format to compared with the relative performance format.

However, there is AI player A28 has rather high rank throughout the experiments though that AI player ranked 28/34 in the relative performance format.

6.2 Discussion

For RQ1.1, the population of mean standard deviation of the bootstrap has quite a range considering the rank is only from 1 to 20. Most of the rank difference is about 3 to 4 ranks. Hence, we fail to reject H_{0Turns}. However, the results for 10-round tournament for both random and same start-up groups are quite similar. This suggests that the random start-up groups may not be the reason for large standard deviation but the non-deterministic factor of the game.

For RQ1.2, the ranking of the players seems to be not so stable with the same start-up groups in 10-rounds tournaments. The spread of the mean deviation of bootstraps data is quite large.

The ranking of the 50-round tournaments seems to have smaller standard deviation, the distribution of mean standard deviation of bootstrap data is also more compact. Though, the confidence interval of the MAD of the boot-

strap data is still quite large: 4.32 out of 20 ranks. We can not reject $H_{0Non-deterministic}$ that the ranking are similar if the tournaments are run with the same start-up groups. This mean the non-deterministic factors of the game can affect the final ranking.

The distribution of mean absolute deviation of bootstrap samples of 50-rounds tournaments are smaller than 10-round tournaments. Figure 10 also suggests the ranking gets more stable in 50 round. However, difference in confidence interval is too small to have a definite conclusion. Therefore, we fail to reject $H_{0Stability}$.

Even though the ranking of the AI programs in all the experiments are not stable, we can argue that the ranking is affected by the players' rating. Since the ratings are relatively close together, it can lead to unstable ranking.

For the overall reliability of the new champion challenge format, we do not have enough evidence to conclude it's reliability. More researches needs to be performed and data collected in order to prove the new format's reliability.

Same with RQ1, there is not enough evidence to conclude anything for RQ2 as the top AI player did not take part in any tournament due to compilation issue. Hence, we fail to reject $H_{0Efficiency}$. However, the case of player A28 suggests that there is AI player that does not do well fighting versus pre-written AI but did well fighting other AI.

6.2.1 Limitations of the research

First of all, the implementation of the new champion challenge tournament format is still not complete. In figure 11, the straight line in ranking of some player did not always indicate that the AI player keeps its rating. It was due to failure in compiling code for that certain group of players that make them keeps their ranking and be on top for many rounds. Their rating only changed when another AI program knocked down one of the top AI programs and fight against them instead.

Secondly, as mentioned in result and discussion part, some AI players could not compile and be part of the test. Unfortunately, the top 3 players are among them and not having the top 3 player fight in the new champion challenge format could affect how the data turn out.

Moreover, sample size is small as there are only 20 AI programs that are in the experiment. It is harder to see much of a clear difference in ranking compared to having 100 contestants. Also, the experiment was done only with one game which can be biased.

Elo rating supposes to be more stable the more games are played. The larger confidence intervals and standard deviations for the 10-round tournaments compared to the 50-round one could be result of too little rounds were played.

However, the factor that affect the large confidence intervals and standard deviations is the small sample size. It takes a lot of time to run a tournament, the more rounds a tournament has, the longer the time taken. Therefore, only limited amount of data was collected hence the small data sample.

6.2.2 Technical limitations

The time spent on implementing champion challenge format in Honeypot system exceeded what was planned in the beginning of the research. It was both due to technical depth of the researcher as well as some technical problem.

First of all, Honeypot needs preparation to run, such as, to have running NodeJS, MongoDB, as well as Linux or OS X operating system. It is also a big system with a previous thesis as the only documentation of its workings. Therefore, it took some time to understand the system before implementation process could be started. Honeypot is implemented in Python and Javascript, which created additional challenges in the implementation process.

There were also technical problems with Honeypot's code submitting process where the level did not get initialized and scored. Furthermore, a problem with missing data in the database in the start-up process took time and hindered the development process.

The code compiler made the testing process cumbersome. It can process up to five programming languages. This was limited to two on my system to increase the development speed of the research software. Code written in Javascript, C++, and C# all had problems executing.

There was some problem with the graphical part of the system. To change the Liar's dice challenge from the original *relative performance* to champion challenge, the code and configuration of the challenge got duplicated into the developer's local computer but failed to run properly due to some graphical errors which was not included in the change of source code that was done for the new champion challenge.

6.3 Future work

Since the top 3 AI players in the relative performance tournament cannot execute properly with the new champion challenge code, it would be very interesting to see how they fight against other AI. Therefore, making the system works for Javascript, C++, and C# will be a good plan for the future.

Also one of the problem mentioned is that if a group players' code did not work together, their ranking stays constant. It is a big problem, especially if the group are the top players. It takes time for other players to get to the top and get the game complied. This problem could be solved in the future by using proper Swiss-system tournament where players should not faced each other twice. For a game of 4

players, it may not be possible but avoiding facing the same players too many times would be a good practice.

Experiments run with more rounds per tournament will be a good way to evaluate the stability of the new champion challenge format.

7 Conclusion

The purpose of this study is to evaluate the reliability of a tournament format called champion challenge. It is a tournament where AI players will be grouped by their rating to fight. The outcomes are new ratings of the players based on JDPR system, or multilayer Elo rating. The study also aims to compare the differences in ranking between the old relative performance format and champion challenge format.

To evaluate the reliability, tournaments were run using different metrics to compare: 10-round tournaments with different start-up groups, 10-round and 50-round tournaments with same start-up groups. The final ranking and rating are recorded and processed for the evaluation.

There are still lack of evidence to prove the reliability of the new tournament format. One conclusion can be made is that the non-deterministic factor of the game can affect the final ranking. There are also some interesting differences in ranking of the new format compared to the old format. Technical limitations lead to lack of time to collect data, hence lack of evidence. The small amount of players to test as well as using only one game to experiment are also the limitation of this research.

The prototype of the champion challenge format can help Software Skills in further development of a new AI tournament format where contestants' AI programs can fight against each other in a game. The ranking of the champion challenge format can be applicable for other non-deterministic multiplayer games. This new format will also help SWS in making more AI games without having to spend time writing AI opponents to match versus contestant AIs. This will certainly save the company time and also encourage the players to fight to the top.

Acknowledgment

I would like to express my deepest gratitude to Henrik Enström at Software Skills, who has been the commissioner of this product. It is thanks to him that I have been successfully able to do this research. A special thanks to Luuk van Egeraat, a developer at Software Skills and a classmate, who helped me with the development of the product.

Furthermore, I would also like to acknowledge the post-doctoral researcher from Chalmers University of Technology, Michal Palka, who helped me with planning the research, understanding the system and providing feedback to improve the research.

References

[1] The ai games. http://theaigames.com/. Accessed: 2016-03-22.

[2] Elo inspired diplomacy rating system. http://diplom.org/Zine/S1998R/Nichols/ratings2.html. Accessed: 2016-04-30.

[3] Ftg ai competition. http://www.ice.ci.ritsumei.ac.jp/ ftgaic/. Accessed: 2016-03-22.

[4] Gomocup. http://gomocup.org/. Accessed: 2016-03-22.

[5] Ieee cig starcraft ai competition. http://cilab.sejong.ac.kr/sc$_{c}ompetition/$. $Accessed$: $2016 - 03 - 22$.

[6] Judge diplomacy player ratings. http://diplom.org/Email/Ratings/JDPR/describe.html. Accessed: 2016-05-14.

[7] Liar's dice. https://boardgamegeek.com/boardgame/45/liars-dice. Accessed: 2016-04-22.

[8] Software skills homepage. https://softwareskills.se/. Accessed: 2016-03-22.

[9] Student starcraft ai tournament. http://sscaitournament.com/. Accessed: 2016-03-22.

[10] S. Bellevik, T. Tikka, D. Bäckström, M. Olsson, J. Petersson, and N. Alexandersson. Investigation of the feasibility and development of a user-friendly platform for creating and hosting programming contests as a recruitment aid for software companies, 2015. 65.

[11] L. Csato. Ranking by pairwise comparisons for swiss-system tournaments. *Central European Journal of Operations Research*, 21(4):783–803, 12 2013. Copyright - Springer-Verlag Berlin Heidelberg 2013; Document feature - ; Tables; Equations; Last updated - 2014-08-30.

[12] A. Elo. *The rating of chessplayers, past and present*, volume 3. Batsford, 1978.

[13] L. M. Frank Harary. The theory of round robin tournaments. *The American Mathematical Monthly*, 73(3):231–246, 1966.

[14] N. Othman, J. Decraene, W. Cai, N. Hu, M. Y. H. Low, and A. Gouaillard. Simulation-based optimization of starcraft tactical ai through evolutionary computation. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 394–401, Sept 2012.

[15] S. Rosen. Prizes and incentives in elimination tournaments. *The American Economic Review*, 76(4):701–715, 1986.

[16] N. Veček, M. Črepinšek, M. Mernik, and D. Hrnčič. A comparison between different chess rating systems for ranking evolutionary algorithms. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 511–518, Sept 2014.

[17] E. Yucesan. A tournament framework for the ranking and selection problem. In *2007 Winter Simulation Conference*, pages 297–302, 2007.

[18] S. Ólafsson. Weighted matching in chess tournaments. *The Journal of the Operational Research Society*, 41:17–24, 1990.