UNIVERSITY *of York*

This is a repository copy of *Player Preference and Style in a Leading Mobile Card Game*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/114017/

Version: Accepted Version

Article:

Cowling, Peter I orcid.org/0000-0003-1310-6683, Devlin, Sam orcid.org/0000-0002-7769-3090, Powley, E. et al. (2 more authors) (2014) Player Preference and Style in a Leading Mobile Card Game. Computational Intelligence and AI in Games, IEEE Transactions on. pp. 233-242. ISSN: 1943-068X

https://doi.org/10.1109/TCIAIG.2014.2357174

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Player Preference and Style in a Leading Mobile Card Game

Peter I. Cowling, *Member, IEEE,* Sam Devlin, Edward J. Powley, Daniel Whitehouse, *Member, IEEE,* and Jeff Rollason

***Abstract*—Tuning game difficulty prior to release requires careful consideration. Players can quickly lose interest in a game if it is too hard or too easy. Assessing how players will cope prior to release is often inaccurate. However, modern games can now collect sufficient data to perform large scale analysis post-deployment and update the product based on these insights.**

AI Factory Spades is currently the top rated Spades game in the Google Play store. In collaboration with the developers, we have collected gameplay data from 27 592 games and statistics regarding wins/losses for 99 866 games using Google Analytics. Using the data collected, this study analyses the difficulty and behaviour of an Information Set Monte Carlo Tree Search player we developed and deployed in the game previously [1].

The methods of data collection and analysis presented in this study are generally applicable. The same workflow could be used to analyse the difficulty and typical player or opponent behaviour in any game. Furthermore, addressing issues of difficulty or non-human-like opponents post-deployment can positively affect player retention.

***Index Terms*—Game Analytics, Data Mining, Artificial Intelligence, Monte Carlo Tree Search**

I. Introduction

When deploying an Artificial Intelligence (AI) in a game, it is challenging to balance the difficulty and the fun suitably to maximise players' enjoyment. If the AI is too difficult players will give up and leave the game, too easy and the players will become bored and quit the game. Somewhere between these two extremes is a compromise that is essential for successful games [2]. Player enjoyment is particularly important in mobile games, where player acquisition is driven largely by user-submitted ratings and word-of-mouth: a few negative reviews can significantly damage a mobile game's download figures.

Play testing before deployment can reduce the chances of either of the extreme cases occurring, but cannot guarantee they will not occur. Therefore, analysing player data from games after the AI was deployed is essential to understand how the game's users are coping. Subsequently tweaking the settings and parameters of the AI with regard to this analysis can positively impact player retention.

Manuscript received Xxxxx XX, 20xx; revised Xxxxx XX, 20xx.

P.I. Cowling, S. Devlin and D. Whitehouse are with the Department of Computer Science and York Centre for Complex Systems Analysis, University of York, UK. E.J. Powley is with Orange Helicopter Games, York, UK. J. Rollason is with AI Factory Ltd., Pinner, Middlesex, UK. E-mail: peter.cowling@york.ac.uk, sam.devlin@york.ac.uk, ed@orangehelicopter.com, dw830@york.ac.uk, jeff.rollason@ntlworld.com.

This work is funded by grants EP/H049061/1 and EP/K039857/1 of the UK Engineering and Physical Sciences Research Council (EPSRC).

DOI:

Furthermore, if an AI player performs behaviours distinctively uncharacteristic of human players, gamers will notice and their enjoyment of the game will be affected. This occurs often when AI players believe a number of actions have the same value and, therefore, see no reason to perform a specific one despite tendencies that are expected by humans. For example, if an AI can assess far earlier than a player that it has lost the game, it may behave randomly as it knows all moves lead to a loss. To the human player, however, this appears irrational as they are not aware they have already won the game. Therefore, collecting data from real users of a game and using this to influence the design and behaviour of the AI can increase players' enjoyment by making AI-controlled opponents and partners more engaging and more human-like.

Previously, we deployed an AI in the game AI Factory Spades [1]. The AI has been very popular but questions have been raised at times regarding occasional non-human-like behaviour. AI Factory have addressed these complaints with some success by adding hand-designed and hand-tuned heuristic knowledge to the AI. However this approach is ad-hoc and relies on expert knowledge of Spades. To investigate whether a more systematic approach is possible, we collected a significant amount of game data to explore the differences in strategies employed by real users of the game and the AI we had implemented. This data also highlighted potential issues regarding the default difficulty of the game.

The contributions of this paper include specific recommendations for future revisions of AI Factory Spades and insights into good strategies of play for any instance of the classic card game Spades. More generally, our intention is to present a detailed analysis of game data from an active and popular commercial product so that others may be able to recreate this study for their own games. The methods of data collection we have used can be used with any other mobile games and the analysis techniques could be used more widely on any game given access to similar data.

The remainder of this paper is organised as follows. In Section II, we detail the relevant background material. Then in Section III, we discuss the implementation details of collecting data from the game. Section IV covers the analysis of AI difficulty and human/AI play styles. Finally the paper concludes in Section V.

II. Background

This section covers all necessary background material, starting with details of the algorithm Information Set Monte Carlo Tree Search used in the AI we have deployed in Section II-A.

Section II-B explains the rules and details of Spades. Section II-C discusses the AI Factory Spades implementation and our previous collaboration. Finally, Section II-D discusses the emerging area of game analytics and work related to this study.

A. Information Set Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is a family of game tree search algorithms invented in 2006 [3], [4], [5] and notable for successes in Go [6], [7], General Game Playing [8] and many other domains [9]. MCTS combines the precision of tree search with the generality of random simulation, providing reasonable play in the absence of game-specific heuristics and strong (in some cases world champion level) play when enhanced with some domain knowledge.

A game has *imperfect information* if the state is only partially observable, i.e. if there is some aspect of hidden information. Many games of imperfect information have *information asymmetry*: different players can observe different parts of the state. For example in a card game such as Spades, a player can observe her own cards in hand but not the cards held by the other players. Games of imperfect information are particularly challenging targets for AI: the inability to predict the exact outcome of a sequence of moves is a problem for many tree search methods.

Information Set Monte Carlo Tree Search (ISMCTS) is a variant of MCTS for games of imperfect information [10]. ISMCTS is based on the idea of *determinization*: sampling from the set of possible game states that are consistent with current observations. In ISMCTS, each iteration uses a different determinization, with the result that the statistics collected in the search tree are averaged over many determinizations. ISMCTS performs well in several games of imperfect information [10], [11], [12], [1].

B. Spades

Spades is a four-player trick-taking card game, especially popular in the USA but played worldwide [13]. Spades is a partnership game, with North and South in coalition against East and West. (It is common to name the four players after the compass points.) Spades has some similarities with, but somewhat simpler rules than, the game of Bridge.

A game of Spades consists of several *rounds*. A round begins with each player being dealt a hand of 13 cards from a standard deck. The players must then *bid* on how many *tricks* they expect to take. A trick consists of each player in turn playing a single card from their hand. The *leader* plays first, and the other players must *follow suit* if they are able, i.e. they must play cards of the same suit as the leader's card. If the player has a *void* suit, i.e. no cards of the required suit, then they may play any other card from their hand. Once every player has played a card, the trick is won by the player who played the highest ranking card of the same suit as the leader's card. The exception are ♠ cards, which are *trumps*: if any ♠ cards are played in a trick, then the highest ranked ♠ card wins the trick regardless of the suit of the leader's card. Furthermore, the leader cannot lead with a ♠ until they have been *broken* by a player in an earlier trick within the same round playing one due to having a void suit.

Each partnership's goal is to win a total number of tricks equalling their total bid. If the partnership wins at least that many tricks, they receive 10 times their total bid in points; if not, they lose that many points. For each trick over their total bid, the partnership receives a single point and a *bag*: for every 10 bags the partnership accumulates, they lose 100 points. A bid of 0, or *nil*, is treated differently: in this case the player himself is aiming to win no tricks, and his partnership gains or loses 100 points depending on whether he is successful or not. The game ends when either partnership exceeds 500 points, at which point the highest-scoring partners win.

C. AI Factory Spades

AI Factory[1] is a UK-based independent game developer, currently specialising in implementations of classic board and card games for Android mobile devices. AI Factory's implementation of Spades has been downloaded more than 2.5 million times, with an average review score of $4.5/5$ from more than $90\,000$ ratings on the Google Play store[2].

AI Factory Spades is a single-player game, in which the user plays with an AI partner against two AI opponents. The user always plays as South. The user may choose the partner and opponents from several AI characters. Some of the parameters in the AI's heuristics are influenced by the choice of character (for example some characters are "cautious" while others are "aggressive"), but the main feature of the character profile is a level rating from 1 to 5 stars which determines the number of simulations used by the AI player.

There are many variations on the rules of Spades: for example a target score other than 500 can be used, or players may be allowed to pass cards to their partners after bidding nil. AI Factory Spades supports these and other variations.

In previous work [1] we collaborated with AI Factory to implement ISMCTS-based AI players for Spades. We found that our knowledge-free ISMCTS player was objectively stronger than AI Factory's knowledge-based player, but the non-human-like and sometimes counterintuitive playing style of the ISMCTS player caused beta testers to perceive the player as weak. It was necessary to inject heuristic knowledge into the ISMCTS player to produce play that was perceived as strong, even though this knowledge had no measurable impact on win rate. However subsequent refinement of the heuristics has produced an increase in playing strength, resulting in an ISMCTS player that is objectively and subjectively stronger than the previous AI.

D. Game Analytics

The emerging field of game analytics has been growing rapidly as evident from its coverage in *Science* [14] and the publication of the first textbook specifically on this topic [15]. There have also recently been multiple startups dedicated to

[1]http://www.aifactory.co.uk

[2]https://play.google.com/store/apps/details?id=uk.co.aifactory.spadesfree&hl=en

providing game analytics as a service and a wealth of job advertisements requesting data scientists at large AAA games companies. Futhermore, a recent review of game AI [16] identified large scale game data mining and gameplay-based player experience modelling as key areas for ongoing research.

In particular, a previous study of Starcraft [17] also considered game data to advise the design of AI players by modelling human strategies. This work supports our argument that after the release of a game, it is beneficial to collect player data to create better AI. However, the authors of this previous study intended to use the derived model to make harder opponents whilst our aim is to balance an already deployed AI, making the game more enjoyable and the opponents more human-like.

Another study [18] used video replays to learn an AI that imitates human-like behaviour in Quake 2 by combining neural networks and self organizing maps. Our approach differs from this, as we are interested in tweaking an existing AI not developing a new one. Neural networks and self organizing maps can create sophisticated AI, as demonstrated by this previous study, but the resultant model is not easily human readable. The analysis and workflow presented here is focussed on giving insight into how humans play the game and where the AI conforms to or deviates from these patterns.

Data collected from Madden NFL 2011 [19] was mined previously to predict features of a game that would maximise player retention. This study made various conclusions related to alterations of game mechanics such as reducing the number of options available to players in game and presenting the controls more clearly. Unlike our study, they did not explore the effect of AI behaviour and difficulty as a method of increasing player retention.

The Madden NFL 2011 study also concluded that, whilst the recommendations they presented were specific to the game analysed, their workflow of analysis was domain independent and, therefore, generally applicable to other games. We share this motivation and believe that the methods of data collection and analysis presented in the following sections could be used to increase player retention in many other games.

III. Data Collection

AI Factory Spades was already a mature released product before data collection was implemented, so it was essential that the new functionality did not disrupt the existing game experience or require too much re-engineering. In particular, AI Factory were keen only to report information that the game already tracked and stored locally. Additionally, the data is collected via Google Analytics, which places restrictions on the amount of data that can be reported both in terms of the number of bytes in an individual “event” and the number of events per day. Thus it was important both to find a compact data representation and to limit the volume of data collected. The data is exported from Google Analytics as a CSV (comma-separated values) file, so the data representation is limited to printable ASCII characters.

The first time the game is run, it generates a random 32-bit identifier. This provides a way to identify the player while still retaining anonymity. To reduce the volume of data collected, games are only reported from player identifiers where the lowest five bits are all zero, thus for only $\frac{1}{32}$ of players. AI Factory have the ability to tune this proportion at will by altering the bit-mask applied to the identifier. The only piece of demographic information collected for the player is their country, which is not used in the present study.

The game keeps track of statistics about the player’s past performance, namely the number of games they have won and lost with each AI character as a partner and against each as an opponent. The combination of AI characters for a game gives a *level* between 2 and 14: if the star ratings for the opponents are r_w and r_e and for the partner is r_n, then the level is $r_w + r_e + (5 - r_n)$. The default setting is $r_w = r_e = r_n = 5$, which corresponds to level 10. It is possible to achieve the same level with different choices of characters, however the level gives us a single number to describe the game difficulty that arises from the strength of the opponents and/or the weakness of the partner. The game tracks the player’s wins and losses at each level. This information was already tracked locally and accessible for the player to view from the game’s main menu.

The events reported to the analytics server are completed games. A game of Spades consists of several rounds and generally takes between 10 and 60 minutes to play. Reporting only completed games helps to reduce noise in the data; for example, users who try the game once and decide they do not like it (or do not know how to play) are unlikely to play a game to completion because a full game of Spades takes a significant amount of time. However, reporting only completed games may introduce some selection bias towards games where the human player wins or loses by only a narrow margin, as players may decide to abandon the current game and start a new one if they begin to fall behind.

Upon completion of a game, the following information is sent to the analytics server:

- The player’s anonymised identifier and country;
- Historical win and loss counts for each game level and for each AI character;
- The version number of the game;
- The random seed used for this game (the same pseudo-random number generator is used for card deals and for ISMCTS simulations);
- Parameters for the chosen AI players;
- Rule settings for this game;
- The final score;
- The sequence of bidding and trick play moves.

An example of a game record is shown in Figure 1. Note that the historical statistics are sent along with every game. This increases the size of a game record, but has the advantage that player statistics do not need to be reported separately from game records.

For the sequence of moves in the game, each of the 52 cards is represented by a single alphanumeric character. It would be possible to devise a more compact encoding than this, but the game records currently are typically smaller than 2kb (well within the limits of Google Analytics event data) and so there is nothing to gain from making the representation more compact and thus harder to parse.

U:-1498973648# C:US#	Player ID -1498973648 Player country (USA)
L8:20-8 L10:55-50#	Player has won 20 games and lost 8 at level 8 Player has won 55 games and lost 50 at level 10
Don:26-40-0-0 Mary:24-15-1-2 ···	Player has won 26 games and lost 40 against AI character Don, but never played with Don as a partner Player has won 24 games and lost 15 against AI character Mary, and won 1 and lost 2 with Mary as a partner More AI character statistics
V:0H9# M:15265014# L:NA,28,28,28#Y:NA,0,0,0#T:NA,1,1,1# PT:500#TB:100# ···	AI Factory Spades version number Random seed AI settings(Internal Difficulty Rating, Style, Algorithm) Game rule settings (score limit, ten bag penalty, etc.)
S1:517#S2:367# WB:5NB:2EB:3SB:2# W:d148c2wa ··· Sqgp# NB:2EB:3SB:6WB:2# N:nflprhgk ··· xm9N# ···	North/South won with 517 points to 367 Bids for the first round: West=5, North=2, East=3, South=2 Card play for the first round, with West leading the first trick (d = $A\heartsuit$, 1 = $2\heartsuit$, 4 = $5\heartsuit$, etc.) Bids for the second round Card play for the second round, with North leading the first trick (n = $10\diamondsuit$, f = $2\diamondsuit$, l = $8\diamondsuit$, etc.) More rounds

Fig. 1. Example of a game record, with explanation of each field. Line breaks and ellipses have been added for readability; in the actual data, the text in the left column forms one continuous string.

The results reported in this paper are based on data collected between 1st April 2013 and 12th November 2013. The data contains 27 592 complete games, and win/loss statistics for 99 866 historical games, played by 690 unique players.

IV. Data analysis

Using the data collected, we present two analytical studies. The first explores the difficulty of the game and the difficulty settings chosen by players, whilst the second looks into understanding how people play the game at the level of individual moves. Both of these studies give insight into the design of the game and provide important feedback for possible future AI card game improvements.

A. Difficulty and Play Level

Figure 2 shows, for each AI level, the number of players who played at least one game at that level. By default, the game offers an assignment of partner and opponents equal to an AI level of 10. We note that 45.5% of players always changed from the default and completed no games at this level. The slight majority, however, are content to try this difficulty for at least the duration of a full game. Therefore, careful consideration should be taken when implementing the default difficulty to ensure first time players are not immediately thrown into a game they cannot compete in, but also that more experienced Spades players do not get the initial impression that the AI is weak.

Figure 3 shows the total number of games played at each AI level. We see that 59.8% of all games are played at the default level. More games are played at the default level than at all other levels combined, which again emphasises the importance of carefully tuning the default difficulty.

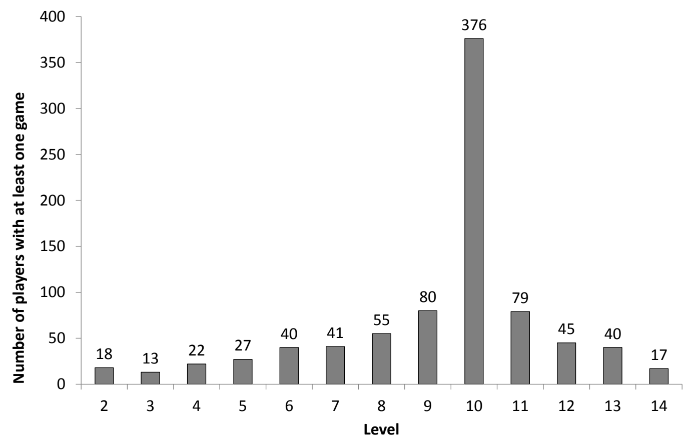


Fig. 2. Number of players with at least one game at each AI level.

We also see a disproportionately large number of games played at the easiest difficulty level. Five of the 690 players have played more than 180 games each at this level, with one player having logged 1360 games at level 2 and only two games at any other level. Presumably these players enjoy the satisfaction of crushing the weakest AI opponents. However these players are the minority, with most players opting for a more challenging game.

To explore whether the default level for this game was too high, we did a significance test for each player using their games at each level to determine whether they were significantly better, worse or no different on average than the AI. We consider a game at a particular level to be a Bernoulli trial, and take the player's number of wins and total number of games at that level as numbers of successes and trials. We use these to compute a Clopper-Pearson interval [20] at the

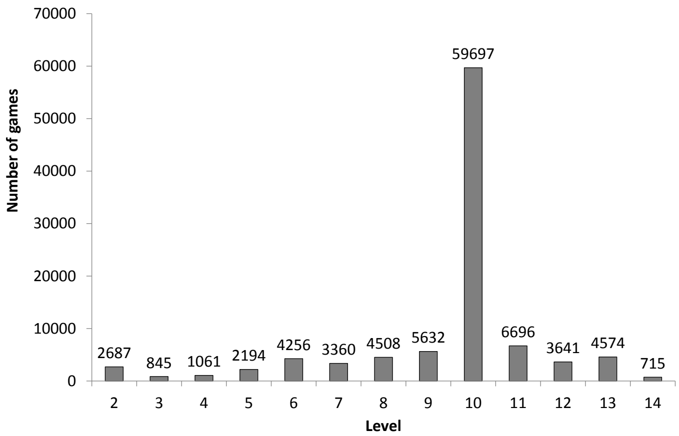


Fig. 3. Total number of games played at each AI level.

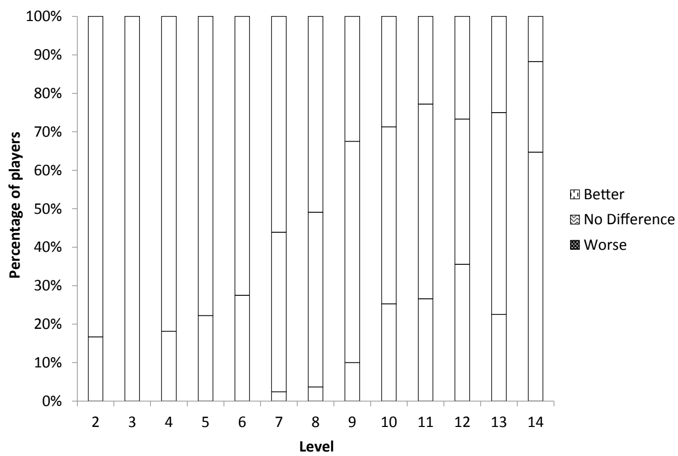


Fig. 4. Comparison of player performance at each AI level, classifying players based on confidence intervals on their win rates. Each bar shows the number of players classified as better than, no different from and worse than the AI opponents at that level, as a proportion of the number of players with at least one game at this level (see Figure 2).

95% confidence level. If the lower bound is greater than 0.5, we categorise the player as **better** than the opponents at that level; if the upper bound is lower than 0.5 we say the player is **worse**; otherwise we conclude there is **no difference**. Figure 4 shows the proportion of players who fall into each category for each level. This approach assumes that the skill of a player in a specific level does not change over time. By averaging across all players in a short time window, the effects of player learning are unlikely to be significant.

At level 10, 25.2% of players are worse than the AI. Therefore, the vast majority of players will have either a competitive or easy game with the default AI. Provided it is obvious that the game can be made harder by reducing the ability of your partner, this should not cause players to think the game is too easy. However, for those players that are worse than the default AI, losing their first few games could put them off. Figure 5 suggests this occurs often, as there is a positive correlation between win rate and average games per player and a high number of players in the low win rate bin. Given this data, perhaps reducing the default AI level may be a worthwhile update. Aternatively, the game could assess the

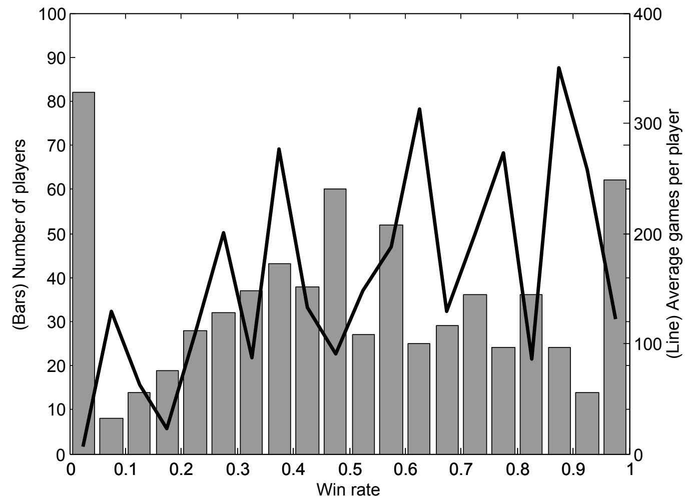


Fig. 5. Comparing win rate of players.

skill level of the player and dynamically adjust the “default” difficulty accordingly. Further work on player modelling and difficulty adjustment is surveyed in [21].

Furthermore, with due consideration for noise in the data, it appears that the correspondence between level and win rate (i.e. difficulty) is somewhat linear. There is probably some bias given that better/worse players might select easier/hard difficulties and skew the results, but that would have a “flattening” effect if the bias occurred in the logical direction. Therefore, we conclude the harder difficulty levels are legitimately harder, even for better players. This also validates the use of level as a single number to estimate the difficulty of the game, since the difficulty scales linearly with level despite there being many different AI character combinations possible at each level. It is still possible that the difficulty of different choices of AI characters at the same level is not the same, but the level provides a good indicator of the relative difficulty of two set-ups provide the difference is sufficiently large.

Next, we filtered the players to those who have played at more than one level, are in the “no difference” category for at least one level, and either “better” or “worse” for at least one other level. That is, we consider only players who are able to make a free choice whether or not they play at a “no difference” level. For each of these 72 players we looked at the level of their most recently played game. The majority (47, or 65.3%) chose a level at which they were in the “no difference” category, with 18 and 7 players choosing “better” or “worse” respectively. This suggests that most players, given the choice, prefer to play at a level where they are evenly matched with the AI. The remainder of players tend to select a level where they can comfortably win, although a small number of players opt instead for a challenge beyond their current abilities.

Figure 5 shows a histogram of win rates for all players, as well as the average number of games played by the players in each bin of the histogram. The histogram suggests a “normal-like” distribution with mean around 40–50%, although spikes at both ends of the distribution show a large number of players with win rate less than 5% or greater than 95%. The average number of games played by the players in the less than 5%

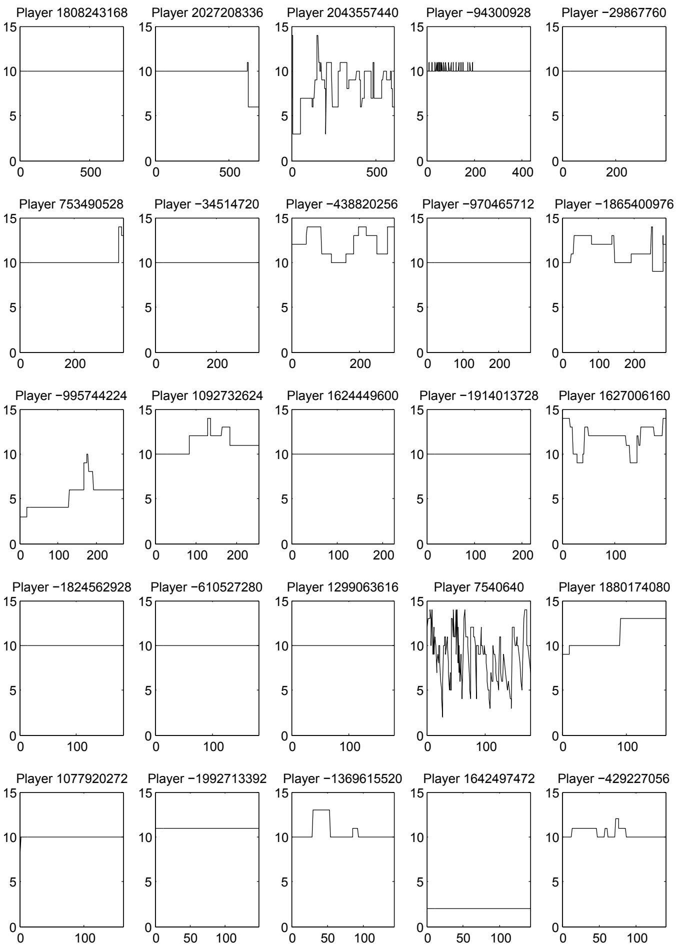


Fig. 6. Change in game level (y-axis) over games played (x-axis) for individual players.

category is low, indicating that retention of these players is particularly poor. This may be due to players losing their first few games and then quitting due to frustration, or it may be due to players unfamiliar with Spades downloading the game, trying it once and then deleting it.

The average number of games is much higher, although still relatively low, for players with greater than 95% win rate. These players are firmly in the "better" category of Figure 4, but the average number of games suggests that winning almost every game is not a source of frustration (at least for some players). We see a general upward trend of average number of games with respect to win rate. This has three possible explanations: players who consistently lose games are more likely to get frustrated and stop playing; players become more skilled the more they play; or players choose AI opponents where they win more than they lose. The real situation is likely to be a combination of all these reasons.

Figure 6 shows the levels of games played by individual players. The players in question are the top 25 with respect to number of games played over the data collection period. We see that 10 of these players only ever play at the default level 10, with a further two (players 1992713392, fifth row second column, and 1642497472, fifth row fourth column) playing at a constant level other than the default (levels 11 and 2 respectively). Of the remaining players, only three (players 2043557440, first row third column, -995744224, third row first column, and 1880174080, fourth row fifth column) show a clear upward trend in level over the data collection period, with the other players seeming to switch between levels more freely.

The analysis in this section shows that a high proportion of players play at the default level. There is evidence that the default level is too challenging for some players, causing them to stop playing after losing their first few games. This suggests that adjusting the default level may improve player retention. Amongst players who play the game over a longer period of time, some are content to stick with the default level whilst others switch frequently between levels. Those who try different AI characters generally settle on a level at which they are equal to or slightly better than the AI opponents.

B. Understanding Human Play Style

To understand human play from the data, it was important to group moves based on the effect they have in a trick. Playing 10♠ has a significantly different effect in a trick where it is the highest ♠ still in play than in a trick where A♠ is in play.

Therefore, we assigned all moves into the following, mutually exclusive and exhaustive, categories of abstract moves:

- **Follow Steal:** follow suit, and play a card of higher rank than the current highest card in the trick;
- **Follow Duck:** follow suit, and play a card of lower rank than the current highest card in the trick. If a trump card has been played in this trick, all cards in the trick suit are considered to be in this category;
- **No Follow:** fail to follow suit, instead playing a card of a non-trump suit (♡, ♢ or ♣);
- **Trump Steal:** fail to follow suit, instead playing a trump card (♠). The card is either the first trump to be played in this trick, or is a higher rank than the current highest trump card in the trick;
- **Trump Duck:** fail to follow suit, instead playing a trump card (♠) that is lower than the current highest trump card;
- **Lead ♠:** begin a trick with a trump card;
- **Lead ♡/♢/♣:** begin a trick with a non-trump card.

Figure 7 show the frequency with which each category is played, for the human player and for the partner AI respectively. Each graph shows a total of 3 636 854 moves for the human player and the same number for the AI partner, over the 27 592 games (279 758 rounds) in our data set. Figure 7 (a) shows the total number of moves played per category, while Figures 7 (b)–(f) show, within each category, how frequently the player or AI plays:

- The **single** card in a category of size $= 1$;
- The **lowest** card in a category of size > 1;
- The **highest** card in a category of size > 1 (if this card is not a boss card as defined below);
- Some **other** card (i.e. none of the above) in a category of size > 2;
- A **boss** card, i.e. a card where no other player can possibly hold a card in the same suit of higher rank.

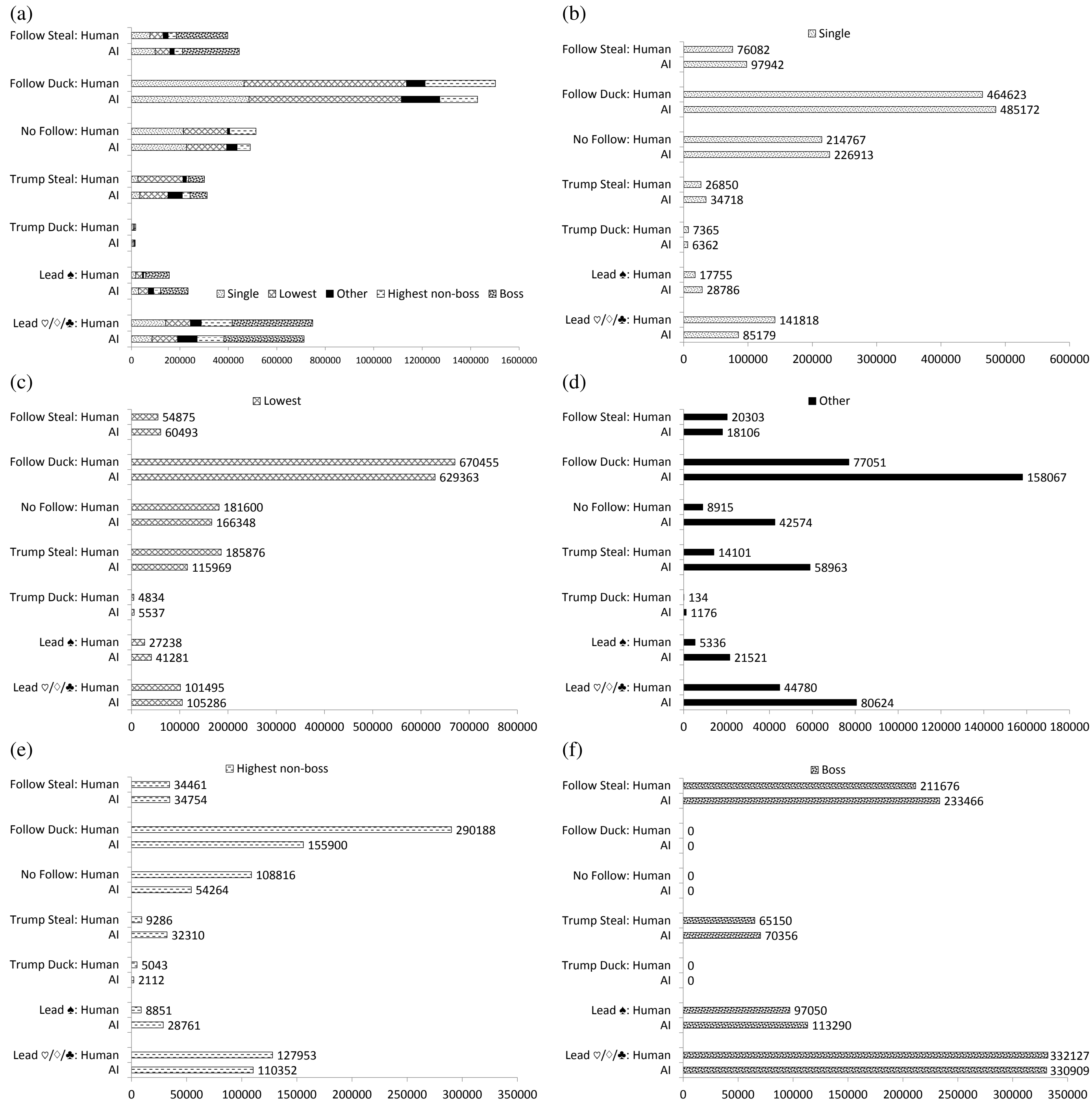


Fig. 7. Comparison of human and AI usage of abstract moves: (a) shows the total number of moves played per category, (b)–(f) show,within each category, how frequently the player or AI plays their (b) only, (c) lowest, (e) highest non-boss, (f) boss or (d) any other card in that category.

Figure 7 shows some interesting similarities and differences between human and AI play. Specifically, Figure 7 (d) shows that the AI plays "other" moves much more frequently than humans. Common strategies for Spades often recommend playing the highest or lowest card that achieves a particular outcome, whereas the ISMCTS-based AI player has no such bias. The AI player can also count cards perfectly, so may be able to see in certain situations that playing the second highest card is equivalent to playing the highest. Furthermore, Figure 7 (e) shows that the AI player is less likely to duck or no-follow with the highest card in situations where tricks are not needed. This is a common strategy amongst human players, on the principle that the highest card presents the greatest risk of taking an unwanted trick later in the round and so should be discarded as soon as the opportunity arises. However the AI may see that the highest card can safely be discarded later. From Figure 7 (f), human and AI players play boss cards with approximately the same frequency, suggesting that the AI can see the value of such plays. However it seems that the AI plays boss cards slightly more often, possibly because the AI has a better understanding of which cards are bosses due to its superior card counting ability.

Figure 7 (e) shows that the AI is more likely to steal with its highest non-boss trump card. This is an aggressive style of play: while human players seem to prefer playing their lowest trump (Figure 7 (c)) in an attempt to steal the trick with a low card, the AI plays higher to have a better chance of taking the trick or forcing an opponent to play a higher trump card. One of the problems with our ISMCTS player is that it often cannot see more than one or two tricks into the future [1], so may be failing to assess the value of holding onto a high trump card until later in the round. Figure 7 (b) indicates that human players are more likely to lead a trick with their only card in a non-trump suit. Voiding suits is generally a good idea as it produces opportunities to discard off-suit cards or play trumps, and this shows that humans are more likely to engineer situations where suits can be voided and more likely to do so when the opportunity arises. It may be that the AI sees voiding as less valuable than the human players.

To gain a deeper understanding of the differences between human players and the AI, we generated a decision tree to classify what abstract move humans typically make given the following features representative of the current state of play:

- **Tricks Needed:** How many tricks the player needs (assuming the player is focussing solely on making her own bid);
- **Partner Tricks Needed:** How many tricks the player's partner needs;
- **Turn Number:** Is the player playing 1^{st}, 2^{nd}, 3^{rd} or 4^{th} in the current trick;
- **Can Play Abstract Move:** Multiple features, one for each abstract move crossed with category {single,lowest,highest,other} and {boss,no boss}.

For example, the feature "Can Play Follow Duck Single" means the player has a single card lower than a card already played of the suit that was led with. As a move that does not steal or lead a round (i.e. all duck or no follow moves) cannot be a boss that feature is excluded from this node. For an example node that does, consider the last node on the far right bottom of the tree; "Can Play Follow Steal Lowest No Boss". This node means the player has multiple cards from the leading suit higher than those already played in the trick but not the highest of the suit still in play.

The tree (illustrated in Figure 8) was generated using version 2.15 of R [22] and version 4.1-3 of the package rpart [23]. Rpart is open-source software based on the now commercialised concept of Classification and Regression Trees (CART) [24]. The parameters were set as minsplit = 20, minbucket = 7, complexity parameter = 0.01 and maxdepth = 30. These parameters were set through preliminary experiments on a subset of the data. The 10-fold cross-validated error rate was 0.430. For comparison, the error rate of a dumb classifier that always predicts the most common class is 0.816.

Some interesting and well advised strategies emerge from the data when presented in this format. For example, if a player has already won all the tricks they need this round, a good strategy is to play their highest card in the current suit that will not win the round. By doing so the player gets rid of cards more likely to win tricks later in the round, which would incur a higher penalty for having gone over the player's bid.

Other strategies apparent from Figure 8 include leading with boss cards when it is your turn to go first in a trick and have the necessary card to do so. This strategy will cause a guaranteed win if you are playing the boss ♠, or if all players have at least one card of the suit you are leading with if playing the boss ♡, ♢ or ♣.

Interestingly, there is a high preference to play the lowest trump card a player has that will steal the trick when they have higher valued trump cards that could do so. This strategy is rational if you are the last player in a trick, as you are guaranteed to then win the trick whilst simultaneously using the lowest value card you have to do so. However, within the data analysed, this move was typically played regardless of the player's turn in the trick. This may be representative of players being cautious, unwilling to part with a high valued ♠ in case it gets trumped later in the trick. This strategy also allows the player to force the remaining players to play higher valued ♠ cards or lose the trick. Doing so can plausibly make the initial player's remaining ♠ card(s) the boss and, therefore, a guaranteed trick win later in that round.

Another worthwhile observation is to note that the feature "Partner Tricks Needed" is not used in the average strategy. This implies that an average player assumes his partner will meet their own bid. It also shows that average players typically will not alter their play style in cases where they could assist. Given that tricks won by the player and their partner are summed at the end of a round to see if they match the sum of their bids, failing to assist a struggling partner is a poor strategy.

Figure 8 also includes some less expected strategies. In particular, the high preference to duck in a trick without first considering if you could steal and the absence of considering to steal with a trump when the player has multiple ♠ cards. This may be caused by flaws in the strategies played (the players are human after all), more complex Spades strategies incomprehensible to us, or simply the inaccuracies of the tree.

Overall, in our study we found a decision tree was a useful representation of human behaviour. The ease of comparing two trees generated from different subsets of the data greatly simplified the process of determining typical behaviours. For example, we also generated a decision tree for only good strategies by reducing the data set based on all of the following criteria being met:

- Only close games or games where the player wins ($FinalScore > 0.9 * TargetScore$);
- Only games against tough AI (AI level $>= 10$);
- Only games by players that win more than average (win rate > 0.5).

The resultant tree was very similar to the one illustrated in Figure 8, but the error rate was significantly higher. This suggests that better players use a wider variety of strategies. Without a human readable model, such comparisons would be significantly more complex.

It would be interesting, in further work, to learn different decision trees for clusters of players found by unsupervised

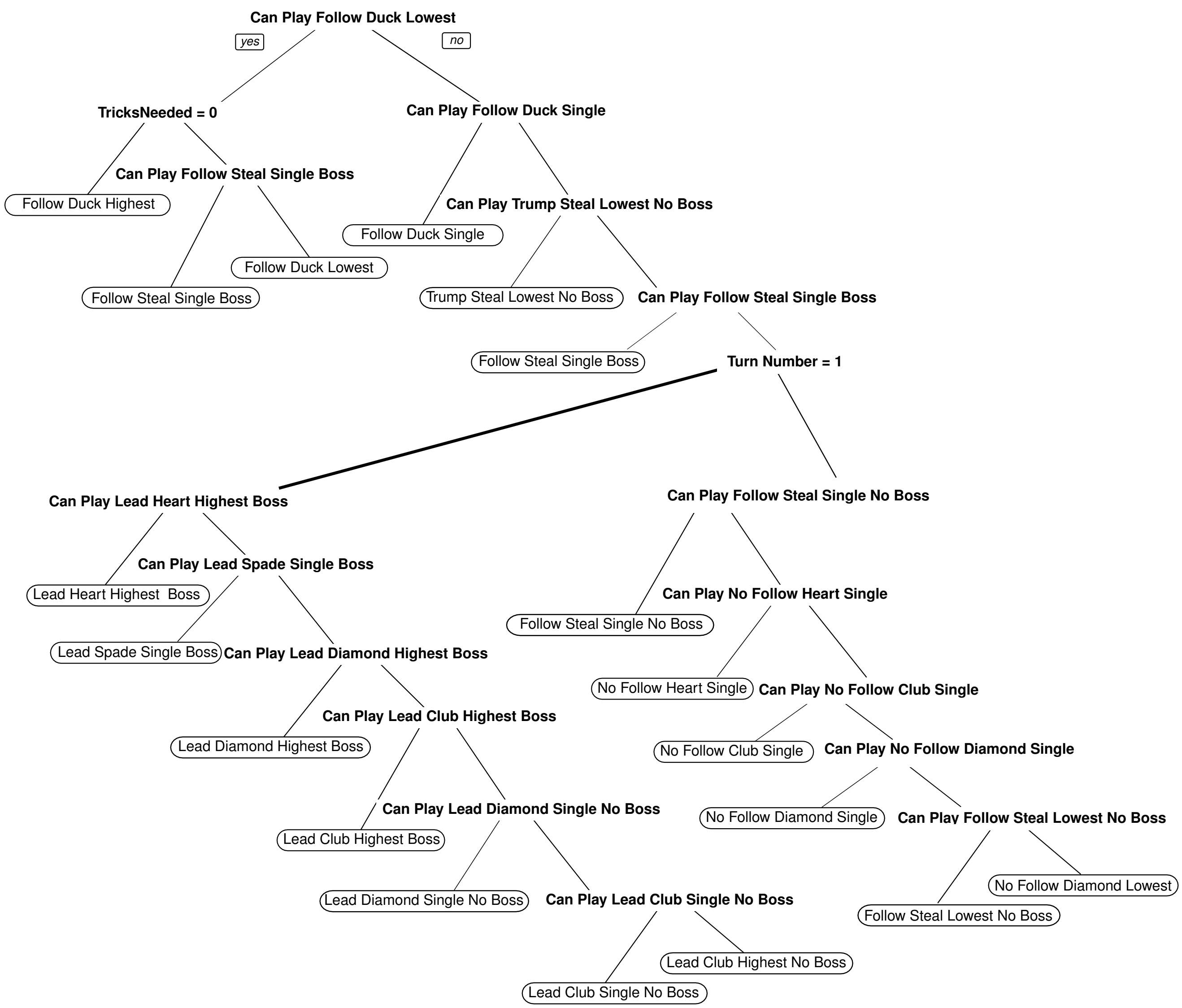


Fig. 8. Model of average Human Strategy.

learning algorithms. Previous work has found different player types in many games using various methods of clustering [25], [26], [27]. Decision trees, as demonstrated in this paper, are a suitable representation of play style and could help with the understanding of the differences between these player types due to the ease of comparing their human readable model. Clustering players may also give further insight into different responses to difficulty.

V. Conclusion

In this study we have collected a large volume of data from the commercial game AI Factory Spades via Google Analytics. This data was a key resource for analysing the difficulty of the game, the ISMCTS based AI players and the strategies of human players.

Our analysis suggests that the default level of AI Factory Spades is too high and that some players may have been put off by this quickly. It is tempting to conclude from this that the default level should be lowered. However, doing so may cause stronger players to assess the AI as weak. This may be even more harmful in the long run as, in contrast to beginners, experienced Spades players are more likely to play the game over a long period of time if they enjoy it (resulting in increased advertising revenue) and are more likely to be vocal about the product's strengths or weaknesses in reviews (impacting new players' decision whether to download the game). A balance must be struck to keep both groups of players happy.

Adaptive opponents offer a potential solution, but care needs to be taken that the player's skill level is being assessed on "fair" card deals. If the AI's first few hands of cards happen to be strong, causing a skilled human player to lose, then the wrong adjustment may be made. Whatever measures are taken to improve player satisfaction and retention, it will be important to perform further data analysis to assess the true effect of the changes.

We have presented a number of approaches for analysing the difficulty of a game that are generally applicable. In particular, the idea of using binomial confidence intervals as in Figure 4 could be used in any game to provide a statistically sound method for assessing player strength.

Furthermore, we have observed from the data that the ISMCTS based AI players behave differently to humans. In particular, the usages of their highest non-boss cards is significantly different. The ISMCTS based AI player is aware whether their highest card is the highest of that suit still in play and can, therefore, make rational choices regarding this. Human players, however, can get confused regarding this and may think they have a boss card when a better card is still in play. This suggests a novel method of possibly weakening AI; causing them to forget or make mistakes when tracking the cards that have been played before.

We have also modelled a typical human strategy as a decision tree. The resultant model highlights a number of good strategies for playing Spades, relevant both to the commercial product used to obtain the data and any other Spades game both digital and card based. In future work these strategies could be used to influence the behaviour of the AI players, perhaps making them appear more human. These strategies also may contribute to playing strength as it demonstrates where humans saw different strategies to those being exercised by the AI.

As with most card games, Spades is a game of tactical and strategic skill. Player feedback indicates that satisfaction correlates with difficulty: the appeal of the game for the majority of players is that the AI provides a competent partner and challenging opponents. The methods and conclusions of this paper are likely to hold for other similarly strategic games, most notably digital versions of board and card games but also turn-based and real-time strategy games. For competitive games where dexterity is a factor (such as first-person shooters and racing games) similar ideas could be applied, but the likely result of this type of analysis would be the need to "dumb down" the AI players to place the majority of players in the "no difference" category. For games which are not competitive in nature (such as story-driven action games or arcade-style games) the "no difference" category tends to be frustrating rather than fun; here the aim should instead be to ensure the human player is firmly in the "better" category but feels that getting there was an accomplishment.

This study demonstrates post-deployment analysis of player behaviour for the fine tuning of AI difficulty and behaviour. The methods presented here are directly applicable to any strategic competitive game, potentially applicable with modifications to a much wider class of games, and have the potential to significantly improve player retention.

References

[1] D. Whitehouse, P. I. Cowling, E. J. Powley, and J. Rollason, "Integrating Monte Carlo Tree Search with Knowledge-Based Methods to Create Engaging Play in a Commercial Mobile Game," in *Proc. Artif. Intell. Interact. Digital Entert. Conf.*, Boston, Massachusetts, 2013, pp. 100–106.

[2] M. Csikszentmihalyi, C. Kolo, and T. Baur, "Flow: The psychology of optimal experience," *Australian Occupational Therapy Journal*, vol. 51, no. 1, pp. 3–12, 2004.

[3] R. Coulom, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search," in *Proc. 5th Int. Conf. Comput. and Games, LNCS 4630*, Turin, Italy, 2007, pp. 72–83.

[4] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo Planning," in *Euro. Conf. Mach. Learn.*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Germany: Springer, 2006, pp. 282–293.

[5] G. M. J.-B. Chaslot, J.-T. Saito, B. Bouzy, J. W. H. M. Uiterwijk, and H. J. van den Herik, "Monte-Carlo Strategies for Computer Go," in *Proc. BeNeLux Conf. Artif. Intell.*, Namur, Belgium, 2006, pp. 83–91.

[6] C.-S. Lee, M.-H. Wang, G. M. J.-B. Chaslot, J.-B. Hoock, A. Rimmel, O. Teytaud, S.-R. Tsai, S.-C. Hsu, and T.-P. Hong, "The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments," *IEEE Trans. Comp. Intell. AI Games*, vol. 1, no. 1, pp. 73–89, 2009.

[7] S. Gelly and D. Silver, "Monte-Carlo tree search and rapid action value estimation in computer Go," *Artif. Intell.*, vol. 175, no. 11, pp. 1856–1875, 2011.

[8] Y. Björnsson and H. Finnsson, "CadiaPlayer: A Simulation-Based General Game Player," *IEEE Trans. Comp. Intell. AI Games*, vol. 1, no. 1, pp. 4–15, 2009.

[9] C. Browne, E. J. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Trans. Comp. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, 2012.

[10] P. I. Cowling, E. J. Powley, and D. Whitehouse, "Information Set Monte Carlo Tree Search," *IEEE Trans. Comp. Intell. AI Games*, vol. 4, no. 2, pp. 120–143, 2012.

[11] D. Whitehouse, E. J. Powley, and P. I. Cowling, "Determinization and Information Set Monte Carlo Tree Search for the Card Game Dou Di Zhu," in *Proc. IEEE Conf. Comput. Intell. Games*, Seoul, South Korea, 2011, pp. 87–94.

[12] E. J. Powley, D. Whitehouse, and P. I. Cowling, "Bandits all the way down: UCB1 as a simulation policy in Monte Carlo Tree Search," in *Proc. IEEE Conf. Comput. Intell. Games*, Niagara Falls, Ontario, Canada, 2013, pp. 81–88.

[13] Pagat, "Spades," http://www.pagat.com/boston/spades.html.

[14] J. Bohannon, "Game-miners grapple with massive data," *Science*, vol. 330, no. 6000, pp. 30–31, 2010.

[15] M. S. El-Nasr, A. Drachen, and A. Canossa, Eds., *Game Analytics, Maximizing the Value of Player Data*. Springer, 2013.

[16] G. N. Yannakakis, "Game ai revisited," in *Proceedings of the 9th Conference on Computing Frontiers*, ser. CF '12, 2012, pp. 285–292.

[17] B. G. Weber and M. Mateas, "A data mining approach to strategy prediction," in *Proceedings of the 5th international conference on Computational Intelligence and Games*, ser. CIG'09. IEEE Press, 2009.

[18] C. Bauckhage, C. Thurau, and G. Sagerer, "Learning human-like opponent behavior for interactive computer games," in *Pattern Recognition*. Springer, 2003, pp. 148–155.

[19] B. G. Weber, M. John, M. Mateas, and A. Jhala, "Modeling player retention in madden nfl 11." in *Innovative Applications of Artificial Intelligence*, D. G. Shapiro and M. P. J. Fromherz, Eds. AAAI, 2011.

[20] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934.

[21] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. Andre, "Player Modeling," in *Dagstuhl Seminar on Artificial and Computational Intelligence in Games*, 2013.

[22] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2012, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org/

[23] T. Therneau, B. Atkinson, and B. Ripley, *rpart: Recursive Partitioning*, 2013, r package version 4.1-3. [Online]. Available: http://CRAN.R-project.org/package=rpart

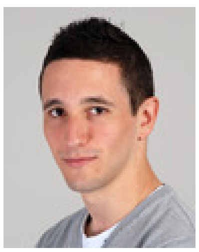
[24] L. Olshen, J. Breiman, R. Friedman, and C. J. Stone, "Classification and regression trees," *Wadsworth International Group*, 1984.

[25] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in Tomb Raider: Underworld," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*. IEEE, 2009, pp. 1–8.

[26] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 2012, pp. 163–170.

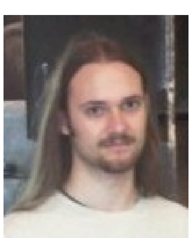
[27] J. Gow, R. Baumgarten, P. Cairns, S. Colton, and P. Miller, "Unsupervised modelling of player style with lda," *Computational Intelligence and AI in Games, IEEE Transactions on*, 2012.

Peter Cowling (IEEE) is a 50th Anniversary Professor of Computer Science and Management at the University of York (UK). His work centres on computerised decision-making in games, scheduling and resource-constrained optimisation. He has worked with a wide range of industrial partners, been a director of 2 research spin-out companies, and has published over 90 scientific papers in high-quality journals and conferences. He is an Associate Editor of the IEEE Transactions on Computational Intelligence and AI in Games.

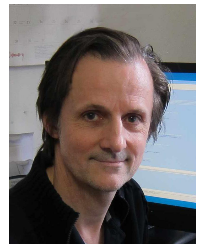


Sam Devlin received an MEng degree in Computer Systems and Software Engineering from the University of York, UK, in 2009. In 2013, he received a PhD in Computer Science from the University of York and visited Oregon State University funded by a Santander International Connections Award. He is currently a Research Associate in the York Centre for Complex Systems Analysis (YCCSA), and is working on data mining/analytics as part of the EPSRC project ‘New Economic Models and Opportunities for Digital Games.’

Edward Powley (IEEE) received an MMath degree in Mathematics and Computer Science from the University of York, UK, in 2006, and was awarded the P B Kennedy Prize and the BAE Systems ATC Prize. He received a PhD in Computer Science from the University of York in 2010. His post-doctoral research applied MCTS to games and sequential decision problems with hidden information and stochastic outcomes. He is now an independent game developer, currently developing his first title for Android and iOS, but still has research interests in using AI and analytics to build better games.



Daniel Whitehouse (IEEE) received the Master of Mathematics degree in Mathematics from the University of Manchester, UK, in 2010. He is currently pursuing a Ph.D. in Artificial Intelligence in the Department of Computer Science at the University of York, UK. and is funded as part of the EPSRC project *UCT for games and Beyond*. He is also a member of the York Centre for Complex Systems Analysis (YCCSA) at the University of York. His research interests include games and game AI and he is investigating the application of Monte Carlo Tree Search methods to games with chance and hidden information.



Jeff Rollason currently heads up and is co-founder of AI Factory Ltd, with some 70 million users. AI Factory was founded in 2003 with the premise of creating modular AI and games that would be licensed to 3rd parties. The business has been very successful, with clients such as Microsoft, which, with other licensees, has seen AI Factory game engines in consoles, PCs, In-flight entertainment and mobile devices. His primary contribution is game AI. His education includes Combined Sciences B.Sc (1st) at Westfield UOL and Computer Science M.Sc UCL UOL. His 17 years in Academia included teaching core computer architecture courses at Kings College UOL. His programs have also competed in computer tournaments for Chess, Go and particularly Shogi (Japanese Chess), where his program Shotest twice ranked 3^{rd} in the world.