A Reliable Information Acquisition Model for D&D Players

João Gabriel Gama Vila Nova(✉), Marcos Vinícius Carneiro Vital, and Roberta Vilhena Vieira Lopes

Universidade Federal de Alagoas, Maceió, AL, Brazil
{jgg, rvvl}@ic.ufal.br, marcosvital@gmail.com

Abstract. This work has the objective to present a combat simulator model to D&D players that considers the possibility of actions formalized in game rules added with players' capability to acquire strategies that will guide them to improve their performance during game simulations. The quality improvement of the strategy adopted by the player with the proposed model, called Adaptive, is evidenced through combat simulations between a team with and without the ability to acquire information, or different levels of this ability, considering that each player can have different levels of experience. It was noticed that the ability to acquire trustworthy information can make the player's strategy improve exponentially depending on the rate of information acquirement.

Keywords: Role-playing games · Strategy · Player behavior · Artificial intelligence

1 Introduction

Game Theory is a mathematical theory developed to describe situations that can be observed when two or more players interact, and constitutes a language to describe intentional and objective processes [1]. It contributes to amplify the ability to act strategically, allowing to foresee the best results for players using the available strategies. It has been applied to several areas of knowledge, such as in Economics, Psychology, Sociology, Biology and Administration [2].

Regardless of the specific situation, players must make decisions that aim at maximizing their gains, rather than cooperating as suggested by the Nash equilibrium [1, 2]. Of course that, in some games - as in a role-playing games - gain can be many things, such as building a captivating narrative. Yet, when game mechanics are analyzed in those games from a combat perspective, gain can be defined as players besting their enemies with minor losses. But how to model the behavior of an inexperienced player behavior? This can be done considering that sometimes an inexperienced player, after some playthroughs, can win against an experienced player. The hypothesis to explain this is the ability of an inexperienced player, besides learning from his own errors and experiences, to seek reliable information during or between gaming matches.

Reliable information can be acquired throughout the evolution process of the inexperienced player, and it can be obtained by various means: conversations with other players, reading manuals, observing opponent's strategies, etc. To test the validity

P. Moura Oliveira et al. (Eds.): EPIA 2019, LNAI 11804, pp. 73–85, 2019.
https://doi.org/10.1007/978-3-030-30241-2_7

of the hypothesis presented, this article will show, in a combat simulation environment, the behavior of a team of players who knows the rules but are inexperienced when confronted by an experienced adversary team, considering that the inexperienced players can acquire reliable information between sessions.

1.1 Board Games and Tabletop Role-Playing Games

Board games have a long history in mankind, and can be traced back to 3 to 4 thousand years old, when Go was created in ancient China. Beyond just entertainment, board games have also been used as educational material in a wide range of situations including learning numerical-spatial relations in young children [3], medical training and health education [4], and computer science and programming skills [5].

A much more specific hobby, considered to be a very important evolution from board games and complex wargames, was born in 1974, when the first edition of Dungeons & Dragons (D&D) - considered to be the first "tabletop" Role-Playing Game (RPG) - was created. The first edition of D&D and the first RPG experiences that happened during the 70s were still very similar to the wargames they came from, but the development during the late 70s and early 80s expanded the gaming ideas to something closer to what we can see today [6].

Tabletop RPGs attempt to simulate imaginary situations set on real or fantasy settings, using game rules that usually relies on dice rolls and pre-set game statistics. In most games (as in D&D), each player creates a character following one or more rulebooks, describing their gaming statistics, the character history and personality. The game happens as one player, known as Dungeon Master (DM) in D&D games, describe a situation and the players describe and role-play their characters' actions.

Depending on the group play style and game system, this can escalate into complex collaborative storytelling and acting, and imagination plays an important role on a session. This led the development of research on areas as diverse as teaching, semiotics and psychology [7]. There is also evidence that tabletop RPGs players perform higher in creativity tests than digital RPGs players and non-players [8].

On the top of storytelling, acting and imagination, all RPGs relies on game rules to determine the performance and outcomes of a character's actions. In D&D this is usually resolved with a 20-sided dice roll, where the player must attain a target number that is calculated based on the result of the roll combined with the potential effects determined by the game rules (such as the character's abilities, the task difficulty, etc.) that can be added or subtracted. As D&D focus in heroic fantasy storytelling, combat against monsters and other enemies are commonplace on a gaming session, with conflict resolution requiring a vast set of rules and several dice rolls. As the game progress and the players' characters become more resourceful and obtain more abilities, preparing a session becomes a complex and challenging task for the DM, as the desired outcomes of combat situations becomes harder to foresee.

The tool that was used as a background for this work is being developed to tackle typical DM's challenges concerning combat balancing, as a successful D&D gaming session relies on presenting the players with combat and conflicts in a way that is challenging but not impossible to overcome. The SCURDD tool (in Portuguese, SCURDD stands for combat simulator using D&D rules) was created using the ruleset from D&D as presented on D&D Players' Handbook 3.5 edition, published by Wizards of the Coast in 2003, and its current stage simplifies the game ruleset and simulate straightforward combat situations that can be used by the DM to easily create and test encounters planned to be used on a session.

1.2 Related Work

The idea of combat simulations is not new to the D&D context, and other attempts were made to achieve a tool that can be used by the DM to create balanced encounters. A well-known tool is the D&D Encounter Simulator, that draw the rules from D&D 5th edition and simulates an encounter a thousand times, generating results statistics for the DM to evaluate [9]. This tool, however, uses a fixed set of playing strategies, and does not try to include the different levels of experience that can be held by players, nor the learning process that can be achieved between sessions. Those aspects are the main novelty brought by SCURDD when compared to similar systems.

The field of artificial intelligence have several applications related to games, including digital RPGs. AIs are used to simulate non player character behavior, to generate procedural content and many other tasks [10]. There is no academic record, however, of AI used to simulate behavior or learning in tabletop role-playing games.

2 The Simulation Tool: SCURDD

SCURDD is a system created with the objective of helping Dungeon Masters (DM) to balance their encounters in D&D v3.5. This game edition was chosen because, despite the existence of newer versions, it is still one of the most currently played versions of the system and, furthermore, has some peculiarities not shared with newer editions such as v4 and v5 do not have. In short, v3.5 has much more published content than v4 and v5, thus having more options for character customization, world building and game mechanics to challenge experienced players.

The tool aims to help balancing encounters by simulating combats with user inputted character sheets from a text file. Every character will be simulated in the system with an AI that fits better the user's gaming table. There are two main modules of AIs in SCURDD: A Skilled AI (SAI) and an Adaptive AI (AAI).

The SAI was created to simulate players with a good knowledge of D&D mechanics, who would know good strategies to control their characters, although not always the best ones. On the other hand, AAIs were made to represent beginner players, with little knowledge of the game, that could learn more after each session.

Since a real D&D game can potentially encompass an infinite set of possible player's decisions and character actions (as in most RPGs), SCURDD abstracts many rules and possibilities to a small set of character actions that are usually used during combat. Combat simulations starts as the game rules of D&D v3.5 states, by an Initiative Roll, that will determine the order of turns of characters actions from both teams. Following that, character actions at SCURDD current version are limited to: attacking, casting spells that does damage or heal (on a single or multiple targets) and moving at the battlefield. Character can act as their AI type commands or by following their team's messages. After doing its main turn action, such as an attack or casting a spell, a character may send a message to his team asking for healing or help to defeat his target. When a character can not get in range of an enemy in one movement action, it can spend its main action as an extra movement and then end its turn. When a character is reduced to 0 HP it is defeated and removed from that simulation, together with all messages concerning him.

The "help to defeat" message is sent when a character does damage to an enemy and makes the enemy get near death, while "need healing" is sent when the character is low on HP and ask for teammates to heal him. During combat simulations the user can follow what is happening in combat step-by-step with the output of SCURDD.

The team that wins a combat is the one with any number or characters alive after the defeat of all enemies. Character's statistics are recorded in logs with different complexities and information depending on their type of AI after each combat. At the end of all simulations, a log containing information about each character performance is generated in a text file so that the user can have further analysis of that simulation.

It this work, we will analyze the impact that a reliable information acquisition model can have into the strategies a player can use during a combat at a D&D game.

3 Skilled Artificial Intelligence

The Skilled Artificial Intelligence is one of the AI modules from SCURDD, it was created to represent a player with some knowledge about D&D rules and good strategies, but that is stagnant and can not evolve as the second AI model. In this way, it can also represent DM's NPCs and monsters, giving them good but fixed level of strategy. The SAI has two submodules, also known as Roles: Fighter and Spellcaster. Each one of these has a series of behaviors linked to their function.

Both SAI Fighters and Spellcasters have a straightforward vision for combat, following a basic strategy of fighting enemies using the best possible attack available from his list. This AI chooses his targets every turn based on a priority order (α) that focus: Proximity (β), Health Condition (δ) and Role (η) of opponent set X, as shown in the following equations:

$$\beta_X^R = \{i | \exists i \in X \left(\forall j \in X - \{i\} \left(\beta_i^R \leq \beta_j^R \right) \right) \} \tag{1}$$

$$\delta_X^R = \{i | \exists i \in X \left(\forall j \in X - \{i\} \left(\delta_i^R \geq \delta_j^R \right) \right) \} \tag{2}$$

$$\eta_X^R = \{i | \exists i \in X \left(\forall j \in X - \{i\} \left(\eta_i^R \geq \eta_j^R \right) \right) \} \tag{3}$$

Equation (1) reads as: there is an opponent i closer or equally distant as other opponents to the acting character of role R. Thus, β is the set of closest opponents to this character. Equation (2) reads as: there is an opponent i that is in worst or in equally bad health condition than other opponents of the acting character of role R. Thus, δ is the set of worst health conditioned opponents for this character. Equation (3) reads as:

there is an opponent i that has a more important or equally important role as the other opponents from the acting character of role R. Thus, η is the set of most important role opponents for that character.

The target's choice for a Fighter SAI is done first by searching for an opponent that is part of all three sets (β, δ and η). If that opponent does not exist, the AI will try to find one that fits in at least two other sets and, in the last case, the SAI will choose the closest enemy (β). This is demonstrated on Eq. (4):

$$\alpha_O^{Fgt} = \begin{cases} i \in \beta_O^{Fgt} \cap \delta_O^{Fgt} \cap \eta_O^{Fgt} & if\left(\beta_O^{Fgt} \cap \delta_O^{Fgt} \cap \eta_O^{Fgt}\right) \neq \emptyset \\ i \in \beta_O^{Fgt} \cap \delta_O^{Fgt} & if\left(\beta_O^{Fgt} \cap \delta_O^{Fgt}\right) \neq \emptyset \\ i \in \beta_O^{Fgt} \cap \eta_O^{Fgt} & if\left(\beta_O^{Fgt} \cap \eta_O^{Fgt}\right) \neq \emptyset \\ i \in \delta_O^{Fgt} \cap \eta_O^{Fgt} & if\left(\delta_O^{Fgt} \cap \eta_O^{Fgt}\right) \neq \emptyset \\ i \in \beta_O^{Fgt} & Otherwise \end{cases} \tag{4}$$

When close to death, Fighters tend to get closer to Spellcasters allies that have healing spells, asking for their help. The target priority of Spellcasters is equal to Fighters, but before deciding to attack, they need to choose if they will heal himself or an ally with spells, if the character has this ability.

In this way, Spellcasters have, besides the previous sets of players, a set of allies A and the function ϕ that returns the health condition of allies, $i \in$ A, that is used to construct the set of most wounded allies, defined in Eq. (5):

$$\Phi_A^{Spc} = \{i | \exists i \in A\left(\forall j \in A - i\left(\Phi_i^{Spc} \geq \Phi_j^{Spc}\right)\right)\} \tag{5}$$

The choice to heal an ally includes the own acting Spellcaster as the top priority if he is badly wounded. If it is not the case, the Spellcaster will try to heal the most wounded ally. If there is no ally badly wounded, the acting Spellcaster will choose to attack with magic or spells instead of healing. Spellcasters always choose their best available spell, since spells are a finite resource and eventually end. The choice of the target follows the same rules as the Fighter SAI as shown in Eq. (6):

$$\alpha_A^{Spc} = \begin{cases} Spellcaster & if\,\Phi_{Spc}^{Spc} \neq 0 \\ i \in \Phi_A^{Spc} & if\,\Phi_A^{Spc} \neq \emptyset \\ i \in \beta_A^{Spc} \cap \delta_A^{Spc} \cap \eta_A^{Spc} & if\left(\beta_A^{Spc} \cap \delta_A^{Spc} \cap \eta_A^{Spc}\right) \neq \emptyset \\ i \in \beta_A^{Spc} \cap \delta_A^{Spc} & if\left(\beta_A^{Spc} \cap \delta_A^{Spc}\right) \neq \emptyset \\ i \in \beta_A^{Spc} \cap \eta_A^{Spc} & if\left(\beta_A^{Spc} \cap \eta_A^{Spc}\right) \neq \emptyset \\ i \in \delta_A^{Spc} \cap \eta_A^{Spc} & if\left(\delta_A^{Spc} \cap \eta_A^{Spc}\right) \neq \emptyset \\ i \in \beta_A^{Spc} & Otherwise \end{cases} \tag{6}$$

The SAI was developed based in production rules using author's vast knowledge about D&D after many years of playthrough, helped by researches and surveys with other players that share the hobby. It was created to represent not an expert and unbeatable player, but one with some skill in the game.

4 Adaptive Artificial Intelligence

The Adaptive Artificial Intelligence (AAI) was developed with the intent to simulate a player in the learning stages of the game, demonstrating the difficulties of learning how to use a character sheet without any experience in D&D. This AI consists of a series of behaviors that will represent the play style of a certain character under its control. These behaviors are separated in two types: Main Behaviors (MBs) and Sub Behaviors (SBs).

Main Behaviors are abstractions of combat behaviors that D&D characters usually have, and Sub Behaviors are inside details and specifications that some MBs need. The created MBs for the purpose of this simulator are:

- **Priority Attack Order (PAO):** Consists on the use priority of each attack in the character's attacks list over one another.
- **Priority Offensive Spell Order (POSO):** Equivalent to PAO, but for offensive spells, those that cause damage, from the character's spells list.
- **Priority Defensive Spell Order (PDSO):** Same from POSO, but for healing magic (magic that restore HP).
- **Priority Enemy Target (PET):** Priority for the purpose of deciding the enemy targets of character's attacks and spells. This MB has SBs that have priority among them, and the order of this priority is chosen during character's evolution. The SBs for PET are: *By Role*: Priority for targeting enemy characters of a determined role (e.g. Spellcasters > Fighters); *By Health*: Prioritize enemies with more or less HP; *By Proximity*: Prioritize enemies closer or farther away from the character.
- **Priority Allied Target (PAT):** It is the equivalent of PET but for allies, used for targeting characters when using healing spells or choosing allies for protection. This MB has the same SBs that of PET but are used for allies instead of enemies.
- **Movement Pattern (MP):** This MB specifies the way that a character moves in the battlefield. This way is specified by the SBs: *Closer to Enemy*: Characters tends to move in the direction of enemies during combat; *Closer to Ally*: Characters fight moving alongside allies or fight enemies close to them; *No Moving*: Characters tend to stay in place during combat, keeping their initial position in the simulation.
- **Message Reading (MR):** This behavior contemplates the attendance of a character about the messages from his team. This attendance is specified by the SBs: *Always Attend*: Character always attend to messages that he is capable of attending in the current turn; *50% to Attend*: Character has 50% to attend to a message that he is capable of; *Never Attends*: Character never attends to messages sent from his team, even if he is capable of doing it. Capability of attending a message is defined by the possibility to attend it in the current turn.
- **Play Style (PS):** The general play style of a character is defined in this MB, directing the method of attack and helping in prioritizing the actions and movement of a character during the simulation. This MB contain the SBs: *Aggressive*: This Play Style

comprehends stubborn and unilateral characters that focus all their effort in trying to cause the most damage possible. Aggressive characters only attack with their most prioritized attack from PAO or spell from POSO; *Balanced*: These characters try to cause damage and be somehow useful every turn, using not always their most prioritized attack or spell, varying it by the situation; *Defensive*: Defensive characters fight to eliminate enemies that are threatening their allies. Utilizing their PAT, they decide an ally to protect, heal and fight enemies that are current engaging this character in combat. It is possible for Defensive characters to have himself as his priority ally.

4.1 Scores and Statistics

After each combat simulation, logs containing statistics of each individual character are generated for that simulation. These statistics includes Physical Damage dealt (PD), Spell Damage (SD), Healing, Health Points left (HP) and Spell Uses (SU). Other values saved in logs are calculated based on these past statistics. These other values are the character Score and High-Performance Score (HPS).

The Score represents the punctuation a character reached in that simulation, and it is calculated summing Damage, Healing and HP statistics. Where Damage is the sum of PD and SD, Heal is equal to all Healing a character has done, if he is capable of using healing spells, and HP are the health points of this character at the end of combat, possibly 0 if he was defeated.

On the other hand, the HPS is a metric of possible performance for that character in that specific simulation. It is calculated as if the character could achieve 75% of his best performance in every action that he does (e.g. the character can do 10 damage at max per turn, his HPS would count as he has done at least 7.5 damage every turn in that simulation). The HPS calculate the possible performance for all character actions that influence his Score, including healing and damage capabilities in addition to health left at the end of combat.

In summary, the HPS is a possible reachable Score of 75% efficiency for that character in that simulation. We will be using all these statistics presented in this subsection to help the character to intuit a new set of MBs and SBs so that he can perform better. One important thing to notice is that the calculation of HPS is different based in the character' Role, because only Spellcasters can cast spells, and this ability changes the way we need to look at this HPS, affecting directly it's calculus.

4.2 AAI Evolution

After every simulation, characters bearing AAIs evolve using their logs from previous simulations. There are two main ways that the system can intuit changes on characters behaviors using their logs, and the first and simplest is to use only the last log to do it. The first way is used always until a character has at least two Best Scores, that are logs in which their Score reached at least half of the HPS for that simulation.

Using only their last log, the system calculates character's performance using those MBs and SBs, comparing it with the statistics contained in the log such as PD, SD, Healing, HP, and others depending on character's Role. Analyzing how well the character has done using those MBs and SBs, relating each one of them to a set of those statistics, the system applies probabilities of changes to this set of behaviors.

The parameterizations used in this method were made by simply dividing the odds of behavior changing equally between all related MBs, for example, the HP left at the end of a combat is related to the MBs PAT, MP and PS. This was a simple approach, but part of the future work is to analyze how can this parametrization impact in characters' performance. To demonstrate how the structure of probability attribution is working, the pseudocode presented in Algorithm 1 shows the logic behind the changes of behaviors related to the final HP of a Fighter character, being HPF the final HP in a simulation:

Algorithm 1: Probability Attribution related to HPF of Fighters

```
procedure HPFBehavioralChanges(character, log)
  float chances[]{PAT, MP, PS}
  if character.Role = Fighter then
      HPF = log.HP
      HPMax = character.HPMax
      if HPF < 0.5 * HPMax then
          Chance of change for the PAT, MP and PS raise in 45%
      if 0.5 * HPMax <= HPF and HPF < 0.75 * HPMax then
          Chance of change for the PAT, MP and PS raise in 25%
      if 0.75 * HPMax <= HPF and HPF < HPMax then
          Chance of change for the PAT, MP and PS raise in 15%
      if HPF = HPMax then
          No extra chance is added to PAT, MP and PS
  if character.Role = Spellcaster then
      Similar comparisons are made
  return chances[]
```

The MBs presented in the algorithm are also affected by other analysis, like damage related ones, possibly reaching up to 95% chance of changing. The changing of an MB or SB is nothing more than the change to another random MB or SB different than the last. So, it is possible that an MB or SB is not changed if the random chance does not meet the probability of changing that behavior.

The other way to intuit changes on character's behaviors is using the list of Best Scores. When a character has at least two Scores that go over half of their respective HPS, the logs containing those Scores are added to a list of Best Score logs. The system will search for two of those logs in particular, first the one with the biggest numerical Score and second the log which has the smallest difference between its Score and it's HPS.

In that way, we call the first log the Best Score (BS) and second one the Smaller Difference from HPS (SDHPS). The choice of the SDHPS in detriment of searching for the second biggest numerical Score is done because the biggest numerical Score is not always the representation of a great performance. This happens because the longer a simulation is, the bigger will be Scores and HPSs for that simulation, and if the HPS is still much higher than the log Score, the character did not have a good performance.

Finally, have the system picked the BS and SDHPS among the Best Score logs, it will compare the set of MBs and SBs from both logs and assign the minimum changing chance for the MBs and SBs that are equal in both logs. This minimum changing chance is equal to 5%. The behaviors that are not equal in both logs are calculated as shown before, using comparisons with simulation metrics and assigning probabilities of change.

The only case when the chance of changing a behavior is 0% is when, apart from having equal behaviors among BS and SDHPS, this behavior is also listed in the Best Behaviors list for that character. This list and the subsystem of Insights and Learning Speed (LS) will be discussed in the next subsection.

Thus, using these two methods after every simulation character evolve each in a unique way, directing themselves to an "ideal" set of behaviors. The time that a character needs to reach this ideal set of behaviors depends mainly on his initial random set of MBs and SBs, his performance in combat and the Learning Speed chosen by the user.

4.3 Insights and Learning Speed

For the purpose of representing external knowledge acquirement such as, in the case of D&D, forum discussions, rule books and another player, we created the Insights and Learning Speed system. This system has also the function to help SCURDD give a reasonable output to the user in less than 1000 combat simulations.

The LS is a parameter chosen by the user after starting the simulations, when among the characters there is at least one with an AAI. This parameter varies from 1 to 5, the bigger the LS more chance to a character receive an Insight. Insights are like clues to ideal behaviors for a particular character, that are generated based on his character sheet and D&D knowledge from the authors and added to that character Best Behaviors list. This Best Behaviors list is a hardcoded list of best MBs and SBs combination for each type of character (Spellcaster and Fighter), created by analysis of many simulations done in the tool. As future work, we will develop a method to generate this list automatically with a more profound analysis of every character sheet, generating a unique list for every character.

After every simulation, apart from having the common evolution methods described in the last section, characters with AAI have a chance to receive an Insight based on their LS. This Insight will guide one of their MBs or SBs that still is not a Best Behavior to one. This makes that in future evolutions will be a greater chance of that behavior appearing among the Best Score logs, and in the BS or SDHPS.

5 Evaluation

To test the potential of both AIs a set of experiments were made, putting teams of SAIs and AAIs to battle against each other. Two teams (A and B) were generated with identical characters (3 Fighters and 1 Spellcaster each), them 1000 combat simulations were done between both teams in these three specific situations, where Team A is always composed of SAI characters and Team B of AAIs with different LSs: SAI vs AAI (LS = 1), SAI vs AAI (LS = 3) and finally, SAI vs AAI (LS = 5).

These simulations were made in both fixed and dynamic map environments, with a grid size of 8 × 10 squares. As both results turned to be very similar, we present here only the results from the fixed map tests.

The expertise from SAI over AAI can be noticed in Fig. 1, as Team A maintain a better win rate over Team B through all simulations, having a 54% win rate at the last 100 simulations. This happens because LS = 1 is the lowest rate of Insights receiving, so AAI characters will take longer to converge to their Best Behaviors making so that they lose more against a team that is already quite experienced.

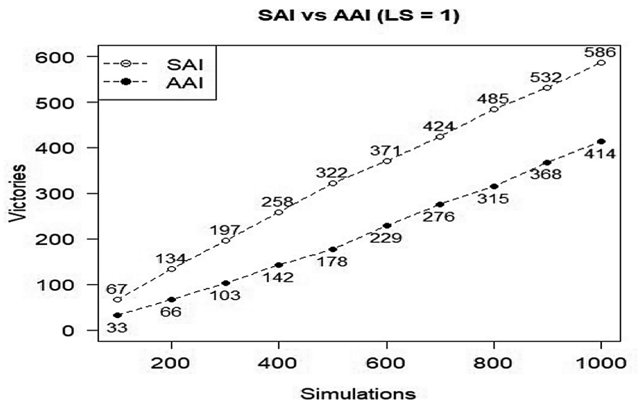

Fig. 1. Victories graph between a SAI team and an AAI team with LS = 1

The impact of LS is noticeable when we start using higher LSs. At LS = 3, the team composed of AAIs starts to overcome Team's A win count at around 300 simulations, as seen in Fig. 2. But still, the team with SAI put up a good competition, finishing 1000 simulations with only 10.6% less wins than the AAI Team, but with only 39% win rate in the last 100 simulations.

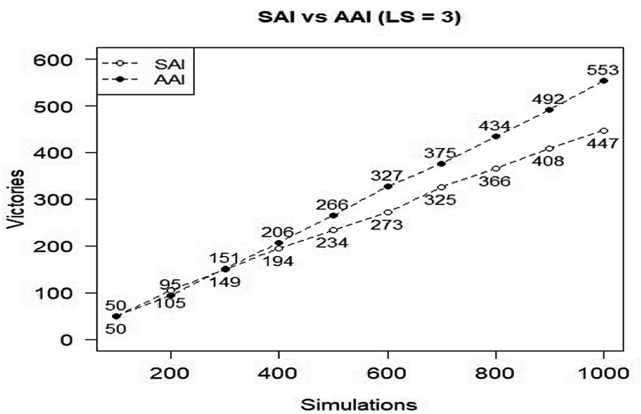

Fig. 2. Victories graph between a SAI team and an AAI team with LS = 3.

The impact of a higher LS from AAIs versus a fixed, but good, knowledge from SAIs can be totally perceived in Fig. 3, where LS = 5 is used. The team with AAIs bests their adversary even on the first 100 simulations, ending with over 35% more wins in comparison with Team A at 1000 simulations, that finish their last 100 simulations with only 25% win rate.

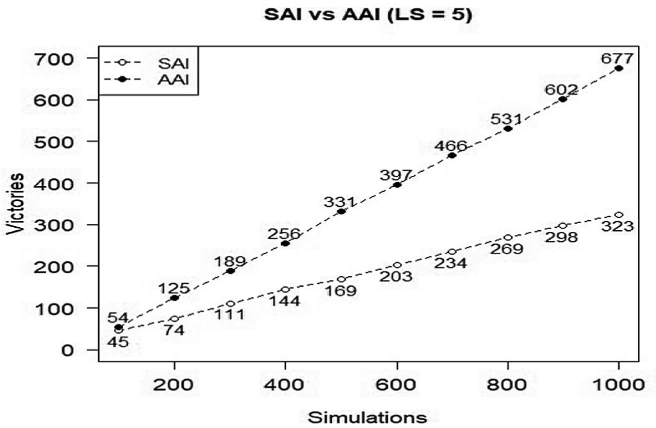

Fig. 3. Victories graph between an EAI team and an AAI team with LS = 5.

It is worth pointing out that with a sufficiently great number of simulations, an AAI Team always tend to overcome a SAI team using identical characters. Besides that, the exact victory count of each team can vary even in identical experiments since D&D is a game based in probabilities and dice rolls and SCURDD does represent it.

6 Discussion

Based on the experience gained with the work described in this paper, some insights can be taken about the AI modules presented and the hypothesis raised. First of all, using our described representations of an Skilled AI and an Adaptive AI for D&D v.3.5 players, we confirmed our hypothesis that a player with the ability to acquire reliable knowledge about the game could beat a more experienced player without this ability after a certain number of matches.

It was noticed that other than the rate of knowledge acquirement (Learning Speed), the overall complexity of a character's sheet and it's set of abilities directly impacts in the number of simulations needed to get to an ideal behavior set. This is noticeable in any field of work, since the effort to learn a subject is directly proportional to the complexity of this subject.

7 Conclusion and Future Work

In In this paper, we presented a model to represent a D&D player that is capable of learning not only through his experiences but through a reliable source of knowledge. This model, called AAI, was based in behaviors that abstracted the possible strategies of a player in battle. We then tested this model in our D&D combat simulation tool, SCURDD, against another group of characters controlled by a different type of AI.

Our idea was to test if a group of AAIs with no experience and a bad strategy set could learn through experience and external knowledge acquirement to beat a group of SAI characters that already have good experience but have no way to acquire knowledge. This way, our experiments demonstrated that, given a good rate of knowledge acquirement, the first group can prevail over the second, in a setting where both teams have identical characters and abilities.

However, the abstractions made in both AI models can have consequences in their performance. Further work is needed to investigate some of these issues, and testing the AI models performance versus groups of human players with different experiences level could be a good future experiment. Many abstractions were made to represent a D&D player within the constraints of our AAI and overall battle simulation, and there is still work to be done in order to improve the simulation tool, SCURDD, and our AI models to better represent real human players on a D&D combat environment. This future work includes optimization to implemented algorithms, usage of Neural Networks for training and evolution of player strategies and focus more on the role-playing aspect that influences player's choices in battle, such as his character's alignment or general way to deal with conflicts and certain situations.

References

1. Straffin, P.: Game Theory and Strategy, 5th edn. The Mathematical Association of America, United States of America (2004). https://doi.org/10.1137/1038146
2. Colman, A.: Game Theory & its Applications, 2nd edn. Routledge, London (2003). https://doi.org/10.4324/9780203761335
3. Laski, E.V., Siegler, R.S.: Learning from number board games: you learn what you encode. Dev. Psychol. **50**(3), 853–864 (2014)
4. Bochennek, K., Wittekindt, B., Zimmermann, S.Y., Klingebiel, T.: More than mere games: a review of card and board games for medical education. Med. Teach. **29**(9–10), 941–948 (2007). https://doi.org/10.1080/01421590701749813
5. Drake, P., Sung, K.: Teaching introductory programming with popular board games. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, pp. 619–624. ACM, USA (2011). https://doi.org/10.1145/1953163.1953338
6. Mason, P.: In search of the self: a survey of the first 25 years of anglo-american role-playing game theory. In: Montola, M., Stenros, J. (eds.) Beyond Role and Play: Tools, Toys and Theory for Harnessing the Imagination, pp. 1–15. Ropecon ry, Finland (2004)
7. Montola, M., Stenros, J.: Beyond Role and Play: Tools, Toys and Theory for Harnessing the Imagination, 1st edn. Ropecon ry, Finland (2004)

8. Chung, T.S.: Table-top role playing game and creativity. Thinking Skills Creativity **8**, 56–71 (2013). https://doi.org/10.1016/j.tsc.2012.06.002
9. DnD Encounter simulator GitHub page, https://github.com/matteoferla/DnD-battler. Accessed 29 Apr 2019
10. Yannakakis, G.N., Togelius, J.: A panorama of artificial and computational intelligence in games. IEEE Trans. Comput. Intell. AI Games **7**(4), 317–335 (2015). https://doi.org/10.1109/TCIAIG.2014.2339221