Universität Heidelberg
Institut für Informatik
Sommersemester 2019
Seminar Artificial Intelligence for Games
Dozent: Prof. Dr. Ullrich Köthe

Seminararbeit

Opponent modelling for case-based adaptive game AI

Name: Alexander Römelt
Matrikelnummer: 3482270
Studiengang: Master Angewandte Informatik (5. Fachsemester)
Email: Roemelt@stud.uni-heidelberg.de
Datum der Abgabe: August 6, 2019

Hiermit versichere ich, **Alexander Römelt**, dass ich die Seminararbeit mit dem Titel **Opponent modelling for case-based adaptive game AI** im **Seminar Artificial Intelligence for Games** im **Sommersemester 2019** bei Prof. Dr. Ullrich Köthe selbstständig und nur mit den in der Arbeit angegebenen Hilfsmitteln verfasst habe. Zitate sowie der Gebrauch fremder Quellen, Abbildungen, Texte und Hilfsmittel habe ich nach den Regeln guter wissenschaftlicher Praxis eindeutig als solche gekennzeichnet.

Mir ist bewusst, dass ich fremde Texte und Textpassagen nicht als meine eigenen ausgeben darf und dass ein Verstoß gegen diese Grundregel des wissenschaftlichen Arbeitens als Täuschungs- und Betrugsversuch gilt, der entsprechende Konsequenzen nach sich zieht. Diese bestehen in der Bewertung der Prüfungsleistung mit “nicht ausreichend” (5,0) sowie ggf. weiteren Maßnahmen.

Außerdem bestätige ich, dass diese Arbeit in gleicher oder ähnlicher Form noch bei keiner anderen Prüfung vorgelegt wurde.

Mainz, den 6. August 2019 A. Römelt

Abstract

This seminar paper discusses the paper "Opponent modelling for case-based adaptive game AI" by Bakkes, Spronck and van den Herik and covers general aspects of opponent modelling as well as the proposed architecture for a case-based adaptive game AI. After introducing basic concepts of opponent modelling and related approaches applied to all sorts of games, the components of the solution designed specifically for real-time strategy games are explained. The evaluation of the proposed system's performance through two experiments is then described before a more recent method by Farouk et al. is briefly outlined that includes more sophisticated ways of determining opponent models from features. The conclusion completes the report summarizing the beneficial effects of adaptive game AI and particularly incorporating opponent models.

Contents

1 Introduction **1**

2 Related Work **2**

3 Opponent modelling **3**
3.1 Variants of opponent models 3

4 Case-based adaptive game AI **4**
4.1 The case base 4
4.2 The game environment 5
4.3 The evaluation function 6
4.4 The adaptation mechanism 7
4.5 The opponent models 7

5 Experiments **8**
5.1 Setup 8
5.2 Results 9

6 Approach by Farouk et al. **9**

7 Conclusion **11**

1 Introduction

Game AI has come a long way since the first algorithms were able to play classical games such as Tic-Tac-Toe, Checkers or Chess. Nowadays, game AI is a central component in large-scale video games and contributes significantly to an authentic and immersive gaming experience. In addition to shooters or role-playing games, in which human interaction and confrontation or path finding must be simulated accurately, real-time strategy (RTS) games are particularly worth looking at. The RTS genre poses the challenge that AI agents have to process a large amount of data and at the same time make a number of strategic decisions, such as building economic infrastructure or executing military attacks simultaneously. Weber, Mateas and Jhala name particularly the challenges of decision-making under uncertainty, spatial and temporal reasoning, adversarial real-time planning and also the enormous decision complexity as key features for RTS games that make them interesting to study in the context of AI research [WMJ11].

One approach that has been used widely in game specific AI is opponent modelling. The main purpose of this concept is to represent the behavior of an opponent in a model, on the basis of which strategic decisions can be made by the AI. The model is hereby utilized in order to adapt the strategy to the opponent in a way that supports a desired outcome, which can for instance be most optimal play with regard to the goal of winning the game, or playing a challenging game against a human opponent while acting intelligent, but not unbeatable.

In this context, „Adaptive game AI“ is a broader term to be mentioned as it describes the general ability of AI agents to adapt to changing circumstances in the game environment. In this seminar paper, the paper *„Opponent modelling for case-based adaptive game AI“* by Sander C. J. Bakkes, Pieter H. M. Spronck and H. Jaap van den Herik, published in 2009 will be discussed. As the title suggests, different basic concepts of dynamically adaptable AI are combined in one approach. Adaptive game AI in the most different aspects was already extensively addressed by Spronck in his dissertation under van den Herik as doctoral supervisor [Spr05]. The presented concept is now complemented by a case base and mechanisms of opponent modelling. Both aspects will be described in detail in sections 3 - „Opponent modelling“, respectively 4 - „Case-based adaptive game AI“. The experiments carried out to evaluate the method are then discussed in the following section. Subsequently, an extended concept from a more recent research work will be presented. A general conclusion will complete the report. Preceding this, however, in the following section a brief overview of the history and related approaches of the topic will be given.

2 Related Work

According to [BSH09], opponent modelling has been considered an interesting research topic for a long time in the context of classic games. A simple version of an opponent model for a chess program from the 1970s is mentioned, in which a so-called "contempt factor" was implemented. The model was in this case suitable for estimating the opponents strength and making a decision on this basis about whether to accept or decline an offered draw. In 1983, Reibman and Ballard proposed a method of utilizing information about an opponent that explicitly aimed at the opponents fallibility and imperfect play [RB83]. In their work, they also introduce a notion of playing strength as a measure for performance in actual competition against an opponent that does not play a perfect strategy as opposed to theoretical scenarios. By considering the chance that an opponent performs a non-rational decision, incorporating a mechanism that makes a random proportion of imperfect choices, a rudimentary opponent model is established.

In 1993, research groups from Haifa and Maastricht independently proposed methods for opponent model search. Similar to Reibman and Ballard, Carmel and Markovitch from the Technion in Haifa argue that previous research has mainly been concerned with increasing the efficiency of game tree search while better utilization of information was neglected [CM93]. Their opponent model search approach is an algorithm capable of learning an opponent‘s strategy and integrating with a generalization of the minimax algorithm, which enables them to acquire accurate models and perform better than non-learning agents. Iida et al. from the University of Limburg in Maastricht and Delft University of Technology refer to the minimax principle as well, proposing an opponent model search strategy that models an opponent‘s evaluation function.

In 2001, Donkers, Uiterwijk and van den Herik propose a new approach for probabilistic opponent model search which incorporates uncertainty of a player about the opponent‘s behavior [Don01]. This is done by constructing multiple models as representations of different opponent types with diverging evaluation functions and a probability distribution over the types.

Other than the methods of Bakkes et al.[BSH09] in 2009 and Farouk at al.[] in 2013, which will be discussed in detail in the following sections, there have been some recent approaches of advanced opponent modelling techniques. In [GS11], the authors describe an approach that relies fundamentally on game theoretic reasoning and apply it to large imperfect-information games with an explicit focus on poker. Another attempt of establishing an artificial agent capable of playing poker and utilizing opponent models is presented in [SBLPBBR12], with the introduction of a Bayesian probabilistic model. Furthermore, opponent modelling approaches using machine learning have emerged, amongst others a method for modeling human behavior in strategic settings using deep learning [HWLB16] and a deep reinforcement learning powered opponent modelling approach [HBGKDI16].

3 Opponent modelling

The term „opponent modelling" generally refers to the process of building models as an abstraction of a player's behavior in games. Van den Herik et al. accordingly define an opponent model as „an abstracted description of a player or a player's behavior in a game" [Her05]. Farouk et al. state similarly, that an opponent model is „a generalization of his strategy" [FMA13]. In all cases, the purpose of the created models is the utilization for actual play [BSH09]. Bakkes et al. use the simple example of a rock-paper-scissors game to illustrate a basic opponent model in a game theoretical environment. If both players play an optimal strategy of choosing one of the three moves at random each round, they both have an equal chance of winning. If one of the player's was to choose another strategy, e.g. only choosing scissors in each round, the other player could obtain an advantage by modeling the opponents strategy and adapting his own strategy accordingly — in this case, play rock in each round.
In addition to opponent modelling applied in classic games, the topic has received attention in the context of modern video games. As opposed to game AI approaches that try to achieve maximum playing strength in order to perform as good as possible against human players, Bakkes et al. view the primary role of opponent modelling in raising the entertainment factor for the human player. This explicitly refers to an artificial agent's playing style, that tries to adapt to human performance not playing too strong to be beaten, but keeping difficulty on a challenging level.
Establishing opponent models in modern video games can be particularly challenging due to the complexity of the environment holding a huge amount of information that has to be considered and usually processed in a short period. Practical approaches must therefore not be too computationally expensive, since only a small part of the resources is available for the special application of opponent modelling within the game AI.

3.1 Variants of opponent models

For video games, there is a difference whether an opponent model models a player's *preferences*, or the concrete *actions* resulting from the fixed preferences. The preference-based approach

focuses on decisions the player to be modeled makes during the game and treats the selection of an appropriate model as a classification problem. In practice, this means that a large amount of data is collected during the game and clustered, resulting in different opponent models. Subsequently, in a new game, similarity measures can be applied to classify the current opponent regarding one of the available opponent models and adapt game AI accordingly.
Another two categories relevant for opponent modelling are the ones of *explicit* and *implicit* opponent models. In this context, explicit refers to the separation of a specification of an opponent‘s attributes and the game AI‘s decision-making process. In implicit opponent models, the attributes are represented by the parameters of the game AI being fine-tuned to the respective opponent model.
Also, the role of the opponent model does not have to be limited to pure opposition. Variants in which an AI agent in a companion role works together with the human player and adapts to his actions using a model of the player‘s behavior are feasible as well.

4 Case-based adaptive game AI

The AI system proposed by Bakkes et al. combines the main concepts of a game AI capable of adapting to changing circumstances in general, a case base which is used to extract constructive strategies and the opponent modelling feature to a complete system that can be applied to an RTS game. While adaptive game AI typically refers to systems with learning abilities implemented through machine learning techniques, this approach encounters some difficulties. The main problem with online learning, i.e. learning behavior about an opponent while the game is in progress, is that it usually takes too many learning trials to have a noticeable effect before either the game finishes or the respective character dies. This creates the need to access data collected offline that is immediately available during the game. A key element to accomplish this goal is the *case base* which collects a large amount of data from previous games and is therefore especially suitable for games that are permanently connected to the internet. The architecture of the resulting case-based adaptive game AI can be seen in figure 1. The most important individual components are briefly described in the following sections.

4.1 The case base

In general, the use of a case base is inspired by the Case-based reasoning paradigm. Case-based reasoning is strongly tied to the principle of learning from previous experiences represented in

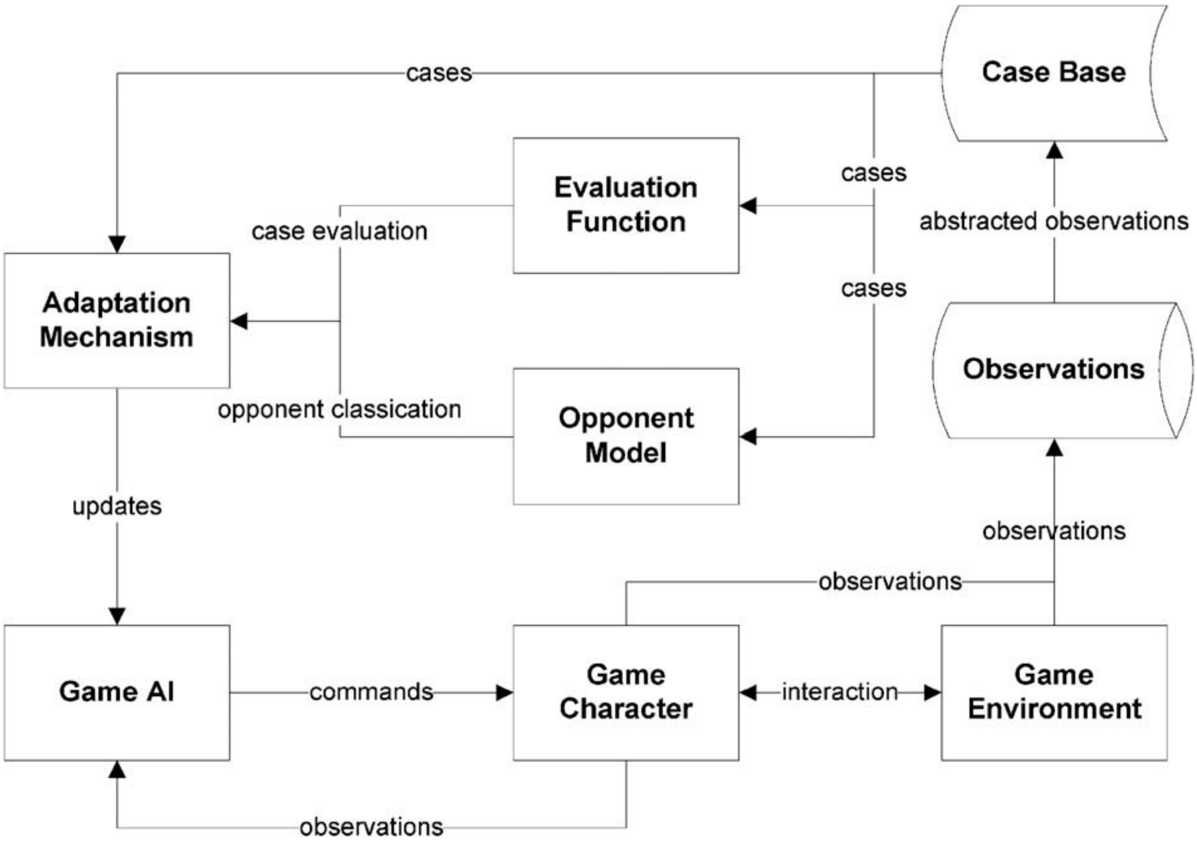

Figure 1: The architecture of the proposed case-based adaptive game AI[BSH09]

the form of cases [AP94]. In the present cases, the cases are, as depicted in figure 1 extracted from character and environment observations and brought into an abstract representation. The data in the case base is then structured in a standard format with a timestamp and made accessible for the adaptive game AI. As the number of games played increases, the case base grows accordingly. The main components derived from the case base are an *evaluation function* and the *opponent models*. Both are then subsequently utilized by the *adaptation mechanism* that updates the game AI.

4.2 The game environment

The game used by Bakkes et al. to implement and evaluate the case-based adaptive game AI is the real-time strategy game *Spring*. It essentially offers a highly customizable open source engine available on github[1] which is extremely suitable for research purposes. In 2019 there is still an active community based on the project‘s website[2] developing and modding

[1]https://github.com/spring/spring
[2]https://springrts.com

various aspects of the game. Its features are essentially standard repertoire of the genre including mining of resources, building and advancing infrastructure for economical and military purpose, creating different units and eventually destroying an enemy‘s commander unit in order to win a match.
The experiments conducted by Bakkes et al. were each carried out on three different maps with different conditions. In figure 2 an overview of the three maps „SmallDivide“, „TheRing“ and „MetalHeckv2“ can be seen. The three maps differ considerably in their tactical requirements. „SmallDivide“ offers only a narrow passage which has to be passed in order to get to the opponent. In "The Ring" both opponents can choose two round ways to attack and "MetalHeckv2" is characterized by widespread occurrences of metal resources.

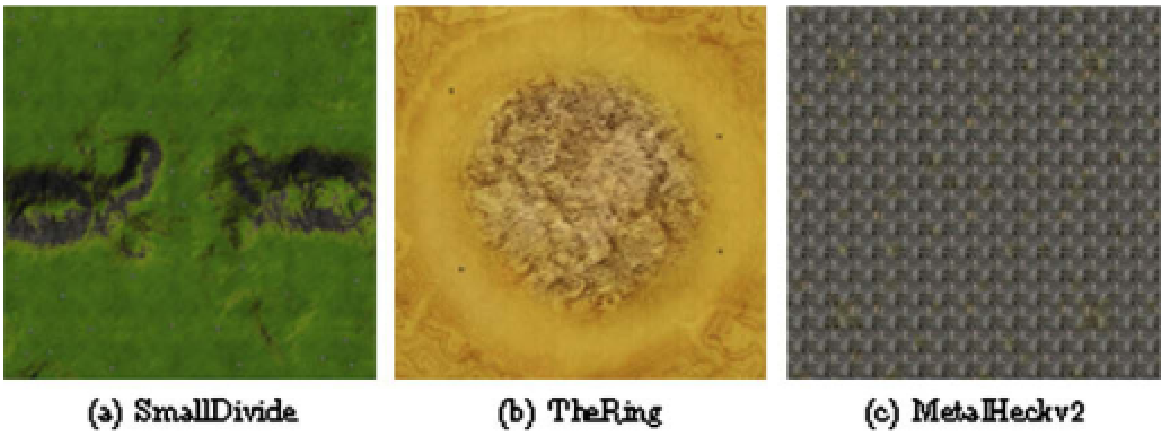

Figure 2: The three maps used for the experiments[BSH09]

4.3 The evaluation function

The evaluation function‘s purpose is express the current game situation in a value that indicates which player is to what extent likely to win the game. As indicated in [FMA13] the used evaluation function can be expressed by the following equation:

$$V(p) = W_p V_1 + (1 - W_p) V_2 \tag{1}$$

There are two evaluative terms where V_1 represents a player‘s material strength and V_2 the safety of the respective player‘s commander unit. The parameter p indicates the phase of the game, referring to a value that can e.g. represent a phase near the opening stage or the ending stage. The parameter $W \in [-1...1]$ is a free parameter representing the weight of the

Nr.	Feature	Meaning
1.	number of observed k-bot units	Global strategic preference
2.	number of observed tank units	
3.	number of observed air units	
4.	number of tech. adv. constructions	Technological development
5.	number of metal extractors	Economy strength
6.	number of solar panels	
7.	number of wind turbines	
8.	time of first attack on metal extractors	Aggressiveness
9.	time of first attack on solar panels	
10.	time of first attack on wind turbines	

Table 1: High-level strategic parameters for opponent modelling

evaluative terms.

4.4 The adaptation mechanism

The functioning of the adaptation mechanism is significantly characterized by two different phases, an *offline* phase and an *online* phase. In the offline phase, collected games are indexed and observations are clustered. Being online, i.e. during the game, the adaptation mechanism chooses a matching game strategy using similarity measures. As an appropriate strategy can not be chosen accurately from the beginning of the game, the AI is initialized with a previously successful game strategy. The strategy itself is expressed by the configuration of 27 different parameters that determine strategic behavior. For instance, the parameter MAX_STAT_ARTY equals the maximum number of stationary artillery that is build by the respective player. MIN_AIR_SUPPORT_EFFICIENCY indicates the minimum efficiency a hostile unit must reach for air support to be called.

4.5 The opponent models

In order to find an efficient way of building the opponent models, Bakkes et al. choose 10 high-level strategic parameters which are suitable from their experience for modeling a player‘s behavior accurately. The parameters and their meaning can be seen in table 1.

The generation of the opponent models relies on feature data clustered using the k-means algorithm. Determining differences in the feature data is done using the Euclidean distance

measure. The process of utilizing the resulting opponent models then follows a sequence from the offline preprocessing steps to the online strategy selection. During the offline phase, each game in the case base is labeled with collected information about the opponent. After the clusters have been identified, each opponent is assigned one of the clusters using the nearest neighbor method. A new game is then initialized with the cluster, respectively the opponent, that was observed most of the times in previous games in order to start with an opponent model that is approximately resembling the opponent the player is most likely to be pitted against. The game AI is accordingly initialized with a strategy that has proven itself against the particular chosen model in previous games.
During gameplay, also referred to as the online phase, the opponent model and corresponding game AI strategy are validated and continuously updated depending on the game state. The desired state is indicated by the value of the fitness function, therefore it can be derived whether the current game state is likely to lead to a win eventually, or if the strategy has to be changed. A main challenge in establishing an accurate opponent model is finding the right time. A sufficient choice has to be made early enough to have a significant impact on the game‘s result, but not too early to miss observations about important strategic choices of the opponent which allow choosing the best fitting model. Usually, accurate opponent models are established after about 150 game states which conforms to roughly 10 minutes of real-time play.

5 Experiments

5.1 Setup

In order to evaluate the performance of the proposed solution, a series of experiments is conducted. The general setup is based on the open-source game AI „AAI“ for the Spring game which was enhanced by the authors‘ case-based architecture with incorporated opponent models. Two different experimental setups were used independently in order to address the following two research questions:

1. How well does the case-based AI adapt to the original AI?
2. How well does the case-based AI adapt to a previously unobserved opponent?

For the experiments addressing the first research question, the original AAI was set to medium strength. To test the second research question, AAI was initialized with randomly generated strategies that can be viewed as an unseen opponent.
For each experiment, there were three different modes which should highlight the different

impact of the individual features added to the game AI. The first mode is a baseline for the purpose of comparing results and contains no adaptation mechanism at all. An initial strategy is selected randomly and there is no intelligent adaptation during online play. The second mode („basic mode“) implements case-based adaptive behavior by dynamically choosing an appropriate strategy, but without making use of opponent models. This is done by first selecting games with features similar to the current game and from this subset selecting games with an outcome that satisfies a goal criterion defined by the fitness value determined through the evaluation function. This process aims to find a similar game with a positive outcome and apply the respective strategy, that has proven to be successful in a similar situation. In the third mode, this adaptation is complemented by the consideration of opponent models, which is incorporated in the process of strategy selection described before. Additionally, in this mode adaptation takes place when predicted opponent models do not match the current opponent anymore and reclassification has to be performed.

5.2 Results

The results of the first experiment are depicted in table 2. Each mode was tested with 150 trials on each map, documenting the amount of games won against the respective opponent AI. As the results show, the incorporated opponent models lead to an increase in performance on all three maps. The effect is expressed most strongly on the map SmallDivide with 90% won matches compared to 77% in basic mode with only strategy adaptation, whereas on the other two maps, incorporating opponent modelling does not lead to an effect as strong, but sill considerable. The results for the second experiment, depicted in table 3, are similar for the maps SmallDivide and MetalHeckv2. For the map TheRing, there is no increase in performance noticeable with opponent models included. In both experiments, the game AI enhanced with the proposed solution is always performing better than the baseline.

6 Approach by Farouk et al.

As stated in [BSH09], there are several aspects in the proposed architecture that leave room for improvement. The features considered for modeling the opponents’ behavior are limited to the fixed amount specified beforehand. Enhancing the number of features or enabling the system to choose suitable features dynamically could lead to an improvement as more accurate opponent models would resemble actual players in a better way. Another imperfect aspect of the proposed method is the lack of weights that indicate the importance of a specific

Adaptation mode	Trials	Goal achv.	Goal achv. (%)
SMALLDIVIDE			
Disabled	150	59	39
Basic	150	115	77
OM	150	135	90
THERING			
Disabled	150	90	60
Basic	150	122	81
OM	150	127	85
METALHECKV2			
Disabled	150	70	47
Basic	150	124	83
OM	150	130	87

Table 2: Results of experiment 1 (original game AI)

Adaptation mode	Trials	Goal achv.	Goal achv. (%)
SMALLDIVIDE			
Disabled	150	71	47
Basic	150	96	64
OM	150	136	91
THERING			
Disabled	150	76	51
Basic	150	93	62
OM	150	93	62
METALHECKV2			
Disabled	150	54	36
Basic	150	60	40
OM	150	79	53

Table 3: Results of experiment 2 (random strategy)

feature.
An approach similar to Bakkes et al., but with further enhancements regarding dynamic feature selection and feature weighting has been presented by Farouk et al. in 2013[FMA13]. The main objectives of the proposed system are the ability of being applied generally to any RTS game, without needing knowledge specific for the game, robust adaptability to opponents that switch strategy and efficiency in terms of avoiding slow online learning processes. In analogy to Bakkes et al., there is an offline phase where a case base of previously collected game data is preprocessed and opponent models are clustered, and an online phase that handles continuous adaptation to the current opponent and switches opponent model and strategy on the basis of the evaluation function described before.
A key aspect of the newer approach is the ability to learn features from data collected from previous games that is highly distinctive for the respective opponent‘s behavior during the offline phase. Additionally, the learned features are weighted regarding their importance for the opponent model derived from the set of features. Both processes are assigned to a responsible module as a step in the offline preprocessing sequence. The feature selection module searches the relevant subset of features from a state space and determines the state with the highest accuracy to be considered for constructing the opponent model. Afterwards, the feature weighting module evaluates each feature and assigns weights representing the relative importance using the conjugate gradient method. Following the weighting process, the case base is clustered based on the selected set of features and their respective weights using the k-means algorithm. Compared to Bakkes et al., the extension of the offline phase should allow a higher accuracy of the constructed opponent models that resembles actual opponents more precisely.

7 Conclusion

The experiments conducted by Bakkes et al. show the superiority of a game AI enhanced with adaptation capabilities over a non-adaptive one, as well as the effects of incorporating opponent modelling into a case-based adaptive game AI. Both achieve a clearly better performance winning more often against the ordinary non-enhanced version of the game AI. With regard to opponent modelling, it must be noted that the effect is not equally pronounced in every case. In the second experiment there is in fact a case on the map TheRing where no improved performance can be observed compared to the basic mode. Bakkes et al. attribute this to the fact that opponent modelling primarily provides a greater improvement in highly strategic environments. Their assumption is, that improving the opponent models’ level of detail by enhancing features and weighting them could provide a remedy for this limitation. In this context, the approach by Farouk et al. could offer a solution to the shortcomings discovered. Nevertheless, the system proposed by Bakkes et al. has proved its effectiveness in many respects. The case-based approach avoids the shortcomings of resource-intensive online learning approaches. Also, opponent modelling has proven to be an effective method for abstracting an opponent's behavior and increasing the winning rate by adapting the performed strategy to the estimated model.

References

[AP94] Agnar Aamodt and Enric Plaza. "Case-based reasoning; Foundational issues, methodological variations, and system approaches." In: *AI COMMUNICATIONS* 7.1 (1994), pp. 39–59.

[BSH09] Sander C.J. Bakkes, Pieter H.M. Spronck, and H. Jaap van den Herik. "Opponent modelling for case-based adaptive game AI." en. In: *Entertainment Computing* 1.1 (Jan. 2009), pp. 27–37. ISSN: 18759521. DOI: 10.1016/j.entcom.2009.09.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1875952109000044 (visited on 03/15/2019).

[CM93] David Carmel and Shaul Markovitch. "Learning Models of Opponent's Strategy in Game Playing." In: *IN PROCEEDINGS OF THE AAAI FALL SYMPOSIUM ON GAMES: PLANNING AND LEARNING*. The AAAI Press, 1993, pp. 140–147.

[Don01] H Donkers. "Probabilistic opponent-model search." In: *Information Sciences* 135.3-4 (July 2001), pp. 123–149. ISSN: 00200255. DOI: 10.1016/S0020-0255(01)00133-5. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020025501001335 (visited on 08/02/2019).

[FMA13] G. M. Farouk, I. F. Moawad, and M. Aref. "Generic opponent modelling approach for real time strategy games." In: *2013 8th International Conference on Computer Engineering Systems (ICCES)*. Nov. 2013, pp. 21–27. DOI: 10.1109/ICCES.2013.6707164.

[GS11] Sam Ganzfried and Tuomas Sandholm. "Game theory-based opponent modeling in large imperfect-information games." In: *AAMAS*. 2011.

[HBGKDI16] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. "Opponent Modeling in Deep Reinforcement Learning." In: *arXiv:1609.05559 [cs]* (Sept. 2016). arXiv: 1609.05559. URL: http://arxiv.org/abs/1609.05559 (visited on 08/02/2019).

[Her05] H. Jaap van den Herik. "Opponent Modelling and Commercial Games." In: 2005.

[HWLB16] Jason S Hartford, James R Wright, and Kevin Leyton-Brown. "Deep Learning for Predicting Human Strategic Behavior." In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 2424–2432. URL: `http://papers.nips.cc/paper/6509-deep-learning-for-predicting-human-strategic-behavior.pdf`.

[RB83] Andrew L. Reibman and Bruce W. Ballard. "Non-Minimax Search Strategies for Use Against Fallible Opponents." In: *AAAI*. 1983.

[SBLPBBR12] Finnegan Southey, Michael P. Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. "Bayes' Bluff: Opponent Modelling in Poker." In: *arXiv:1207.1411 [cs]* (July 2012). arXiv: 1207.1411. URL: `http://arxiv.org/abs/1207.1411` (visited on 03/15/2019).

[Spr05] Pieter Hubert Marie Spronck. "Adaptive game AI." English. OCLC: 66511329. PhD thesis. Maastricht: UPM, Universitaire Pers Maastricht, 2005.

[WMJ11] Ben G. Weber, Michael Mateas, and Arnav Jhala. "Building human-level ai for real-time strategy games." In: *In 2011 AAAI Fall Symposium Series*. 2011.