

Nested Look-Ahead Evolutionary Algorithm Based Planning for a Believable Diplomacy Bot

Markus Kemmerling¹, Niels Ackermann², Mike Preuss²

¹ Robotics Research Institute, Section Information Technology, Technische
Universität Dortmund, Germany

² Chair of Algorithm Engineering, Computational Intelligence Group, Dept. of
Computer Science, Technische Universität Dortmund, Germany

Abstract. With regard to literature, improved estimations for the number of possible moves and placements are provided, showing that the complexity of Diplomacy is enormous, making it a good candidate for machine learning and evolutionary learning techniques. To enhance the playing strength of an existing Diplomacy bot and alleviate the distance to the presumed best current bot, a look-ahead planning component based on nested evolutionary algorithms, is then implanted into an already existing bot. The experimental investigation shows that the resulting bot is significantly improved.

1 Introduction

The game of Diplomacy is an interesting object of study for artificial intelligence techniques, as it may be regarded as pure multiplayer (on the standard Europe pre-World War I map, every game has 7 factions) strategy game with a real-time component. As moves are executed simultaneously, one cannot employ standard planning schemes as one could e.g. for chess. It is in most cases not possible to predict if a move one faction chooses can indeed be executed or not; it may conflict with the move of another faction.

However, this is only one side of the game and probably not the one that still makes it popular with human players although Diplomacy was invented already in 1954 by Allan Calhamer (see his book from 1999 [3]). The aspect that fascinates human players is the one of negotiation. Typically, every faction negotiates with every other, and is free to promise whatever sounds favorable, but not bound to any of its promises. In fact, betraying at least one ally when it is least expected seems to be an almost necessary tactic for reaching a dominant position in the game. In human games, a betrayal is only reasonable when the payoff is high and the betrayed does not get a reasonable chance to pay back the bad behavior. In AI-controlled Diplomacy bots, one usually does not get to

a point where such decisions are taken because many bots do neither possess a well developed negotiation module nor a long-term planning component.

Works on Diplomacy bots date back to the 1990's and usually focus on either the negotiation aspect (e.g. Kraus et al. [10] suggest the *Diplomat* bot which is modelled as *Multi-Agent System* (MAS)) or rather on the strategy aspect which is often approached with game theoretic techniques as suggested by Loeb [11] (introducing the *Bordeaux Diplomat*). After some years of standstill in publication history, Shapiro et al. [14] present their bot concept of bottom-up self-learning Diplomacy Strategy by means of *Temporal Difference Learning* (TDL) and pattern-weights. Shaheed [13] suggests another bot named *Diplomat*, combining strategy and negotiation with an opponent model, but without look-ahead into the future. Booijink [2] establishes an evolutionary algorithm based strategy forming module for his MAS representation of Diplomacy, without negotiation component.³ Johansson and Håård [8] also focus on the no-press variant, suggesting the HA AI bot, which establishes strategies by valuing provinces and unit threats and also provides different bot characters (berzerk and vanilla) with different properties in range of sight and blurring strength⁴. Johansson [7] further discusses similarities and differences of different strategic board games as Diplomacy and Risk and the consequences for the bot design.

Currently, the interest in Diplomacy as game and/or testbed for agent models seems to be renewed, as Fabregues and Sierra [4, 5] state. Presumably, this is because modern AI techniques have improved to a level that makes them more capable to deal with negotiation and opponent modeling in near real world situations. This direction has also been taken up by others. Where Ribeiro et al. [12] mainly focus on the different personalities a bot may express, Kemmerling et al. [9] provide the *Stragotiator*⁵, a bot that has been reported to sometimes trick humans in a Turing test (at least for short games). It has been developed with focus on believability and negotiations, modeling a specific human playing style described by Windsor [16] as *Classicist*. The Classicist is a loyal team-player, tries to maintain the groups welfare and does not violate contracts. While the Diplominator [15] was used as initial code-frame, it was enhanced according to the archetype of the Bordeaux Diplomat [11] and by using an *Evolutionary Algorithm* (EA) for movement optimization. However, the best playing no-press bot to date is van Hal's *Albert* [6] which uses a mixture of clever pruning techniques in a main algorithm that may be described as competitive coevolution.

In this work, we pursue two goals. After giving a small primer concerning the Diplomacy rules, we establish an improved estimation of the complexity of no-press Diplomacy in terms of movement possibilities. Secondly, we implant a look-ahead planning method into the Stragotiator bot and experimentally assess its success against the old Stragotiator and the best currently available bot, Albert.

³ Diplomacy games without negotiation are usually called *no-press* games

⁴ Blurring distributes province values to the neighboring provinces by gradual decrease

⁵ the Stragotiator is available at http://www.irf.tu-dortmund.de/cms/de/IT/Mitarbeiter/Wissenschaftliche_Mitarbeiter/Kemmerling.html

2 Diplomacy Primer

The board game Diplomacy⁶ is a strategy war game focusing on negotiations between players who may negotiate moves, contracts and alliances which are not binding. Action takes place in Europe just before World War I. The goal is to conquer most of Europe either alone or together with allies. The board is divided into 56 land and 19 sea regions (provinces) whereas 34 land regions contain supply centers, which can support one army or fleet each. Each of the players commands the entities of one the powers Austria (AUS), England (ENG), France (FRA), Germany (GER), Italy (ITA), Russia (RUS), and Turkey (TUR). The game ends when a single power or an alliance of countries attains supremacy with at least 18 supply centers.

Diplomacy consists of 2 rounds (spring and fall) per year, starting from 1901. Each round consists of a negotiation and a movement phase (negotiation phases are disabled in no-press games). In every round, a unit can either hold its position, move (which is an attack if the move goes to a region that is neutral or belongs to another faction), or support another unit in moving or holding. All moves are simultaneously executed. As in each province one unit can remain at most, conflicts are resolved by the majority rule, that is, a province can be conquered only with a superiority of units against an enemy. New supply centers may be conquered in fall moves only and the number of units is adapted to the number of supply centers in winter. New units can only be built in one of the 22 home supply centers, these are the ones occupied at the beginning of the game. The game does not incorporate any random effects.

Several computer based versions of Diplomacy exist, more or less supporting the development of Artificial Intelligence (AI). The best known of these is the Diplomacy AI Development Environment (DAIDE)⁷ that consists of a communication protocol and a language (syntax and semantics) for negotiations and instructions. The diplomacy server (AIserver) and a human interface program (AImapper) allow for games played by bots and humans.

3 Diplomacy Complexity

Even without considering the negotiation phase, simultaneous moves of 7 players undoubtedly make Diplomacy a complex game in terms of possible moves, despite its simple rules. Booijink [2] already reasons that Diplomacy is more complex than *Go*, but we have not yet seen a more concrete estimate rather than for the first move, which is given by Loeb [11] as $\approx 4.4 \cdot 10^{15}$, and for the szenario of all units which is estimated by Shapiro [14] to $\approx 1.7 \cdot 10^{27}$ possible moves. Making use of some simplifying assumptions, we develop a more accurate estimate for any number of units in the following. Such estimates are useful for setting up any look-ahead planning, as different branching factors will require different techniques.

⁶ <http://www.wizards.com/default.asp?x=ah/prod/diplomacy>

⁷ <http://www.daide.org.uk>

3.1 Total Positions on the Map

In standard Diplomacy, 75 regions exist, of which 19 are sea, 42 coastal, and 14 interior provinces. Equation (1) gives the number of possibilities to place units on the map, ignoring unit affiliations to factions and using exchangeability of individual units. However, coastal provinces may be occupied either by a fleet or an army, which has to be recorded as it affects the result. Interior provinces may only hold an army, whether sea provinces are only allowed to fleets.

$$P(n) = \sum_{u=0}^{\max\{n,33\}} \binom{42}{n-u} \cdot 2^{n-u} \cdot \binom{33}{u} \quad (1)$$

$P(n)$ gives the number of placements of $n \leq 34$ units without repetition, $u \leq 19 + 14 = 33$ the number of units on non-coastal provinces ($n - u$ on coastal provinces). On non-coastal provinces, the unit type is determined by the region type, otherwise 2 possibilities exist. Placing u units on 33 provinces without replacement and order is given by $\binom{33}{u}$, the remaining $n - u$ units have to be distributed to the coastal provinces, $\binom{42}{n-u}$ where 2^{n-u} is the number of possible army-fleet combinations. For $n = 34$, (1) results in a number of $4.09 \cdot 10^{27}$.

3.2 Unit Moves

During the movement phase, a unit may either execute a hold, move or support command, and in case of a fleet (on a sea region) also a convoi command. As hold commands are always valid, the number of moves depends on the number of neighboring provinces, and the possible support moves for units in these provinces. In the following, we presume possessing complete move information concerning units taking part in any support or convoy operation. Within the game, it is also possible to give support to units without informing the respective faction beforehand. However, we ignore this case and also disregard convoys. Implementing techniques to enable a bot to execute convoys is a very complex task and not done in the majority of bots including the Stragotiator.

Setting up some definitions enables us to establish a movement combination formula. Let p_i be the set of neighbors of any province i and $\varphi_{i,l} = \{p_i \cap p_l\} \setminus \{i, l\}$ the set of common neighbors of provinces i and l . Without considering convoys, a unit may thus execute $M(i)$ different moves as expressed by equation (2): We add up one hold 'move' plus $|p_i|$ moves to neighboring provinces, plus a hold support for all occupied neighboring provinces, plus a move support for all common neighbors of i and l , plus move supports for indirect neighbors. The set $\{j \in p_k \setminus p_i | j \neq i\}$ with $k \in p_i$ characterizes the indirect neighbors of i (excluding $j = i$, the province itself, and the direct neighbors p_i). Employing operator δ_i (gives 1 if the province is occupied and 0 otherwise), we sum over all neighbors $k \in p_i$, conditional on occupation ($\delta_j = 1$), in this case $\varphi_{i,j}$ move supports are possible.

$$M(i) = 1 + |p_i| + \sum_{l \in p_i} \delta_l \cdot (|\varphi_{i,l}| + 1) + \sum_{k \in p_i} \sum_{\{j \in p_k \setminus p_i | j \neq i\}} \delta_j \cdot |\varphi_{i,j}| \quad (2)$$

Using the mean number of neighbors ($|p_i| = 4.31$), common neighbors of neighbors ($|\varphi_{i,l}| = 1.91$), and common neighbors of indirect neighbors ($|\varphi_{i,j}| = 1.41$) and assuming that 33 other units are placed on the remaining 74 provinces, we obtain $M(i) = 7.24$.

4 Evolutionary Algorithm-Based Move Planning

At first, an EA is used to obtain the best move combinations for the controlled units and allies (utilizing a map and move evaluation function similar to the one of [9]). At most 20 move combinations with a value of at least 85 % of the best obtained combination are saved. For all remaining units, we then also use an EA to compute the best move combinations for every single faction separately or for all factions at once (this is later on referred to as *option-1*, meaning that we assume that all enemies and neutral factions play together). Of course, this step is essential for a good look-ahead planning, and mistakes at this stage may lead to bad plans. According to parameter tests, we employ a (30+60) evolution strategy with mutation rate $\frac{5}{n}$. The genome simply consists of one move per unit, and a mutation means to randomly generate a new valid move for one unit.

Further on, the outcome of each move combination saved in the first step is calculated regarding the obtained enemy moves. Then, several EA instances are adopted to calculate for all distinguished board states the next move combination for own and allied units. To distribute computing time to the different EA instances, a component named EA-controller is used. Starting with an initial number of $\max\{1, \lfloor \frac{400}{|EA|} \rfloor\}$ generations after initialization, the single EAs self-terminate if they cannot obtain improvements for some generations. After the initial round, the controller sorts the still active EAs according to their current best move combination and provides them one after the other (as long as there is at least one active EA and there is still time available) with another $\max\{2, \lfloor \frac{200}{|EA_{active}|} \rfloor\}$ generations of computation time. The value 2 is a result of manual adjusting during parameter tests. $|EA|$ is the number of all EAs and $|EA_{active}|$ is the number of EAs not self-terminated.

Selecting one of the active EAs for further computation employs a random component and a quadratic weighting (EAs with the better results get more chances for another round). All active EAs get a number i according to their reversed ranks, so that the current worst obtains $i = 1$. Now we assign the interval $[\frac{(i-1)i}{2}, \frac{i(i+1)}{2}]$ to each i and draw a uniformly distributed random number r from $[1, \frac{n(n+1)}{2}]$. The EA that belongs to the interval covering r is executed next. This procedure stops if the stopping time is reached (externally provided by the game server) or no active EA remains.

In summary, several move combinations for own and allied units are calculated, selecting the one that leads to the most advantageous game state for the next round regarding a specific opponent behaviour. The goodness of a game state is measured by accumulating the evaluation values of the move combination resulting in this state and the best move combination calculated for the next round based on this state.

5 Planning vs. Non-Planning Stragotiator

The first versions of the Stragotiator exhibited a near human-like behavior, but was easily beaten by other bots. Kemmerling et al. [9] increased the playing strength by adapting an EA for move optimization. Here, we provide a further enhancement by planning one step ahead. To investigate the effect of planning, we compare the playing ability of planning and non-planning Stragotiator.

Pre-Experimental Planning. In order to validate the implemented planning technique, an appropriate opponent for both the planning and the non-planning Stragotiator had to be chosen. Albert [6] is well-established in the DAIDE community and probably the best currently available bot concerning playing strength. We therefore use it as a reference.

A drawback of the Albert bot is the amount of time needed to calculate and submit orders. As result of various test games we decided to use 45 sec. for movements, 5 sec. for retreats and 3 sec. for builds as time limits. In games with one planning Stragotiator and 6 Alberts, this time limits are sufficient for all bots to submit orders and, in case of the planning Stragotiator, to finish planning. "Essentially Albert will deeply think about each power's best moves and in an iterative process make adjustments to the probability of any order set being selected for each power" [6]. This "thinking depth" can be adjusted and defaults to 50 (denoted as "-d50"), affecting not only the time needed for calculation but also the playing strength. According to Albert's README -d20 or -d30 should already yield good results, and still be relatively quick. Therefore, we investigate the parameters -d20, -d23, -d26 and -d30 in order to find a setting enabling Albert to submit orders within given time limits and to provide an opponent that is neither too strong nor too weak, which is important to show the effect of the Stragotiator's planning. Figure 1 shows the supply centers owned by the Stragotiators playing one against 6 Alberts, 20 games for each combination of planning (Niels) and non-planning Stragotiator (Markus) as AUS. One can easily see that the Stragotiator performs the best against Albert with "thinking depth" -d26. Nevertheless, Albert -d26 is still challenging as the Stragotiators lose centers in the end of the games. Because all Stragotiators are able to conquer and defend centers in the beginning and obviously do not dominate the games, we consider -d26 as appropriate setting.

According to Ackermann [1] the best strategy for the planning Stragotiator playing AUS is to consider all enemies as one single power (option-1). For all other powers, the opponents move should be predicted separately. We first conduct games with one Stragotiator (planning and non-planning) against 6 Alberts and compared the supply centers owned by the two Stragotiators, using option-1 only for AUS. The results lack the desired superiority of the planning Stragotiator with exception of AUS. Therefore, we used option-1 for all powers in the following experiments.

Setup. In each game, one Stragotiator plays against 6 Alberts -d26. All powers are played by the non-planning (Markus), the planning (Niels) and the planning Stragotiator with option-1 (Niels paranoid) for at least 49 times. Games are

conducted with 45, 5 and 3 sec. as time limits for movements, retreats and builds, respectively, comprises of 6 years 1901 to 1906 divided in a startup phase (1901, 1902), ruling phase (1903, 1904) and end phase (1905, 1906). The Stragotiators are compared by the supply centers owned. To give a neutral impression of the playing capabilities of our bots, we also conducted 300 games with all powers played by Albert -d26.

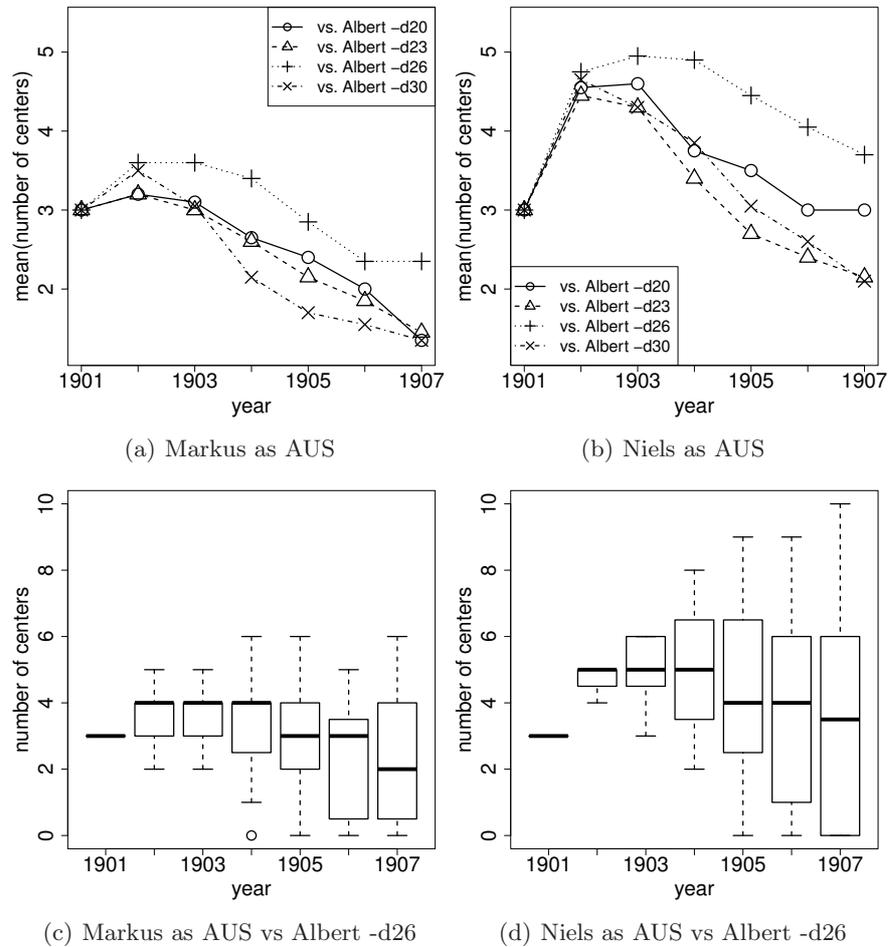


Fig. 1. (a) and (b) show the mean of supply centers owned by the planning (Niels) and the non-planning Stragotiator (Markus) while playing against 6 Alberts over 20 games each. (c) and (d) show exemplarily the distribution of supply centers owned by the Stragotiators while playing against Albert -d 26.

Task. We require that the planning Stragotiator performs better than the non-planning Stragotiator at least for some powers and does not performs worse for all powers.

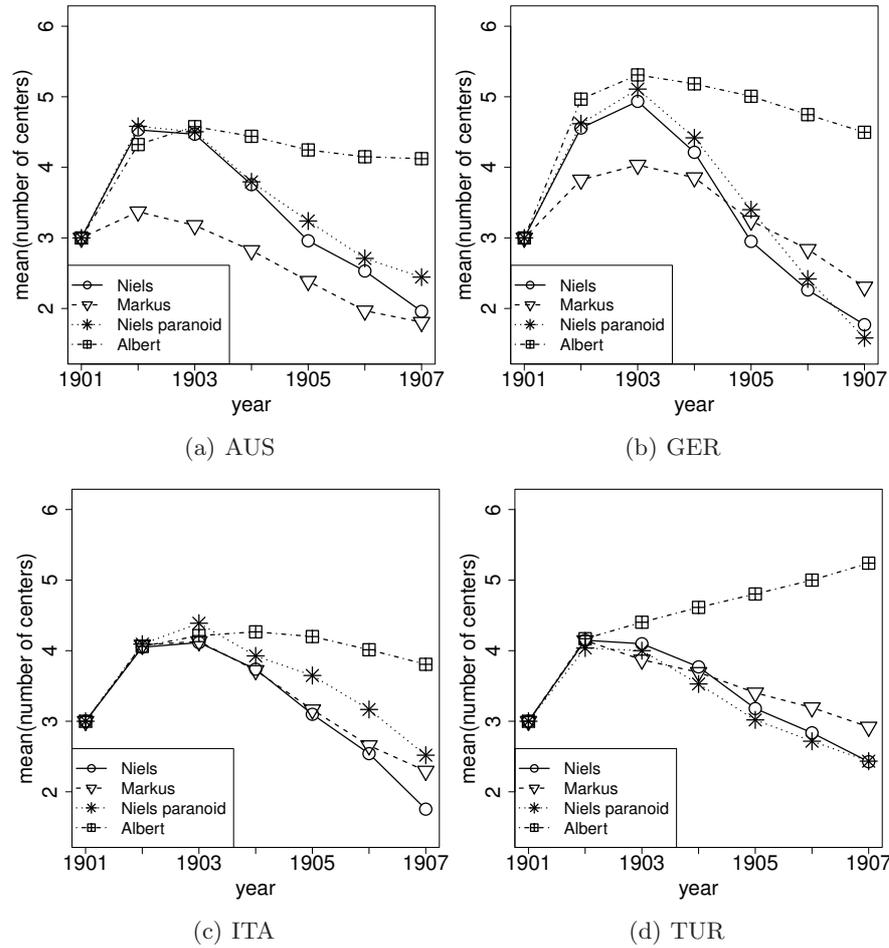


Fig. 2. Mean of supply centers owned by the non-planning (Markus), the planning (Niels) and the planning Stragotiator with option-1 (Niels paranoid) while playing against 6 Alberts -d26.

Results/Visualization. Figure 2 shows the mean of supply centers owned by the different Stragotiator types and the Albert (all playing against 6 Alberts). The figure depicts the ownership at the beginning of a year. Since the ownership changes at the end of a year, we included 1907 in order to show the ownership at the end of 1906. RUS, ENG, and FRA are not pictured, as they exhibit characteristics similar to TUR, ITA, and AUS respectively regarding the course of centers owned by the Stragotiators. Figure 3 shows exemplarily the distribution of supply centers for the non-planning (Markus) and the planning Stragotiator with option-1 (Niels paranoid) when playing FRA..

Observations. With exception of TUR and RUS, figure 2 clearly shows the advantage of the planning Stragotiator in comparison with its non-planning coun-

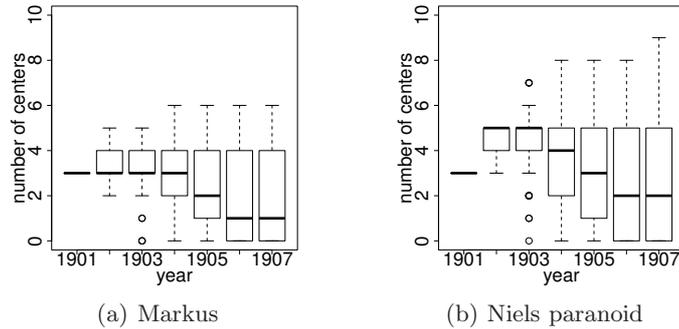


Fig. 3. Distribution of supply centers owned by the non-planning (Markus) and the planning Stragotiator with option-1 (Niels paranoid) while playing with FRA against 6 Albert -d 26.

terpart. But due to the small gap, we can not conclude an advantage of the non-planning Stragotiator for TUR and RUS. For ITA and TUR both planning versions exhibits a greater variance of the owned centers than Markus. While playing GER, the planning capability results in clearly more conquered centers. Unfortunately, both planning Stragotiators loose their centers much faster than Markus during the ruling and the end phase. For AUS, ENG, GER, RUS and TUR, no difference can be observed between Niels and Niels paranoid. In particular, this means that option-1 does not decrease the playing strength. But for FRA and ITA, the usage of option-1 can obviously be advised as Niels paranoid dominate both other bots while Niels without option-1 is slightly dominated by Markus. The Albert performs better, on average, than the Stragotiators during the ruling and the end phases. However, in the startup phase, the average number of conquered centers is mostly the same for the planning Stragotiator and the Albert.

Discussion. Our guess is that the planning Stragotiator assumes moves for his opponents in the same way he calculates his own moves. This implies the assumption, that the opponents are non-planning Stragotiators. This assumption is weakened by enabling option-1. Therefore, the decrease in playing strength for ITA and FRA when not using option-1 may be explained. Considering the results obtained with option-1, we have considerably improved the playing capabilities of the Stragotiator. Especially for AUS and ENG, Niels paranoid exhibits much greater playing strength than Markus. This is also the case for GER, at least in the startup phase. In comparison with Albert, we can conclude that Niels shows equal playing strength in the startup phase. But there is still need for improvements during the ruling and the end phase, especially for FRA and TUR.

6 Summary and Conclusions

The two main goals of this paper have been achieved: We have provided an improved move/placement complexity estimation for Diplomacy, and, probably

more important, have shown that enhancing an existing (already quite believable) bot with a look-ahead planning mechanism improves its playing strength. However, more questions have turned up in the course of this investigation, the most important ones being: How to predict opponent moves with some reliability if the opponent has a very different playing style? This is of course essential for any look-ahead planning. And, what are the effects of the different powers in Diplomacy? Does it make sense to adapt a bot to playing a specific power?

References

1. Ackermann, N.: Evolutionäre Spielbaumsuche für das Strategiespiel Diplomacy. Dipl. Inf. Diploma Thesis, TU Dortmund, Department of Computer Science (2010)
2. Booijsink, B.: Strategy Evolution and Resolution Deduction in Diplomacy. Master's Thesis, University of Groningen, Artificial Intelligence (2005)
3. Calhamer, A.B.: Calhamer on Diplomacy: The Boardgame Diplomacy and Diplomatic History. Authorhouse (1999)
4. Fàbregues, A., Sierra, C.: Diplomacy game: the test bed (2009), <http://www.perada-magazine.eu/view.php?source=1761-2009-08-03>
5. Fàbregues, A., Sierra, C.: A testbed for multiagent systems. Tech. rep., IIIA-CSIC, Bellaterra, Barcelona (10/2009 2009)
6. Hal, J.v.: Diplomacy AI - Albert. <http://sites.google.com/site/diplomacyai/albert> (2010), <http://sites.google.com/site/diplomacyai/>, 20. November 2010
7. Johansson, S.J.: On using multi-agent systems in playing board games. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. pp. 569–576. AAMAS '06, ACM (2006)
8. Johansson, S.J., Håård, F.: Tactical coordination in no-press diplomacy. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. pp. 423–430. AAMAS '05, ACM (2005)
9. Kemmerling, M., Ackermann, N., Beume, N., Preuss, M., Uellenbeck, S., Walz, W.: Is human-like and well playing contradictory for diplomacy bots? In: Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2009). pp. 209–216 (2009)
10. Kraus, S., Lehmann, D.J.: Designing and building a negotiating automated agent. Computational Intelligence 11, 132–171 (1995)
11. Loeb, D.E.: Challenges in multi-player gaming by computers. The Diplomatic Pouch Zine (S1995M) (1995)
12. Ribeiro, J.a., Mariano, P., Seabra Lopes, L.: Darkblade: A program that plays diplomacy. In: Proceedings of the 14th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence. pp. 485–496. EPIA '09, Springer (2009)
13. Shaheed, J.: Creating a Diplomat. Master's Thesis, Imperial College, London, UK (2004)
14. Shapiro, A., Fuchs, G., Levinson, R.: Learning a game strategy using pattern-weights and self-play. In: In Proc. of the Third International Conference on Computers and Games (2002)
15. Webb, A., Chin, J., Wilkins, T., Payce, J., Dedoyard, V.: Automated negotiation in the game of diplomacy. <http://www.daide.org.uk/external/TheDiplominator.pdf> (2008), 30. November 2010
16. Windsor, P.D.: What's your point. The Diplomatic Pouch Zine (S1999M) (1999)