

# Tactical Coordination in No-press Diplomacy

Stefan J. Johansson  
School of Engineering  
Blekinge Institute of Technology  
Soft Center Box 520, 372 25 Ronneby, Sweden  
sja@bth.se

Fredrik Hård  
Softhouse  
Campus Gräsvik  
Karlskrona, Sweden  
fredrik.haard@softhouse.se

## ABSTRACT

While there is a broad theoretic foundation for creating computational players for two-player games, such as Chess, the multi-player domain is not as well explored. We make an attempt to apply a multi-agent approach to a multi-player game with huge search spaces and multiple adversaries, namely no-press Diplomacy. We tested our solution against other available bots in an open competition and show that our solution outperforms its competitors in score while being competitive in speed.

## Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Coherence and Coordination, Multiagent Systems*

## General Terms

Algorithms, Experimentation

## Keywords

Diplomacy, Multiagent system, Contract net, Game bots, Board games

## 1. INTRODUCTION

The creation of computational players (in short: *bots*) for two-player games is a well known domain of computer science and mathematics. For games with manageable search spaces, these methods include extensive searching for possible moves, and weighting of found positions to find the best move from the current position. Such methods can then be improved by using heuristic methods, such as *Iterative deepening* [19] and *Alpha-Beta pruning* [8, 11]. However, such solutions are not applicable to all kinds of games. In Diplomacy<sup>1</sup>, the search space is huge — the exact number

<sup>1</sup>Diplomacy (©Hasbro Inc.) as described here is the "No-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.  
Copyright 2005 ACM 1-59593-094-9/05/0007 ...\$5.00.

of unique openings is 4,430,690,040,914,420 (not counting so called useless supports) [13]. This clearly makes exhaustive search intractable even one step ahead (not to mention to search in a search tree with a branching factor of  $4.4 \cdot 10^{15}$ ). This makes Diplomacy a very interesting game to study, since the uncertainty and unmanageable search spaces are properties that it shares with the real world, while the software environment and unambiguous rules still make it a viable domain for experiments.

This work explores the possibility of using a multi-agent architecture to create a Diplomacy bot, in an attempt to discern whether a distributed solution can successfully compete with centralized solutions in games with high complexity and huge search spaces.

Our method consists of the development of a Multi-Agent System (MAS) based bot for Diplomacy, and evaluation of the performance of that bot compared to existing bots, by arranging a Diplomacy tournament.

In the following section, Diplomacy will be described and the bot environment will be introduced. We also briefly describe some previous attempts to create Diplomacy bots. In Section 3, the bot developed is described, and in Sections 4–5 the experiment setup and the results are presented. We finish by discussing our results, draw some conclusions and point out some possible directions for future work.

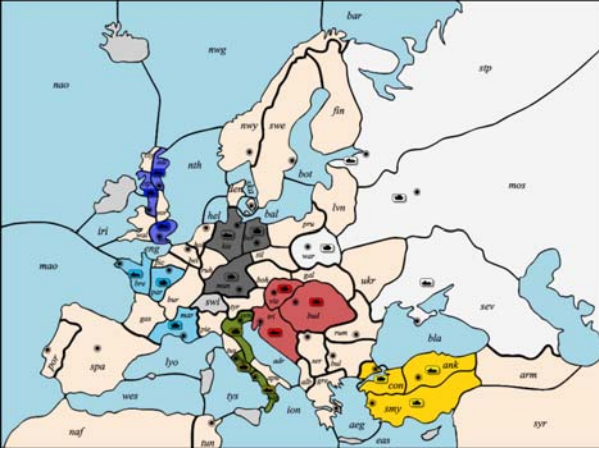
## 2. DIPLOMACY AI DEVELOPMENT ENVIRONMENT (DAIDE)

Diplomacy is played on a map resembling Europe in 1901 where the overall goal is to take control over Europe by leading England, France, Germany, Russia, Turkey, Italy or Austria-Hungary to the victory, see Figure 1.

The different types of provinces are defined in how they interact with units. There are two kinds of units: *fleets* and *armies*, denoted 'F' and 'A' respectively. Armies can only traverse land provinces and only over land borders. Fleets on the other hand may move only in sea and coastal provinces, and may never enter a landlocked province.

Some (land) provinces are *supply centers* - marked with a dot on the map. Each such supply center allows its owner to construct and maintain one unit - in other words, the maximum number of units a player can control is the number of supply centers it controls.

press Standard" variant - it is played on the original map, but without negotiations. More information on Diplomacy variants can be found in [4] and [5]



**Figure 1: The Diplomacy map in the beginning of a game.**

Season	Activities
Spring movement	Move, Hold, Support, Cut support
Summer retreats	Retreats, Disbands
Fall movement	Move, Hold, Support, Cut support
Winter retreats	Retreats, Disbands
Winter adjustments	Owners build new units

**Table 1: The activities in the Diplomacy seasons**

Turns in Diplomacy are measured in *seasons* (or phases) which are: *spring movement*, *summer retreats*, *fall movements*, *winter retreats* and *winter adjustments* (or *winter builds*). During each *movement* season, all players submit orders for their units. During *retreats*, units that have been dislodged are retreated or disbanded, and during the adjustment phase center ownership is decided and units are built or disbanded. When the game begins, all players control three centers except Russia, which controls four. Initially all players have units on those control centers. These centers are considered the players' *home centers*. The significance of home centers is that they are the only places where new units can be constructed.

A player can take control of an uncontrolled center or a center of another player by occupying it with a unit during the winter adjustment phase. That player then controls the center until another player takes it over in a similar fashion.

The aim of the game is to dominate Europe by controlling at least 18 centers - the first player to do so is the winner (by a *solo* victory). Games ended prematurely (i.e. after a predefined number of seasons) are always considered a draw between all surviving players.

## 2.1 Orders

Units can in a movement season only move to, and interact with, units in provinces bordering the province they reside in. A fleet may not directly interact with units in landlocked provinces, and armies may not interact with fleets in a sea. Any unit can also be ordered to *hold* — stand ground, or

to *support* the action of another unit, see Figure 2. In any season, orders for all units are entered secretly by each player and then revealed and carried out simultaneously.

All units are of equal strength, and whenever two units try to enter the same province, a standoff occurs (*bounce*) and the involved units do not move. This also happens when a unit tries to enter a province where there already is another unit (that is not moving away). A unit can give support into any province it can move to. Support is given either to *hold* (defend) or *move* (attack) and is only valid if the supporting unit is not attacked itself - even if that attack is unsuccessful. The act of attacking a supporting unit is known as *cutting support*. If a supported unit moves to a province where there is an unsupported unit, the attacked unit is *dislodged*. During the following retreat phase, dislodged units must retreat to an unoccupied province other than the one the attacker came from, or (if a retreat is not possible) be *disbanded*. The general rule of movements is that equal strength bounces, superior strength prevails. For a more in-depth discussion of the rules and their implications, we refer to the rule book [2] and/or *The Game of Diplomacy* [18].

## 2.2 Previous Approaches to Diplomacy Bots

There have been several previous attempts to create an automated Diplomacy player:

*The Israeli Diplomat* (by Kraus and Lehmann) was primarily concerned with the diplomatic aspect of the game [12]. In no-press Diplomacy the possibility to negotiate at the player level is removed, while in ordinary Diplomacy, this aspect is a major part of the game. The Israeli bot was reportedly quite successful — it played better than its human counterparts. It used an agent based approach, and distributed tasks between agents that were ordered in a hierarchical fashion.

*The Bordeaux Diplomat* (by Loeb) was based on an optimized best-first searching algorithm, seeded with best-guess moves [14]. It used scripted "book openings" to increase the performance, and an evaluation method that created areas of varying importance that the bot should try to control. The strategic and tactical planning seemed to be done through searching with heavy pruning to offset the huge search space.

*The LA Diplomat* (by Shapiro, Fuchs and Levinson) uses pattern-weights and temporal difference (TD-) learning to learn successful strategies in playing no-press Diplomacy [17]. By playing a large amount of games against itself, it managed to learn some strategic aspects of the game although it started without any such knowledge.

None of the above bots were available in versions compatible with the DAIDE system on which we ran the experiments and they will therefore not be commented further.

*DumbBot 2* (by Norman) works by first calculating values for all provinces, and then creating orders based on those values [3]. When evaluating provinces, it takes into account supply centers, owner size and proximity, as well as the attack strength it has on the province. Then it tries to move units to the highest ranked province, with random chances at moving towards lower ranked provinces with the chance declining proportionally to the values computed. If the unit is already at the best place it can reach, it holds, and if another unit is already occupying the province or is moving there, it either supports the unit (if it is not already



HaAI consists of a MAS and a communications interface through which the system communicates with the DAIDE platform. The MAS hosts one unit agent for each unit the bot controls in the game. The MAS also has a common world model based on the jDip adjudicator and the jDip world model, an open source implementation of Diplomacy [9]. This model gets updated through the communications interface and that is available to the agents of the system, see Figure 3. The units of the opponents are not represented by any *agents* in the system; their current positions however, are of course shown in the world model.

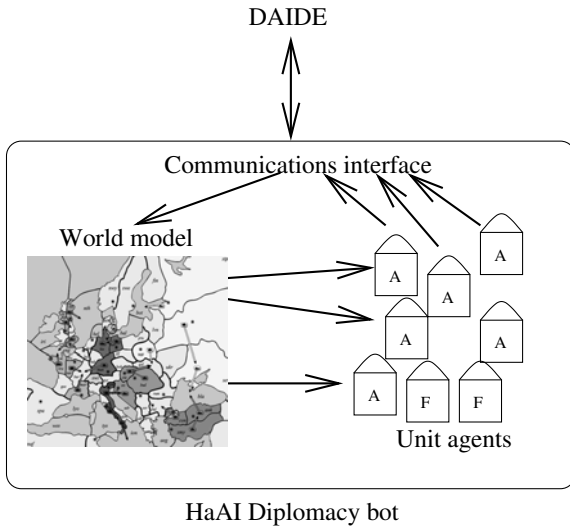


Figure 3: An overview of the architecture of the HaAI bot.

### 3.1 The Unit Agents

The unit agents represent the units of the player. At the start of the game, one unit agent is created for every starting unit by the bot engine. Further unit agents are created when the bot is allowed to build additional units, by letting a unit agent create itself at the best available location and the best available type for a new unit. The unit agents have the ability to evaluate their surroundings and create goals of movements using those evaluations. They also have the ability to support goals created by other unit agents.

#### 3.1.1 The Goals of the Unit Agents

The MAS maintains a set of (one step movement) goals proposed by each agent.

Each goal in the set contains information about:

- the expected value of a successful move,
- the threats possibly preventing the success of the goal
- the amount of support that is needed to guarantee an achievement of the goal
- the amount of support that may be enough to try to achieve the goal

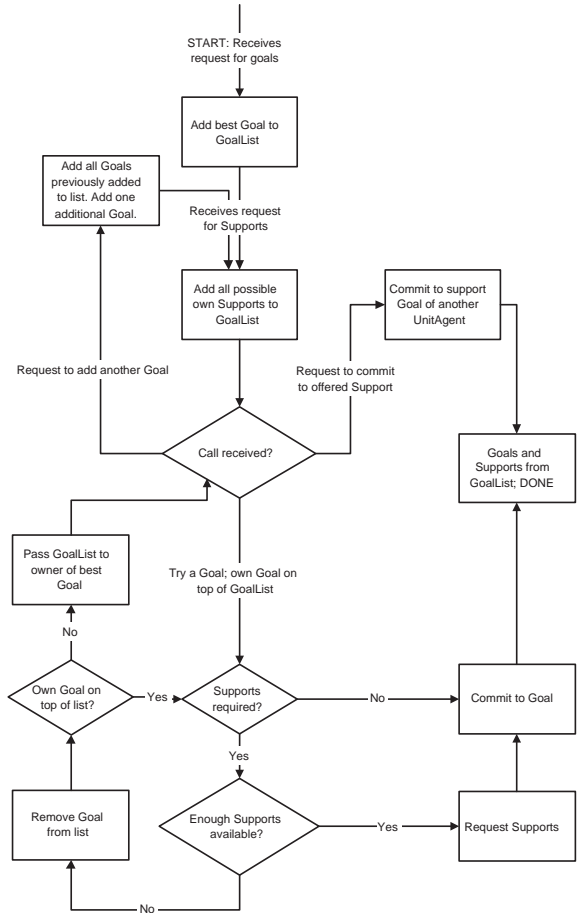


Figure 4: Flowchart of the information passed between the agents of the system.

- the agent who proposed the goal, and
- a sorted list of the unit agents that are able to offer support. The list is sorted in increasing order based on the values of alternative goals of these agents.

#### 3.1.2 Unit Agent Movements Coordination

The coordination model is loosely based on the *Contract Net* [20], with modifications to suit our domain. In the terminology of Smith, the *task announcements* used are the goals created by the unit agents, and the *bids* are the support offers that are attached to those goals. In this way, every agent will act as both a *Contract Net manager* and *contractor*.

When the movements coordination is initiated, the following steps are taken (see Figure 4):

1. All unit agents submit their highest valued individual goals to the common set of goals
2. The unit agents are then requested to add all support (bids) they can give to the goals in the list. For each goal these supports are sorted in order of their cost

Parameter	Explanation	Vanilla	Berserk
EVAL_SCOPE	How far the agent should look in its evaluation.	8	12
RANGE_DECLINE	How many % of the value of a province that should transfer to neighboring provinces.	12	10
E_C_FALL	The value of an enemy center in the fall.	1000	1200
E_C_SPRING	The value of an enemy center in the spring.	800	1000
O_C_THREAT_FALL	The value of an owned, threatened center in the fall.	1000	900
O_C_THREAT_SPRING	The value of an owned, threatened center in the spring.	800	800
O_H_CENTER	Extra value given to own threatened home centers.	1000	1200
E_H_CENTER	Extra value given to an enemy home center.	500	1000
RETRY_FACTOR	Retry tweak factor.	1.05	0.90
O_OCC_UNTHREAT	Chain avoidance factor. (Unthreatened units)	0.60	0.30
O_OCC_THREAT	Chain avoidance factor. (Threatened units)	0.70	0.50

**Table 2: HaAI parameters and values for Vanilla HaAI and Berserk HaAI.**

(i.e. the value of an alternate goal for the offering unit agent).

- While the list of goals is not empty, the unit agent with the highest valued goal examines if there are enough support offered.
  - If the needed support is not available, the goal will be removed from the set of goals.
  - If there is enough support, the supporters that have offered the cheapest support will be notified to commit to this goal, and all of them as well as the attacking unit agent will purge the rest of their goals and all of their support bids from the set of goals (cf. the *award message* in the Contract Net - however, there are no *tasks completed*, or *report* messages sent.).
- All unit agents not committed to a goal are then requested to (if possible) add a new goal. In addition, any previously submitted goal(s) will get a random chance of lowering their required support, based on the difference between the value of the considered goal and the value of the last added goal of that same unit agent.

### 3.1.3 Building New Unit Agents

The creation of a new unit agent is initiated by the system, but the rest is handled by the unit agent to be built. First, it will decide whether an army or a fleet is most needed; it will always try to make the army/fleet ratio match the ratio of non-home centers reachable by a unit. Then, it will build at the best available home center for that kind of unit. If it attempts to build a fleet, and no fleet builds are possible, it will (if possible) build an army instead.

### 3.1.4 Disbands and Retreats

When disbanding, the unit at the position with the lowest value will be disbanded. Retreats are decided by letting the retreating units pick their best retreat; conflicts are handled by comparing the second best retreats available, and so on.

### 3.1.5 Evaluation

When evaluating a province, only those provinces with a supply center have their own value. That value is based

on whether the bot is owner of the center or not, as well as what season it is. No value is given to own centers that are unthreatened. After computing this *base value*,  $v$ , the values of provinces are also increased with a fraction ( $v \cdot (\text{RANGE\_DECLINE}/100)^{\text{distance}}$ ) of the base value of all provinces within EVAL\_SCOPE moves. In addition to this, in order to avoid *chains* of several moves that depend on each other (which generally is a bad thing in Diplomacy) the value for a province occupied by a friendly unit is decreased (using O\_OCC\_UNTHREAT or O\_OCC\_THREAT).

## 3.2 Weights & Variants

HaAI was created in two different variants, with slightly different characteristics, listed in Table 2.

*The HaAI Vanilla* has basic weighting, this bot made little difference between offense and defense.

*The HaAI Berserk* prioritizes centers owned by enemies, and especially their home centers. It is more likely to try a move with low chances of success than HaAI vanilla.

## 4. EXPERIMENT SETUP

The experiments were conducted as a series of games where all seven participating bots were assigned their starting player uniformly distributed at random. Each game was played until there were:

- a solo victory,
- no change in supply center ownership had taken place for five consecutive game years, or
- three minutes had passed.

592 games were played on a single AMD 1700xp 768 DDR running Windows XPpro. The scores awarded were  $(7/(\text{number of survivors}))$  points for a draw, and 7 points for a solo victory. Less than 2% of the games were aborted because of the time limit.

### 4.1 Participants

Both of the two described variants of HaAI participated in the tournament, as well as Man'Chi AttackBot 7, Man'Chi DefenceBot 7, RandBot, DiploBot v1.2 and DumbBot v2.

## 5. EXPERIMENT RESULTS

We will now present the results of the tournament both regarding the scores of the bots, their ability to survive an elimination and their speed.

### 5.1 Scores

There are two ways of scoring in Diplomacy, as discussed in Section 4: to win games (i.e. to *solo*), or to still be in the game when it ends prematurely (i.e. be part of a *draw*).

In Tables 3–4 we see the individual sums of scores of the bots of the 592 matches that were run (of which 487 were solos, the rest of them ended in draws). The total score is shown in Table 5.

Clearly, HaAI Berserk performs well, winning 18.4 percents of the games, although DiploBot, DumbBot, HaAI Vanilla and Man’Chi AttackBot all are between 12.7 and 16 percent (see Table 3). Although Man’Chi DefenceBot was only able to win 40 matches, it was best on the draws, closely followed by HaAI Vanilla. DumbBot was by far the bot that was least able to survive to a draw, reaching barely over the level of RandBot as shown in Table 4.

In total, HaAI Berserk performed best with DiploBot and HaAI Vanilla on second and third place. As expected, random movements (represented by RandBot in our tournament) is not a successful strategy in this game (either).

### 5.2 Elimination

Another aspect of Diplomacy is the ability to survive. Even though a solo implies that it is just a matter of time before the soloing player will be able to conquer the rest of the provinces, it may still be of interest to see what bots are able to survive an elimination. In Table 6 we see that the very same bots that are among the best in reaching draws (the Man’Chis and the HaAIs), also survive elimination to a higher degree than e.g. DumbBot.

### 5.3 Performance

Each bot was set to play against itself for ten minutes of effective game time, and the total number of orders submitted per second for the seven participating bots was recorded. In Table 7 we can see that the DumbBot was by far the fastest bot, leaving HaAI and Man’Chi far behind. Slowest of them all was DiploBot, only managing 3.7 orders per second, about 15 times as slow as DumbBot.

## 6. DISCUSSION

Bot	Matches won	Percent	Solo score
HaAI 0.63 Berserk	109	18.4	763
DiploBot 1.2	95	16.0	665
DumbBot 2	87	14.7	609
HaAI 0.63 Vanilla	81	13.7	567
Man’Chi AttackBot 7	75	12.7	525
Man’Chi DefenceBot 7	40	6.8	280
RandBot	0	0	0

Table 3: Solo score totals.

Bot name	Draw Score
Man’Chi DefenceBot 7	164.27
HaAI 0.63 Vanilla	157.64
Man’Chi AttackBot 7	133.32
HaAI 0.63 Berserk	103.05
DiploBot 1.2	98.10
DumbBot 2	42.34
RandBot	36.29

Table 4: Draw score totals

Bot name	Total Score	Score per match
HaAI 0.63 Berserk	866.05	1.46
DiploBot 1.2	763.10	1.29
HaAI 0.63 Vanilla	724.64	1.22
Man’Chi AttackBot 7	658.32	1.11
DumbBot 2	651.34	1.10
Man’Chi DefenceBot 7	444.27	0.75
RandBot	36.29	0.06

Table 5: Average score totals

Bot	Matches eliminated	Percent eliminated
HaAI 0.63 Vanilla	59	10.0
Man’Chi DefenceBot 7	76	12.8
HaAI 0.63 Berserk	87	14.7
Man’Chi AttackBot 7	115	19.4
DiploBot 1.2	116	19.6
DumbBot 2	348	58.8
RandBot 1	530	89.5

Table 6: Elimination records

The focus of our discussion will be on the validity of the results, the reasons for them, and their relevance, especially to the area of MAS.

### 6.1 Validity

We invited the no-press Diplomacy community to submit bots to an open tournament in May. Since only two submissions were acceptable (DiploBot and DumbBot; Man’Chi initially crashed too often), we extended the deadline and ran the tournament in September including the Man’Chi bots. There were at least three more bots that were not able to participate due to compatibility problems: the Israeli, the Bordeaux, and the LA Diplomats [12, 14, 17]. We have no reason to believe that our HaAI solution would not be competitive against them (in no-press Diplomacy) as well as their later competitors, DiploBot, DumbBot and Man’Chi. It would however require too much effort to recreate these early bots just based on their descriptions in the available sources. In all, we think our bots performed well in the (up to this date) best possible available bot resistance, although preliminary results show that they still are beaten by good human players.

Another aspect is the consideration of processing time. The DAIDE server does not support neither limitations in

Bot type	Orders/s	Implementation Language
DumbBot 2	55.9	C++
HaAI 0.63	19.6	Java
Man'Chi 7	12.7	Java
DiploBot 1.2	3.7	Java

**Table 7: Bot performance (orders/s)**

bot processing time, nor the ability to force a draw after a certain number of years.

## 6.2 Qualitative Analysis

We believe that there are several factors contributing to the good performance of HaAI:

1. The short look-ahead allows HaAI to spend more time on negotiating and reaching a good solution one step ahead, than e.g. DiploBot which struggles with the inherent complexity of *planning* in this rather unpredictable environment.
2. The handling of uncertainties is essential. If HaAI was to restrict itself to choose between the moves that were guaranteed to succeed, it would not survive very long (unless it ran an extremely defensive strategy, but then it would never solo). HaAI, especially the Berserk version, seems to do this better than its opponents.
3. The clustering of goals implements an iterated *best move first* strategy, whereas, e.g., DumbBot uses a roulette wheel based selection of moves, thus it has a higher probability of choosing a worse move compared to HaAI in the same situation.

Even though the iterated best move first strategy is a heuristic that may lead to non-optimal allocation of available resources, the problem of finding the optimal solution is a bin packing problem, and thus NP-complete.

## 6.3 Relevance

So, how relevant is this demonstration of multi-agent solutions? It can be argued that the chosen domain is far from reality. On the other hand, no-press Diplomacy holds a number of properties that it shares with domains that are considered to be hard (such as robot football) e.g.:

- The number of possible future states (even one step ahead) is too large to facilitate exhaustive search.
- The environment is hostile, in the sense that several actors try to gain from your loss.
- Coordination of actions is needed in order to succeed in performing certain tasks.

We argue that the domain has passed the level of toy problems (even though there are of course more complex environments available) and that our solution is novel in the chosen domain. There have been other attempts to make agent based solutions to other games such as Chess [1, 6, 7] and Risk [10]. In the case of Chess, Drogoul [6] and Fransson [7] have tried to create MASS playing Chess based on various

kinds of negotiations between the Chess pieces, rather than traditional minimax search algorithms, but the performance of these systems is not overwhelming. AntChess takes an artificial ant approach to the problem [1]. Keppler and Choi used only one agent that they trained using neural nets to learn good strategies for playing Risk [10].

Even though Diplomacy share several properties such as being competitive, demanding supportive actions at piece level, etc. with Chess and Risk, there are great differences too, as (partly) shown in Table 8.

Property	Diplomacy	Risk	Chess
Moves	parallel one step	sequential multi-step	sequential one-step
Outcome	depends on opponents	dice-based	deterministic
Pieces	two types	single type	six types
Board	map	map	symmetric
Pieces/prov.	0-1	1-∞	0-1

**Table 8: Differences between the games Diplomacy, Risk and Chess**

These differences force us to approach the problems from different directions when designing agent systems able to play the games. For instance, Chess allow (and force) us to plan the support several steps ahead, something that is very hard to do in both Risk and Diplomacy. Risk is a game involving rolling dice. When calculating the possibility of reaching a far goal within the turn of the bot, it has to keep multiple models representing the possible outcomes of conquering intermediate provinces on its way to the final goal in order to replan if the losses on the way are too high. Still there are parts of the systems that are common:

- A MAS is easily modularized in pieces or provinces.
- The agents can, by negotiating with other agents within reach of it, make common efforts to achieve (possibly partly) common goals.
- The agent metaphor provides a good way of modelling utilities at the individual agent level.

Is then MAS techniques the only way of implementing HaAI? Although the functionality of HaAI could be implemented by means of traditional centralistic approaches, we have found that agents are well suited to model the units of Diplomacy.

## 7. CONCLUSIONS

We have presented a novel distributed solution for playing no-press Diplomacy that is based on a multi-agent system. The agents of the system represent the units currently under control of the player. By using a contract net based protocol for negotiating about how to coordinate the joint actions, our solution showed good performance reached at an acceptable amount of time in competition with available state-of-the-art solutions.

## 8. FUTURE WORK

Future work will include:

- Calibrating the parameters of the model in order to find a near optimal setting for the Diplomacy domain.
- Try our approach in domains, such as Risk, Go, etc.
- Adding a module of strategic analysis. At the moment, HaAI performs well without paying any attention to strategic matters. While DumbBot for instance prioritize to attack the strongest opponent, AttackBot and DefenceBot selects a target player to attack, and DiploBot works with dynamically prioritized areas, HaAI only looks at the tactical information. We believe that some strategic consideration would improve HaAI even further.
- Improve the method of analysis of the strategies, e.g. by using replicator dynamics [21].

## Acknowledgments

The authors would like to thank the members of the jDip group and the DIPAI Yahoo group, especially Zach DelProposto, Henrik Bylund, David Norman, François McNeil, Andrew Rose and Brian Roberts for providing bots, platforms and libraries. We would also like to thank Blekinge Institute of Technology and the Swedish Knowledge foundation for sponsoring parts of this work.

## 9. REFERENCES

- [1] The antchess home page. <http://www.antchess.hu/>.
- [2] A. B. Calhamar. *The rules of Diplomacy*. The Avalon Hill Game Company, 4th edition, 2000.
- [3] DAIDE web page, <http://www.starblood.org/daide/>, December 2004.
- [4] The Diplomacy archive, <http://www.diplomacy-archive.com>, December 2004.
- [5] The Diplomatic pouch, <http://www.diplom.org>, December 2004.
- [6] A. Drogoul. When ants play chess (or can strategies emerge from tactical behaviours?). In C. Castelfranchi and J.-P. Müller, editors, *From Reaction to Cognition — Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW-93 (LNAI Volume 957)*, pages 13–27. Springer-Verlag: Heidelberg, Germany, 1995.
- [7] H. Fransson. Agentchess — an agent chess approach — can agents play chess? Master’s thesis, Blekinge Institute of Technology, 2003.
- [8] T. Hart and D. Edwards. The Tree Prune (TP) algorithm. Technical report, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1961. Artificial Intelligence Project Memo 30.
- [9] jDip web page, <http://jdip.sourceforge.net>, December 2004.
- [10] D. Keppler and E. Choi. An intelligent agent for risk. Technical Report CS 473, Computer Science Department, Cornell University, 2000.
- [11] D. Knuth and R. Moore. An analysis of alpha-beta-pruning. *Artificial Intelligence*, 6(4):293–326, 1975.
- [12] S. Kraus and D. Lehmann. Designing and building a negotiating automated agent. *Computational Intelligence*, 11(1):132–171, 1995.
- [13] D. E. Loeb. Challenges in multi-player gaming by computers. *The Diplomatic Pouch Zine*, S1995M, 1995.
- [14] D. E. Loeb. Stable winning coalitions. In R. J. Nowakowski, editor, *Games of No Chance*, Mathematical Sciences Research Institute (MSRI) Publications, pages 451–471. Cambridge University Press, 1996.
- [15] The Man’Chi home page, December 2004. <http://ca.geocities.com/bmr335/manchi.html>.
- [16] F. McNeil. Personal email communication with the authors. 2004.
- [17] A. Shapiro, G. Fuchs, and R. Levinson. Learning a game strategy using pattern-weights and self-play. In *Proceedings of Third International Conference on Computers and Games*, July 2002. Edmonton, Canada.
- [18] R. Sharp. *The Game of Diplomacy*. 1978.
- [19] D. Slate and L. Atkin. Chess 4.5 — the northwestern university Chess program. In P. Frey, editor, *Chess skill in Man and Machine*, pages 82–118. Springer Verlag, Berlin, 1977.
- [20] R. G. Smith. The contract net: A formalism for the control of distributed problem solving. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI-77)*, 1977.
- [21] W. Walsh, R. Das, G. Tesauro, and J. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *Proceedings of the AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents*, pages 109–118. 2002.