J. Expt. Theor. Artif. Intell. 1(1991)255–281.

Research Paper

Negotiation in a non-cooperative environment

SARIT KRAUS,[1] EITHAN EPHRATI[2] and DANIEL LEHMANN[2]

[1]*Department of Mathematics and Computer Science, Bar Ilan University, Ramat Gan 52900, Israel and* [2]*Department of Computer Science, The Hebrew University, Jerusalem 91904, Israel,*
e-mail: {sarit, tantush, lehmann}@cs.huji.ac.il (972-2)-585258

Abstract. The area of automated negotiation has been of particular interest in artificial intelligence due to the important role negotiation plays in facilitating understanding and achieving co-operation among entities with differing interests. These entities may be individuals, organizations, governments, or automated agents. This paper presents methods for solving different aspects of automated negotiation: with whom to negotiate, evaluation of suggestions and the way to offer suggestions. These methods were successfully used to develop the system *Diplomat*, that may be one of the players in a board game, Diplomacy. This game is characterized by intense negotiation, a very large set of possible strategies and the absence of a trusted intermediary. Although Diplomacy players may break their promises, close co-operation is needed for a success.

Received 1 September 1990; revised 25 June 1990

1. Introduction

The process of negotiation is one in which two or more parties in a dispute attempt to reach a settlement acceptable to all sides. The area of automated negotiations has been of particular interest in artificial intelligence due to the important role negotiations play in facilitating understanding and the achievement of co-operation among entities with differing interests. These entities may be individuals, organizations, governments, or automated agents. Many of the processes involved in negotiation are the same regardless of the subject of the negotiation, and hence the development of models of the negotiation process can be useful in analysing many important forms of human interaction.

The need for negotiation arises in a multi-agent environment if co-operation between the agents may be beneficial; it becomes even more crucial in the presence of conflict among the agents. The agent will derive even more benefits from negotiations if some or all of the following conditions exist:

1. The co-operation requires precise division of tasks between the participants.
2. Information is incomplete, and the negotiations enable the agents to gain more information.
3. There are difficult problems to analyse and a single agent is unable to solve its problems alone. The analytic power may be increased through co-operation and exchange of ideas.
4. A multi-agent environment where the other human (or automated) agents

have the capability to negotiate and an agent that does not have this capability is at a disadvantage.

In this paper we describe some aspects of a project for developing the capabilities that are needed for building an automated negotiator. In the course of our research some general principles and algorithms for solving different aspects of automated negotiation were developed. In order to check the validity of these principles and algorithms an automated negotiator was implemented. As a testbed, a specific domain was chosen, rich enough to include most aspects of negotiation. Given a (restricted version of) natural language which covers this domain, our agent was confronted with human agents and even showed an advantage over its human negotiation partners.

We will begin with an examination of previous works. Our objective is to distinguish our work from previous works. This will be followed by a description of the game of Diplomacy, which is the environment for which the automated negotiator, *Diplomat*, was developed. Then we will describe the general structure of *Diplomat*. Subsequent sections address three aspects of negotiation: with whom to negotiate, evaluation of suggestions and ways to offer suggestions. At the beginning of each of those sections we will describe our general ideas concerning the relevant problem. Later we will describe the algorithms *Diplomat* uses while dealing with these three aspects and demonstrate them with a game in which *Diplomat* participated. Finally, we will describe the experiments we conducted with *Diplomat*.

2. Previous work

Some previous work in distributed artificial intelligence (DAI) has dealt with negotiation strategies. Davis and Smith's work on the contract net (Davis and Smith 1983) introduced a form of simple negotiation among co-operative agents, with one agent announcing the availability of tasks and awarding them to other bidding agents. Malone refined this technique considerably by overlaying it with a more sophisticated economic model (Malone *et al.* 1988), proving optimality under certain conditions. While Davis and Smith's original work assumed some autonomy among agents, these agents willingly bid for tasks without explicit motivation. Malone's work introduced a motivational framework in the language of economic theory, and at the same time provided a more theoretical language in which to discuss the task-sharing algorithm.

Davis and Smith and other researchers in DAI investigated negotiations in the case of co-operative systems. These systems were designed to achieve a common general task, or dealt with agents that belonged to the same organization or unit (see for example Sathi *et al.* 1986 which deals with project management, Durfee 1988 which deals with the vehicle monitoring domain, and Sathi and Fox 1989 which deals with resource reallocations). Conflicts among the agents in such environments may arise while each tries to achieve its own sub-tasks (for example, they may need to share the same resources), but their overall task is the same.

In the present case, we deal with a non-cooperative environment where each agent has its own task, and this task usually contradicts in part or in full the task(s) of the other party(ies). Nevertheless, they need to achieve some co-operation in order to fulfill their own tasks.

Sycara (1987) presented a model of negotiation that combines case-based reasoning and optimization of a multi-attribute utilities of the agents. She implemented her ideas in a computer program called the PERSUADER which resolved adversarial conflicts in the domain of labour relations, and tested her system using simulations of such domains. While she concentrated on the perspective of the mediator, we analyse such situations from the point of view of the participants.

Matwin *et al.* (1989) developed an expert system shell called *Negoplan* to support single party participants in a negotiation situation. *Negoplan* simulates the changes in the positions of the parties during the negotiation, based on their anticipated behaviour. Basic negotiation issues are represented as Prolog-like facts. Their method does not simulate the entire process of negotiation since they give one party a competitive advantage. They mainly support the negotiator while dealing with multiple issues in the negotiation. We simulate the overall negotiation process, and our purpose is not to support a negotiator, but mainly to let him negotiate with an automated negotiator in the environment in which he participates.

Rosenchein and Genesereth (1985) used certain game-theoretic techniques to model communication and promises in multi-agent interaction. There, the process of negotiation was severely restricted (the agents could only make single, simultaneous offers), and it assumed that each agent knew the complete payoff matrix associated with the interaction. Also, for large games involving many agents and outcomes, the kind of environments in which we are interested, the size of a payoff matrix may quickly become intractable. This work was extended by Zlotkin and Rosenschein (1989, 1990), and in this last work there is no need for the players to know the full payoff matrices, but the players still need to explore all possible strategies. Kraus and Wilkenfeld used different game theory techniques to model negotiation of automated agents under time constraints (Kraus and Wilkenfeld 1990, 1991a,b). They also assume that the set of possible agreements is limited and that there is full information, but made fewer restrictions on the negotiation procedure.

We examine a complex environment, where the set of possible strategies is very large, a mediator is not available, the agents may break their promises, close co-operation between different agents is needed and possible coalitions between other agents must be taken into account. For that purpose we developed a system that actually negotiates with human partners.

3. Diplomacy

The environment we chose to deal with is a board game called *Diplomacy*, marketed by Avalon Hill Company. Diplomacy is an environment of intense negotiation. We implemented and tested our ideas by building an automated Diplomacy player called *Diplomat* (see also Kraus and Lehmann 1988b, 1989, Kraus 1988 and Kraus *et al.* 1989).

Diplomacy is a board game played on a map of Europe during the years just before World War I (Figure 1). Each player represents one of seven European powers: England, Germany, Russia, Turkey, Austria-Hungary, Italy and France. The countries are depicted on the board by heavy, solid black lines (Figure 1). The Great Powers are also subdivided into provinces. The seas are divided into 'bodies of water'. Certain provinces on the board are designated 'supply centres'.

Figure 1. Diplomacy's map.

The supply centres are important places in Europe and are marked by black dots on the map (Figure 1; e.g. London and Berlin are supply centres). Each of the supply centres produces supplies needed to maintain an army or fleet. A player may have only as many armies and fleets on the board as the number of supply centres he controls.

The object of the game is for one power to gain control over the majority of the board, i.e. to control 18 supply centres. Since gaining control of the board can take a long time, the game can also be played when the exact number of moves to be made is predetermined. The winner in such an abbreviated game is the player with the most units on the board. The stages of the game are divided into two seasons each year: a Spring season and a Fall season, beginning in the year 1901.

Coalitions and agreements among the players significantly affect the course of the game. Agreements include mutual promises for co-operation; an agreement may be detailed as to include detailed military co-operation. The coalitions and agreements are determined during the diplomacy periods which take place before each move. During these diplomacy periods a player may say anything he wishes.[1] The communications usually consist of bargaining or joint military planning, but they may also include exchanges of information, denunciations, threats, rumours, etc. The communications between players are usually secret to the other players. The rules of the game do not determine in advance which powers need to be allies or enemies; co-operation is reached only through negotiations. The rules of the game do not bind a player to anything he says. Deciding whom to trust as situations arise is part of the game.

At the end of the negotiation period, all players secretly write the orders for all of their units simultaneously. A unit may be ordered to do only one thing in each season: to hold, move, or provide support. A fleet may also be ordered to convoy an army from one coast to another. Only one unit may be in any space at one time. Therefore, an army (resp. a fleet) may move to any adjacent province only if this move does not cause it to conflict with another unit.

An army or a fleet may support the move of another army or fleet of that country or any other country in making a move. Support can also be given on a defensive basis to protect a paying piece from being dislodged from the space it occupies.

If one piece has the support of another in making the move and the opposing unit does not, then it moves into the empty space—or it may force an unsupported unit to retreat out of a space. To 'support' a move, the supporting unit must itself be able to move into the space under attack. Opposing units with equal support do not move. An advantage in force of only one is sufficient to win.

The number of units that a player has on the board is determined by the number of 'supply centres' which he controls. Adjustments in strength are made after the Fall moves have been completed. Each player writes his desires. If he increased the number of supply centres under his control, he writes whether he wants an army or a fleet and, if so, where. The unit must be placed in a supply centre in his home country. If he has lost strength, he decides which army or fleet is to be removed. The following example demonstrates some of the rules of Diplomacy. Further details of the rules of the game can be found in Avalon Hill Co. 1976.

Example 3.1: It was Fall 1902. The state of the game appears in Table 1. We will examine part of the orders that were given:

[1]In the games we conducted, we restricted the negotiations to a special formal negotiation language that we have developed; the negotiations were conducted through electronic mail.

Table 1. A situation in Fall 1902: *Diplomat* played Russia and England (independently). Supply centres are places that produce supplies sufficient to maintain an army or a fleet. Units are armies (A) or fleets (F)

Power	Supply centres	Units
Russia *Diplomat*	Warsaw, St Petersburg, Sweden, Rumania, Moscow, Sevastopol	A Bulgaria, F St. Petersburg (NC), F Black Sea, F Sweden, A Rumania, A Silesia
Austria	Venice, Vienna, Serbia, Budapest	A Serbia, A Tyrolia, A Budapest, F Trieste
England *Diplomat*	London, Edinburgh, Liverpool, Norway	A Edinburgh, F English Channel, F North Sea, F Norway
France	Portugal, Brest, Marseilles, Paris, Spain	F Mid-Atlantic Ocean, A Burgundy, A Picardy, A Gascony, A Marseilles
Germany	Kiel, Berlin, Munich, Belgium, Holland	A Kiel, A Ruhr, F Berlin, A Munich, F Belgium
Italy	Rome, Naples, Greece, Trieste	A Albania, F Tyrrhenian Sea, F Apulia, A Greece
Turkey	Ankara, Bulgaria, Constantinople, Smyrna	A Constantinople, A Smyrna, F Ankara

England:
1. F North Sea Convoys A Edinburgh to Denmark
2. A Edinburgh Moves to Denmark

Russia:
3. F Sweden Supports A Edinburgh to Denmark
4. F Black Sea Supports A Bulgaria to Constantinople
5. A Bulgaria Moves to Constantinople

Germany:
6. A Kiel Moves to Denmark

Turkey:
7. A Smyrna Moves to Constantinople
8. F Ankara Supports Smyrna's Move to Constantinople

Italy:
9. F Tyrrhenian Sea Moves to Tunis

Results:
England's army at Edinburgh entered Denmark, (1,2) despite Germany's attempt to enter Denmark too from Kiel (5). The success was due to support from Russia's fleet in Sweden (3). Russia's attempt to enter Constantinople from Bulgaria (4,5) failed because of Turkey's attempt to enter there from Smyrna (7,8) with equal forces. Italy entered Tunis (9).

Although each player in Diplomacy tries to win the game, players can benefit from co-operation; there are co-operative strategies that improve both players' situation, compared with being enemies. For example, if there is a strong player, who each player by himself can not defeat, they can work together against him, gain control over his supply centres and share them amongst themselves.

The need for negotiation increases when certain moves require close co-operation between different allied powers. The units of one power may help the moves of another power only if they are explicitly ordered to do so. For example, if Russia wants to help England to enter Denmark (as in Example 3.1), it must specify the unit that will perform the incursion.

The game is quite complex; there are 75 spaces on the board that can be occupied by 34 units. At the end of each season each unit is ordered simultaneously to perform one of four possible actions (move, support, convoy/valid hold). All of the actions, beside 'hold', are related to the spaces adjacent to the space that the unit occupies (usually around five spaces). We conclude that there are approximately 34^{16} possible moves in each step of the game.

Since the game is so complex, it is difficut to evaluate the preferences of the other players given different possible situations of the board. It is even harder to estimate the preferences of the players given different possible strategies. Therefore, a Diplomacy player negotiates in order to obtain information about the goals of the other players. He also tries to receive information from one player on his evaluation of the goals and strategies of a third party.

The need for negotiation increases in Diplomacy since other players negotiate. The negotiations between other players may lead to coalitions against a player and if he will not negotiate, he can find himself without any help against those players. Actual experience shows that a power cannot last long without taking part in extensive negotiations.

In Diplomacy each player represents a different country, and therefore the initial situation and the prospects and problems of each player are different. Nevertheless, all players have almost equal strategic possibilities to win the game. To be a successful Diplomacy player one needs some technical skills in moving military units on the board according to the reasonable but complex rules of the game, but above all, one needs the ability to communicate and negotiate with the other players, to make agreements with the others and, possibly, to decide to ignore prior agreements.

There are several common attributes that are important for a negotiator regardless of the environment he negotiates in (see Karrass 1970). We assume that methods and theorems that are applicable to negotiations in Diplomacy will be applicable in any similar environments in which negotiation is an important issue.

4. *Diplomat*'s general description

The general structure of *Diplomat* can be described as a government which includes a prime minister and its secretary, the Ministry of Defence, the Foreign Office, the military headquarters and the intelligence (Figure 2).[2]

[2]*Diplomat* is implemented in Ylisp (Levy and Dimitrovski 1984) (a dialect of Franz-Lisp) on a Vax 11/785 running Unix, Berkeley 4.3.

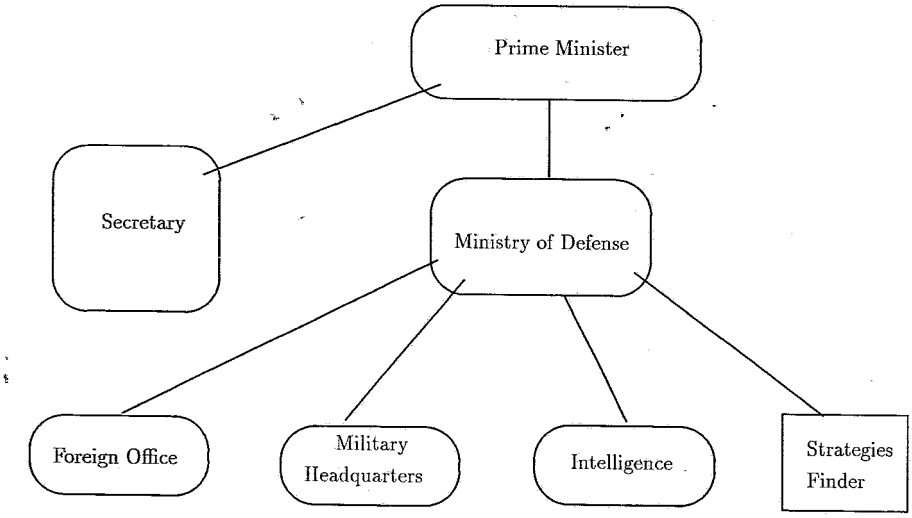

Figure 2. *Diplomat*'s general description.

The personality of the prime minister influences the behaviour of the government's members. The government's secretary maintains the government knowledge base. The Ministry of Defence is responsible for planning and situation analysis and it directs the behaviour of the Foreign Office, the military headquarters and the intelligence. The Foreign Office is responsible for communicating with the other powers. The military headquarters gives *Diplomat*'s orders at the end of each season and the intelligence acquires information about the other players. Another module is the strategies finder (SF) which provides strategies.[3]

One can see that we suggest separate modules for the negotiation and for planning. This makes *Diplomat* more modular. We found out that this modularity helps in the development period, because each module can be developed independently. This separation is also a characteristic of a good negotiator (see Fisher and Ury 1981).

We will describe shortly these modules.

- **The Prime Minister** – directs *Diplomat* activities. It is the only module that has 'personality' traits such as aggressiveness, willingness to take chances and loyalty. The 'personality' traits are given to it at installation or during the negotiation. They allow *Diplomat* to change 'personality' from one game to another, and therefore it is more difficult for other players to figure out its intentions.

[3]We define a strategy, in the Diplomacy game, to include a list of orders and the expected loss or profit from them. Each order includes the active unit, the type of the activity (Move, Support, Convoy, Valid and Hold), and the locations on the board that are related to this activity. The purpose indicates whether the action is part of a plan to attack a specific area or part of a plan to defend a specific area. For example, 'The Rumanian's army in Bulgaria moves to Greece in order to attack Greece'. The strategy also includes the expected average profit from carrying out the strategy for each power that participates in the strategy (i.e. at least one of its units is active in the strategy), and the common expected profit for all of the powers.

The government's secretary, which is directed by the prime minister, keeps and maintains *Diplomat*'s knowledge and belief base according to the information which flows from all the other modules. The knowledge and belief base (KBB) includes the following parts:

—Diplomacy's rules (including general information about the board).
—The situation of the game (see an example in Table 1).
—The messages that were exchanged between *Diplomat* and other players.
—Agreements table (see an example in Table 2).
—Information about the other powers and their relations (see an example in Table 3).

Details about the KBB and the way it is maintained can be found in Kraus and Lehmann (1988b) and Kraus (1988).

• **Ministry of Defence** – This module is responsible for the planning and analysis. It is influenced by the prime minister's personality traits. It directs the behaviour of the Foreign Office, the headquarters and the intelligence and it uses the strategies finder for finding strategies. While analysing a new situation of the game, the Ministry of Defence considers different possible fronts[4] and coalitions and searches for possible strategies.

An important mission of this module is to find partners for negotiation. A 'desk' at the Foreign Office is established for each partner for negotiation,

Table 2. *Diplomat*'s agreements table at the beginning of Fall 1902 when *Diplomat* played Russia. The intentions of *Diplomat* and its allies to keep the agreement (columns 4 and 5) are on a scale between 0 and 10

Agreement	Allies	Enemies	Intention of *Diplomat*	Intention of the ally	Details
1	Russia England		4	4	England won't enter to Barents Sea and St. Pet.
2	Russia Germany		0	4	
3	Russia Austria	Turkey Germany	7	7	Russia will support by A. Bulgaria Austria's move from Serbia to Greece in Fall 1902

[4]A Front includes possible enemies and possible allies. For example: Russia and Turkey against Austria and Germany or *Diplomat* against Italy.

Table 3. *Diplomat*'s beliefs about the relations among the powers in the beginning of Fall of 1902

Power	Friends	Enemies
Russia	Austria, England, Germany	Turkey, Italy
Austria	Russia	Italy
England	Russia, France	Germany
France	England	Germany
Germany	Russia	France, England
Italy		Russia, Turkey, Austria
Turkey		Italy, Russia

which is responsible for the negotiation with this partner. The Ministry of Defence provides the 'desk' with a strategy that can be a basis for negotiation with this partner. We will describe the algorithm for finding possible partners for negotiation in Section 5. The Ministry of Defence also helps the Foreign Office to evaluate different suggestions received from the other powers.

- **Foreign Office** – This module directs the communications of *Diplomat* with the other powers. It includes different departments ('desks') that are responsible for the relations with the different powers.[5] These desks conduct negotiation according to strategies received from the Ministry of Defence. During the negotiation it passes and analyses messages, evaluates suggestions received from the other powers (this algorithm is described in Section 6) and makes suggestions to the other powers (see Section 7). It also decides whether to sign agreements, and if so whether to keep or to break them. After signing an agreement it announces the details to the Prime Minister and its secretary updates the agreements table.

- **Military Headquarters** – This module gives *Diplomat*'s orders at the end of each season. It uses *Diplomat*'s KBB to find out its allies, its enemies (according to the agreements). It also gathers the activities that *Diplomat* promised its allies to perform and decided to keep, and the activities *Diplomat*'s allies promised to carry out and *Diplomat* estimates that they will really keep. Using this information, the Military Headquarters searches for strategies that fit the set of allies and enemies and include the above promised activities. The strategies do not include any other orders to be performed by its allies beside the orders above and they do not include attacks against

[5]The negotiations are conducted using a formal negotiation language. *Diplomat* sends and receives messages in a Lisp-style representation. A special directed editor that we have developed helps the players using the formal negotiation language and transfers the English-style representation into the Lisp-style representation and vice versa.

powers that are friendly or neutral (i.e. does not appear either in the enemies set or the allies set).

- **The Intelligence** – This module tries to estimate the relations between the powers and the other powers' characters. The estimation is mainly done by analysing the powers' orders at the end of every season.
Diplomat may also receive messages concerning other powers' character and behaviour or concerning the relations between other powers. The intelligence decides whether or not to believe the messages that were received. It reports its conclusions to the Prime Minister.

- **Strategies Finder (SF)** – gets a set of allies, a set of enemies and the configuration of the board as an input. Based on this input it finds some plausible strategies and provides different values that serve for their evaluation. The Ministry of Defence, the Military Headquarters and the Intelligence use this module. They can influence the generation of strategies by providing the SF with portions of strategies that were already agreed upon (due to existing agreements), areas of special interest and various degrees of willingness to take chances. The SF was developed, based on observation and interrogation of human Diplomacy experts in the spirit presented by PARADISE (Willkins 1983).

Before going into the details of some of the algorithms that are used by *Diplomat* let us describe a typical behaviour of *Diplomat* during a season in a Diplomacy game.

At the beginning of a season *Diplomat* analyses the situation and looks for possible partners for negotiation (this is the Ministry of Defence's responsibility). Then it negotiates with those partners. The negotiation may be in order to reach a beneficial agreement, to prevent the foundation of an opposing coalition and in order to gather information. More negotiations may be initiated during the season (this is the Foreign Office's responsibility). If an agreement is signed, its details are kept in the agreements table (this is the Prime Minister's responsibility). At the end of the season *Diplomat*'s orders are given taking into account the agreements signed until then (this is the Military Headquarters' responsibility). After the orders of the players are made known, *Diplomat* analyses them, in order to estimate the other players' relations and characters (this is the Intelligence's responsibility).

5. With whom to negotiate

The first step (not only in the context of Diplomacy but in general) is to choose carefully a partner for negotiation. A good partner for negotiation is an agent to whom the negotiator[6] has a beneficial proposal to both partners to offer. The purpose of the negotiation with such a partner is to achieve a beneficial agreement, to get information or to fool the other party about the negotiator's intentions. Even when there is no intention to follow the agreement at all, the proposal must

[6]We use the notion 'negotiator' as a general name for negotiators.

look serious, i.e. profitable for both parties. Other good partners are agents that, of their own initiative, open negotiation with the negotiator.

Experts in Diplomacy claim that it is worthwhile to negotiate intensively in order to win the game (see for example Walker 1984). One may suggest that a player will negotiate in each period with each of the other six powers. This suggestion is difficult to follow since the number of possible coalitions is 3^6 (each player can be considered as a friend, enemy or neutral). In addition, a joint operation plan for co-operation between the negotiators should be found and the desire of a player to keep his credibility to some extent must be taken into consideration (negotiation with all players yields contradictory agreements). Therefore, the player must try to limit the number of players he negotiates with.

The algorithm used by *Diplomat* to find its partners for negotiation is presented below. We note that this algorithm, in addition to finding with whom to negotiate, provides a strategy that can be a basis for negotiation with those partners and a set of possible enemies. This algorithm was designed by trial and error. We presented to *Diplomat* situations that were taken from experts' human games, let it provide the set of powers it negotiates with, compared it with the behaviour of the human Diplomacy experts and updated the algorithm.

The algorithm uses extensively *Diplomat*'s knowledge and belief base (KBB) which was described above and in particular the agreements table. The agreements table includes an entry for any agreement reached by *Diplomat* and another power. It includes the set of allies and the set of enemies of the agreement and the details of the promises for mutual co-operation (if there are such details). It also includes *Diplomat*'s intention to keep the agreement and *Diplomat*'s estimation of the intentions of its allies to keep the agreement. The intentions parameters are on a scale between 0 and 10, where 0 indicates that there is no intention to keep the agreement.

Algorithm 5.1. *Finding possible partners for negotiation at the beginning of a season*

NegoSet = Ø;

1. For every agreement in the agreement table do the following:

   (a) Check whether it is worthwhile to continue the agreement either exactly as it is or with some changes (use (i) to (v) below). If so, add the allies of the agreement to *NegoSet*.
   The algorithm for checking whether it is worthwhile to continue an agreement:

   (i) Apply to the strategies finder (SF) for finding strategies for *Diplomat* and allies against enemies of the agreement and for finding strategies for *Diplomat* and allies against new set of common enemies.[7]

   (ii) If the set of details of the agreement is not empty, compile it into SF format and repeat on (i) by sending to the SF the compiled

[7]Possible common enemies are powers that according to *Diplomat*'s beliefs are enemies of *Diplomat* and are not friends of *Diplomat*'s allies and vice versa. For it to be worthwhile for *Diplomat* and its allies to act against these common enemies, they have to be neighbours of *Diplomat* or its allies.

details together with the appropriate sets of the allies and the enemies.

(iii) Choose the best strategy from all the strategies SF provided in (i) and (ii) and denote it as $strategy_1$.[8]

(iv) Apply to the SF for finding strategies for *Diplomat* and the enemies against a set of new possible enemies. Choose the best strategy from the strategies returned by SF and denote it by $strategy_2$.

(v) If the difference between the expected profits for *Diplomat* from $strategy_2$ and the expected profit for *Diplomat* from $strategy_1$ is greater than *loyalty** 1000 then break the agreement, else continue the agreement.

(b) Otherwise, add the enemies of the agreement to *NegoSet*. Decide to pretend to keep the agreement and add the allies of the agreement to *NegoSet* with probability 1—*loyalty* and decide to tell the other party with probability *loyalty*.

2. For every power in the set of friends of *Diplomat* in its KBB; if the power does not already belong to *NegoSet* do the following:

(a) If there is a possibility for co-operation between *Diplomat* and the power add the power to *NegoSet* (use (i) and (ii) below).

The algorithm for finding whether there is a possibility for co-operation:

(i) Apply to the SF for finding strategies for *Diplomat* and the power against a set of possible new enemies.

(ii) Choose the best strategy from those that were returned by the SF. If *Diplomat* will not lose from keeping the agreement according to this strategy then decide that there is a possibility for co-operation.

(b) Otherwise, decide whether to start negotiation without an intention to keep the agreement with probability 1—*loyalty*. If so, add the power to *NegoSet*.

3. For every power which is *Diplomat*'s neighbour[9] but does not belong to the enemies' set of any agreement in *Diplomat*'s agreements table and is not already in *NegoSet* do the following:

(a) If there is a possibility for co-operation between *Diplomat* and the power add the power to *NegoSet* (use (i) and (ii) of 2 above).

■

[8]The comparison and the choice between strategies are done using a special algorithm (see Kraus 1988). Since a negotiator wants to 'win', one may suspect that the only criterion that will guide him while comparing and choosing between stategies will be his own benefits from the strategies. We found out after observing and interviewing human negotiators that this is not the case. The reason for that phenomenon is that in order for the agreement to last, it should be beneficial to *all* parties involved. Otherwise, a leftbehind partner may be tempted to reach a more appealing agreement, even without informing the negotiator. For that same reason, the other partner should be convinced that the agreement is profitable to the negotiator (see Fisher and Ury 1981) otherwise he will suspect that the negotiator will break the agreement.

[9]Two powers are neighbours if a unit of one of them is adjacent to a unit of the other power or to a supply centre which is controlled by the other power.

In step 1 of algorithm 5.1, *Diplomat* considers the continuation of existin agreements. The examination of possible changes in the agreement (step 1(a) makes *Diplomat*'s behaviour more dynamic.

Diplomat's willingness to break an agreement (step 1(a)(v)) is determined b the difference between the expected profits from breaking the agreement and th expected profits from keeping it. Here we follow experts in negotiation an experts in Diplomacy. For example Walker (1984) says: 'The best rule for an stab is this: will it yield a decisive advantage?. . . Never practice treachery for it own sake, but never refuse it if it's worth the price'.

If *Diplomat* decides not to keep an agreement, it still needs to decide whethe to pretend to keep it or to tell the other party. Its decision is also influenced b its loyalty trait (step 1(b)). *Diplomat* decides with probability 1—*loyalty* to preten to keep the agreement and with probability *loyalty* to tell its partner. Since th decision whether to pretend to keep an agreement is taken semi-randomly, it i more difficult for other players to figure out *Diplomat*'s intentions.

In step 2 *Diplomat* searches among the powers that it believes to be its friends In step 3 it also considers its neighbours that it believes not to be its enemies *Diplomat* decides not to open negotiations with these powers, only if it may los from any agreement with them. Since it is an early stage in the negotiation, it i not yet necessary to profit from reaching an agreement. It may turn out durin the negotiation that *Diplomat* can benefit, and it does not want to lose such a opportunity.

Sometimes, *Diplomat* starts negotiation with its friends even if it does no intend to keep the agreement at all (step 2(b)). The purpose of this negotiatio is to try to prevent the creation of an opposing coalition. We will now give a example which demonstrates Algorithm 5.1.

Example 5.1: *Choosing parties for negotiation*

It was Fall 1902. The situation's details appear in Tables 1, 2 and 3.

Step 1:

Diplomat checked whether to uphold the agreements with Austria, German and England. We note that actually the non-aggression agreement witl Germany contradicted the agreement with Austria against Turkey and Germany but *Diplomat*'s intention to keep the agreement was zero, i.e., *Diploma* decided which agreement to follow. Since *Diplomat* did not make any hostil movement against Germany in the previous season, it assumed that German did not realize its intentions. *Diplomat* estimated that Germany's intentions t keep the agreement was greater than zero and left it in the agreements table

Austria: The agreement with Austria was against Turkey and Germany an stated that Russia would support Austria's attempt to enter Greece. *Diploma* considered the following possibilities (in 1(a)):

1. Continuing the agreement exactly as it is.
2. Continuing the agreement against Turkey and Germany but not supportin Austria's attempt to enter Greece.
3. Adding Italy to the enemies set and supporting Austria's attempt to ente Greece.
4. Adding Italy to the enemies set but without supporting Austria's attemp to enter Greece.

After examining the possible strategies that could be the basis for the above agreements, *Diplomat* decided that the fourth possibility is the best one.
Diplomat also considered betraying Austria (also in 1(a)) by reaching an agreement with Turkey against Austria and Italy or with Germany. The possible agreement with Austria was more profitable to *Diplomat* than any of the other agreements and therefore, *Diplomat* decided to negotiate with Austria in order to reach an agreement against Turkey, Italy and Germany.

England: The agreement with England was a non-aggression agreement which stated that England would not enter the Barents Sea and St. Petersburg. *Diplomat* of decided to add Germany to the agreements' enemies set. It also considered the possibility of betraying England by forming a coalition with Germany against England, but decided to continue the agreement with England.

Germany: The agreement with Germany was a non-aggression agreement. *Diplomat* considered the possibility of adding England to the enemies set and compared it with possible agreements with England against Germany, with Austria against Germany and with acting alone against Germany. *Diplomat* decided not to continue any agreement with Germany but to continue the negotiation with Germany and to *pretend* to continue the non-aggression agreement with her.

Step 2:
In this example, *Diplomat*'s friends were Austria, England and Germany (see Table 3). All of them belong to the *NegoSet* at the end of step 1 and therefore step 2 was not applicable.
Step 3:
Italy was a neighbour of *Diplomat* (Italy's army at Greece was near Russia's army at Bulgaria) and did not appear as *Diplomat*'s enemy in any agreement (even though it appeared as *Diplomat*'s enemy in its relations table). *Diplomat* considered starting negotiation with Italy and decided to do so.

Diplomat may start negotiation with other powers in the middle of the diplomacy period after receiving a message about a coalition that is being formed. It also needs to respond when other powers start negotiation. We will describe those cases in algorithm 5.3. We now describe the algorithm for finding whether potential benefits to *Diplomat* exist as a result of forming a coalition with some powers, possibly against some other powers. This algorithm is used in algorithm 5.3 when *Diplomat* decides how to respond when another power starts negotiation with *Diplomat*. A modified version of this algorithm is also used when *Diplomat* receives a message announcing the existence of an agreement among other powers and it tries to estimate its correctness.

Algorithm 5.2: *Deciding whether there exists a potential benefit from a coalition*
Input: A set of allies and a set of possible enemies.

1. Find the internal relations among the allies of the suggested agreement and their relations with the enemies of the agreement (if enemies exist). Three parameters weigh in favour of the agreement:

(a) If the allies of the suggesteed agreement are friends (according *Diplomat*'s KBB).
(b) If the enemies of the agreements are among the enemies *Diplomat*.
(c) If common friends or common enemies exist for the allies of agreement.

The parameters that weigh against the agreement are:

(a) If the allies of the agreement are enemies according to *Diplomat*'s KE
(b) If the enemies of the agreement are friends of *Diplomat*.
(c) If *Diplomat*'s enemies are friends of the other allies of the agreement

2. Make a general evaluation of the expected profit and loss for *Diplo* from the suggested agreement. (See algorithm below). The estimation t the agreement is profitable for the allies is necessary for *Diplomat* to dec whether the agreement is worthwhile.

3. If a strong agreement[10] exists between the enemies of the agreement enemies exist) and *Diplomat*, then *Diplomat* will conclude that the suggest agreement is not worthwhile.

The general evaluation algorithm of the expected profit and loss is as follo

• For each power of the allies of the suggested agreement, compute t difference between the number of units between this power that adjoin t units or the supply centres of *Diplomat* and the number of *Diplomat*'s ur that adjoin the partner. The danger for *Diplomat* from the agreement increa with this number, and the expected profit decreases with it. The reason that it increases the willingness of the allies to attack *Diplomat* and decrea their willingness to help it.

• Find all the powers that have borders both with *Diplomat* and the suggest allies. These powers can be possible common enemies of an agreement. T expected profit for *Diplomat* from an agreement against such a power increas monotonically with the difference between the number of *Diplomat*'s un that adjoin this power and the number of this power's units that adjc *Diplomat*.

There are two main factors that influence the ability to reach an agreeme the relations among the negotiators and possible benefits to both sides (see Fisl and Ury 1981). In step 1 of the algorithm, *Diplomat* checks the possibility

[10]The strength of an agreement depends on the intention of *Diplomat* and its ally to ke the agreement which is on the scale between 0 and 10. If both intentions are higher th 5, the agreement is strong.

reaching an understanding and achieving good relations among the allies. *Diplomat* checks whether the allies are friends according to its KBB since this is an indication if previous friendly relations among the allies existed. The fact that the enemies of the proposed agreement are enemies of *Diplomat* and that the possible allies have common friends may contribute to the development of good relationships. This is because it reduces the number of possible conflicts among *Diplomat* and the allies.

The idea behind the general evaluation of the expected profits or loss from the agreement is the following: if the partner is stronger than *Diplomat*, he will gain more if he tries to betray *Diplomat*. As we mentioned above, the willingness of a power to betray the other party increases with her possible gains from doing so. In the second part of the general evaluation, *Diplomat* checks the relations balance with the enemies. Since in Diplomacy, the number of units participating on a front strongly influences the possible gains for a power in this front, at least in the near future, *Diplomat* compares between the number of its units and uses it as the general estimation.

We will now describe the algorithm *Diplomat* uses while considering whether to add partners for negotiation in the middle of a negotiation period.

Algorithm 5.3: *Adding possible partners for negotiation in the middle of a negotiation period*

1. If *Diplomat* receives a message that declares the existence of an agreement against it and other powers, and if the agreement is beneficial to the allies of the agreement (use an algorithm similar to algorithm 5.2) add the enemies of the agreement to *NegoSet*.
2. If *Diplomat* receives a message announcing the existence of an agreement in which *Diplomat* does not belong to the enemies, then if the agreement is beneficial to the allies of the agreement (use an algorithm similar to algorithm 5.2), broadcast this message among *Diplomat*'s friends, who are threatened by this agreement.
3. If *Diplomat* receives a message from another power offering it a co-operation agreement then

   (a) If the power belongs to *NegoSet* already but *Diplomat* has not started negotiations with it yet, continue the negotiation after considering the details of the suggestions of the other party.
   (b) If the other power does not belong to *NegoSet*, make a general evaluation as to whether it is worthwhile continuing the negotiation with this power, using algorithm 5.2.
   If it is worthwhile to continue the negotiation with this power, add it to *NegoSet*. Otherwise, decide whether to continue the negotiation without an intention to keep it with probability 1—*loyalty*. If so, add the power to *NegoSet*.

In steps 1 and 2, *Diplomat* receives messages concerning the existence of agreements. It believes that those agreements really exist, if they are beneficial to the allies of the agreements.

In step 1, if it believes that the agreement exists it starts to negotiate with the powers that are threatened by the agreement in order to try to form a coalition against the threatening agreement.

In step 2, the motivation behind broadcasting the message in case it believes that the agreement really exists, is to strengthen the friendships with those powers.

In step 3 another power starts negotiation with *Diplomat*. Its message can be either a formal 'co-operation' message or any other message with a suggestion for co-operation.[11] In cases when *Diplomat* needs to decide whether to continue the agreement (in step 3(b)) it does not apply the strategies finder to find strategies (as in algorithm 5.1), because it likes to see what the other party suggests. Actually, if *Diplomat* could have found a good strategy for co-operation, this power would have already been in *NegoSet*. Rather, *Diplomat* tries to find out whether there is any chance that it could benefit from the negotiation (using algorithm 5.2). If so, it will continue the negotiation.

We will demonstrate algorithms 5.2 and 5.3 in the following example.

Example 5.2: In Fall 1902 (the details are presented in Tables 1, 2 and 3), Turkey, which did not belong to *Diplomat's NegoSet* (see example 5.1), sent the following message to *Diplomat*:

MESSAGE 1 FROM Turkey TO Russia:
I would like to suggest to you a Peace agreement between Russia and Turkey now.
END OF MESSAGE.

This message falls under category 3(b) in algorithm 5.3 and therefore *Diplomat* checked whether cooperation with Turkey was beneficial using algorithm 5.2.
Step 1: (in 5.2) *Diplomat* and Turkey appeared as enemies in *Diplomat*'s KBB but Italy was a common enemy, and therefore there was almost equal weight for continuing the negotiations as for not continuing them.
Step 2: There were two units of *Diplomat* (A. Bulgaria and F. Black Sea) that adjoined the units of Turkey, and two units of Turkey (A. Constantinople and F. in Ankara) that adjoined *Diplomat*'s units. There was no power that bordered both *Diplomat* and Turkey. *Diplomat* estimated that an agreement was not profitable.
Step 3: *Diplomat* had a strong agreement with Austria against Turkey.
Using the results of steps 1, 2 and 3 *Diplomat* estimated that forming a coalition with Turkey would not be profitable. It also decided not to stab Turkey and did not continue the negotiation.

6. Evaluation of suggestion

The negotiator may receive detailed proposals from other agents, and he has to decide how to react. Usually, such suggestions are received after the negotiator has decided that he may benefit from the negotiation (or at least pretend to) as described in Section 5. First he has to understand what the other party really means. He has to try to find out whether the message should be considered an

[11]The message could be either of the form 'I would like to suggest to you a co-operation agreement against...' or 'I would like to suggest to you that you will support my action and I will support your action...' and etc.

answer, a suggestion, a declaration or a question. This is not always easy since players may sometimes use, for example, a declarative sentence for a suggestion. When the negotiator receives a message where details are missing, he has to try to fill the gaps, when possible, like a human player. Only when he fails to fill the gaps will he ask for more details. Then the negotiator has to evaluate the proposal and decide whether to accept it or not, by comparing it with other possibilities, taking into account the state of the negotiation.

The negotiator should translate any suggestion into a set of strategies that *fit* the given suggestion, taking into account the current situation and the views of the agent about the other agents and the environment. This translation changes each suggestion received or sent by the negotiator into a unique strategy format, which allows it to be compared with other suggestions or strategies. The translation is done using the strategies finder of the negotiator, and the strategy found is used as a basis for further negotiations when needed. The next step is to examine the expected profit from the fitted strategy and compare it with the expected profits from other possible strategies.

A strategy *fits* a suggestion when all the specific activities that are precisely mentioned in the suggestion appear in the strategy and the strategy does not include activities whose negations appear in the suggestion. If general activities are mentioned in the suggestion, the strategy must include at least one order that implements every such general activity.

If the parties had agreed upon some details during previous steps of the negotiations, where messages were exchanged between them, and the current suggestion does not contradict these details, then the *fitted* strategy has to fit these details too.

Two details of a suggestion or an agreement contradict one another in the following cases:

1. One detail negates the activity of the other detail.
2. The two details include different actions for the same object.

The negotiator assumes that a detail of an agreement is valid until the other party says the opposite or does something that contradicts this assumption. We found out from human negotiators that they make the same assumption. Therefore, an automated agent that negotiates with humans should follow this line of reasoning, in order to avoid unnecessary conflicts with the other players.

Diplomat uses the following algorithm in order to find fitted strategies to a suggestion it received.

Algorithm 6.1: *Finding fitted strategies for a suggestion*
Input: A suggestion and a set S of stored strategies.
Output: A set of strategies *FittedStrategies* that fit the suggestion.

EnemiesSet = Enemies(Suggestion);
AlliesSet = Allies(Suggestion);
DetailsSet = Details(Suggestion);
If Previous Agreement between *Diplomat* and AlliesSet exists then
DetailsSet = *DetailsSet*∪
{*Details(Agreement)**Contradictory(Details(Agreement),DetailsSet)*};

Step 1: $S' = \{s|s \in S, s$ is a strategy for AlliesSet against EnemiesSet$\}$
SpecificDetails = $\{d|d \in$ DetailsSet, d is an order such as *Move*, *Support*, *Con*
or *Valid*$\}$
GeneralDetails = $\{d|d \in$ DetailsSet, d is a general suggestion such as *Atta*
Defend, *Enter*, Help$\}$
NegationDetails = $\{d|d \in$ DetailsSet, d is a negation of an order or a gene
suggestion$\}$;
FittedStrategies = $\emptyset$;

For any $s \in S'$ if the following conditions hold

1. SpecificDetails $\subseteq s$
2. *for any general detail in* GeneralDetail *exists an order in* s *that impleme*
the detail
3. *for any detail in* NegationDetails *there is no order in* s *that implement*
then FittedStrategies = FittedStrategies $\cup \{s\}$;

Step 2: If FittedStrategies == $\emptyset$
then SFdetails = Compilation of DetailSet into SF Format;
If SFdetails $\neq \emptyset$
then apply to SF for strategies for AlliesSet against EnemiesSet w
SFdetails.

Diplomat stores all the strategies that it finds during a season for further
in the current season (usually strategies for one season are not applicable in ot
seasons). It first tries to look for a fitted strategy among the stored ones (S
1). If it fails, it tries to compile the details into SF format[12] and to apply to
SF in order to find fitted strategies (Step 2). The first search among the sto
strategies which are grouped according to the allies and the enemies of
strategies saves *Diplomat* the time needed to generate and evaluate strategi
Also, there are cases where *Diplomat* cannot compile the details into a SF,
can search for fitted strategies among the stored strategies.

If *Diplomat* receives general suggestions such as Attack, Defend, Enter, Le
or Help, while trying to look for a fitted strategy among the stored ones, it t
to figure whether there exists a strategy that already implements this suggest
((1) in the algorithm).

The Attack, Leave, and Defend sentences include the following informati
the active powers, the places, and possible directions from where to perform
activity (see Kraus 1988). The Leave sentence includes the active powers and
site that must be left. The Help sentence includes details about the active po
and the kind of help that will be given: Attack, Defend and Enter. It also inclu
all the details for these actions.

In looking for the fitted order for an Attack or for a Defence senten
Diplomat looks for an order that has an attack purpose or a defend purpose,
'space' field of this order is one of the places mentioned in the sentence, and

[12]SF format is the format of the orders of the strategies we defined above, i.e. each or
includes the active unit, the type of the activity, and the locations on the board that
related to this activity. The purpose indicates whether the action is part of a plan to att
a specific area or part of a plan to defend a specific area.

active power of the order is among the powers of the sentence. If the 'directions' field of the suggested sentence is not empty, the active units must be located in one of them. The Enter sentence is treated like the Attack sentence. In the case of a Leave sentence, the fitted order can be a Move order or a Valid order which moves from the site of the Leave sentence and is owned by one of the specified powers. We will demonstrate this algorithm in the following example.

Example 6.1: *Finding fitted strategy to a suggestion*
Diplomat (Russia) received the following message in the beginning of Fall 1902 (the details of the situation are presented in Tables 1, 2 and 3).

MESSAGE 10 FROM Austria TO Russia:
I would like to suggest to you the following facts:
Austria will support with its army at Serbia,
Russia's attempt to move from Bulgaria to Greece now, (a)
and Russia will support with its army at Rumania,
Russia's attempt to move from the Black Sea to Bulgaria, (b)
and Austria will not attack Rumania from Budapest from now on. (c)
END OF MESSAGE

The previous agreement between Russia and Austria was against Turkey and Germany. Russia promised Austria to support Austria's move from Serbia to Greece in Fall 1902. Detail (a) of the suggestion contradicts this promise and since there are no more previous agreed details, the details set does not change as a result of the first 'If' in Algorithm 6.1 (1).
The first and the second details (a,b) are specific and therefore in *Step 1* they are assigned to SpecificDetails together with the orders that are supported in those details, i.e. *A. (R) Bulgaria moves to Greece.* The third detail (c) is a negation and therefore assigned to NegationDetails.

Any strategy that includes the orders (1), (2), (3) and (4) below and does not include (5) fits this suggestion.

1. A. (R) Bulgaria moves to Greece in order to attack Greece.
2. A. (A) Serbia supports A. Bulgaria moves to Greece in order to attack Greece.
3. F. (R) Black Sea moves to Bulgaria.
4. A. (R) Rumania supports F. Black Sea moves to Bulgaria.
5. A. (A) Budapest moves to Rumania.

Diplomat found several strategies that fit the suggestion, and compared them with other strategies. *Diplomat* preferred another strategy in which the fleet in the Black Sea did not move and without attacking Greece, and suggested it to Austria.

After finding fitted strategies for the suggestion, *Diplomat* chooses the best strategy among them. The chosen strategy gives *Diplomat* an estimation of its expected benefits from reaching an agreement based on the suggestion it received. Using this estimation, *Diplomat* compares the suggestion to other possibilities. First it compares it with the best strategy for co-operation with this ally against the enemies.

If the expected profit from the chosen strategy (that fits the suggestion) is not less than the expected profit from the best strategy, *Diplomat* considers the

suggestion positively. Otherwise, if *Diplomat* already offered the best plan i previous step of the negotiations, and he was not accepted, then if the expec profit from the partner's suggestion is not significantly smaller than the expec profit from the best plan, *Diplomat* will also consider the suggestion positive

Before deciding whether to consider the suggestion positively, *Diplomat* compares it with possible strategies for other coalitions.

If *Diplomat* decides to react positively to the suggestion, it will send a posi message to its partner and may add more details when needed (see Kraus 1 Kraus and Lehmann 1988a). *Diplomat* will also update its agreements table, adding the newly agreed upon details. If a new detail contradicts an old then the old detail is removed from the agreement table.

When the negotiator does not want to accept the suggestion it may suggest own better plan (if this has not been done yet), or pretend that he likes suggestion in order to mislead the other agent (see Kraus and Lehmann 198 or send a negative answer.

7. The way to offer suggestions

Suppose the negotiator has a good co-operative plan for negotiations with ano agent, he still has to decide what details from that plan to reveal to his part If the negotiator wants specific help from his partner, or is willing to give help to his partner, or he wants to keep his partner from doing something, obvious that he has to discuss it with him. In such cases it is worthwhile gi detailed offers, since vague propositions do not sound as reliable as the n detailed ones. On the other hand, it is not advisable to discuss parts of the that are of no importance to the partner, in order not to commit oneself and reduce one's reliability. Limiting the amount of information given to a partn also advantageous if the partner turns into an enemy.

It must be decided whether to discuss those parts of the plan that are s related to that partner. Independent of the quality of the plans, every discussion can be helpful to the partner, since it gives him new ideas. There the negotiator has to decide whether he wants to help his potential ally or

Diplomat's willingness to give advice to its ally increases when the ally is relative to their common enemies, since *Diplomat* is more interested in hel a weak ally to prepare a good plan than in helping a strong ally get even stro Nevertheless, in the games in which *Diplomat* participated, we observed tha human players do not like to receive advice not related at all to *Diplomat*'s u

We will demonstrate *Diplomat*'s way of offering suggestions in the follo example.

Example 7.1: *Diplomat* chose the following strategy as the basis for negotia
A. Greece (I) Moves to Serbia in order to attack Serbia.
A. Rumania (R) Supports A. Greece to Serbia in order to attack Serbia.
A. Bulgaria (R) Moves to Constantinople in order to attack Constantinopl
F. Black Sea (R) Supports A Bulgaria to Constantinople in order to a Constantinople.
A. Albania (I) Supports A. Greece to Serbia in order to attack Serbia.
F. Tyrrhenian Sea (I) Moves to Tunis in order to attack Tunis.
F. Apulia (I) Moves to Venice in order to attack Tunis.
Expected outcomes: Aver: 19581 Max: 33284 Min: 2573 Russia: 7273 Italy: 1

Diplomat decided to discuss with Italy details that are relevant to the attack on Serbia since it was willing to support Italy's move in this attack. It decided not to discuss the details of its attack on Constantinople and Italy's attacks on Tunis and Venice since co-operation between the parties was not required.

After making those decisions, *Diplomat* sent Italy the following message:

MESSAGE 34 FROM Russia TO Italy:
Yes, absolutely.
Now, my suggestion contains the following points:
Italy will support with its army in Albania,
Italy's attempt to move from Greece to Serbia in this season, and Russia will support by its army in Rumania,
Italy's attempt to move from Greece to Serbia in this season, and Italy will move from Greece to Serbia in this season.
END OF MESSAGE.

8. Testing *Diplomat*

In general, there are several ways to evaluate whether a negotiator is performing well in an environment of intensive negotiations. The main indications are whether he achieved his goals and the advantages he gained through his agreements. In addition, one can analyse the negotiator's behaviour during the negotiations and determine how well he did on specific tasks.

The goal of a Diplomacy player is to win the game. In long games he needs to gain control over the majority of the board, i.e. to control 18 supply centres. In short games, in which it is predetermined exactly how many moves will be played in the game, the winner is the player with the largest number of units on the board.

In all our experiments the negotiation was done by electronic mail and the identities of the players were kept secret. The messages were written using the formal language we have developed. The human players used a special editor which helped them compose their messages.

Diplomat took part in the following games:

1. A long game with 6 human players. The human players were experienced Diplomacy players. *Diplomat* won this game.
2. A short game. Two powers played by *Diplomat* and five by human players; powers did not know which players were human and which were computers. One of the *Diplomat* entities won the first place and the other won the second (i.e. had the second largest number of supply centres under its control).
3. A short game with 6 human players. *Diplomat* won this game.
4. A short game where *Diplomat* played all the powers. This was only a test game.
5. We presented *Diplomat* with various subgame situations that were taken from human games. In each of those subgames, we converted the situation into *Diplomat*'s format and let him play. We presented *Diplomat* with the messages that the human players sent in the real game as adequately as possible. Each such subgame lasted a season. *Diplomat* played in a reasonable way, but not always the same way the human players did.

Our findings show that *Diplomat* played well in the games it participated More games are required in order to gain statistical results about *Diplo* abilities in full games.

On the other hand, since *Diplomat* played in more than 100 seasons games 1, 2, 3, and 4 above), we were able to collect enough data to exan different aspects of *Diplomat*'s negotiation behaviour.

With whom to negotiate: We examined *Diplomat*'s behaviour in deciding v whom to negotiate (Section 5). In general, *Diplomat* needs to consider 3^6 poss coalitions in each season. Even after restricting the number of possible allies the number of possible enemies to 3, there are still 461 possibilities.

We checked 13 seasons in the game where *Diplomat* played the role of Ru and *won* (game 2). It considered 74 combinations of possible coalitions in beginning of those seasons (Algorithm 5.1). In those seasons, *Diplomat* considered 2 coalitions as a result of a message it got from another po (Algorithm 5.3). On average it considered 5.8 coalitions (from 461 possibilit in a season. In those seasons *Diplomat* (Russia) was involved in 47 negotia processes,[13] (average 3.6 per season) where in 30 negotiation processes (63 % the cases) *Diplomat* was the initiator of the negotiation (see Table 5).

In the same game (game 2) where *Diplomat* played the role of England finished second, it considered 46 coalitions in 13 seasons. Two coalitions v considered as a result of a message it got from another power. In this case average is 3.5 coalitions per season. In those seasons, *Diplomat* (England) involved in 35 negotiation processes (average 2.69 per season) where in 17 c *Diplomat* (England) was the initiator (48 %).

We compared those findings with the behaviour of human players. The ave number of negotiation processes for a power varied between 3.28 and 1.72 the average of 2.5 negotiation processes for a power for a season. On aver each power was the initiator of the negotiation process in 45 % of those ca Those findings indicate that *Diplomat* negotiates with other parties above average number of times human players do.

Evaluation of suggestions: In order to analyse *Diplomat*'s behaviour w evaluating suggestions (see Section 6), we examined 114 detailed suggestions

Table 4. Summary of performance while evaluating suggestions. The messages we count as 'messages that were understood' are those that the player did not ask for clarifications from the sender of the message

Player	Messages that were checked	Message was understood	
		Messages	%
Diplomat	114	101	89 %
Humans	90	83	92 %

[13]We considered a negotiation process in each season that *Diplomat* exchanged at one message with the other party.

Table 5. *Diplomat*'s and other players' behaviour during the negotiation period. The second column (from left) indicates how many coalitions *Diplomat* considered. No data is available about the number of possible coalitions the human players considered. The third column indicates the number of negotiation processes in which a specific player was involved. The last column is the percentage of the negotiation processes initiated by the specific player. Germany was eliminated from the game after 6 seasons. Turkey was eliminated from the game after 10 seasons.

Player	Coalitions considered	Negotiation processes in 13 seasons	average per season	Initiator out of all processes
Diplomat Russia	76	47	3.61	63 %
Diplomat England	46	35	2.69	48 %
Austria	—	42	3.2	52 %
France	—	27	2.07	33 %
Germany	—	23 (in 6 seasons)	3.83	60 %
Italy	—	30	2.3	33 %
Turkey	—	18 (in 10 seasons)	1.72	47 %
Diplomat average	61	41	3.15	55 %
Human average		33 without (G) and (T)	2.62	45 %

were received by *Diplomat* during negotiation periods (see Table 4). In 13 (11.4 %) cases, *Diplomat* asked for more details from its opponent because it could not find a strategy that would enable it to evaluate the suggestion. We also examined 90 suggestions that were received by human players. We found out that in 7 (7.7 %) cases they asked for more details. In this task *Diplomat* did slightly worse than human players, but still succeeded in evaluating 88.6 % of the suggestions we have examined.

Estimation of other powers' interrelations: We also tried to evaluate out how good *Diplomat* was in estimating the other powers' interrelations (see Section 5). We examined 260 relations between powers in a Diplomacy game. In 223 cases (85.7 %) *Diplomat* estimated correctly the existence of those relations. In 25 cases (9.6 %) *Diplomat* did not discover relations that existed and in 12 cases (4.6 %) *Diplomat* estimated that relations existed and was wrong. According to our experience, *Diplomat*'s estimations are at least as good as human players'.

Prediction of the ally's intention to keep an agreement: We also examined 63 agreements that were signed between *Diplomat* and another power, and *Diplomat* predicted successfully the ally's intention to keep an agreement in 92 % of all the agreements it had signed (see Table 6). The human players predicted successfully *Diplomat*'s intention to keep an agreement in 86 % of all the agreements they had signed with *Diplomat*.

We conclude that in the tasks that we examined, *Diplomat* performed at least as well as human players. Since all these tasks are common to any negotiation

Table 6. *Diplomat* and its allies' behaviour after signing agreements. 63 agreements were checked during 23 seasons. We distinguish between 4 cases (from left): *Diplomat* and its allies kept the agreement, *Diplomat* and its allies broke the agreement, *Diplomat* broke the agreement and its allies kept it and vice versa. The number of agreements in which *Diplomat* estimated correctly the intention of its allies to keep an agreement is listed in the second row.

	Both kept	Both broke	*Diplomat* broke	Other broke	Sum
Sum	51	4	6	2	63
Diplomat correct	51	3	4	0	58

situation, these findings seem to indicate that these algorithms will be useful i other environments.

9. Conclusion

We have proposed methods for solving different aspects of automated negotiation with whom to negotiate; evaluation of suggestions and the way to offer suggestions We used our methods to develop the system *Diplomat*, which plays Diplomac as one of the players. Diplomacy is a complex environment, where the set o possible strategies is very large. A mediator is not available, the agents may breal their promises, close co-operation between different agents is needed and possibl coalitions between other agents must be taken into account.

From the statistics we presented above, one can see that *Diplomat* is a goo negotiator. Further work is needed to extend the use of the tools and the idea of *Diplomat*, to other domains.

Acknowledgements

This work was partly done while the first author was at the Institute for Advance Computer Studies and Department of Computer Science at University of Maryland College Park and at the Department of Computer Science at the Hebre University, Jerusalem. We would like to thank Arcot Rajasekar and Jonatha Wilkenfeld for their comments and the anonymous referees for their constructiv criticism that greatly improved the presentation of this paper. We would also lik to thank Avalon Hill Game Company for allowing us to reproduce their Diplomac map in this article.

References

The Avalon Hill Game Co. (1976) *Rules for Diplomacy*.

Davis, R. and Smith, R. G. (1983) Negotiation as a metaphor for distributed problem solving *Artificial Intelligence*, 20:63–109.

Durfee, E. H. (1988) *Coordination of Distributed Problem Solvers* (Boston: Kluwer Academi Publishers).

Fisher, R. and Ury, W. (1981) *Getting to Yes: Negotiating Agreement Without Giving In* (Boston: Houghton Mifflin).

Karrass, C. L. (1970) *The Negotiating Game: How to Get What You Want* (New York: Thomas Crowell Company).

Kraus S. (1988) *Planning and Communication in a Multi-Agent Environment*. Ph.D. Dissertation, Hebrew University, Jerusalem. (Written largely in Hebrew).

Kraus, S. and Lehmann, D. (1988a) Automated negotiator. Technical Report 88-7, Leibniz Center for Computer Science, Hebrew University, Jerusalem.

Kraus, S. and Lehmann, D. (1988b) Diplomat, an agent in a multi-agent environment: An overview. In *Proceedings of the Seventh Annual IEEE Phoenix Conference on Computers and Communications*, Arizona, 434–438.

Kraus, S. and Lehmann, D. (1989) Using interacting goal-processes for automated negotiation. In *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, 209–219.

Kraus, S. and Wilkenfeld, J. (1990) The function of time in cooperative negotiations. In *10th International Workshop on DAI*, Texas.

Kraus, S. and Wilkenfeld, J. (1991a) The function of time in cooperative negotiations. In *Proceedings of AAAI-91*, California, 179–184.

Kraus, S. and Wilkenfeld, J. (1991b) Negotiations over time in a multi agent environment: Preliminary report. In *Proceedings of IJCAI-91*, Australia, 56–61.

Kraus, S., Ephrati, E. and Lehmann, D. (1989) Evaluation of suggestions during automated negotiations. In *Proceedings of the 11th Cognitive Science Conference*, 299–306.

Levy, J. and Dimitrovski, Y. (1984) *The Ylisp 2 Manual*. Hebrew University, Jerusalem, Israel.

Malone, T. W., Fikes, R. E., Grant, K. R. and Howard, M. T. (1988) Enterprise: A marketlike task schedule for distributed computing environments. In B. A. Huberman (ed.), *The Ecology of Computation* (North Holland), 177–205.

Matwin, S. , Szpakowicz, S., Koperczak, Z., Kersten, G. E. and Michalowski, W. (1989) Negoplan: An expert system shell for negotiation support, *IEEE Expert*, **4**(4):50–62.

Rosenschein, J. and Genesereth, M. R. (1985) Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, California. 91–99.

Sathi, A. and Fox, M. S. (1989) Constraint-directed negotiation of resource reallocations. In L. Gasser and M. N. Huhns (eds), *Distributed Artificial Intelligence*, Volume II (London: Pitman/Morgan Kaufmann) 163–194.

Sathi, A., Morton, T. E. and Roth, S. F. (1986) Callisto: An intelligent project management system, *The AI Magazine*, **7**(5): 34–52.

Sycara, K. P. (1987) Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods. Ph.D. Dissertation, School of Information and Computer Science, Georgia Institute of Technology.

Walker, R. (1984) *A Gamer's Guide to Diplomacy* The Avalon Hill Game Company, Baltimore, Maryland.

Willkins, D. (1983) Using pattern and plans in chess, *Artificial Intelligence*, **14**(2):165–203.

Zlotkin, G. and Rosenschein, J. (1989) Negotiation and task sharing among autonomous agents in cooperative domain. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI. 912–917.

Zlotkin, G. and Rosenschein, J. (1990). Negotiation and conflict resolution in non-cooperative domains. In *Proceedings of AAAI-90*, Boston, MA. 100–105.