Thoughts on Programming a Diplomat

Michael R. Hall
hall@rocky.bellcore.com
Bellcore
445 South St., Room 2D-335
Morristown, NJ 07960

Daniel E. Loeb
loeb@nestor.greco-prog.fr
LABRI
Université de Bordeaux I
33205 Talence, France

Abstract

In contrast to two-player sequential move games, multiplayer games with simultaneous move have not been extensively studied. Diplomacy [1] is a multi-player simultaneous move game that supplies some new challenges in creating a computer player, namely: the difficulty of determining the true strength of a position, the tremendous size of the search tree, and the necessity of negotiation. The Diplomacy Programming Project presents here some of its ideas for overcoming these problems.

Première étude sur la programmation d'un diplomate

Contrairement aux jeux à deux joueurs dans lesquels chacun joue à tour de rôle, on n'a jamais vraiment étudié les jeux à plusieurs joueurs jouant chaque coup simultanément. Le jeu de *Diplomacy* est un exemple type de cette dernière categorie de jeux. Il est particulièrement interessant de l'étudier en vue de l'elaboration d'un "diplomate informatique" : il faut en effet pour cela déterminer quelle est numeriquement la veritable force d'une position, cela à l'interieur d'un arbre de recherche gigantesque. Il faut d'autre part accorder un rôle essentiel à la négotiation. Nous répondons ici, dans le cadre du *Diplomacy Programming Project*, à un certain nombre de ces questions.

1 Introduction

The theory of two player games is beginning to become well understood. On the theoretical front, we have such works as *Winning Ways* [3]. Meanwhile, on a more practical level advances in game theory and artificial intelligence have given birth to computer programs which play such games as chess, checkers, and othello amazingly well.

In a multi-player game (and in particular the game diplomacy), the results are nowhere as clear. There are three issues at stake here:

- First, the "paradox" that in a multiplayer game, the best move is not always one that gives you the "strongest" position. As an example of this paradox, consider the following game involving three players.

  > The three players A, B, and C are involved in a gunfight. They are to take turns shooting one bullet until only one player remains. C is a perfect shooter; he kills his target every time. B has an accuracy of two kills in three attempts. Finally, A only hits his target one out of every three times. To be fair, it is decided that A should begin the gunfight, to be followed by B, and then C (if they are alive), and so on.

  It is amazing but true that the best strategy for player A is to purposely miss. If by chance, he hits one of his opponents, then he now becomes the target for the remaining one. By keeping both opponents alive, he guarantees himself a chance to win on the second round!

  Similarly, in Diplomacy [1], the goal is to accumulate a total of eighteen *supply centers*, yet it is a general maxim that Germany should not attempt to capture three supply centers during the first couple of turns. By doing so, he would threaten too many other countries. Germany is thus in a stronger position if he is "weaker" by one unit. He thus grows at a decent rate while not provoking any attacks.

  As a result of this paradox, the evaluation of the strength of a Diplomacy position is no longer as straightforward as it is in a game such as chess. In chess, white never avoids taking black's queen for fear of black "ganging up on him." However, in Diplomacy, Germany often decides to avoid taking *Belgium* for fear of the consequent formation of a coalition against him.

- The other issue involved in all multi-player games is that of negotiation. It is this facet of the game which gives Diplomacy its name. No player can hope to win a game of Diplomacy by remaining silent. He only owns one-seventh of the force on the board. Even if his strategy is superior to the other players, he could not win against a force six times greater than his own.

  The analogy in chess would be a grandmaster starting a game of chess with only a knight and three pawns. The grandmaster's advantage in ability would not be enough to overcome the corresponding lack of material.

- Finally, Diplomacy is distinguished among all multi-player games by the fact that all players submit moves for each of their units simultaneously. Since each of the seven players begins with at least three units, and each unit has at least four possible legal moves, a quick computation reveals that even a "one-ply" search is impossible.

On the other hand, in chess there is no position with over 203 possible next-moves [16]. In fact, most common positions have about three dozen possible continuations. Thus, it is fairly easy to execute an extensive search of reasonable depth.

The Diplomacy Programming Project seeks to create automated diplomacy players, or *diplomats*.[1] A program called the *Diplomacy Interface* has been created to provide a common environment for diplomats, so that they may communicate with each other and have their moves carried out. We are currently designing actual diplomats. In so doing, we are pressing simultaneously on three fronts of computer game-playing: First, positions must be evaluated using new techniques; next, search techniques must be investigated to narrow down the possible options; finally, negotiation heuristics must be devised.

In some sense, the game Diplomacy provides a vehicle to study the interactions among intelligent agents - be they human or computer. Issues of negotiation, cooperative/antagonistic behavior, problem-solving, learning, tactics, strategy and so on all arise in this rich environment, in sharp contrast to one person or two person games with no mechanism for communication outside the game board. The *Diplomacy Interface* can be used by humans as well as computer diplomats, and we thus have a laboratory where theories of human and machine intelligence may be tested, against each other if desired. This means a limited form of the Turing test is also possible.

Before delving into the problem of building diplomats, we present a short review of the rules of the game Diplomacy.

2 Rules

2.1 Basic Rules

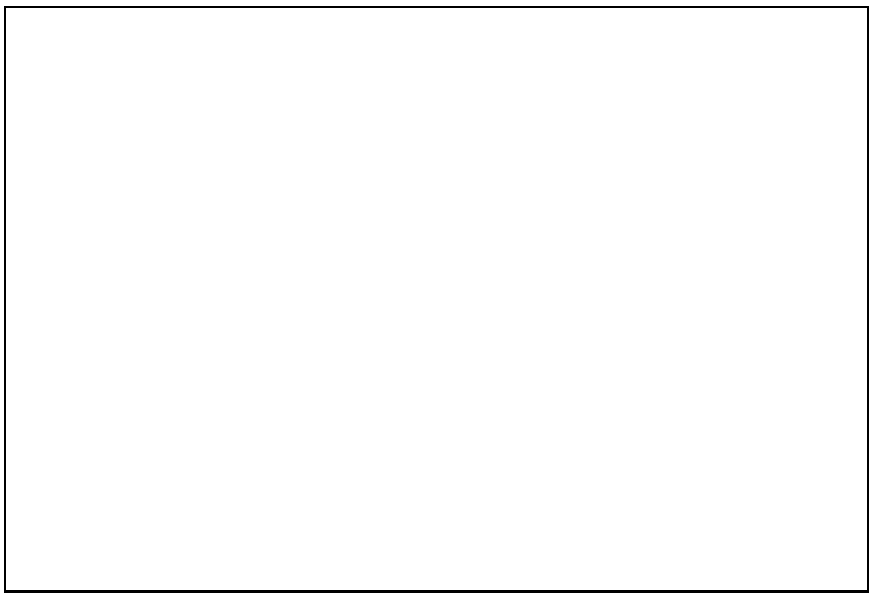
In 1959, Allan Calhammer created Diplomacy with an eye towards recreating the political intrigue of World War I Europe. It is played on a map of Europe (see Table 1) which has been divided into *regions* (called *seas* and *provinces*). The seven players represent the great powers of World War I (Austria, England, France, Germany, Italy, Russia, and Turkey). Each starts with three home cities known as *supply centers*,[2] and an equal number of *units* (either *armies* or *fleets*). Certain *neutral* supply centers are unowned at the beginning of the game.[3]

At the end of each *year* of play, the number of units is again compared to the number of supply centers, and units are *built* or *removed* in order to bring the two equal. Thus, by capturing territory a player can become more powerful. A player wins by capturing 18 supply centers which represents a little more than half the board. The game can also end at any time with a draw by the mutual consent of the remaining players.

Each year is divided into five *seasons*: *Spring* movement, *Summer* retreat, *Fall* movement, *Autumn* retreat, and *Winter* build. The first turn on the game is considered Spring 1901.

[1] Short for "Diplomacy automata"
[2] Except Russia, which has four.
[3] All supply centers are marked by a • in Table 1.

Table 1: Diplomacy Map



2.2 Submitting Orders

The heart of the game lies in the movement seasons. Each movement season is preceded by fifteen minutes of negotiations among the players during which the players devise their strategies. Each player then writes orders secretly for all of his units. Each unit can do one of the following actions:

Move to an adjacent province with the restriction that armies can not enter seas, and that fleets can not move inland and must follow the coasts when moving between coastal provinces.

Hold in place.

Support a unit attempting to remain in or move to a neighboring province on the condition that the neighboring province is accessible by the supporting unit (see the restrictions on movement above for details).

Convoy: An army can move between any two coastal provinces on the condition that there is intervening path of fleets at sea which are ordered to make the convoy.

In this article, all diplomacy orders, units, and regions are written in italics. The symbols *A, F, S, C, H,* and $\rightarrow$ respectively are used to represent armies, fleets, supports, convoys, holds, and moves to. For example, *A Paris S F Brest*$\rightarrow$*Picardy.*

2.3 Resolving Conflicts

After all moves are ready, moves are read and the results are computed following a number of simple rules:

- All moves are successful except as noted below.
- There are three types of conflicts:
  - When two units attempt to switch places (without being convoyed).
  - When two units attempt to enter the same region.
  - When a unit attempts to enter a region whose occupant has not successfully moved out.

  When equally supported units participate in a conflict, they remain in their original regions.
- In case of unequal support of units participating in a conflict, only the best supported unit succeeds in its order.
- If a unit attempts to remain in its region (or fails to leave it) while another unit succeeds in entering that region, then the first unit is *dislodged* and must retreat.
- If a fleet in a convoy is dislodged, then the convoy has no effect.

Additional rules not particularly relevant to this article cover under what circumstances a support can be *cut*.

During the retreat seasons, orders are given secretly and simultaneously for all dislodged units without any preceding negotiation period. Retreats are made according to the same rules as ordinary moves except that one can not retreat to an occupied region, the region the attacker came from, or a region in which two equally well support units *bounced*. There is the possibility of *disbanding* instead of retreating. If two units are ordered to retreat to the same region, then they are both disbanded. Note that any unit disbanded would then be replaced during the upcoming Winter turn if the player retained the same number of supply centers.

The Winter builds immediately follow the Autumn retreats without any intervening negotiations. Players with excess units (in comparison with their number of supply centers) are obliged to *remove* the excess. Players with surplus supply centers can construct new units in their empty *home* supply centers (unless another player owns those supply centers).

3 Search Techniques

3.1 Game Trees

It is difficult to program intelligent machines even in circumscribed domains such as Diplomacy. However, we need not throw up our hands in despair; the fields of game theory and artificial intelligence (A.I.) have at least some of the answers.[3, 4, 5, 7]

Nearly all A.I. or game-playing programs search. Search is merely the exploration of different alternatives. Humans search, but very slowly and with many mistakes. For example, in the Diplomacy game "Tokugawa" [13], when the authors—playing Austria and Turkey—were at war, Turkey was surprised when Austria forced him to disband one of his armies capable of gravely harming Austria; Turkey had not foreseen the move because he had not done a thorough enough search of Austria's possible moves. Computers, on the other hand, have the ability to search quickly and as completely as desired.

There are two main search paradigms: *breadth-first* and *depth-first*. With breadth-first search, one examines all the nodes at the same level at the same time. Depth-first search involves going deep as quickly as possible. In a game, the nodes of the tree represent possible states. These states are reached by making moves (which are the arcs connecting the nodes.)

What are we searching for? An evaluation function is needed to evaluate the value of the resulting positions. For example, the value of the final position in tic-tac-toe is easy to analyze—if you have three in a row, you win; if your opponent has three in a row, you lose; otherwise, when all the spaces are filled, it is a tie. These final position evaluations can be rolled back assuming that O plays optimally and tries to minimize X's score.

Unfortunately, for nontrivial problems, it is impossible to search completely within a reasonable amount of time. Chess is an example of a game that cannot be solved completely because each piece has so many so options, and there are so many turns in a chess game. The combinatorial explosion that occurs during the search is potentially deadly; even on the world's fastest hardware it would take longer than the history of the universe to "solve" chess.

Because the units move simultaneously in Diplomacy, analyzing Diplomacy completely is even more hopeless! Imagine trying to evaluate all the possible opening moves for all the players simultaneously—since each unit has at least four possible orders, there are well over $4^{(3*6+4)} = 17,592,186,044,416$ possible sets of orders for Spring 1901,[4] so we can not even analyze the first turn completely, at least not within a reasonable amount of time on today's hardware. We are again confronted with two of the problems mentioned in the introduction. First, we must find ways to reduce the amount of searching, and second, we must find a position evaluation function that does not depend on searching until the end of the game.

3.2 Best-First Search

Leaving the problem of evaluation aside until Chapter 4, assume we have a human available to evaluate the board positions, and focus on the search itself. There are a few techniques to speed up the search. First of all, obviously you should only look at your units and nearby units of other players. If you want to look at the whole board, then do it in pieces (divide and conquer). Second, you can also use a combination of breadth-first and depth-first search called *best-first* search.

The idea of best-first search is that if you keep looking in what appears to be the right direction at any given time, then you will find the best set of orders more quickly. Generate all the valid orders for each of

[4] The exact number of openings is 4,430,690,040,914,420 (not counting openings in which useless support is given) [9].

the units to be ordered, while pruning out any illegal or stupid orders.[5] Next, evaluate the partial position generated by each of the orders. Only one unit is ordered at a time; the other units are assumed to hold for the moment. (See section 3.4 for other possibilities here.) Choose the order that seems to produce the best position, and fan out to the subsequent partial positions. Then choose the partial position that seems best—this may be in a completely different part of the search tree than you were last working on, and so on.

To implement best-first search, only the current leading edge (i.e., the leaves) of the tree need to be stored in a list. This list is called an *agenda*. More formally, best-first search may be implemented as follows:

1. Initialize the agenda.
2. Loop:
   - Take the first node on the agenda.
   - If the node is a goal node, then return it or store it and keep searching.
   - If the agenda is empty, then stop.
   - Get the successors of the node and evaluate them.
   - Insert the successors into the agenda and keep the agenda sorted by position value.
   - Continue looping.

An example search is given in section 4.2.

3.3 Even Better Best-First Search

Of course, this best-first approach cannot be applied directly to analyze the effects of the enemy's moves. You could take the paranoid conservative approach of evaluating a complete set of orders according to the worst position that can be obtained by some combination of enemy orders. Or better you could assume that the enemy is attempting to maximize his own position given your orders, which does not necessarily undermine your position.[6]

In other words, in order to generate your best orders, it is helpful to first compute good moves for the other players. To do so, you may wish to run seven processes in parallel (one for each country), which each try to devise orders against the best current orders of the others. The processes would communicate the current "best-move-being-considered" to each other on rendezvous. Of course, the best move being considered for your own country would also be sent into the "game master" every time it changed. By so doing, you are guaranteed to have submitted the best move so far discovered by the deadline.

[5] Practically speaking, we can avoid generating illegal orders. However, stupid orders are harder to avoid. A certain number of them can be avoided by the use of an "expert system" weeding out sets of orders violating certain maxims (such as "Never cut your own support.") (See section 4.3.3.) However, a certain number of stupid orders remain in our search to be examined and rejected by the position evaluation routine.

[6] Since Diplomacy is a multi-player game, it is not necessarily zero sum. That is to say, cooperative behavior is possible.

The process associated with each country would occasionally stop and demand the results of the other processes. The search would then be continued if these moves were already those being used as the "background" for the initial search. In the more likely case that one of the other countries is now considering a new set of moves, we restart the search using our last set of moves as our *seed*. (See section 3.4.)

If you can come up with a number of alternative sets of enemy orders, then you can evaluate a candidate set of orders according to how well it does on average given these various sets of enemy orders. This should result in a very robust set of orders that would work well in a wide variety of situations, but would also allow the diplomat to take some worthwhile risks.

Best-first search will not necessarily arrive at a complete set of orders quickly, so you may wish to modify it slightly to *first* do a form of depth-first search, choosing the best position at the current level to expand (but always moving deeper and never jumping to a different portion of the search tree). Thus, it will rapidly find a set of mediocre orders to turn in just in case the rest of the search bogs down and fails to terminate within the available time. After this set of orders is found, it switches to the normal best-first search.

One way to optimize best-first search is to have it prune redundant positions from the search tree. In Diplomacy, this is accomplished by sorting each set of examined orders in some way and then caching them. Any time the diplomat starts to evaluate a "new" set of orders, it first checks the cache to make sure that it is not simply a permutation of a set of orders that have already been evaluated. This results in exponential time savings with no loss of completeness, since whole redundant subtrees get pruned.

Another approach to speed up best-first search is to modify it to become a *beam search*, where a limit is put on the maximum number of nodes in the agenda at any one time. The worst positions are discarded when the beam limit is exceeded.

Given enough search time, we will include Summer retreats in our search for Spring moves, and the Autumn retreats and Winter builds in our search for Fall moves. This is feasible since the number of possible retreats and builds is usually quite limited; we are not facing the same sort of combinatorial explosion described above. Moreover, there are many strategies that can only be seen by looking ahead that far.

One such look-ahead strategy is called the *offensive retreat*, which takes advantage of the fact that a unit can often retreat "forwards." For example, if England attacks the French *F Belgium* with support from *Holland* while the *North-Sea* is open, France can retreat there and menace the entire East coast of England as well as Scandinavia and the Lowlands. It is equally important to look ahead at the builds, if Austria-Hungary is at war with Turkey and wants to build a fleet, then he should take care during the Fall movement that *Trieste* is left open for such a build.

Even if anticipating retreats and builds were not needed for strategic reasons, they are clearly needed on a diplomatic level. Diplomacy is not allowed before retreat and build season. Thus, any message concerning those seasons should be anticipated and sent to allies before the preceding movement season.

However, looking ahead to future movement seasons will often be futile, because the computer is likely to be wrong about the opponents' moves and the mistakes will get compounded for each turn looked ahead. Nevertheless, it is certainly worth an attempt if the computer has identified two or more near optimal sets of orders and no other orders at the current turn seem worth exploring.

3.4 Seeds

So far we have focused mainly on step two of the best-first search: the "loop." However, the initialization of the agenda is also of primary importance. The agenda must start off with a set (or possibly several sets) of orders for all countries which serve as the default for each unit. Possible changes to these orders are considered as the search progresses.

However, a great deal of time can be saved by choosing a good set of initial orders. In theory, we have no idea where to start. For this reason, the example considered below in section 4.2 is arbitrarily initialized with the possibility of all units holding. From this starting point, a reasonable set of orders is devised.

In practice, we can do better sometimes. For example, it is more reasonable to assume that units which remain in the same position as the last turn will repeat their movement, than to assume that they will hold. Moves suggested, promised, or threatened by other players can be easily tested by using those moves as a seed. Moreover, as mentioned above, while computing the orders for one country, we use the results of our computations for other countries as a background along with the previous results in the most recent calculation for that country. As we will see below, predictability can be avoided by choosing a random seed, and conjectured alliances can be verified by using the actual moves of the previous turn as a seed and measuring its stability.

4 Position Evaluation

4.1 Simple Model

A static position evaluator is a necessity for a diplomat. There are many issues at stake in the design of such an evaluator, so the following describes merely our thoughts on the subject. We assume that the diplomat has a module that knows the rules of Diplomacy well, if not perfectly. A position and a set of orders is passed into this module, and it produces a new position. Then this position must be evaluated.

Suppose we are trying to develop a position evaluation function for Turkey. Now imagine a mountain with its peak in Turkey, sloping down to the Balkans, Austria, and Russia, and then down further to Italy and down further to France, maybe a bit higher through the Mediterranean and around Iberia and the critical Straights of Gibraltar, and then down further to Germany, England, and Scandinavia. The position evaluation at any given time could be the sum of the "altitudes" of the territories controlled by Turkey. This means that Turkey is very, very valuable to Turkey, which is what we would hope. However, this does not mean that Turkey leaves all her units there; she still controls the mountain peak when she leaves it to take the lower ground provided no enemy sneaks through the front lines. Therefore, sets of orders that expand Turkey's influence to other high "altitude" areas are generally preferred over ones that would expand to low "altitude" areas.

Here we will give an example of an "influence" measure. If a player has a unit in a territory, it controls that territory. Otherwise, if a territory is vacant, the nearest units share in the control of it. For example, if the closest units to *Bulgaria* are two regions away, say, a Russian *F Sevastopol*, a Turkish *A Budapest* and Turkish *F Ionian-Sea*, then Turkey gets two-thirds of how much she values *Bulgaria*, while Russia gets one-third of how much she values *Bulgaria*. The theory behind this is that if there were suddenly a race for the territory, then

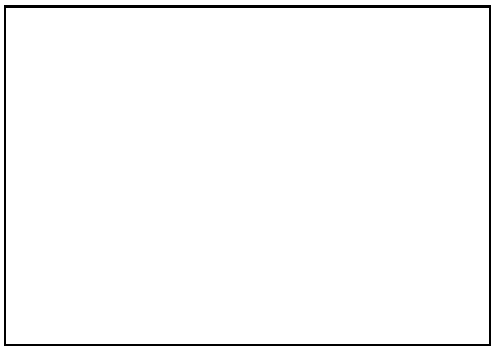
all the units would get there at about the same time. Distance should take into account the accessibility of territories to armies/fleets; thus, for example, any army is infinitely distant from the *Ionian-Sea*.[7]

This distance measure of control will cause the diplomat to desperately try to avoid letting "raiders" slip through its front lines, and to try to slip its own units through enemy lines. However, since the diplomat considers enemy territory less valuable than his own, the diplomat will never try to slip a unit through at the expense of his own security. This is to be desired, since a war without organized fronts between two countries can rapidly lead to mutual destruction.

4.2 Example

In practice, "book openings" will be stored to give the program a little head start, but as an exercise consider the analysis of the opening moves for Turkey. First, the diplomat adjusts the values of holding certain territories according to its current foreign policy, the season, and so on. Let us suppose that the territories have these values: *Constantinople=100, Ankara=90, Smyrna=90, Syria=50, Armenia=60, Black-Sea=60, Sevastopol=70, Rumania=50, Bulgaria=70, Greece=60, Aegean-Sea=60, Eastern-Mediterranean=50, Ionian-Sea=50.* (See Table 2.) Next the diplomat begins the best-first phase. Initially, having chosen "all units hold" as our seed,

Table 2: Turkey and its Neighbors



we start at time $t = 0$ with a one element agenda, the null order set (Table 3). In the above notation, the

Table 3: Agenda $t = 0$

({} = 815)

outer parentheses mark the beginning and end of the agenda, the braces surround the current order set, and

[7] These distances would be calculated one time only, and then stored in the form of a table which could also serve as an adjacency matrix. Thus, these distances do not take convoys in account.

the number after the equal sign is the evaluation of the position achieved by this order set. The null order set is equivalent to "all hold." The position evaluation of "815" was obtained by summing up the values of the territories occupied by Turkey plus a proportion of the values of territories that are partially controlled. (For example, it gets 40 points for *Armenia*, 2/3's of its total's worth of 60, because two of Turkey's units could move there in one turn, while only one of Russia's could move there in one turn.) A 100 point bonus is given for owned supply centers and a 50 point bonus for opponent/neutral supply centers that are occupied in the Spring.

Next, this element is removed from the agenda, and all the legal moves for each individual unit are generated, the resulting positions are evaluated, and the order sets are sorted (Table 4). There is no need

Table 4: Agenda $t = 1$

({*A Constantinople→Bulgaria*}	=	959,
{*F Ankara→Black-Sea*}	=	870,
{*A Smyrna→Armenia*}	=	845,
{*F Ankara→Armenia*}	=	845,
{*A Constantinople→Smyrna*}	=	815,
{*F Ankara→Constantinople*}	=	815,
{*A Constantinople→Ankara*}	=	815,
{*F Ankara→Constantinople*}	=	815,
{*A Smyrna→Constantinople*}	=	815,
{*A Smyrna→Ankara*}	=	815,
{*F Armenia S A Constantinople*}	=	815,
{*F Armenia S A Constantinople*}	=	815,
{*A Constantinople S F Ankara*}	=	815,
{*A Constantinople S A Smyrna*}	=	815,
{*A Smyrna S A Constantinople*}	=	815,
{*A Smyrna S F Ankara*}	=	815,
{*A Smyrna→Syria*}	=	815,
{*A Constantinople H, F Ankara H, A Smyrna H*}	=	815)

to generate support orders for moves before other units have moved. Also, there is no need to generate hold moves explicitly in the search, since units are always assumed to hold if they do nothing else, and holding will not change the position evaluation.[8] Each of the possible moves is tried individually and the resulting partial positions are evaluated. *A Constantinople→Bulgaria* is extremely valuable, giving Turkey new influence in valuable areas as well as occupying a neutral supply center.[9] *F Ankara→Black-Sea* is good, since it gives Turkey influence in *Rumania*, but it is not as good as *A Constantinople→Bulgaria,* especially since it does not grab a supply center. (Remember, the computer does not do all these rationalizations, at least not at this time—it just looks to maximize the evaluation function.)

Now, *A Constantinople→Bulgaria* is made as the first step in the search and this order set is removed from the agenda. Next, all the legal second moves given *A Constantinople→Bulgaria* are generated and inserted into the agenda. (See table 5; asterisks mark the new elements.) Here the moves that increase the position evaluation the most are *F Ankara→Constantinople* and *F Ankara→Black-Sea*, since these are at the top of the list. The

[8] Notice the "all hold" element at the end. It inserts this near-equivalent of {}=815, just in case holding is actually the best thing to do. If this were to ever get to the top of the agenda, it would test holding all its units against enemy orders.

[9] "There is no substitute for *A Constantinople→Bulgaria*." [18]

Table 5: Agenda $t = 2$

(*	{*A Constantinople→Bulgaria,F Ankara→Black-Sea*}	=	1014,
*	{*A Constantinople→Bulgaria,F Ankara→Constantinople*}	=	1004,
*	{*A Constantinople→Bulgaria,A Smyrna→Armenia*}	=	979,
*	{*A Constantinople→Bulgaria,A Smyrna→Ankara*}	=	959,
*	{*A Constantinople→Bulgaria,A Smyrna S F Ankara*}	=	959,
*	{*A Constantinople→Bulgaria,A Smyrna→Syria*}	=	959,
*	{*A Constantinople→Bulgaria,F Ankara H, A Smyrna H*}	=	959,
*	{*A Constantinople→Bulgaria,F Ankara→Armenia*}	=	949,
*	{*A Constantinople→Bulgaria,F Smyrna→Constantinople*}	=	939,
	{*F Ankara→Black-Sea*}	=	870,
	{*A Smyrna→Armenia*}	=	845,
	{*F Ankara→Armenia*}	=	845,
	{*A Constantinople→Smyrna*}	=	815,
	{*F Ankara→Constantinople*}	=	815,
	{*A Constantinople→Ankara*}	=	815,
	{*F Ankara→Constantinople*}	=	815,
	{*A Smyrna→Constantinople*}	=	815,
	{*A Smyrna→Ankara*}	=	815,
	{*F Armenia S A Constantinople*}	=	815,
	{*F Armenia S A Constantinople*}	=	815,
	{*A Constantinople S F Ankara*}	=	815,
	{*A Constantinople S A Smyrna*}	=	815,
	{*A Smyrna S A Constantinople*}	=	815,
	{*A Smyrna S F Ankara*}	=	815,
	{*A Smyrna→Syria*}	=	815,
	{*A Constantinople H, F Ankara H, A Smyrna H*}	=	815)

position with *F Ankara→Black-Sea* is explored first, since it seems slightly better. The allowable subsequent orders are: *F Smyrna→Constantinople, A Smyrna→Ankara, A Smyrna→Syria, A Smyrna→Armenia.*

This yields Table 6 (again, asterisks mark the new elements).

So, finally we have several promising full candidate sets of orders, the best of which is *A Bulgaria→Constantinople, F Ankara→Black-Sea, A Smyrna→Armenia.* This is in fact one of the popular openings for Turkey.[10] However, the numbers in Table 2 were not rigged to make the orders come out right for this one case. Our system should derive decent orders for any position. Since this is at the top of the agenda, it then looks at the enemy units that could have some effect on the outcome: *F Sevastopol, A Moscow, F Trieste, A Budapest, F Naples.* It tries the possible orders for these enemy units. It evaluates the possible final positions, and gets some points knocked off for the possible conflict in the *Black-Sea* with the Russian fleet as well as the advances of Austria and Italy. Since this is the only set of orders thus far that have been tested against enemy orders, it is considered the current best set of submittable orders.

If it has extra time for processing, then it would leave depth-first mode and try the next item on the agenda, namely {*A Constantinople→Bulgaria,F Ankara→Black-Sea,A Smyrna H*}=1014. This is that special element that got put on the agenda when {*A Constantinople→Bulgaria,F Ankara→Black-Sea*}=1014 was exploded into its successors. It is tested against enemy orders, and it gets even more points knocked off for the possible Russian advance. Since this is not better than the current best set of submittable orders, it is discarded.

[10] "This opening poses all sorts of problems for Russia. If she trustingly ordered *F Sevastopol→Rumania*,she is in real trouble. Even if she ordered *F Sevastopol→Black-Sea*, she is going to have difficulty maintaining her position." [18]

Table 6: Agenda $t = 3$

(*	{A Constantinople→Bulgaria,F Ankara→Black-Sea,A Smyrna→Armenia}	=	1034,
*	{A Constantinople→Bulgaria,F Ankara→Black-Sea,A Smyrna H}	=	1014
*	{A Constantinople→Bulgaria,F Ankara→Black-Sea,A Smyrna→Syria}	=	1014,
	{A Constantinople→Bulgaria,F Ankara→Constantinople}	=	1004,
*	{A Constantinople→Bulgaria,F Ankara→Black-Sea,A Smyrna→Constantinople}	=	1004,
*	{A Constantinople→Bulgaria,F Ankara→Black-Sea,A Smyrna→Ankara}	=	989,
	{A Constantinople→Bulgaria,A Smyrna→Armenia}	=	979,
	{A Constantinople→Bulgaria,A Smyrna→Ankara}	=	959,
	{A Constantinople→Bulgaria,A Smyrna S F Ankara}	=	959,
	{A Constantinople→Bulgaria,A Smyrna→Syria}	=	959,
	{A Constantinople→Bulgaria,F Ankara H, A Smyrna H}	=	959,
	{A Constantinople→Bulgaria,F Ankara→Armenia}	=	949,
	{A Constantinople→Bulgaria,F Smyrna→Constantinople}	=	939,
	{F Ankara→Black-Sea}	=	870,
	{A Smyrna→Armenia}	=	845,
	{F Ankara→Armenia}	=	845,
	{A Constantinople→Smyrna}	=	815,
	{F Ankara→Constantinople}	=	815,
	{A Constantinople→Ankara}	=	815,
	{F Ankara→Constantinople}	=	815,
	{A Smyrna→Constantinople}	=	815,
	{A Smyrna→Ankara}	=	815,
	{F Armenia S A Constantinople}	=	815,
	{F Armenia S A Constantinople}	=	815,
	{A Constantinople S F Ankara}	=	815,
	{A Constantinople S A Smyrna}	=	815,
	{A Smyrna S A Constantinople}	=	815,
	{A Smyrna S F Ankara}	=	815,
	{A Smyrna→Syria}	=	815,
	{A Constantinople H, F Ankara H, A Smyrna H}	=	815)

If the deadline has still not yet arrived, the diplomat would examine the position {*A Constantinople→Bulgaria,F Ankara→Black-Sea, F Smyrna→Syria*}=1014 against enemy orders, and presumably discard it. Next, it would come to {*A Constantinople→Bulgaria, F Ankara→Constantinople*}, and this represents a jump to a different part of the search tree. It would look at all the positions generated by the orders for *A Smyrna* in this context. And so on until it ran out of time, at which point the best set of moves that have been tested against enemy moves would already have been submitted.

4.3 Advanced Models

4.3.1 Massive Support

Consider the following problem. Let us assume there is one space much more important than all of the others for Turkey at the moment. Perhaps he is in an end-game position where he has captured all of the Mediterranean, but now his navy is bottled up at Gibraltar. The *Mid-Atlantic-Ocean* now becomes of critical importance to him. Assume England is defending with a fleet doubly supported (*F Mid-Atlantic-Ocean H, F Irish-Sea S F MAid-Atlantic-Ocean, F Gasconoy S F Mid-Atlantic-Ocean*), and that Turkey has four fleets available to attack (*F North-Africa, F Western-Mediterranean, F Spain(sc), F Portugal*).

The evaluation routine described above will give a low score to any attack of the Mid-Atlantic Ocean consisting with two or less supports, since such attacks would fail. Only the triply supported attack is given a high score. Since the search routine only adjusts the move for one unit at a time. Thus, before finding the solution, the diplomat must three times consider orders increasingly low on the agenda.

In order to avoid this problem, we rewrite the agenda creation routine so that whenever an order is placed on the agenda, all of its possible combinations of support are placed there as well.

Obviously the same objections hold for long convoys, so all possible convoys should be immediately placed on the agenda as well.

4.3.2 Limited Raiding

The system above is designed to avoid raiders at all costs. A position containing such a unit which exerts its control over much valuable territory is given an extremely low value. Similarly, by forcing one unit behind enemy lines, our program thinks that it now controls the majority of the enemies territories with that one unit.

However, this is not very accurate. A single raider presented with a variety of supply centers can still only occupy one each year. Thus, the control exerted by the raider must be diluted by the number of supply centers that it controls. Hence, this evaluation function gives a single enemy unit access to seven supply centers if one could by so doing capture two or three supply centers ourselves.

Also, we let a player's owned home supply centers exert control, since units could be built there (but probably the distance should be a little greater, since the units do not exist yet and might never). Furthermore, any owned supply center should give a bonus to the owning player's position evaluation, regardless of the status of "control." A further refinement is to have units exert partial control in any adjacent territories, even those occupied by enemy units.

4.3.3 Advancing Towards the Front

Now comes the most problematic aspect of position evaluation. An example will illustrate the problem best. Suppose Turkey has taken over half the board and builds an *A Smyrna*. 'How do we make the position evaluation encourage the movement of the army towards the "front"? The above notions of "control" would not make the army move towards the front. All of the spaces in the vicinity of Smyrna are already 100% under the control of Turkey, so no move by the army could possibly increase the value of the position which is based solely on control of territory.

However, this is not the right question to ask. Although most of the intelligence should be put into the position evaluator, some intelligence can be put into the order generation and agenda sorting as well. For example, a heuristic like "always prefer a support order in preference to a hold order" can more easily be put into the agenda sorting (or move generation) routine than the position evaluation routine (which is supposed to be "static" and just look at the resulting board position, not the orders themselves *per se*). So, we can add a heuristic to make it prefer orders that move in the "right direction" over all other orders, everything else being equal (i.e., the position evaluation being the same).

One could adopt a gravitational attraction analogy, where the unit would seek to reduce the distance between itself and uncontrolled (or partially controlled) territories, weighting valuable territories proportionally more and distant territories much less. The idea is to have the unit prefer to move to where it can do the most good in the least time. For example, this will ensure that Turkey does not order *A Smyrna→Syria*, because that would move the army away from valuable partially controlled territories; even if its position evaluation were the same as the other orders, the best-first search would put *A Smyrna→Syria* "behind" the other orders.

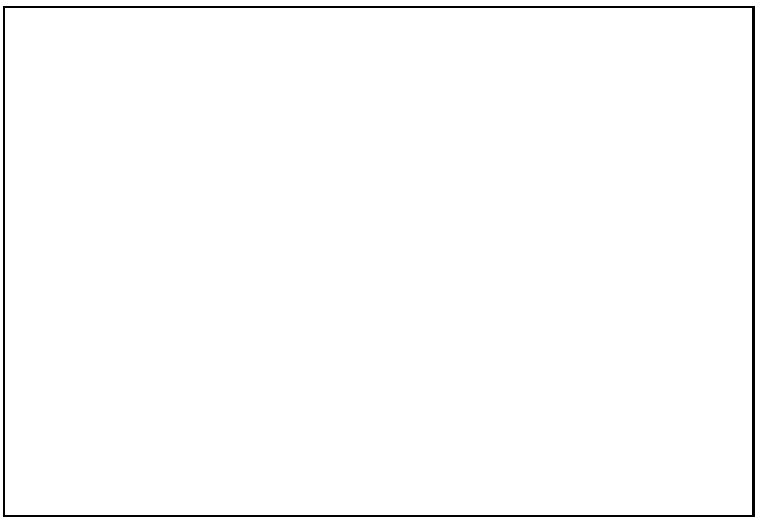
This preference to have units near where they could do the most good *could* instead be incorporated directly into the position evaluation, but then it has to be set up carefully to always be a less important factor than control.

4.3.4 Meta-strategy

Another way to emphasise the regions of the board which are the most important—the front—is to view the game at more abstract levels, with provinces grouped together into meta-provinces, and perhaps meta-provinces grouped together into still larger regions, and so on. (See Table 7) Thus, if the system for some reason decided that it was important to go for Scandinavia, then it could raise the "altitude" of that whole region as well the territories leading up to it. A strategy of a diplomat might be to dominate one portion of this hierarchy before moving on to the next.

However, how would we determine which meta-provinces to concentrate on at any given moment? We should translate the current position to an equivalent position on the meta-map. Then we can recursively invoke our diplomat on this meta-position. In the meta-game, it is much easier to find moves because the number of units and spaces is correspondingly smaller. The meta-moves that we find could then be used to determine the short-term goals of our program.

Table 7: Meta-Map



4.3.5 Large Searches

The above problem-reduction technique is useful for coming up with moves involving many units. Given a constant time limitation, countries with more units must do a more superficial search of their move tree due to the resulting increase in the number of combinations possible. Using a beam search to save time results in the consideration of an agenda in which all of the set of orders being considered are essentially the same. This is good if we are trying to fine-tune a strategy, but not if we are interested in examining fundamentally different tactics: for example, the choice between a static defense and a "the best defense is a good offense" approach.

If the fundamental strategy has already been solved on a meta-game level, then the beam search described above is satisfactory for determining the desired moves to carry out that strategy. However, we can do better than that. If the meta-strategy indicates that we should attempt to capture a certain two meta-provinces which are not adjacent (for example, Turkey pushing westward with his fleets while his armies are busy fighting in the Moscow area), then the real set of moves on each front can be determined independantly in most cases. This technique of "divide-and-conquer" results in a real improvement in speed for large countries fighting on multiple fronts.

Obviously, this is not the only solution. Another possibility is to counter the increase in units with a corresponding increase in the width of the search (to the obvious deteriment of the search depth).

Finally, another possibility is to add a random element to the search path in order to prevent all the search paths from heading the same way. For example, start the search with a variety of seeds (see section 3.4) chosen randomly perhaps. At each step in the search, instead of chosing the highest item on the agenda, chose agenda items on the basis of a decaying probability distribution. The final decision of what set orders to submit

will be made on a basis of the value of the resulting positions. However, the order in which we look at positions will include a slight random factor.

4.3.6 Predictability

This also solves our next problem: "Predictability."

Until now, the behavior of our diplomat was completely predictable. However, sometimes the best move is not well defined. For example, suppose Italy with his *F Tyrrhenian Sea* is trying to counter Turkey's incursion into the *Ionian-Sea*. Italy must guess whether Turkey is going to attack *Naples* or *Tunis*. If Turkey is completely predictable, then Italy will be able to counter the advance. However, if Turkey recognizes the situation and makes a random choice, then he has a 50% chance of gaining a supply center.

In theory, such *guessing games* should be recognized and probabilities chosen according to well-established principles of game theory. However, in practice, the best we hope to do is to add a small random factor to the evaluation of each position in order to avoid this problem.

4.3.7 Province Values

There are a number of refinements we can make to the mountain model of control merely by adjusting the values of regions. For example, territories with supply centers should be more valuable, especially in the Fall when supply center ownership is established. Similarly, territories that border many supply centers should be more valuable in the Spring.

We must also give a value to certain *combinations* of spaces being controlled. The existance of stalemate lines means that a position is more the sum of its parts. If England is holding (*F Portugal, F Mid-Atlantic-Ocean, F English-Channel, F Irish-Sea*) against an alliance of southern powers that have overrun the continent, he can still hold his ground against further advances. [18] England can thus secure his place in a draw, or divide the alliance against him. Hence, England's position is much more powerful than a simple numerical count of his supply centers might indicate. Our program must be able to detect all known stalemate lines, and be able to form them itself.

In the final analysis, machine learning techniques are necessary to lift the tactics of computer Diplomacy players up to a level competitive with even novice human players. For example, the base values of territories could be guessed at by humans and then optimized by machine learning techniques. The computer could play games with various base values and experiment in an attempt to derive better values.

5 Diplomacy

In the example given in the above section, the search wound up with an anti-Russian opening. However, Turkey's neighbors have had no say in the matter. In a real Diplomacy game, Turkey tries to negotiate with all his neighbors. If Russia sounds trustworthy and presented an intelligent plan for attacking Austria, then the

anti-Russian opening might be abandoned in favor of an anti-Austrian one. Similarly, if Austria declares war on Turkey, Turkey would be unwise to risk a two-front war by moving against Russia.

That is to say, so far we have outlined a recipe with which to create a fairly intelligent tactician; however, we have not yet mentioned *diplomacy* which after all is the name of the game. But how can seven diplomats programmed by different people carry out any sort of meaningful dialog? To solve this problem, one of the authors has developed for the Diplomacy Programming Project a program called the *Diplomacy Interface*. Although written in LCS,[11] it is capable of launching programs written in any language compatable with the UNIX operating system. The standard input of each diplomacy is rerouted via the use of an LCS instream to the LCS agent responsible for the communication with that player-diplomat. The agent then waits for a rendezvous with the main program. If the main program has any output to send to the player, it spawns a new process which waits for a rendezvous with that player's communication agent. This agent then can put the output in the corresponding LCS outstream where it is now available on a FIFO basis as input for the diplomat.

To simulate the play of humans against machines, the Diplomacy Interface is capable of opening *X* or *suntools* windows for specified countries. Input and output for these windows are handled in a manner similar to that described above. Actually, it is impossible for a diplomat to determine who is playing the other countries: a human or a machine. Thus, no prejudices against humans can be built into the diplomats, and humans must perform a sort of Turing test if they wish to distinguish the machines from other humans.

All communications are handled by the Diplomacy Interface; however, this interface can then in turn forward messages when desired. The evaluation routines used to compute the strategy of each diplomat can then take into account this communication. Most of the above discussion of position evaluation still applies, except that the holdings of an alliance should be considered together in the evaluation of "altitude controlled," plus some catches need to be put in to prevent accidental stabs. The diplomat should probably work out two sets of orders each turn—one in the case that it remains an ally and one in the case that it stabs.[12]

Similarly, the system should work out two sets of the opponent's orders (one in which he stays allied, one in which he stabs), and evaluate the outcomes using the techniques of game theory. This allows the program to work with an ally while protecting itself.

If the predicted profits for stabbing exceed its estimated benefits for cooperating, then it stabs. Often the table of payoffs forms a version of the *Prisoner's Dilemna*. In that case, we must determine if this stabbing opportunity is likely to repeat itself. In a one-time prisoner's dilemma situation, there is no reason not to stab. However, if such situations are anticipated to arise again many times among the several players, then you should respond with *tit for tat*.[13] The exact frequency and number of players necessary depends on the size of the short-term gains to be won with the stab. In a game with the following payoff matrix:

[11] LCS is a variant of SML developped by Bernard Berthomeiu and the *Outil et Logiciel pour le Communication* group at the *Laboratoire d'Automatique et de l'Analyse de Système* in Toulouse. LCS differs from SML in that it allows the management of parallel processing.

[12] We are ignoring the possibility of *queasling*: that is to say, fulfilling your obligations in a minimal way or undermining the treaty in minor ways in order to maximize your self interests without necessarily entering war with your former ally. Presumably, a weak ally will have no choice but to allow a certain amount of *queasling* to go unpunished. But it is difficult to predict at what point your ally will declare your moves to be a "stab." Our simple model assumes two choice: either complete loyalty, or total war on all fronts.

[13] Given its simplicity, tit for tat is a suprisingly effective strategy for repeated prisoner's dilemma against multiple opponents. Tit for tat begins with cooperative behavior in the first instance. In ensuing rounds, tit for tat adopts the strategy employed last round by the opponent.

	Stab	Cooperate
Stab	0\0	−2\5
Cooperate	−2\5	3\3

corresponding to a specific prisoner's dilemma, the "best" strategy in a certain sense is to be cooperative until the other player stabs, and then after that to always stab. However, the last three opportunities to stab should be taken regardless of what the opponent does.[14]

To extend the strategy to the game Diplomacy, we will instruct our program to consider its worse enemy to be the last player to have stabbed it. However, stabs of allies will be anticipated when so doing is profitable and there are unlikely to be similar future stab possibilities. For example, if the game can be won with a stab, then our diplomat will always do so. However, it will not stab an otherwise trustworthy ally for a single supply center especially if that supply center could be taken in a later stab if necessary.

A complete description of the communication language developed for the Diplomacy Interface is beyond the scope of this paper. However, below are a few examples to give the reader the feel for what is possible to express with this language. English translations are given below each.

FRM ITA (FRM AUS (DWR))
"Italy says Austria has declared war on him."

FRM RUS (DMZ (BLA ARM))
"Russia proposes that the *Black Sea* and *Armenia* be kept empty as a sort of Demilitarized Zone."

SUB (AMY CON MTO BUL)
"I submit the move *A Constantinople—Bulgaria*."

SND RUS (IFF ALY (VSS AUS) THN (YES (DMZ BLA ARM)) ELS (NOP (DMZ BLA ARM)))
"If you, Russia, agree to ally with me against Austria then I accept your proposal to demilitarize the *Black Sea* and *Armenia*, otherwise I reject it."

A diplomat gains some ideas about alliances via the negotiations; then it gets the results of one turn. How can it determine if the two are at odds? This is done in several steps.

First, all of the promises made and received are stored in a truth maintenance system capable of making simple deductions. For example, if Russia had sailed to the *Black-Sea* after having agreed to (DMZ (BLA)), this system would indicate that as a violation of their specific agreement to keep the *Black-Sea* demilitarized. The program would then nullify all agreement between the players (unless there was another independant reason for keeping them), and anticipate hostile actions in the future. Similarly, any conflict between Italian and Turkish units would contradict a general promise of peace such as (FROM TUR (PCE)).

Second, whether or not a player made any promises, it is a good idea to reward him for any help he may have given, and to punish him for any attacks.[14] Sometimes, a country may help with first promising to do so. Usually this is due to a fear of "leaks," but it could also be due to an inability by that player to generate understandable diplomatic messages. In either case, any country whose orders truly increased or decreased the value to Turkey of the resulting position should be raised or lowered on the ally/enemy scale in accordance.

[14] In fact, an attack without warning should be dealt with more harshly than an attack with adequate warning.

Finally, the moves for each country should be used as a seed for a short search of last turn's moves. If given the assumed combination of alliances, we find a much better set of moves that the player could have submitted but did not, then we must reject our assumptions and continue searching the possible combinations of alliances until we find out for which the moves actually made are the best moves possible (or fairly good in any case). For example, a diplomat might assume that Austria and Italy are at war due to a statement by Italy, but then their moves might not make sense given this assumption. After some search, the diplomat could potentially discover that the only reasonable interpretation of their moves is that Austria and Italy are allied against Turkey and Russia.

6 Conclusions

In summary, an efficient search coupled with a good position evaluation function allows a computer diplomat to devise decent tactics. While for humans, diplomacy is often more important than tactics, the first computer diplomats we creates will be better off with good tactics before good diplomacy, because it is much harder for programs (written perhaps by different people) to communicate with each other.[15] Despite having devised a protocol and written an interface for communication between diplomats, much work remains to be done.

However, by attempting to solve these difficult problems resulting from the interactions of intelligent agents, we hope to better understand what sort of capabilities such an agent should have, when it cannot assume that everyone in the world is its friend. Creating programs that can play complex multi-player games is a step towards creating programs that can deal with the complexities of the real world and the craftiness of humans.

References

[1] A. Calhammer, Rules to Diplomacy, The Avalon Hill Game Company, 1956, 1976. "Diplomacy" is a trademark of the Avalon Hill Game Company.

[2] The Avalon Hill Game Company, Diplomacy Conference Map, 1976.

[3] E. R. Berlekamp, J. H. Conway and R. K. Guy, "Winning Ways for your Mathematical Plays," Academic Press, 1982.

[4] Charnkiak, Riesbeck, McDermott and Meehan, "Artificial Intelligence Programming," (2nd edition 1987).

[5] Charnkiak and McDermott, "Introduction to Artificial Intelligence," (1985).

[6] N. V. Findler, G. L. Sicherman, and B. McCall, "A multi-strategy gaming environment," *Computer Game Playing: Theory and Practice*, M. A. Bramer editor, Ellis Horwood Limited, Chichester, England, 1983.

[15] Not to mention the problems of cooperating at strategic and tactical levels, not being too gullible, knowing when to stab, etc. However, for an opposing view on this question see [11, 12] in which a diplomat is described putting the emphasis on negation.

[7] Forsyth and Rada, "Machine Learning: Applications in Expert Systems and Information Retrieval," (1986).

[8] Feng-Hsiung Hsu, Thomas Anatharaman, Murray Campbell, Andreas Nowatzyk, *Un Ordinateur parmi les grands maîtres d'échecs*, Pour la Science, **156**, October 1990, pp. 88–97.

[9] T. Green, private communication.

[10] D. Hofstadter, MetaMagicalThemas.

[11] S. Kraus, D. Lehmann, E. Ephrati, *An Automated Diplomacy Player*, "Heuristic Programming in Artificial Intelligence," Eds. D N L Levy, D F Beal, Ellis Horwood (1989), 136–153.

[12] S. Kraus, E. Ephrati and D. Lehmann, *Evaluation of Suggestions during Automated Negotiations*, Proc of the 11th Cognitive Science Conference, 1989.

[13] Eric S. Klien ed., "Tokugawa," Electronic Protocol, R. L. Cochran Jr. and C. B. McKee game masters, 1989–90.

[14] D. Loeb, Prisoner's Dilemna, LAAS Internal Report.

[15] Elaine Rich, "Artificial Intelligence," (1983/1989).

[16] Warner Smith, Technology Review, July 1990, p. 45.

[17] C. Tzeng, "A Theory of Heuristic Information in Game-Tree Search," Springer-Verlag, New York, 1988.

[18] Rod Walker, "Player's Guide to Diplomacy," The Avalon Hill Game Company, Baltimore, 1979.