Management of Uncertainty in Combinatorial Game Theory

Tristan Cazenave

LAFORIA-IBP
Université Pierre et Marie Curie
4, place Jussieu
75252 PARIS CEDEX 05
Tel: 44 27 70 10
Fax: 44 27 70 00
cazenave@laforia.ibp.fr

Abstract

We describe a method to handle uncertainty in combinatorial game theory. This method has been implemented in Gogol, a general game playing system mainly applied to the game of Go. In complex games such as Go, a search intensive approach is intractable. Gogol uses a lot of knowledge and little search. This knowledge is represented using combinatorial game theory. We have adapted this theory to take into account the uncertainty due to the intractability of search in complex games. The system uses its evaluation of the game to choose a strategy. This strategy is applied by handling the uncertainty resulting from incomplete tree searches.

Key Words : Combinatorial game theory, Uncertainty, Risk, Game of Go.

1 Combinatorial Game Theory

This section is an introduction to Combinatorial Game Theory. It is a mathematical theory which has been developed by J. H. Conway [Conway 1976], and adapted to many games by Berlekamp, Conway and Guy [Berlekamp 1982].

Definition : A combinatorial game x = {G|D} is composed of two sets G and D of combinatorial games. Every combinatorial game is constructed this way.

A combinatorial game defines a tree of possible positions. Each position can be assigned two values. The first value will be either Won for Left (WL) or Lost for Left (LL). The second value will be either Won for Right (WR) or Lost for Right (LR). For each node, we can therefore have four possibilities :

	LL	WL
LR	The player who begins loses, the game is NUL (0)	The left player always win, the game is POSITIVE (+)
WR	The right player always win, the game is NEGATIVE (-)	The player who begins wins, the game is FUZZY (||)

This theory is very efficient in simple games such as the game of Nim. It has been applied with success to the endgame of Go [Wolfe 1991]. But this theory requires the subgames of a same position to be independant, it also requires to be able to forecast all the possible moves until the end of the game. This assumptions do not apply in complex games such as Go and Chess. We show in this article that the introduction and the management of uncertainty in Combinatorial Game Theory permits to use it in complex games.

2 Introducing Uncertainty in Combinatorial Game Theory

In this section, we describe the introduction of uncertainty in Combinatorial Game Theory. It is necessary in complex games to represent what is unknown because it is impossible in such games to forecast all the positions, only a small subset of the positions can be forecasted. Representing uncertainty in Combinatorial Game Theory permits to extend this theory to complex games, and to represent the partial deduction a system can make about a position. It also allows to take strategical decisions on the way to handle what is not predictable in a reasonable time.

Theoretically, a game has only two possible values : Won (G) or Lost (P). However, in practice it is often very costly and most of the time intractable to find the value of a game [Allis 1994]. Therefore, we introduce the Unknown (I) value which accounts for the fact that we do not have enough time to compute the value of a particular game. Let J = {G,I,P} be the set of all possibles states for a game.
The dynamic definition of a game consists in defining the states of the game after the best friend move and after the best opponent move. If $x,y \in J$: xy defines the game for which the best state the friend player can achieve if he plays first is x,

whereas the best state the opponent player can achieve if he plays first is y. We define J_1 = {xy / x,y $\in$ J} and J_n = {xy / x,y $\in$ J_{n-1}}. Computing an element of the set J_n is twice the cost of computing an element of the set J_{n-1}, therefore we limit ourselves to the elements of J_2.
In practice J_1 et J_2 are sufficient to represent the kind of knowledge used in games, the most common games used by players are :

- GG represents a Won game.
- GP represents an unsettled game.
- PP represents a lost game.
- PG represents a Zugswang in Chess or a Seki in Go.
- IIIP represents a threat for the opponent player.
- GIII represents a threat for the friend player.

3 Evaluation of the moves and of the states

A game is associated with a goal to achieve. In a Go position, there are between 500 and 10000 subgoals which can be achieved. Each move is given a value according to the subgoals it achieves. Therefore, we give values to the various states of J. We define :

Val (G) = 1, Val (P) = -1.

To evaluate the global position, we multiply the value of each combinatorial game by the importance of the corresponding subgoal. The evaluation is the sum of the results for all the subgoals.

Using the values of the states of J, we can define the values associated with the friend moves for elements in sets J_1 and J_2 :

$\forall$ xy $\in$ J_1 : Val (xy) = Val (x) - Val (y).

Val (xy) is the value of the move which achieves state x. It means that playing a move in a game of J_1 achieves state x and prevents the opponent player to achieve state y.

$\forall$ $xyx_1y_1 \in J_2$:

$$Val_c (xyx_1y_1) = (Val (x) + Val (y))/2 - (Val (x_1) + Val (y_1))/2$$

$Val_c (xyx_1y_1)$ is the value of the move which achieves state xy. This value takes into account the fact that if we choose the xy state, we will approximatively have the same chance to be in state x as in state y. This approximation is due to the great number of subgames in a position, which also implies that a move plays many subgames at the same time and that for these subgames, the moves which enable state x or state y are not the same in the different subgames even if the move leading to state xy is the same. We will win, in average, half the total number of subgames in which a move plays. This approximation holds for games such as Go or Chess, since complex games inherently have a lot of dependent subgames.

For simpler games in which subgames are independent, we can define another evaluation function for moves :

$\forall\ xyx_1y_1 \in J_2$:

$$Val_s\ (xyx_1y_1) = \min\ (Val\ (x), Val\ (y)) - \max\ (Val(x_1), Val\ (y_1))$$

Val_s is the value of the move achieving state xy in a simple game where interactions between subgames can be neglected. If a move only plays in one subgame, its value is the difference between the best state achieved by playing it and the best state achieved by not playing it.

4 Management of Uncertainty

We have defined values for Won (G) and for Lost (L) but we have not defined a value for the state Unknown (I). This value depends on the chosen strategy. We define three main strategies :

1) The risky strategy is a strategy which aims at destroying the positions of the opponent without caring for friends positions. We use it when we are losing the game. We try to destabilize the overall position, therefore maximizing uncertainty :
Val (I) = 1.

2) The safe strategy is a pessimist strategy which aims at saving current positions without caring for the opponents positions. We use it when we are winning by a large margin. We try to stabilize the overall position, and minimize uncertainty :
Val (I) = -1.

3) The average strategy is a neutral strategy which aims both at saving friend positions and at attacking ennemy positions. It is used when the game is even. We give to uncertainty a neutral value : Val (I) = 0.

In the Table 1, we give the values associated with the friend moves in the games of J_1 for the three strategies :

Games\Strategies	Risky	Average	Safe
GG	0	0	0
GI	0	1	2
GP	2	2	2
IG	0	-1	-2
II	0	0	0
IP	2	1	0
PG	-2	-2	-2
PI	0	-1	-2
PP	0	0	0

Table 1

We give in Table 2 the values associated with the moves of some games in the set J_2 for the three strategies in a complex game, using Val_c :

Games\Strategies	Risky	Average	Safe
GGGG	0	0	0
GGGI	0	1/2	1
GGII	0	1	2
GIGI	0	0	0
GIII	0	1/2	1
IIII	0	0	0
IIIP	1	1/2	0
IPIP	0	0	0
IIPP	2	1	0
IPPP	1	1/2	0
PPPP	0	0	0

Table 2

We can use a smoother value of risk if we let the Unknown parameter take real values between -1 and +1. We obtain a continuous model of risk as shown in Figure 1 :

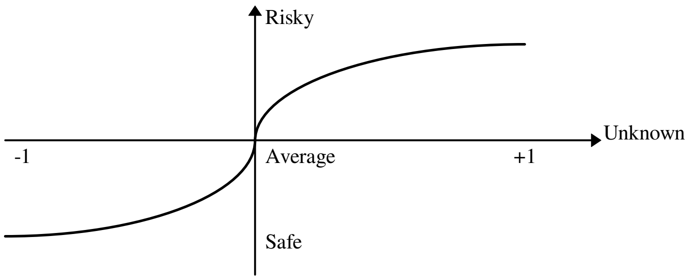

Figure 1

5 A taxonomy of Uncertain Combinatorial Games

We can define a taxonomy on the games we have defined. It is defined by the "is a" relationship. The Unknown (I) combinatorial game is more general than the Won (G) and Lost (P) combinatorial games as well as more general than all the other combinatorial games. Hence, we have GG is a GI. Furthermore, we can construct an inheritance relationship between games of J_1 and J_2. An example of inheritance is : GIPP is an IP. Figure 2 illustrates a partial taxonomy constructed in such a way :

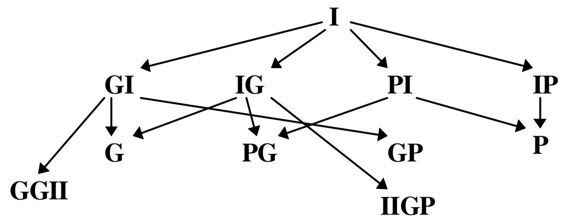

Figure 2

There is a value associated to each element in the taxonomy, depending on the chosen strategy. This value can be used to assess whether continuing to calculate a game of J_1 to generate a game of J_2 is important. In some cases, it can be interesting to stop computing a game as we know it won't change the choice of the global move to play. For example, we choose to prune searching when the two following conditions are observed. On one hand the set of subgoals a move achieves is of low importance and on the other hand there is not enough possible change in the values of the corresponding subgames to make it an interesting move.

To clarify the way these savings are done, let's take the case of the game GG. Its children in the taxonomy of games are : IGGI, IGGG, IGGP, GGGI, GGGG, GGGP, PGGI, PGGG et PGGP.

These games have values between -1 and 1 when choosing the average strategy. If the numerical importance of achieving a goal is N, calculating the game deeper than GG will make the result vary between -N and +N. The interval [-N,+N] is the set of possible values for the game. The important thing to notice is that friends and opponents top moves are the same in a game and in its children.

Moreover, each move may appear in many games. Hence, we define the interval of a move as the sum of intervals of the games which contain that move. This move's interval is used to prune the search in the following manner : let MaxiMin be the maximum, over all the move's intervals, of the minimal value of each interval. We abandon the computations involved in developping subtrees for moves whose maximal value does not exceed MaxiMin. This saves a lot of time of search as we do not have to match the rules concluding on the combinatorial games more specific than the already deduced combinatorial games.

6 Example

We are giving an example of application of management of uncertainty in combinatorial game theory taken from the system Gogol [Cazenave 1995] which uses combinatorial game theory to learn to achieve goals in the game of Go. The game of Go consists in putting alternatively black and white stones on the intersections of a 19*19 grid. A simpler version of the game allows to play on a 9*9 grid, for the sake of simplicity this is the one we have chosen to illustrate our example. Each intersection has a value in the set Color = {Black, White, Empty}.

The goal of the game of Go is to put more stones than the opponent on the board. An interesting subgoal is to replace an ennemy stone by an empty intersection. This subgoal will be noted "Take". The state is G for the goal Take when an intersection which was not Empty before a move becomes Empty after. Figure 3 gives an example of a black block[1] that can be taken playing White in A. It is also possible to define conditions in which the state is P for the goal Take. These conditions are known by Go players to be the two eyes rule : a block containing two eyes cannot be taken and is said to be alive. Figure 4 shows such a black block. The opposite subgoal of Take is Live, this signifies that Live is in state P when Take is in state G and vice-versa. Take will be used for enemy stones and Live will be used for friend stones. Another subgoal frequently used by Go players is Connect for which the G state is achieved when two stones are in the same block. The opposite subgoal of Connect is Disconnect. Connect is used for friend stones and Disconnect is used for enemy stones. In Figure 5, a white move at A connects the two white stones.

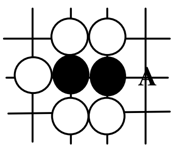

Figure 3

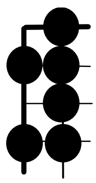

Figure 4

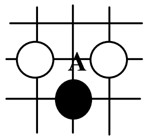

Figure 5

In practice, the conditions fullfilling the G state very rarely apply in the same case as the conditions for the P state. In most of the case, we do not have certain results on all the states in a game.

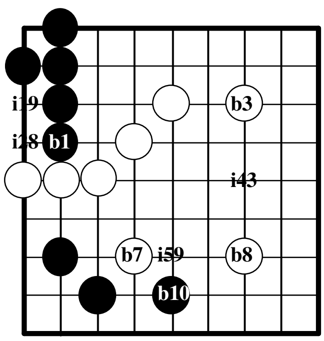

Figure 6

[1] A block is a set of stones in which each stone of the set has another stone of the set on one of the four adjacent intersections (vertically or horizontally adjacent intersections). The five black stones in the upper left corner of the example form a block.

Gogol uses first order logic to represent its knowledge. The uncertain combinatorial games deduced from a position are expressed in form of facts. The predicate Move ($Color_i$ $Goal_i$ $Move_i$ $Game_i$) is the most used to represent uncertain combinatorial games. A goal is composed of its name and of the element on which it applies. The goal Live applies on a block, the goals Connect and Disconnect can applies on two blocks, on one block and one intersection or on two intersections. Constants representing blocks begin with a 'b' and constants representing intersections begin with an 'i'.

Given the example in Figure 6, Gogol deduces the following predicates and calculates the corresponding importances :

Facts deduced on combinatorial games	Importance
Move (Black Live b1 i28 GI)	23
Move (Black Connect b1 i19 i28 GI)	1
Move (Black Connect b1 i28 i28 GI)	1
Move (Black Disconnect b3 b8 i43 IP)	14
Move (Black Connect i43 i42 i43 GIII)	1
Move (Black Connect i43 i34 i43 GIII)	1
Move (Black Connect i43 i52 i43 GIII)	1
Move (Black Connect i43 i44 i43 GIII)	1
Move (Black Disconnect b7 b8 i59 IP)	6
Move (Black Connect b10 i59 i59 GI)	1
Move (Black Connect i59 i60 i59 GIII)	1
Move (Black Connect i59 i50 i59 GIII)	1

Table 3

Choosing a move implies valuating each possible move. It is done using the following formula :

$$\text{Value (Move)} = \sum_{Move_i=Move} (\text{Importance}_i * \text{Val}_c\,(\text{Game}_i))$$

For example, using the average strategy :

Value (i43) = 14*1 + 1*½ + 1*½ + 1*½ + 1*½ = 16

Table 4 shows the values of possible moves under the three strategies :

Move\Strategy	Risky	Average	Safe
Black in i28	0	**25**	**50**
Black in i43	**28**	16	4
Black in i59	12	8	4

Table 4

Gogol evaluates that Black is losing by a large margin, he chooses a risky strategy, according to this strategy he chooses the Black move in i43. If the game were even or if Black were winning by a large margin, Gogol would have chosen in both cases the Black move in i28.

7 Conclusion

We have shown how to combine uncertainty and combinatorial game theory. This method has been implemented in Gogol, a system which plays the game of Go. The system takes into account its evaluation of the position in order to apply a strategy of management of uncertainty on the choice of its moves. It differs from previous studies in the management of uncertainty in the domain of games [Berliner 1979] [Horacek 1989] [Junghanns 1995] for it handles uncertainty after some incomplete And/Or tree searches have been achieved, whereas previous studies use numerical evaluations of the position to choose the nodes to expand in a tree. Furthermore it allows a knowledge oriented approach unlike the usual approach based on intensive tree search and little knowledge. The knowledge oriented approach is necessary in very complex games like the game of Go, for the high number of possible positions (10^{172}), the high number of possible moves in a position ($\approx$200) and the difficulty to evaluate quickly a position make the search intensive approach impracticable.

Gogol participates to the Internet Computer Go Ladder [Pettersen 1994] and it is ranked 5th out of 11 programs on 9x9 boards.

We are presently using our uncertain combinatorial game theory in other complex games. We use it for a general game learning framework.

References

[Allis 1994] - L. V. Allis. *Searching for Solutions in Games and Artificial Intelligence*. PhD. Thesis, Vrije Universitat Amsterdam, Maastricht 1994.

[Berlekamp 1982] - E. Berlekamp, J.H. Conway, R. K. Guy. *Winning Ways (for your mathematical plays)*. Academic Press, New York, 1982.

[Berlekamp 1991] - E. Berlekamp. *Introductory overview of mathematical Go endgames*. Proceedings of symposia in applied mathematics - 43 - 1991.

[Berliner 1979] - H. Berliner. *The B* Tree Search Algorithm : A Best-First Proof Procedure*. Artificial Intelligence 12, pp 23-40, North-Holland 1979.

[Bouzy 1995] - B. Bouzy. *Modélisation cognitive du joueur de Go*. PhD. Thesis, Université Paris 6, 1995.

[Cazenave 1995] - T. Cazenave. *Learning and Problem Solving in Gogol, a Go playing program,* IBP-LAFORIA Technical Report n° 95-10, 1995.

[Conway 1976] - J. Conway. *On Numbers and Games,* Academic Press, Londres/New-York, 1976.

[Horacek 1989] - H. Horacek. *Reasoning with uncertainty in computer chess* Advances in Computer Chess 5, pp 43-63, Elsevier, 1989.

[Junghanns 1995] - A. Junghanns, C. Posthoff, M. Schlosser.*Search with Fuzzy Numbers* International Joint Conference on Fuzzy Systems, pp 979-986, Yokohama, 1995.

[Müller 1995] - Martin Müller. *Computer Go as a Sum of Local Games : An Application of Combinatorial Game Theory*. PhD. thesis, Swiss Federal Institute of Technology Zürich, 1995.

[Pettersen 1994] - Eric Pettersen. *The Computer Go Ladder*. Page WWW : http://cgl.ucsf.edu/go/ladder.html, 1994.

[Wolfe 1991] - D. Wolfe, Mathematics of Go : Chilling Corridors, Dissertation, Université de Californie à Berkeley, Berkeley, 1991.