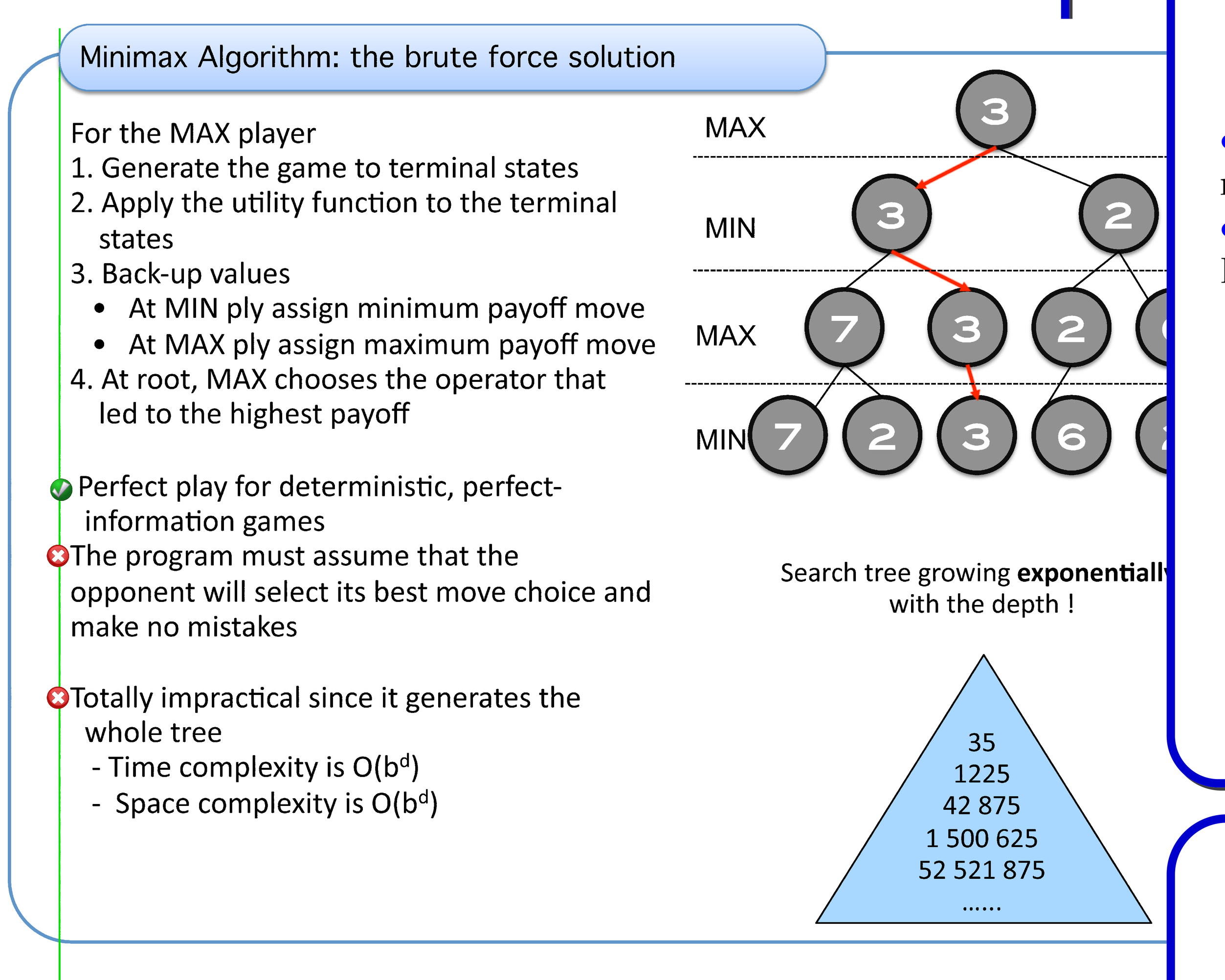
Monte-Carlo Tree Search

Olivier Teytaud, Hassen Doghmen
Tao, INRIA Saclay-IDF

ABSTRACT HERE.................

Multi-Armed Bandit Problem



Minimax algo

EXP3 Algorithm

Parameter : real $\gamma \in]0;1]$
Initialization : define the weight $w_i(t) = 1$ for $t = 1$ and all $i = 1, \dots k$
For each $t = 1, 2, \dots$

1. Set
$$p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{k} w_j(t)} + \frac{\gamma}{K}$$
for $i = 1, \dots K$.
2. Select randomly an arm i_t according the the probabilities $p_1(t), \dots, p_K(t)$
3. Observe the reward $r_{i_t}(t)$
4. Update the weight of i_t by
$$w_{i_t}(t+1) = w_{i_t}(t)\exp(\frac{\gamma}{K}\frac{r_{i_t}(t)}{p_{i_t}(t)})$$
and set $w_j(t+1) = w_j(t)$ for other arms.

• In order to find good actions EXP3 maintains a balance between
- **exploration**: weighting actions according to the reward: $1-\gamma$ term in the probability, and
- **exploration**: the uniform term $\frac{\gamma}{K}$ ensures that unsuffi-

External regret and Zero-Sum Matrix Games

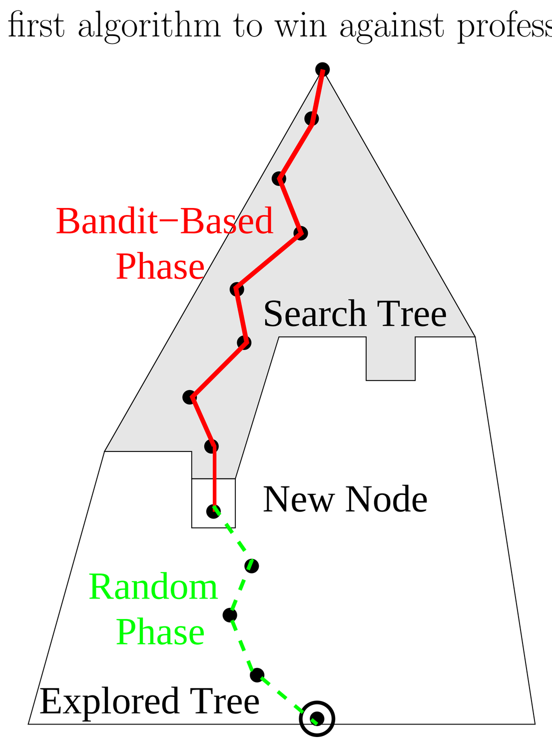
• Two players simultaneously choose a column i and a line j of a given matrix M
• The line player receives $M_{i,j}$ and the column player receives $-M_{i,j}$.
Exemple : the Rock-Paper-Scissors game

$$\begin{pmatrix} & \text{Rock} & \text{Cissors} & \text{Paper} \\ \text{Rock} & 0 & +1 & -1 \\ \text{Cissors} & -1 & 0 & +1 \\ \text{Paper} & +1 & 0 & -1 \end{pmatrix}$$

If both players repeatedly play, selecting their strategies with an algorithm minimizing external regret, then the empirical frequencies of play converge to optimal strategies ("Nash Equilibrium").

Monte-Carlo Tree Search Algorithms

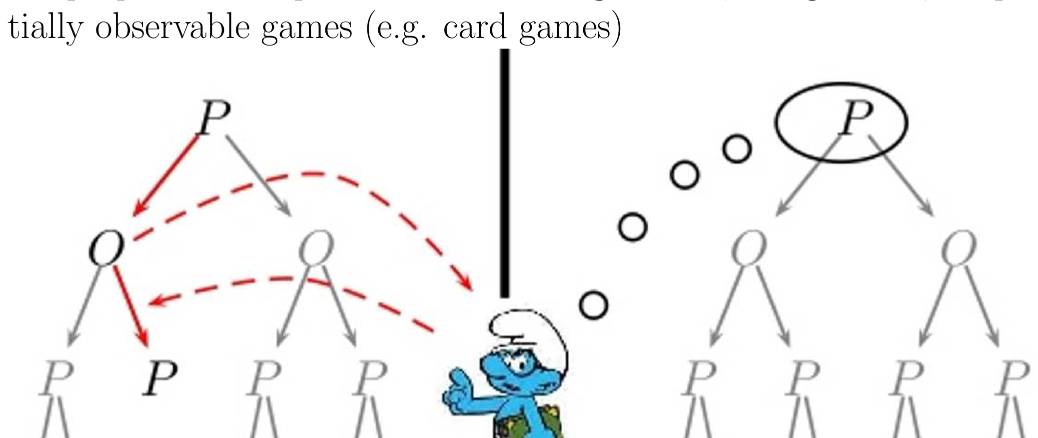
• MCTS Algorithms are efficient algorithms designed for tree-searching with huge inputs.
• They have been used to design computer players in games with full observation, e.g. Go
• Mogo was the first algorithm to win against professional Go players



• As in the multi-armed bandit problem we must balance *exploitation* and *exploration*
• classical implementations [KS06] use the UCB bandit algorithm [LR85], designed for the stochastic setting.

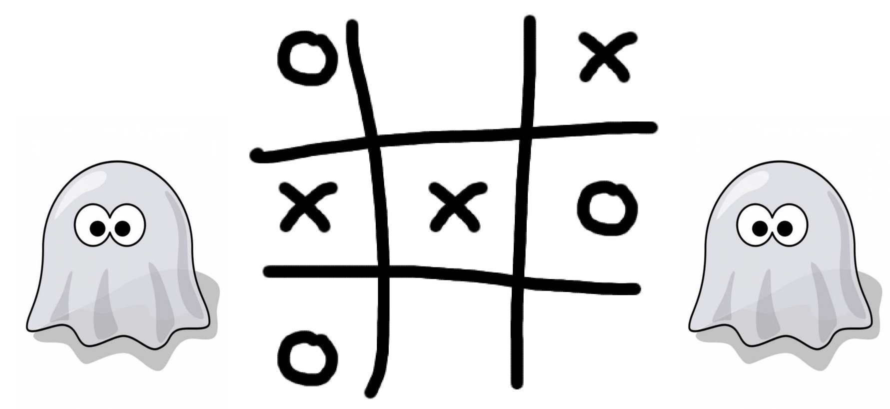
Multiple Tree Monte-Carlo Tree Search

We propose an adaptation of MCTS algorithms, using EXP3, for partially observable games (e.g. card games)

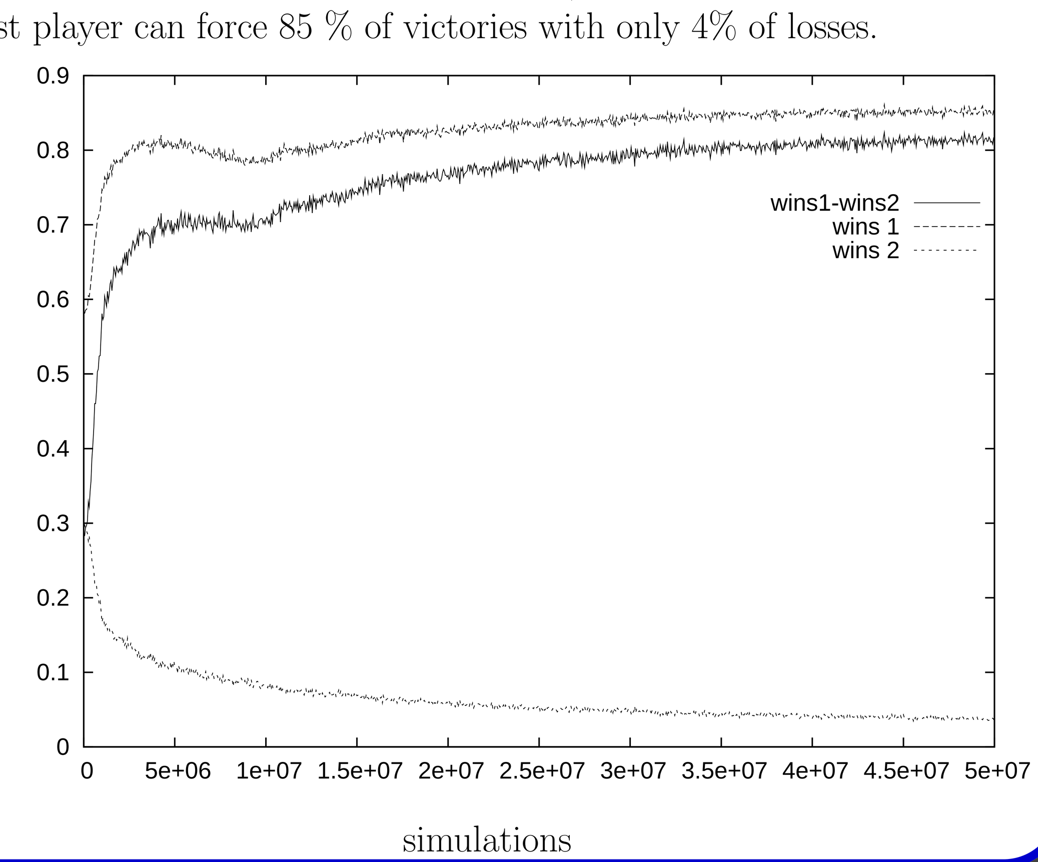


Application to the game of Phantom Tic-Tac-Toe

• Played like standard Tic-Tac-Toe but one does not see where the opponent plays.
• In case of an illegal move the player must play somewhere else



Whereas standard Tic-Tac-Toe is a draw, in Phantom Tic-Tac-Toe the first player can force 85 % of victories with only 4% of losses.



A Phantom Tic-Tac-Toe personal Olympiad

Opponents are :
• MMCTS 500K, 5M and 50M, mixed strategies obtained by our algorithm after 500K, 5M or 50M iterations
• Belief Sampler, who plays standard tic-tac-toe perfectly and randomizes his moves according to optimal strategies compatible with observations (standard approach for P.O. games, see e.g [Caz06])
• Random Player, a dummy uniform random player.

P1 \ P2	MMCTS 500K	MMCTS 5M	MMCTS 50M	Random	B.Sampler
M.500K	65% \ 25%	51% \ 37%	44% \ 47%	67% \ 22%	40% \ 43%
M. 5M	88% \ 06%	82% \ 10%	78% \ 17%	88% \ 05%	78% \ 10%
M. 50M	93% \ 02%	89% \ 03%	85% \ 04%	93% \ 02%	82% \ 03%
Random	55% \ 33%	48% \ 39%	41% \ 47%	59% \ 28%	30% \ 53%
B.Sampler	77% \ 14%	73% \ 18%	68% \ 22%	79% \ 12%	56% \ 28%

References

[ACBFS03] P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

[Caz06] T. Cazenave. A Phantom-Go program. *Advances in Computer Games*, pages 120–125, 2006.

[KS06] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. *Machine Learning: ECML 2006*, pages 282–