Agent-based Systems

Paolo Turrini

www.dcs.warwick.ac.uk/~pturrini p.turrini@warwick.ac.uk

Opponent Modelling

Aggressive Moves

Plan for today

We have seen extensive games and backwards induction. Now we look at situations in which this does not work.

- games are too big to be calculated
- players have weaknesses

We will look at how to play aggressively: maximising the expected reward by tricking the opponents into playing bad moves.

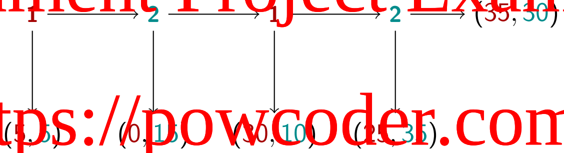
If you wanna know more:

Davide Grossi and Paolo Turrini
Short Sight in Extensive Games
International Conference on Autonomous Agents and Multiagent Systems, 2012

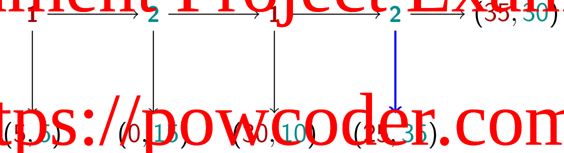
Paolo Turrini
Computing Rational Decisions in Extensive Games with Limited Foresight
AAAI Conference on Advances on Artificial Intelligence, 2015

At the beginning of the twentieth century, Zermelo proved a proposition which can be interpreted, "chess is a trivial game." (...) To see this we can use the backwards induction technique

Ariel Rubinstein
Modelling Bounded Rationality.
MIT Press, 1998

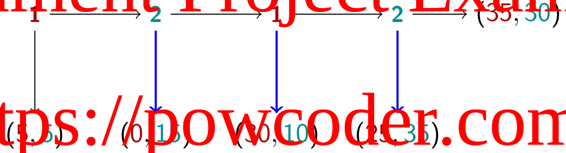

1
2
1
2
(35, 30)
(5, 5)
(0, 15)
(30, 10)
(25, 35)

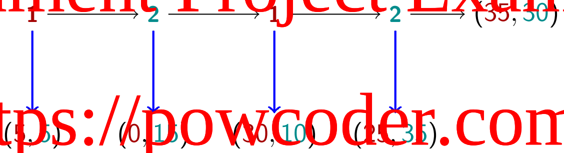
Rethinking backwards induction

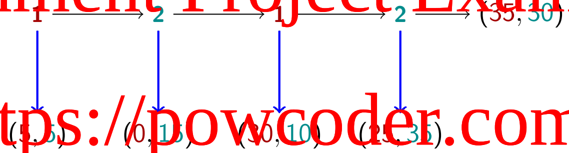


Rethinking backwards induction

$$\begin{tikzcd}
1 \arrow[r] \arrow[d] & 2 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 2 \arrow[r] \arrow[d] & (35, 30) \\
(5, 5) & (0, 15) & (30, 10) & (25, 35) &
\end{tikzcd}$$


1
2
1
2
(35, 30)
(5, 5)
(0, 15)
(30, 10)
(25, 35)


1
2
1
2
(35, 30)
(30, 10)



Backwards Induction implicitly assumes:

- that everyone is playing the same game
- that everyone is playing it rationally
- that all of the above is common knowledge

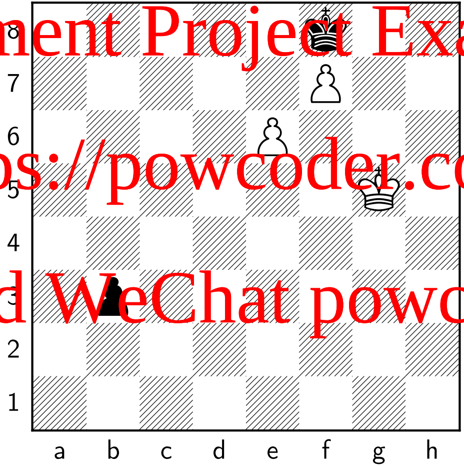
But [in games like chess] this calculation requires going through a huge number of steps, something no human being can accomplish. [1]

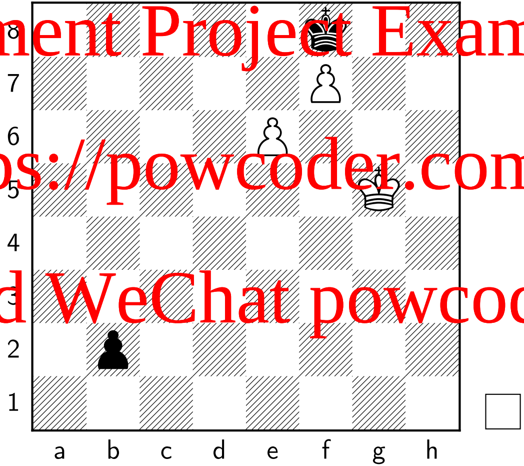
Modeling games with limited foresight remains a great challenge [and the frameworks studied thus far] fall short of capturing the spirit of limited-foresight reasoning.

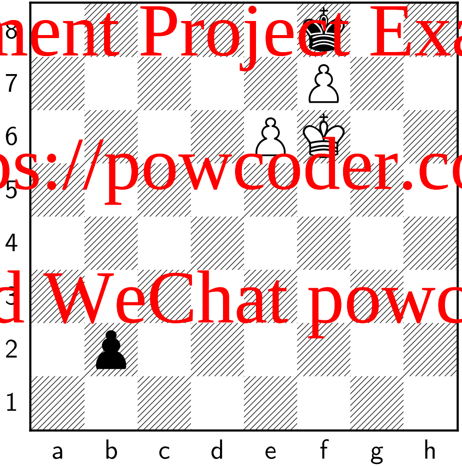
Ariel Rubinstein
Modelling Bounded Rationality.
MIT Press, 1998.

[1] Nor supercomputers for that matter.

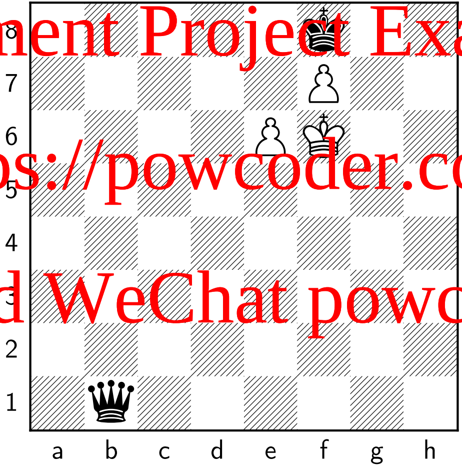
Calculating Variations

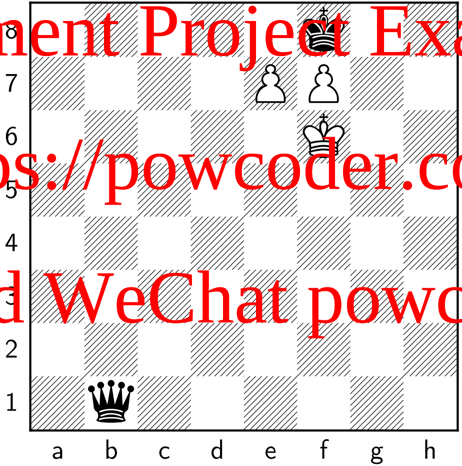



8
7
6
5
4
3
2
1
a
b
c
d
e
f
g
h


8
7
6
5
4
3
2
1
a b c d e f g h

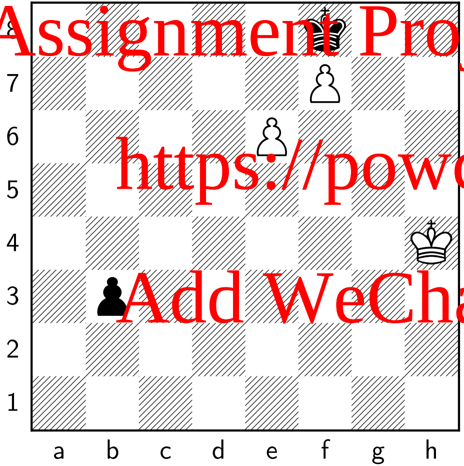
Calculating Variations

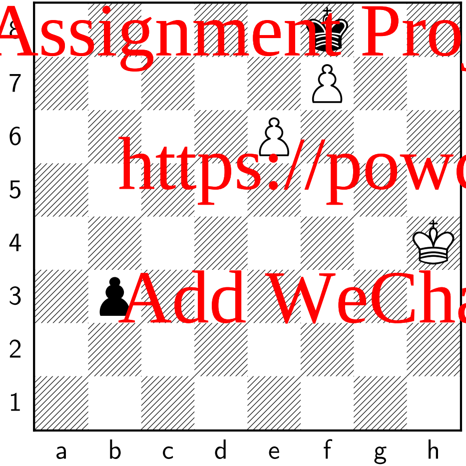



8
7
6
5
4
3
2
1
a b c d e f g h

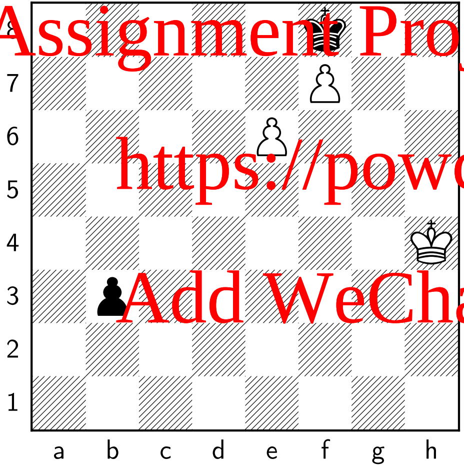
Short Sight

- Black loses for two reasons:
  - Partial view of the game
  - Wrong evaluation
- Chess-like games:
  - Limited foresight
  - Heuristic assessment of intermediate positions


8
7
6
5
4
3
2
1
a b c d e f g h



White is lost



Should he or she resign?

The idea

- White should not resign if (s)he believes that Black won't be able to solve the problem posed by White after Kg5!

- Kg5 is technically as good as any other move, but it is aggressive and might turn out to be winning.

Outline

1. The formal setup

2. An intuitive solution

3. Discussion

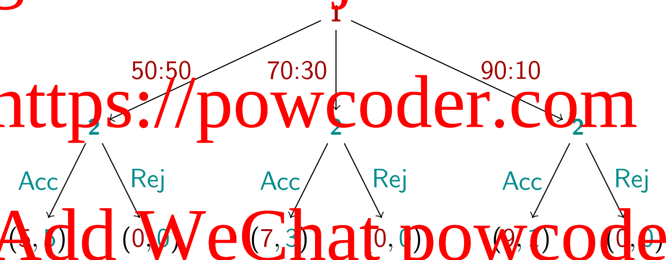
Extensive Games

Definition

An **extensive form game** is a tuple $(N, H, t, \Sigma_i, o, \succeq_i)$ where:

- N is a set of **players**
- H is a prefix-closed set of **histories** (sequences of actions);
- t is a **turn** function, assigning a player to each non terminal history;
- Σ_i is a set of **strategies** available to i i.e., functions that assign an action to each history h with $turn(h) = i$;
- o is an **outcome** function assigning to each strategy profile $\sigma \in \prod_{i \in N} \Sigma_i$ the induced terminal history;
- $\succeq_i$ is a total preorded on the set of terminal histories, i.e., the **preference** of i;

An extensive game



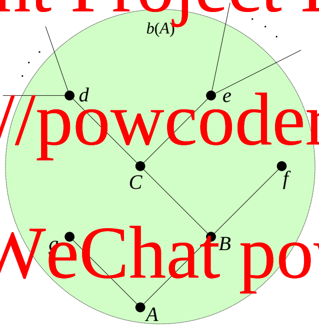
Sight function

Definition (Sight function)

For an extensive game G a **(short) sight function** is a function s associating to each non-terminal history h a finite, non-empty and prefix-closed set of histories in H extending h.

- The idea: at each decision point h, $s(h)$ is the set of continuations that the player is considering before taking a decision;

A sight



Histories sequences

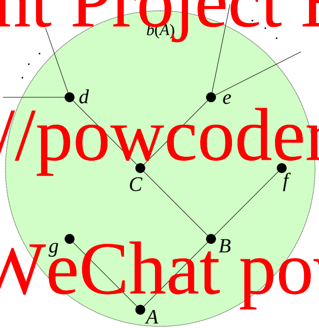
Definition (Histories Sequences)

Let G be an extensive game and s a sight function. A **histories-sequence q** is a sequence of histories of the form $(h_0, h_1, h_2, \cdots, h_k)$ such that

- $h_j \in s(h_0)$ for every $j \in \{1, 2, \cdots k\}$, i.e., histories following h_0 in the sequence are histories within the sight of the player moving at h_0
- $h_j \lhd h_{j+1}$ for each j with $0 \leqslant j < k$, i.e., each history is a prefix of the ones with higher index;

- They encode sight-compatible higher-order points of view. $(h_0, h_1, h_2, \cdots, h_k)$ is what $turn(h_0)$ thinks $turn(h_1)$ thinks $turn(h_2)$ thinks...
- Notice there can be jumps! h_{j+1} need not be h_j plus one action.

A sight



Question: What are the possible histories-sequences?

Belief Structures

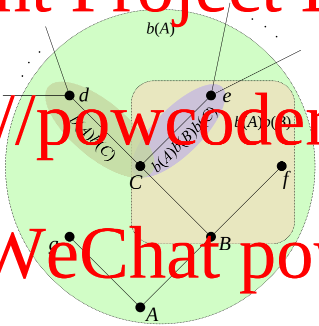
Definition (Epistemic games with short sight)

Let G be a game. To each histories-sequence $\mathbf{q} = (h_0, h_1, h_2, \cdots, h_k)$ we associate an extensive game $G(\mathbf{q})$, that is a subtree of $s(h_0)$ following h_k;

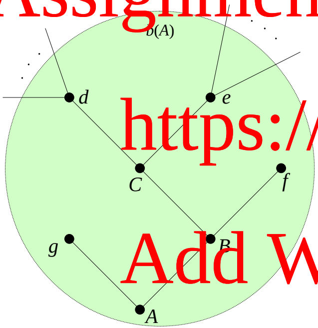
- Intuitively, $G(\mathbf{q})$ is what $turn(h_0)$ thinks $turn(h_1)$ thinks $turn(h_2)$ thinks... $turn(h_k)$ can see and how $turn(h_0)$ thinks $turn(h_1)$ thinks $turn(h_2)$ thinks... $turn(h_k)$ evaluates it;

- Notice that the preference relation on the terminal nodes of $G(\mathbf{q})$ is completely independent of the preference relation in the underlying games.

A sight with beliefs

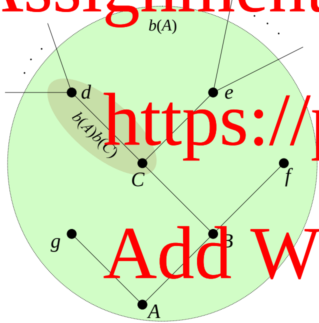


Rethinking Backwards Induction



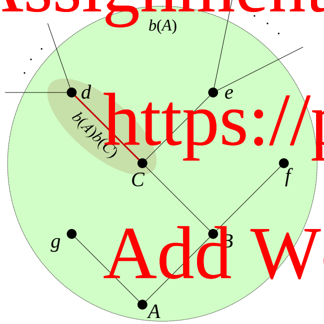
- A can only see b(A)
- She can evaluate its terminal nodes

Rethinking Backwards Induction



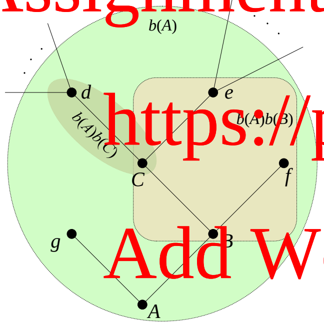
- A needs a belief about what C can see, b(A)b(C)
- ... and about how C evaluates those terminal nodes

Rethinking Backwards Induction



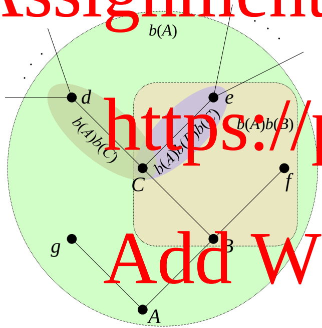
- Then she can start solving the game;

Rethinking Backwards Induction



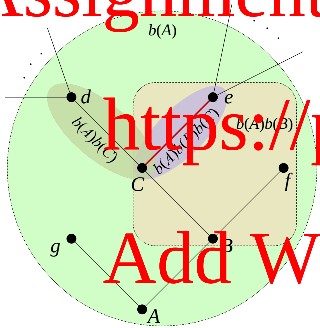
- To determine what B will do A has a belief about what B can see b(A)b(B)
- ... and about B evaluates those terminal nodes

Rethinking Backwards Induction



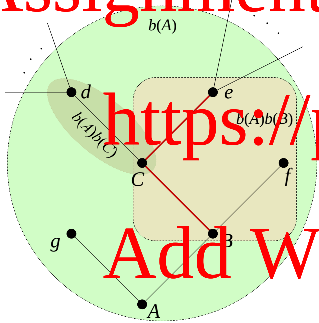
- But on top of that A needs a belief about what B believes C can see
- ... and how B believes C evaluates those terminal nodes

Rethinking Backwards Induction



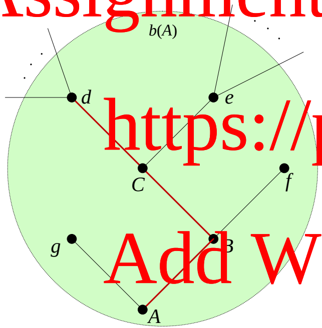
- and then she keeps solving the game...

Rethinking Backwards Induction



- and then she keeps solving the game...

Rethinking Backwards Induction



- and, at last, A will choose her favourite continuation, given her beliefs and her evaluation.

Sight-compatible epistemic solution

- A Sight-Compatible Epistemic Solution is the composition of best moves of players at each history;

- Each such move is a best response to what the current player believes other players will do;
- Each such belief is *supported* by all higher-order beliefs, compatible with the player's sight, about what the opponents can perceive and how they evaluate it.

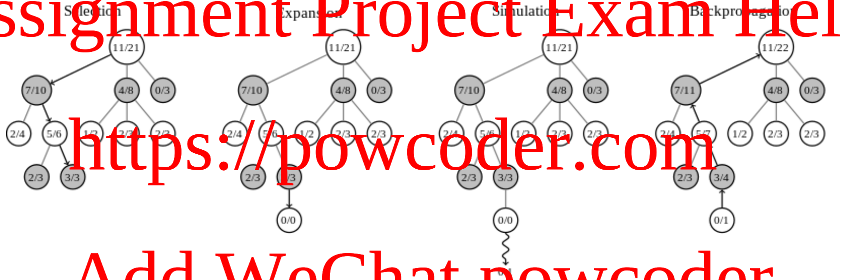
Complexity

Proposition

Computing a SCES is PTIME-complete, on the size of the (finite) game description.

- The algorithm constructs a SCES in PTIME;
- It's PTIME hard because backwards induction is PTIME complete (Szymanik 2014), and we can show that backwards induction is a special SCES.

Monte-Carlo Tree Search



- Selecting random endgames
- Inducing a preference relation over non-terminal histories

Limited Foresight and MCTS

Suppose we are playing Connect 4 and our opponent has a fixed sight: they can see either 1,2,3 or 4 steps ahead.

Then we can start saying things like:
"if she saw 4 steps ahead she wouldn't have made this move"

Using Monte Carlo Tree Search as a way of evaluate non-terminal histories, Cataldo Azzariti showed that MCTS+Limited Foresight+Bayesian Updates can make accurate predictions and plays better than MCTS only.

Cataldo Azzariti
Adaptive MCTS Game-Play Algorithms
Masters Thesis, Imperial College London (2016).

Ongoing research line, with plenty of open questions to answer.

Discussion

- Beyond backwards induction
  - Players try to exploit their opponents' believed weaknesses
  - Built on complex beliefs about game restrictions
    - relativised backwards induction
    - no information sets

- What else?
  - Evaluation criteria the way we do it are still caveman modelling
  - More restrictions and reflection on the computational properties
  - Is *checking* that h is SCES also PTIME complete, can fixpoint logics be of any help?
  - It's unawareness (Halpern, Rêgo, Heifets, Meier, Schipper), but what kind? Specific restrictions allow SCES!