

Modelling an Opponent in Board Games



Julian Jocque

Motivation

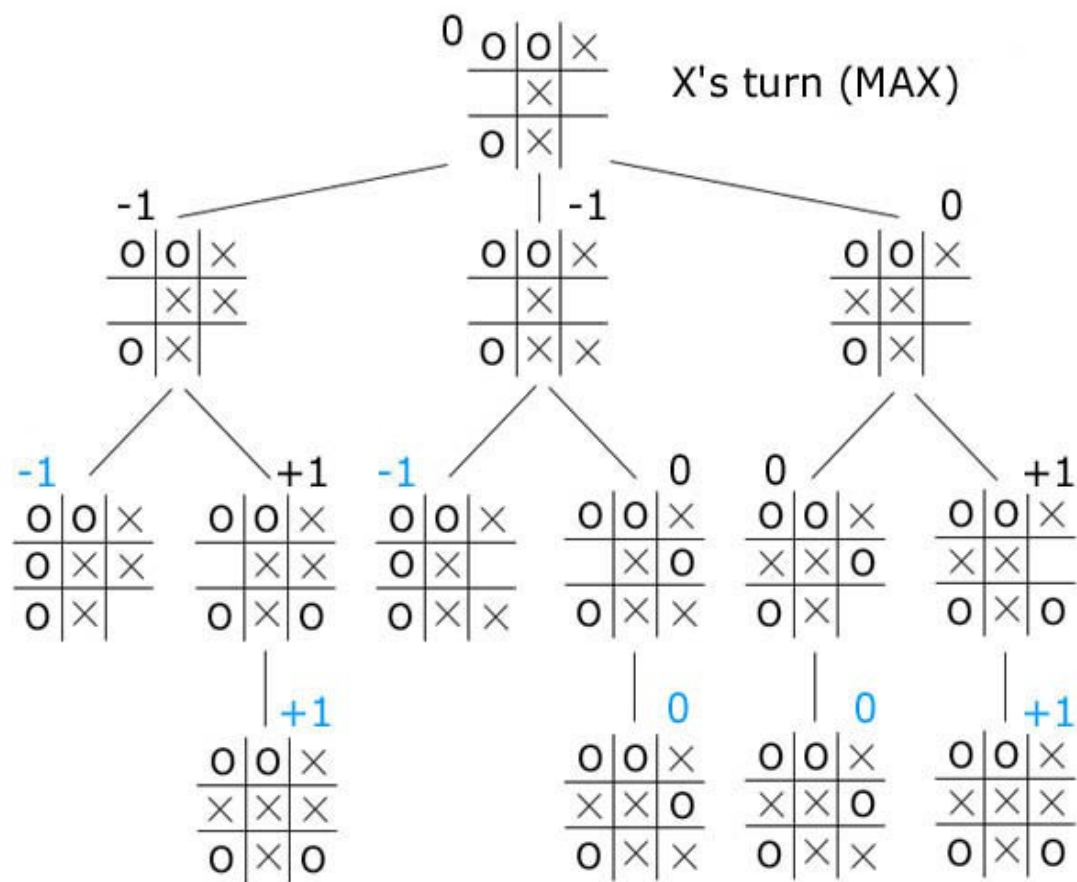
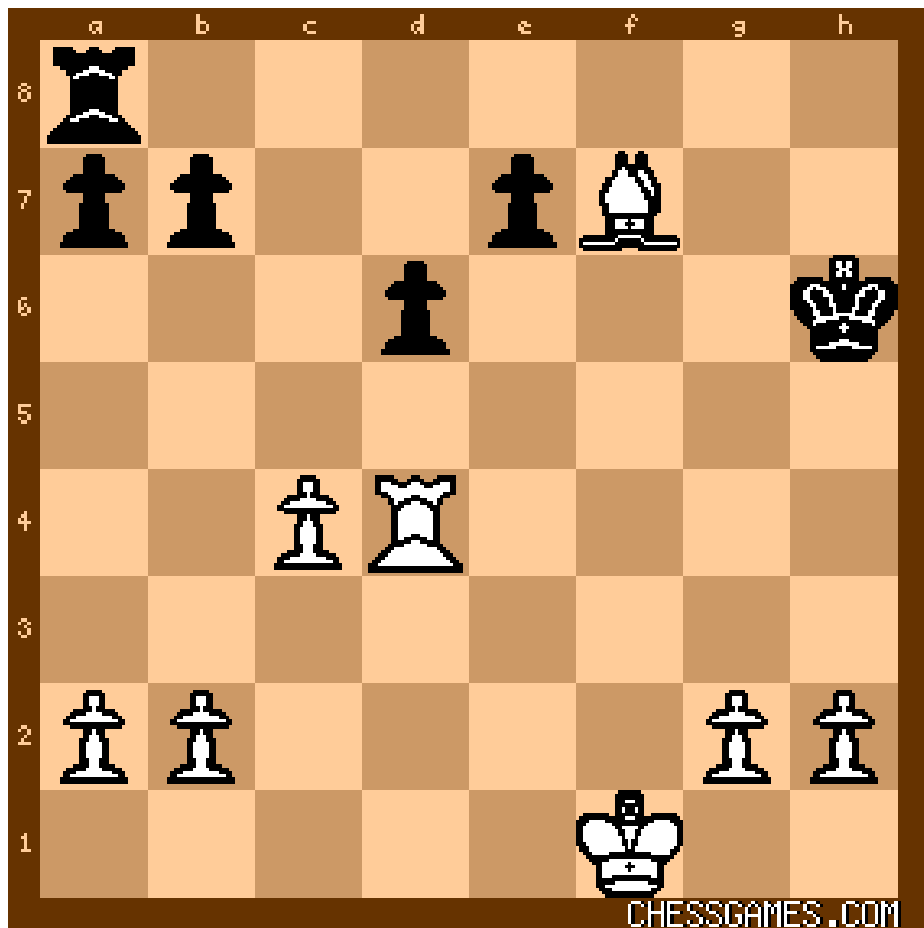
- What if we could create a program to play exactly like Garry Kasparov?



Approach

- Use the Estimation-Exploration Algorithm to model opponents in Konane by presenting the opponent with board states

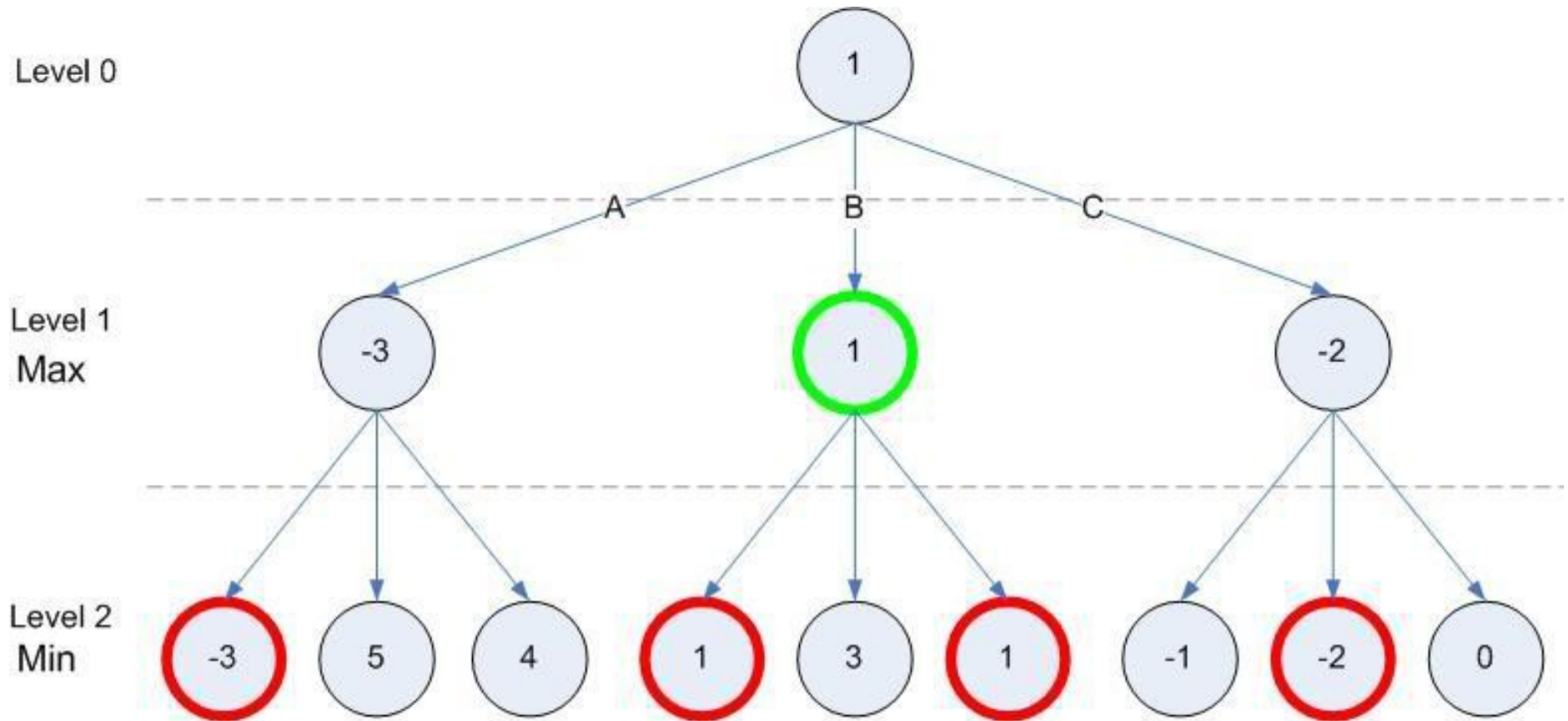
Board State and Game Trees



http://1.bp.blogspot.com/-1VruAI_cdE0/TwccPUduVTI/AAAAAAAAACTg/6hIBEwxXeLI/s400/zz+larsen+petrosian+game+chessboard+r7_pp2pB2_3p3k_8_2PR4_8_PP4PP_5K2.gif

<http://www.ocf.berkeley.edu/~yoseni/extras/alpha-beta/alphabeta.jpg>

Minimax



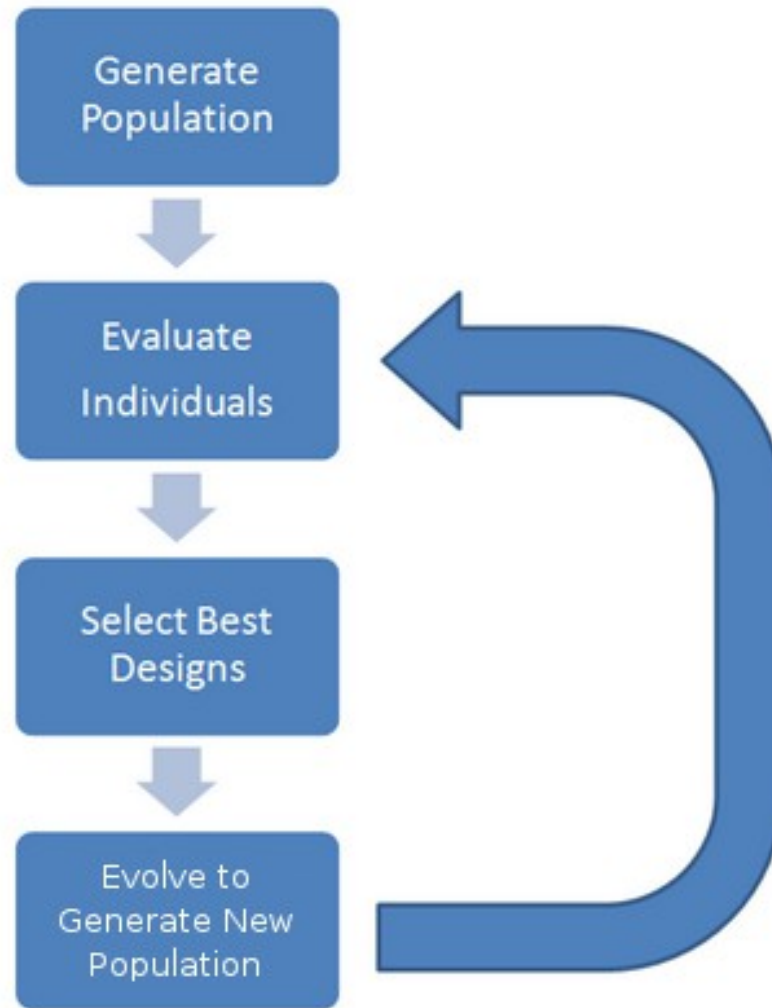
<http://s175.photobucket.com/user/habsq/media/minimax-2.jpg.html>

Static Evaluators

I have a queen	They have a queen	Number of piece I have	Number of pieces they have	I am in checkmate	They are in checkmate
+2.735	-1.4	+3.55	-2.78	-10000	+10000

Allows for Minimax to stop at a particular depth

Evolution



Used with permission from Ben Berger

Estimation Exploration Algorithm 2

1

3.78	2.225	-1.24	1.33	4.3	2.9
		...			
-2.4	-3.6	1.45	1.11	2.0	1.8



3



4

2.41	9.978	1.43	-2.3	3.0	-1.2
		...			
6.89	-1.13	-2.45	-4.1	9.1	-4.2



The System I Built

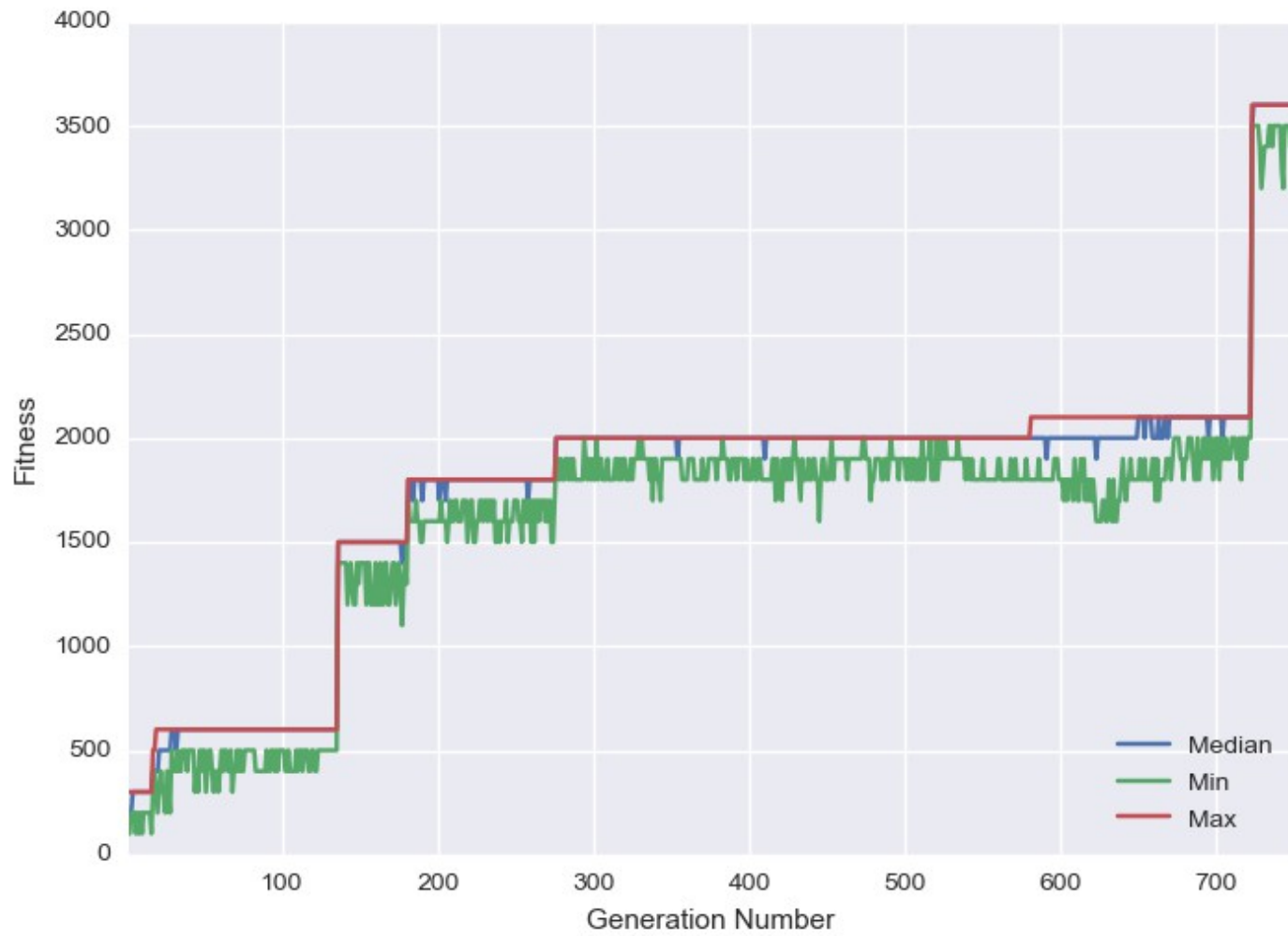
- Konane engine with Minimax, Alpha Beta Pruning
- EEA with evolving static evaluators and evolving sets of board states
- Model evaluator script
- All programmed in Python, all from scratch except Konane starter code

Running The System



Results

Fitness for each generation



Results cont.

- Data still needs interpretation
- Up to 90% accurate on opponents similar to models, only up to 65% on different settings
- About 45% accurate against opponent found on Github

Questions?

