The George Kingsley Zipf AI Engine : Tests

Created by Erin Dahlgren December 1 2011
Modified by Erin Dahlgren January 2 2012

Statistics are well and good, but they are just as much *likelihood* as they are *uncertainty*. That's why extensive testing was crucial for The George Kingsley Zipf Engine. Here are three tests, using code module `testgames.py`.

1. George against George
2. George against a fixed-first-move George
3. George against a random-move player: 10,000 times!

George against George If the George Kingsley Zipf Engine completely plays against itself, from the very beginning (George 'X') to the very end, and if the Engine is constructed never to not lose, a tie should result. Since George will always make the same decision a given a status of the board b, George 'X' always makes the same moves given George 'O's moves, and vice versa. One game results:

O	X	O
O	X	X
X	O	X

player	*move at (row, col)*
X	(1,1)
O	(0, 0)
X	(2, 2)
O	(0, 2)
X	(0, 1)
O	(2, 1)
X	(2, 0)
O	(1, 0)
X	(1, 2)

Indeed, George 'X' ties against George 'O'. The Engine passes the initial test.

George against a fixed-first-move George If George 'X' is forced to take a certain first move, and then plays the game out using the Engine logic against a non-constrained George 'O, two things should results: (1) no losses, and (2) weaknesses of George's opponent if that opponent is not a carbon-copy of himself.

forced start (0,0)

X	.	.
.	.	.
.	.	.

$\downarrow$

X	O	X
X	O	O
O	X	X

player	*move at (row, col)*
X	(0, 0)
O	(1, 1)
X	(2, 2)
O	(0, 1)
X	(2, 1)
O	(2, 0)
X	(0, 2)
O	(1, 2)
X	(1, 0)

forced start (0,1)

.	X	.
.	.	.
.	.	.

$\downarrow$

O	X	X
O	O	.
O	X	X

player	*move at (row, col)*
X	(0, 1)
O	(1, 1)
X	(2, 1)
O	(0, 0)
X	(2, 2)
O	(2, 0)
X	(0, 2)
O	(1, 0)

forced start (0,2)

.	.	X
.	.	.
.	.	.

$\downarrow$

X	O	X
O	O	X
X	X	O

player	*move at (row, col)*
X	(0, 2)
O	(1, 1)
X	(2, 0)
O	(1, 0)
X	(1, 2)
O	(2, 2)
X	(0, 0)
O	(0, 1)
X	(2, 1)

forced start (1,0)

.	.	.
X	.	.
.	.	.

$\downarrow$

O	X	O
X	O	X
O	.	X

player	*move at (row, col)*
X	(1, 0)
O	(1, 1)
X	(1, 2)
O	(0, 0)
X	(2, 2)
O	(0, 2)
X	(0, 1)
O	(2, 0)

for George 'X' forced start (1,1)
see first test section

forced start (1,2)

.	.	.
.	.	X
.	.	.

$\downarrow$

O	X	O
X	O	X
O	.	X

player	*move at (row, col)*
X	(1, 2)
O	(1, 1)
X	(1, 0)
O	(0, 0)
X	(2, 2)
O	(0, 2)
X	(0, 1)
O	(2, 0)

forced start (2,0)

.	.	.
.	.	.
X	.	.

$\downarrow$

O	O	X
O	O	X
X	X	O

player	*move at (row, col)*
X	(2, 0)
O	(1, 1)
X	(0, 2)
O	(1, 0)
X	(1, 2)
O	(2, 2)
X	(0, 0)
O	(0, 1)
X	(2, 1)

forced start (2,1)

.	.	.
.	.	.
.	X	.

$\downarrow$

O	X	X
O	O	.
O	X	X

player	*move at (row, col)*
X	(2, 1)
O	(1, 1)
X	(0, 1)
O	(0, 0)
X	(2, 2)
O	(2, 0)
X	(0, 2)
O	(1, 0)

forced start (2,2)

.	.	.
.	.	.
.	.	X

$\downarrow$

X	O	X
X	O	O
O	X	X

player	*move at (row, col)*
X	(2, 2)
O	(1, 1)
X	(0, 0)
O	(0, 1)
X	(2, 1)
O	(2, 0)
X	(0, 2)
O	(1, 2)
X	(1, 0)

Two crucial things should strike us from these games. First, George 'O' never losses and sometimes even wins. Second, George 'X' can be forced to lose (forked) if we force him to choose a non-corner, non-center position for his first move, *even if he mimics the decision making of George (proper) later on.* This clearly demonstrates George's aggressive strategy. The next section will prove his *no-loss* strategy.

George against a random-move player: 10,000 times! I engineered an artificial player who makes random moves against his opponent. In `testgames.py` see `play_once(p1, p2)` and its dependents. I pitted this player against George and iterated over a high number of games, such that the probability that the random player did not exhaust all possible moves for 'X' would be nearly invisible:

(1) $9*7*5*3 = 945 =$ *total possible moves: George's opponent*

(2) $8*6*4*2 = 384 =$ *total possible moves: George*

Remember, George always goes second.

If we test George and player 'X' 10,000 times, and we know that player 'X' has only 945 possible moves, then we know that there is but a 0.00002551% chance that we got only 944 moves out of 945, if we selected them via random sampling ((944/945)^{1}0000). And this percent chance drops even further if we ask the same question for 943 moves or 942 moves, etc., out of 945.

By the law of large numbers, 10,000 games will provide by far enough chances to exhaust all possible games. Finally the statistics:

wins	*losses*	*ties*
8699	0	1301

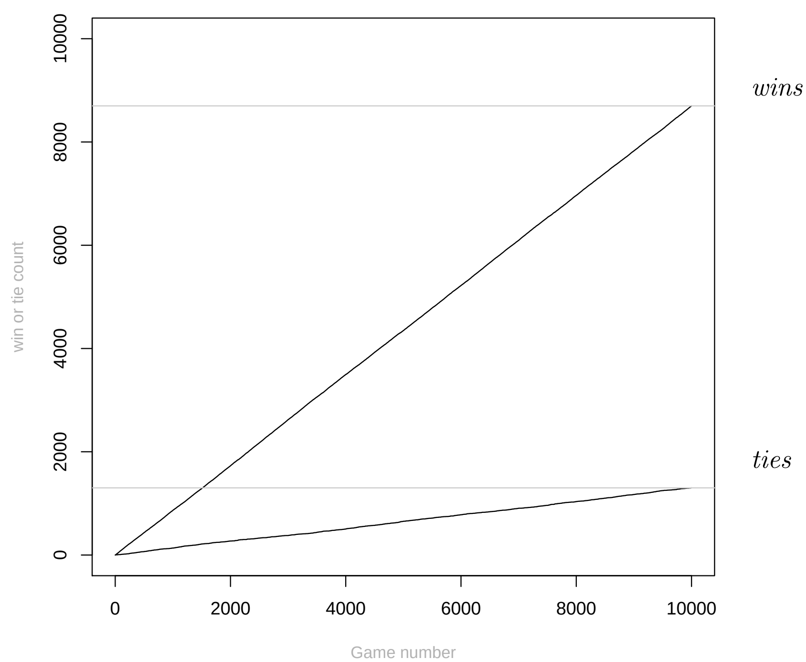
From these statistics, George has passed the no-loss examination.[1] Most interesting, though, is what they tell us about his winning strategy: if George played completely defensively, we would not expect his percent wins to be higher

[1] See `tenthousand_results` doc for raw data.

than 50%, and in general we would expect more ties. Instead, George wins 87% out of a random sampling of all possible games, suggesting that he is an extremely *aggressive* player. This was my original goal.

As wins increased over the 10,000 games, to reach the final 8699, we might wonder what was their rate of increase? As ties increased, to reach 1301, what was *their* rate of increase? If the graphical projections of these rates show strong stability, we would have further confidence in the stability of the *zero X wins* statistic.

I have graphed the cumulative wins and ties for each new game played, to help us answer these questions. plotting done in `R`



This is a fabulous result. This plot shows cumulative counts for all 10000 games, and it should be striking that there is no variation around either the wins plot line and the ties plot line. This is evidence that George's wins and ties each increase at a *steady rate* as the games he plays against a random player increases. This is also strong evidence against the possible disruption of these plots by an 'X' player win: a chaotic or more variable plot would license us to consider this.