Analysis of Voltorb Flip Solver Policies

Multiple people online have posed policies for solvers to the mini-game Voltorb Flip from the game Pokemon Soul Silver. This analysis aims to test each policy and determine if any are sub-optimal.

The premise of this analysis is to test each of the policies against each other. To determine optimality, a reinforcement learning model was trained on the same game and was accepted as the optimal solution. Then, each policy's win rate would be compared to the best win rate achieved across all policies. If any given policy's win rate was below the best by a statistically significant margin, it would be considered sub-optimal.

For each move of each policy, 200 valid boards were generated and used to compute statistics on the likelihood of each cell. Then, each policy was presented the likelihood that a cell in the game was a bomb, one, two, or three. Each policy then used this information to select the best move. Here is the breakdown of each policy:

Avoid Bomb: Take the move that is the least likely to contain a bomb

Greedy: Take the move that is the most likely to be a 2 or 3.

Greedy with penalty: Take the move that is the most likely to be a 2 or 3 and least likely to be a bomb

Max Entropy: Take the move that reveals the most information about the board configuration while avoiding bombs

Reinforcement Learning: The "optimal" policy determined my reinforcement learning.

Each policy played 20,000 games, 2,500 of each level. The win rate of each of these sets of games were used to compute the per-level win rate and the total game win rate. Below are the win rates for each policy. Policies with a number after them signify a policy with the same strategy but different coefficients

Win Rate per Level									
Policy Name	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**	**Level 8**	**Average**
Greedy With Penalty 2	87.81%	77.95%	65.15%	53.50%	39.91%	43.37%	27.81%	29.76%	53.16%
Reinforcement Learning	87.10%	77.01%	64.86%	53.35%	39.13%	42.87%	28.51%	31.15%	53.00%
Greedy With Penalty 3	87.91%	75.86%	63.82%	52.65%	38.46%	43.00%	29.21%	31.70%	52.83%
Greedy With Penalty	87.37%	76.24%	64.92%	55.64%	38.50%	41.78%	27.36%	30.54%	52.79%
Max Entropy	85.51%	76.07%	65.17%	51.98%	36.55%	38.17%	26.57%	28.94%	51.12%
Max Entropy 2	85.25%	75.29%	63.29%	50.20%	38.62%	38.97%	26.85%	30.06%	51.07%
Greedy	80.12%	69.78%	57.43%	49.71%	32.81%	37.88%	25.72%	28.10%	47.69%
Avoid Bomb	76.68%	65.40%	52.13%	43.42%	27.81%	30.75%	21.30%	25.14%	42.83%

	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**	**Level 8**	**Average**
Best Win Rate	87.91%	77.95%	65.17%	55.64%	39.91%	43.37%	29.21%	31.70%	53.16%

To test whether a policy was sub-optimal, I asserted a null hypothesis of “This policy yields the highest win-rate of all policies for this level”. To test this hypothesis, I computed the z-score against the highest win-rate for each level. Below is a table showing all policies that failed a 95% confidence interval and are sub-optimal.

Z-score of "This policy is the best policy for this level"									
Policy Name	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**	**Level 8**	**Overall**
Greedy With Penalty 2	-0.15	0.00	-0.02	-2.14	0.00	0.00	-1.56	-2.13	0.00
Reinforcement Learning	-1.21	-1.12	-0.32	-2.29	-0.79	-0.51	-0.78	-0.59	-0.45
Greedy With Penalty 3	0.00	-2.44	-1.41	-3.00	-1.49	-0.38	0.00	0.00	-0.94
Greedy With Penalty	-0.82	-2.01	-0.27	0.00	-1.45	-1.61	-2.08	-1.26	-1.03
Max Entropy	-3.41	-2.20	0.00	-3.67	-3.49	-5.36	-2.99	-3.04	-5.76
Max Entropy 2	-3.76	-3.08	-1.95	-5.44	-1.32	-4.52	-2.66	-1.79	-5.92
Greedy	-9.76	-8.89	-7.83	-5.93	-7.55	-5.66	-3.99	-4.00	-15.47
Avoid Bomb	-13.29	-13.19	-13.06	-12.33	-13.50	-13.68	-9.66	-7.55	-29.52
Red signifies statistical significance and rejection of null hypothesis									
H0: This policy is the best policy for this level									

The only policy that was not rejected as sub-optimal was greedy with penalty. This implies that the best policy to play the game with is as follows:

Selection score = (probability cell is a 2 or 3) / (epsilon + probability of bomb)

Where the move with the highest selection score is the best move to take, and epsilon is a small number to prevent divide by zero.