Las Vegas Board Game

Jose Luis Mazariegos Ballesteros and Moises Moscoso Valenzuela

Abstract In this project we create an intelligent system to determine what is the best heuristic to win at Las Vegas Board Game.

1 Introduction

We want to program an intelligent system able to play the game with different heuristics. With the information of the winners we will determine what is the best strategy in order to win at Las Vegas Board Game.

The system should consider all the rules of the game and the different heuristics that we program.

1.1 Rules of the game

The game consists in 5 players trying to collect as much money as possible from the casinos.

Set for each round:

- The banknotes are placed in each casino until they have at least 50,000.
- Each player will have 8 dices of one color.

———————————————

Jose Luis Mazariegos Ballesteros

Instituto Tecnologico y de Estudios Superiores de Monterrey, Guadalajara, Mexico, e-mail: estapepe@gmail.com

Moises Moscoso Valenzuela

nstituto Tecnologico y de Estudios Superiores de Monterrey, Guadalajara, Mexico, e-mail: moncho_894@hotmail.com

In your turn roll all your dices and select a set of dice with a single number. Put that set in the corresponding casino. When you run out of dice, pass your turn.

On each casino the player with the most dice receives the highest banknote. In case there are more banknotes, the second place receives the next high banknote and so on.

The game continues for 4 rounds and the winner is the player with the most money. In case of a tie, the winner is the player with the most banknotes.

2 Hypothesis

We believe that executing a fair amount of simulations of a game match will give us enough data. The data will determine if there is an statistically significant bias in the amount of times each match is win by a player using a particular heuristic. This will tell us if the use of that particular heuristic is translated in a better chance to win.

3 Methodology

We modeled each component of the game as an object, using the Python programming language. Included in this objects there's a configurable Player. Each player is configured once at the beginning of the match, and remains with the same configuration until the match ends and the game evaluates the outcome to determine a winner.

In our experiments, the game and players remain in the same configuration during each replica of the experiment. Each replica of the experiment consists of one match, each experiment consists of a given number of replicas, the result of each match consist of the evaluation of the last state at which the game is at the moment that all of the players run out of dice. The discrete unit which changes the state of the game is the turn, at each turn only one player changes the state of the game, a round consists of exactly one turn for each player taken in the same order in every round, a match consists of as many rounds as needed for the all the players to be left with no dice. If a player has run out of dice when its turn comes, the state of the game after its turn will be the same as before its turn. The result of the experiment is a set of tuples, one for each replica. Each tuple contains data taken from the evaluation the game executes at the end of every match and is composed as follows:

$$y = \{x1, x2, ..., xn\} \tag{1}$$

Where xi is the amount of game money the player i is assigned in the final evaluation. For the purpose of analyzing the data, these values where normalized as follows:

$$y' = \frac{y}{|y|} \tag{2}$$

In our experiments there are 5 players with each using a different heuristic.

We determined that, once every other element of the game is set and initialized, the fundamental operation that determines the outcome of a match is the player's decision at each turn to relinquish one subset of dice from the set of all the dice it possesses, this set is partitioned in subsets that have dice with the same top face for the current turn (the player rolls the dice once each turn). Hence our heuristics are nothing but a way for the player to choose which subset of dice to relinquish each turn.

3.1 Heuristics

Alpha	This heuristic will choose the set of dices with the highest amount of dices.
Bravo	This one will choose the casino with the highest banknote in order to get at least one high prize.
Charlie	It will choose the casino with the most banknotes, so it will make sure to receive something.
Delta	It will try to outnumber an opponent in a casino
Echo	It will try to match an opponent so that neither of them will win the prize.

4 Results

We analyzed the results of 1000 replicas of the mentioned experiment (1 match with 5 players, each using a different heuristic). By looking at figure 1, we can observe certain characteristics of the behavior of each heuristic.

Overall, none of the heuristics achieved a mean above 0.5, that alone means that none of those would give one a particular advantage over a random play if the game was simplified to coin toss. On the contrary, the chances of winning using these heuristics alone would overall give the player less than a 50% chance of winning.

However, comparatively, some performed better than others. Particularly Alpha, has an advantage over Charlie and Delta, which performed poorly, as its worst case performance is comparable to Charlie and Delta's mean performance.

We can also observe that Bravo has the highest mean but an slightly higher standard deviation compared to Alpha, which puts both heuristics apart from the rest, but not one of them necessarily superior to the other.

Echo has a surprisingly higher performance than expected because it tries not to surpass its opponents but to match them which makes us believe that its slightly superior behavior compared to Delta and Charlie is due to the fact that the opportunity to match an opponent does not come often and it ends up playing randomly instead.

Lastly we can observe that Delta turns out to be a safe strategy to play because its results do not vary too much and tends to almost secure at least one small prize.

5 Conclusion

With 1000 replicas of the experiment, we couldn't determine with precision which of the heuristics was the best one since all of them won almost the same amount of time. Bravo had a slightly an advantage over the others, but is so small that we think that with more replicas, another one could had won.

References

1. Dorn, R. (2012), in *Las Vegas: Gamble it up to the very last roll! [Board Game]*, Newton, N.H.: alea/Oya/Ravensburger Spieleverlag GmbH, 2012.
2. Wolfram Alpha LCC, in *Wolfram Alpha*, Computational knowledge engine. (Available via WolframAlpha, 2016), https://www.wolframalpha.com/ Cited 28 Apr 2016

	alpha	bravo	charlie	delta	echo
mean	0.47684	0.48430	0.30820	0.30456	0.36812
s.d.	0.20826	0.22116	0.20073	0.16914	0.23205
min.	0	0	0	0.029161	0
median	0.47680	0.50278	0.28807	0.36864	0.35799
max.	0.93646	0.98590	0.91273	0.80498	0.91733
total	476.84	484.30	308.20	304.56	368.12
count	1000	1000	1000	1000	1000

Computed by Wolfram|Alpha

Fig. 1 Statistics: We can observe the characteristics of the 5 heuristics, none of them achieve a mean above 0.5

(log counts)
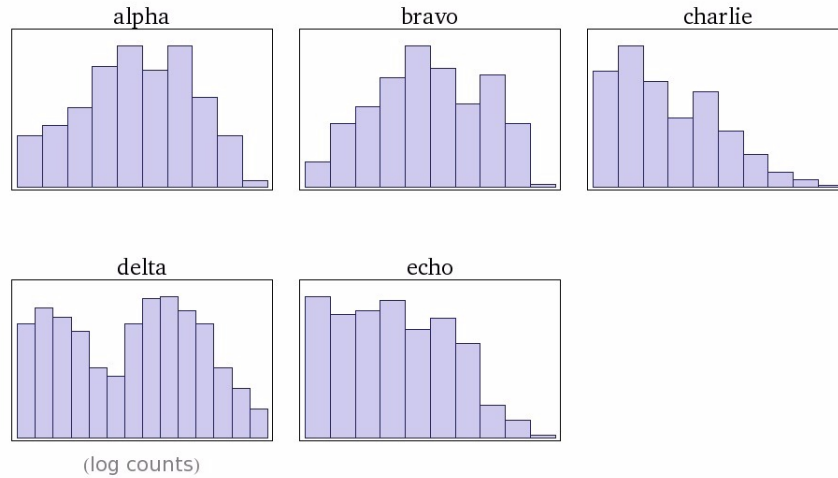
Computed by Wolfram|Alpha

Fig. 2 Histogram: We can observe the variation that each heuristic had over the replicas of the experiment.