Solution to Quant Researcher Puzzle - Optiver

Jon Sveinbjorn Halldorsson

October 19, 2020

The puzzle: Three players A, B, C play the following game. First, A picks a real number between 0 and 1 (both inclusive), then B picks a number in the same range (different from A's choice) and finally C picks a number, also in the same range, (different from the two chosen numbers). We then pick a number in the range uniformly randomly. Whoever's number is closest to this random number wins the game. Assume that A, B and C all play optimally and their sole goal is to maximise their chances of winning. Also assume that if one of them has several optimal choices, then that player will randomly pick one of the optimal choices.

(a) If A chooses 0, then what is the best choice for B?

(b) What is the best choice for A?

(c) Can you write a program to figure out the best choice for the first player when the game is played among four players?

Solution: (a) We will consider each player as a dynamic circle on the line $[0, 1]$ and the player with the highest score (or largest diameter $2r_{(\cdot)}$) at the end of the game is most likely to win. In order to play optimally, each player will pick a point on the line which results in the largest radius while decreasing the opponents' radiuses. To behave like a circle, each player will have its point $p_{(\cdot)}$ which is picked, a midpoint $m_{(\cdot)}$ which will shift and a radius $r_{(\cdot)}$ which will change in size as more players are added to the game.

At first player A will have a midpoint of $m = 0.5$ and a radius $r = 0.5$ regardless of the chosen $p_A \in [0, 1]$ but once player B enters the game the midpoint and radius of A will change so we have to consider a few scenarios. If we consider the edge cases, where A chooses either $p_A = 0$ or $p_A = 1$, clearly B will choose $p_B = p_A \pm \epsilon$, cornering A and creating the maximum radius for itself.

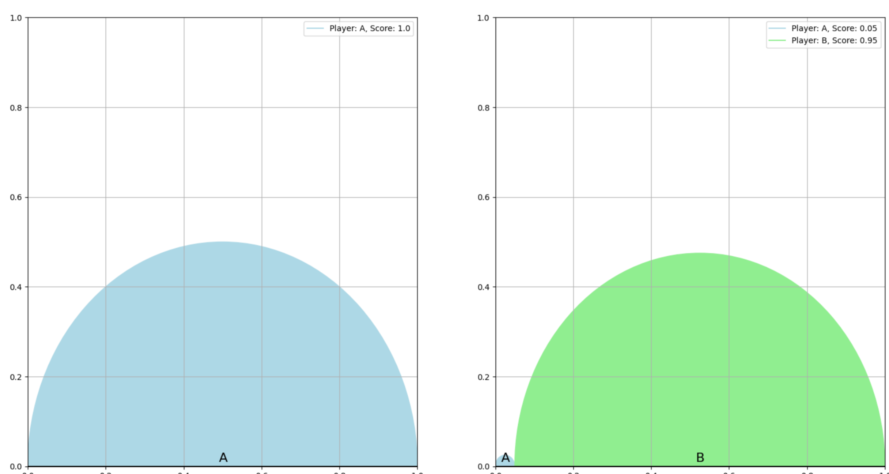

Figure 0.1: Given that A will pick $p_A = 0.0$ the figure above shows how effectively B can counter by picking $p_B = p_A + \epsilon$ where $\epsilon = 0.1$. The same holds for the other edge case where $p_A = 1.0$.

(b) To identify the best choice for A, we will first consider three cases. In the two edge cases discussed above, A will be cornered by B which in turn will be cornered by C and so on. The third case will be picking the expected value, i.e. the middle.

$$x \sim \mathcal{U}([0,1]), \quad \mathbb{E}[x] = \int_0^1 x dx = 0.5$$

In a two-player game, the middle would be the optimal move for the first player since the second player would need to pick either $0.5 \pm \epsilon$ and have a slight disadvantage. However, in a game with more than 2 players the third player would pick whichever side of the middle that the second player didn't pick, restricting the first player to a very minimal range in the middle $[0.5-\epsilon, 0.5+\epsilon]$. In these three scenarios, $p_A = \{0.0, 0.5, 1.0\}$, the score (or probability of winning) of player A tends to 0 as the minimal step size ϵ tends to 0.

We have now identified three minima in the score function $s(x) \in (0,1)$ and can therefore deduce that there must be at least one optimum in the range $(0.0, 0.5)$ and $(0.5, 1.0)$. Following the logic below, we can expect these two ranges to behave in the same way due to the symmetry so we will focus on the former.

We know that as the point picked by A gets further from 0, the score (or probability of A winning) will increase, but as it nears 0.5 it will decrease. This is due to the fact that if p_A is sufficiently low (or high), it will be more worthwhile for B, C and so on to pick a number larger (or lower) than p_A. This means that B will pick $p_B = p_A + \epsilon$ (or $p_B = p_A - \epsilon$) and will cause a chain effect where each player who makes a move after B will choose to $p_C = p_B + \epsilon$ (or $p_C = p_B - \epsilon$).

In this strategy, the n-th player will need to make the decision of picking between $p_A - \epsilon$ and $p_A + n\epsilon$ (or $p_A + \epsilon$ and $p_A - n\epsilon$), i.e. the score available to the n-th player will be the following.

$$s(p_{(\cdot)}) = \begin{cases} \epsilon/2 & \text{if } p_{(\cdot)} = \{0.0, 1.0\} \\ \max\{p_A - \epsilon, 1 - (p_A + n\epsilon)\} & \text{if } p_{(\cdot)} \in (0.0, 0.5) \\ \max\{1 - (p_A + \epsilon), p_A - n\epsilon\} & \text{if } p_{(\cdot)} \in (0.5, 1.0) \end{cases}$$

Given that all players will play optimally, player A will know the consequences of his number so he will need to pick a number satisfying either of the following inequalities:

$$\begin{aligned} p_A - \epsilon <& 1 - (p_A + n\epsilon), \quad p_A \in (0.0, 0.5) \\ 1 - (p_A + \epsilon) <& p_A - n\epsilon, \qquad\quad p_A \in (0.5, 1.0) \end{aligned}$$

Which results in the optimal p_A being in either of the following ranges.

$$0 < p_A < \frac{1-(n-1)\epsilon}{2} \text{ or } \frac{1+(n+1)\epsilon}{2} < p_A < 1.0$$

If player A knows the number of players n and the step size ϵ, the optimal choice is either $p_A = \frac{1 \pm (n \pm 1)\epsilon}{2} \mp \epsilon$. For a small step size ϵ and a reasonable number of players n, this number will be close to 0.5 and is displayed in the figure below.

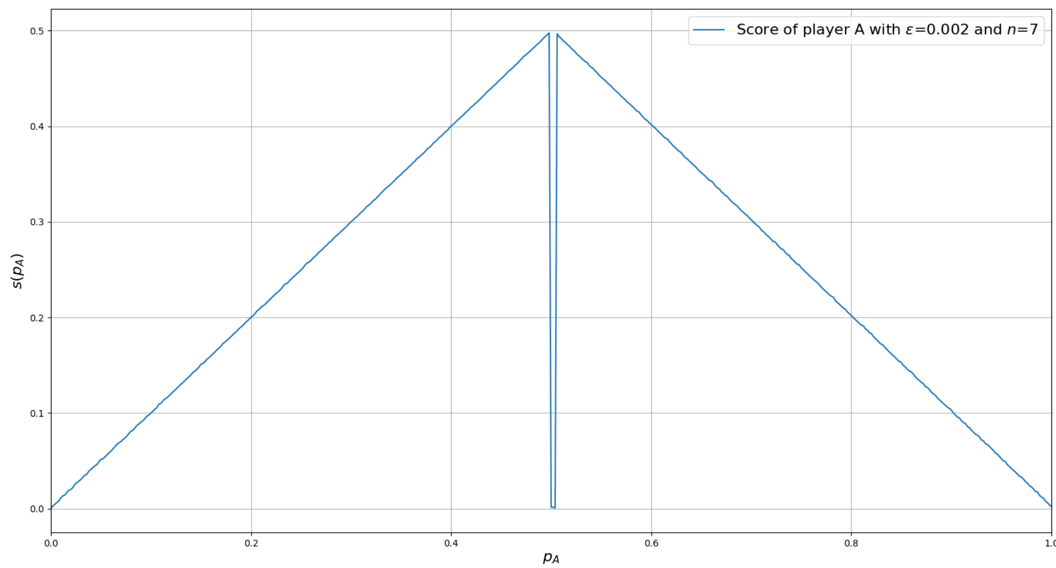

Figure 0.2: The score for player A as a function of p_A

(c) I implemented a simple object oriented version of the code for up to 7 opponents and made it publicly available on GitHub if anyone is interested (`https://github.com/jonsh96/QR_Puzzle/`). The repository will be made private as soon as I know it has been viewed or by request.