All-Pay Bidding Games on Graphs

Bob's Game: Yinjie Li; Jiaying Du; Chengqian Li, Ziyang Sheng, Yuesong Li

ABSTRACT

The paper we meanly focus on describes some all-pay bidding games and gives a new definition : upper-bound. Upper-bound depends on the vertex in the graph. The game progresses to a stage, which corresponds to a vertex in the graph. If the player has money more than upper-bound of this vertex, he has a strategy that guarantees winning with probability 1. In this project, the games described in the paper are all implemented and the game user interface is interactive and intuitive. Also, in this game, two gaming agents are designed to test and player can play with them. An algorithm which is used to calculate upper-bond for every vertex in the graph is implemented, too.

KEYWORDS

All-pay bidding game; Upper bound; Graph user interface; Game value function

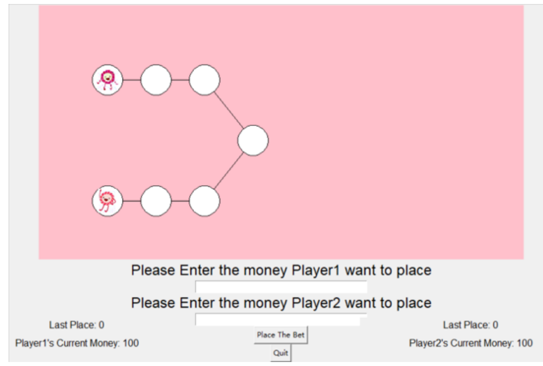

Figure 1: User Interface

1 INTRODUCTION

Two-player graph games are divided into two types: Synchronous game and asynchronous game. Synchronous game means two players determine their actions at the same time. In contrast, asynchronous game means two players take turns to determine their actions. The game in this paper discussed is a synchronous game. Both of two players have their own initial budget and target winning rounds. In every round, they need to choose a value as their bids. The player is the winner in this round if his bid is the larger one. The winner in this game is the player who is the first to complete his winning rounds. This game has three modes: All-pay, Richman and Poorman. All-pay mode means the bank will gets both players' bids after each round. Richman mode means the loser in this round will get the winner's bid. Poorman mode means the bank only gets the winner's bid.

The paper this project meanly focus on researching some attributes of All-pay bidding games and gives a new definition : upper-bound. Upper-bound depends on the vertex in the graph. The game progresses to a stage, which corresponds to a vertex in the graph. If the player has money more than upper-bound of this vertex, he has a strategy that guarantees winning with probability 1. The project implements the game and the game user interface is interactive and intuitive. Two agents are also designed, which can be used as the opponent for the player and used to test the upper-bound in different cases.

2 RELATED WORKS

Colonel Blotto games have been extensively studied; a handful of papers include (Bellman 1969[4]; Blackett 1954[5]; Hart 2008[6]). Most of them focus on the discrete cases, in which the armies are given as individual soldiers. However, with regard to our model, which belongs to the continuous cases[9], the most well-studied objective is maximizing the expected payoff, though recently Behnezhad et al[3] study the objective of maximizing the probability of winning at least k battlefields.

The All-pay bidding games were first discussed by Lazarus et al[7], where it was concluded that optimal strategies require probabilistic choices. Afterwards, all-pay bidding games were studied only with discrete-bidding[8], which significantly simplifies the model, and in the Richman settings. On the basis of first price auction, Avni and his colleagues[1] develop the games into richman, poorman and all-pay modes. Moreover, they also put forward the game value function, which reveals some connections between the winning probability and the initial budget ratio.

3 MODEL DESCRIPTION

3.1 Preliminary Settings

In this project, we focus on all-pay bidding games on directed acyclic graphs. Generally, the game is $G < V, E, L, \omega >$, where V is a finite set of vertices, E $\subseteq V \times V$ is a set of directed edges, $L \subseteq V$ is a set of leaf nodes without any child node, and $\omega : L \longrightarrow Q$ assigns weights to each leaves, which represents the utility. Two players are initialized at the same vertex v_0, each with an initial budget. Then they will have multiple rounds of bidding, and the winner in each round will move its token to one of the child vertices. The game ends once one of the player reaches a leaf, who will obtain the weight attached to it.

To be specific, the game have two parameters: the structure of the graph and the initial budgets of two players. Our research starts with the fundamental cases which the game can be represented as "Win n in a Row". As a fragment of qualitative games, the "Win n in a Row" game are denoted as WnR(n) in which Player 1 needs to win n biddings in a row, otherwise player 2 wins. As figure 2 shows, the simplest case is WnR(2).

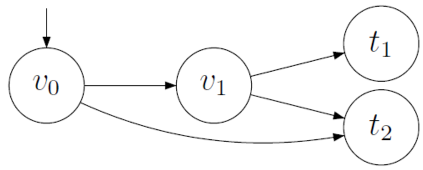

Figure 2: Sample for the game "WnR(2)"

Formally, for $i \in 1, 2$, suppose that Player i's budget is $B_i \in Q^*$. In the kth round, Player i's bid is b_{ik}. Then the higher bidder moves its token Player i's budget is updated to: $B_i - \sum_{j=1}^{k} b_{ij}$. In games of WnR(n), since the structure of the graphs is given by the parameter n, the defining parameter is the initial budget ratio of the two players. A history π in the bidding game can be denoted as:

$$\pi =< v_1, b_1^1, b_1^2 >, \cdots, < v_k, b_k^1, b_k^2 >, v_{k+1} \in (V \times R \times R)^* V \quad (1)$$

where for $1 \leq j \leq (k+1)$, the token is placed at vertex v_j at round j and player i's bid is b_j^i. Let B_i^I be the initial budget of player i. Player i's budget following π is $B_i(\pi) = B_i^I - \sum_{1 \leq j \leq k} b_j^i$. A play π that ends in a leaf l ∈ L is associated with the payoff $\omega(l)$.

According to Avni's research[2], there is some connection between initial budget ratio and winning probability, which can be expressed as game value function. Furthermore, when the ratio exceeds a specific value, which is defined as upper bound, one of the players will have a strategy that can guarantee winning.

3.2 Game Value & Upper Bound

Game value function reflects how the initial budget ratio can influence the player's winning rate. The lower value in an all-pay game G with an initial vertex v_0, and an initial budget ratio r is:

$$val^{\downarrow}(G, v_0, r) = sup_f inf_g \int_{\pi \in D(v_0, r, f, g)} Pr(\pi) pay(\pi) \quad (2)$$

where f is the strategy set for two players, and D is the winning probability.
The upper value $val^{\uparrow}(G, v_0, r)$ $(\in [0, 1])$ is defined similarly. Based on the definition, we always have $val^{\downarrow}(G, v_0, r) \leq val^{\uparrow}(G, v_0, r)$. And when the "=" holds, the game value exists and equals to it. Intuitively, the game value represents that given the initial budget ratio, at least one of the player will have a strategy that guarantees the winning rate no less than the game value. For the the simplest game, we can prove that the game value is $\frac{1}{n+1}$ for the initial budget ratio $r \in [\frac{1}{n+1}, \frac{1}{n}]$. When the game value reaches 1, we say the corresponding budget ratio is the upper bound.

Upper bound is the budget ratio threshold T(v) at vertex v that can be defined as follow:

- If Player *1*'s ratio exceeds T(v), he has a strategy that guarantees winning with probability 1.
- If Player *1*'s ratio is less than T(v), Player 2 has a strategy that guarantees winning with positive probability.

Given the game G<V,E,$\{t_1, t_2\}$>, for v ∈ V \ $\{v_1, v_2\}$, let v_1, v_2 be the neighbors for vertex v such that $T(v^-) \leq T(v) \leq T(v^+)$, the upper bound can be calculated based on the following function.

$$T(v) = \begin{cases} 0, & v = t_1 \\ \infty, & v = t_2 \\ T(v^-) + 1 - \frac{T(v^-)}{T(v^+)}, & otherwise \end{cases} \quad (3)$$

Define G(i,j) as the game where player *1* needs to win i rounds while player *2* needs to win j rounds. The following shows the calculation of upper bound using matrix for G(3,3). In the 3×3 matrix, the initial vertex v_0 is the vertex, the leaf is at the rightmost column and the bottom row. Then we can retrace the upper bound for v_0 based on the recursion equation.

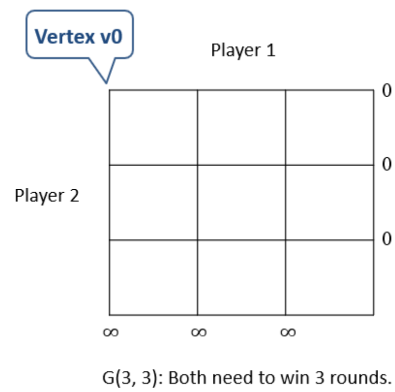

Figure 3: Calculation Upper Bound for G(3,3)

Generally, for the game WnR(n), we can always calculate the upper bound by the matrix method. Although the proof for the existence of upper bound when n is considerably large is quite complicated and barely understandable, we run some experiments indicating that the upper bound for games with $n \leq 5$. The results are shown in the *Result* section.

4 RESULT

4.1 Validation for upper bound

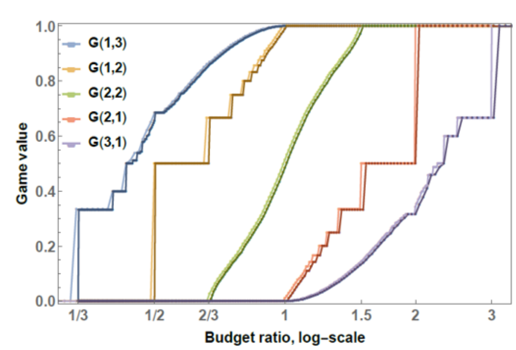

Figure 4: Upper- and lower-bounds on the values of the games G(1, 3), G(1, 2), G(2, 2), G(2, 1), and G(3, 1).

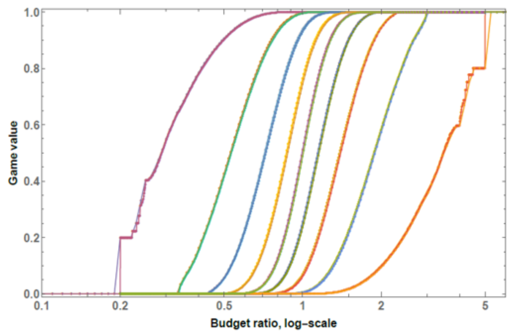

Figure 5: Upper- and lower-bounds on the values of the games G(3, 2), G(3, 3), and G(2, 3).

These two graphs show that with the different winning round, the upper-bound are different. The upper-bound shown in the graphs has been verified and it conforms with the algorithm description in the text.

Since the paper doesn't provided their algorithm to find the winning strategy, we use random function as our testing agent. The following graph shows the winning rate in G(2,2).

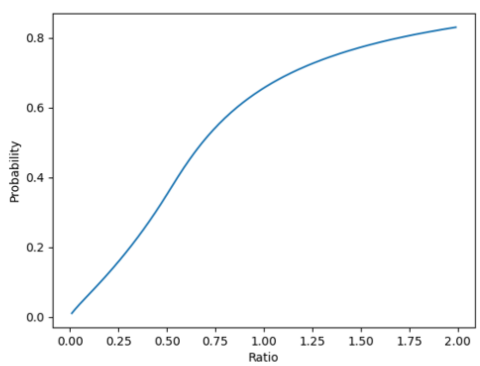

Figure 6: Simulation for G(2,2), Line chart

4.2 Implementation of the game

We use the python library tkinter to implement our all-pay bidding game. Our game provide a optional initial interface where we can change many settings and parameters to help us to process the kind of game test we want. And after the initial setting user interface, we will go to the game process page and input the price we want to place to actually play this game.

In the initial settings interface, we provide several parameter options as showed in Figure 8. At first we need to decide the round of game we need to win, which means the number of rounds of bidding winning needed to win the whole game. Secondly we need to choose whether you want to play with a player or with a pre-coded AI. If you choose play with AI, then there will only have one input field in the game process, otherwise two input field will be

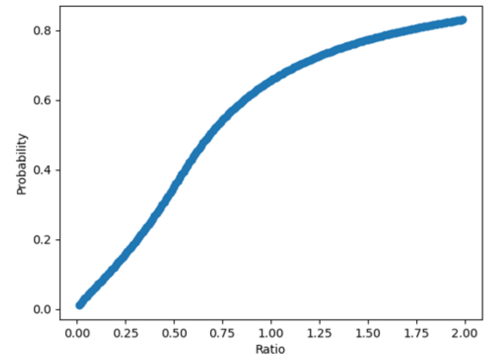

Figure 7: Simulation for G(2,2), Scatter chart

provided. Thirdly, you need to choose the game type which has been described in the previous thesis, and also explained in the interface. Then you need to set the initial money each player have, which helps us to change the initial situation of our experiment and looking for the law of the game. The final option is the kind of AI you want to play with, which will not show up if we choose the Player vs Player mode. In the code we have provided a very convenient API to add more kind of AI method to test, which helps a lot with our experiment.

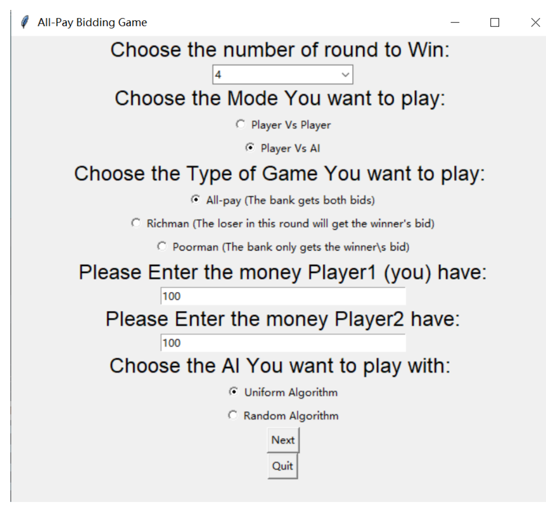

Figure 8: User Interface for initial settings

Then comes the actual game process interface as showed in Figure 9. The above player icon represents player1 and the one below represents player 2 or the AI. If one player wins one rounds, then the icon will go forward for one step, and when one of them reached the rightmost circle, the game ends. Below the canvas is the input filed with the current money and the last round bidding money showed aside. We also provide a AI vs AI mode in the code but not in the user interface, which help us do the experiment thousand of rounds automatically.

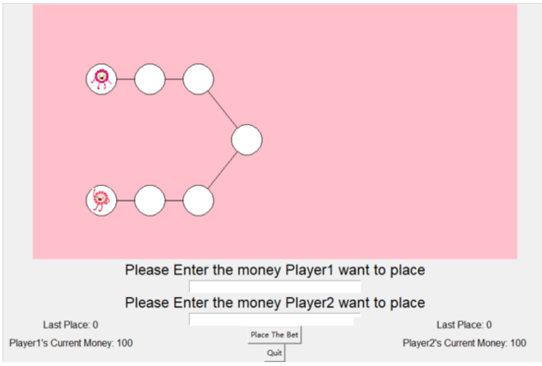

Figure 9: User Interface for game process

REFERENCES

[1] Guy Avni and Thomas A Henzinger. 2020. Bidding Mechanisms in Graph Games. 665385 (2020), 1–11.

[2] Guy Avni, Rasmus Ibsen-Jensen, and Josef Tkadlec. 2020. All-pay bidding games on graphs. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence* 2 (2020), 1798–1805. https://doi.org/10.1609/aaai.v34i02.5546

[3] Soheil Behnezhad, Avrim Blum, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Christos H Papadimitriou, and Saeed Seddighin. 2019. Optimal strategies of blotto games: Beyond convexity. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 597–616.

[4] Richard Bellman. 1969. On "Colonel Blotto" and analogous games. *Siam Review* 11, 1 (1969), 66–68.

[5] John Canny. 1988. Some algebraic and geometric computations in PSPACE. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*. 460–467.

[6] Sergiu Hart. 2008. Discrete Colonel Blotto and General Lotto games. *International Journal of Game Theory* 36, 3 (2008), 441–460. https://EconPapers.repec.org/RePEc:spr:jogath:v:36:y:2008:i:3:p:441-460

[7] H. A. Landman. 1998. Games of No Chance. (1998).

[8] Michael Menz, Justin Wang, and Jiyang Xie. 2015. Discrete all-pay bidding games. *arXiv preprint arXiv:1504.02799* (2015).

[9] Brian Roberson. 2006. The colonel blotto game. *Economic Theory* 29, 1 (2006), 1–24.