Strategy in All-Pay Bidding Games

Michael Menz, Justin Wang, Jiyang Xie

Department of Mathematics, Yale University

Introduction

In all-pay auctions, only one bidder wins but all bidders must pay the auctioneer. All-pay bidding games arise from attaching a similar bidding structure to traditional combinatorial games to determine which player moves next. In contrast to the established theory of single-pay bidding games, optimal play involves bidding from a probability distribution.

Our work has focused on better understanding the structure of these optimal mixed strategies. This poster defines “grounded” and “gap-free” optimal strategies and explains a surprising relationship between opponents’ strategies through a “reverse” operation. Through these results we can implement an algorithm that computes optimal strategies for bidding games.

Definitions

Definition. A **payoff** v_A for player A in $G_{a,b}$ is equal to the probability that player A wins the game under optimal play. A **payoff matrix** M_A for player A is given by $(M_A)_{i,j} = v_A(G'_{a-i+j,b-j+i})$ where G' is the game position after the next move. We let $M_A(k)$ denote the $k \times k$ principal minor of the payoff matrix.

Definition. A **strategy** $S_A(G_{a,b})$ for player A is given by an $(a+1)$-dimensional probability vector where $S_A(i)$ gives the probability that player A will bid i chips. A strategy S is **gap-free** if $S_i, S_j \neq 0 \iff S_k \neq 0 \; \forall i \leq k \leq j$. A strategy S is **grounded** if $S_0 \neq 0$. A strategy S has **length** $l = l(S) \iff S_{l-1} \neq 0$ and $S_m = 0 \; \forall m \geq l$.

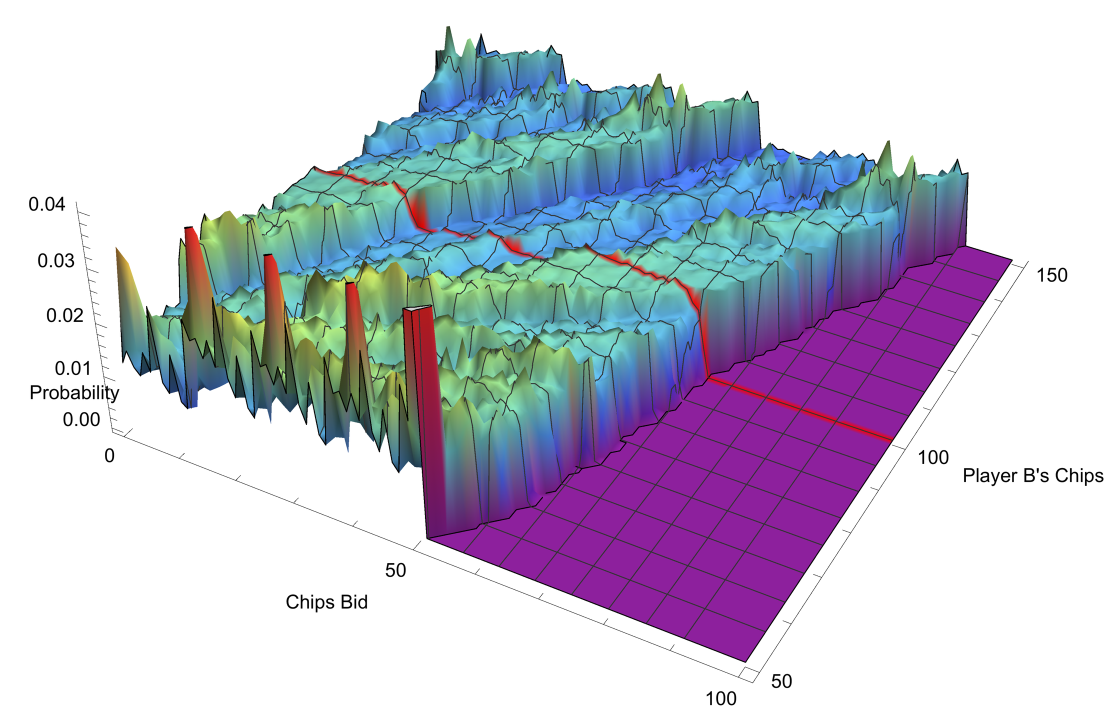
Definition. G is called **precise** if in every successor state to G, it is strictly better to have one more chip.

Definition. A game is in **Nash equilibrium** if both players play strategies such that neither player can further improve his or her payoff.

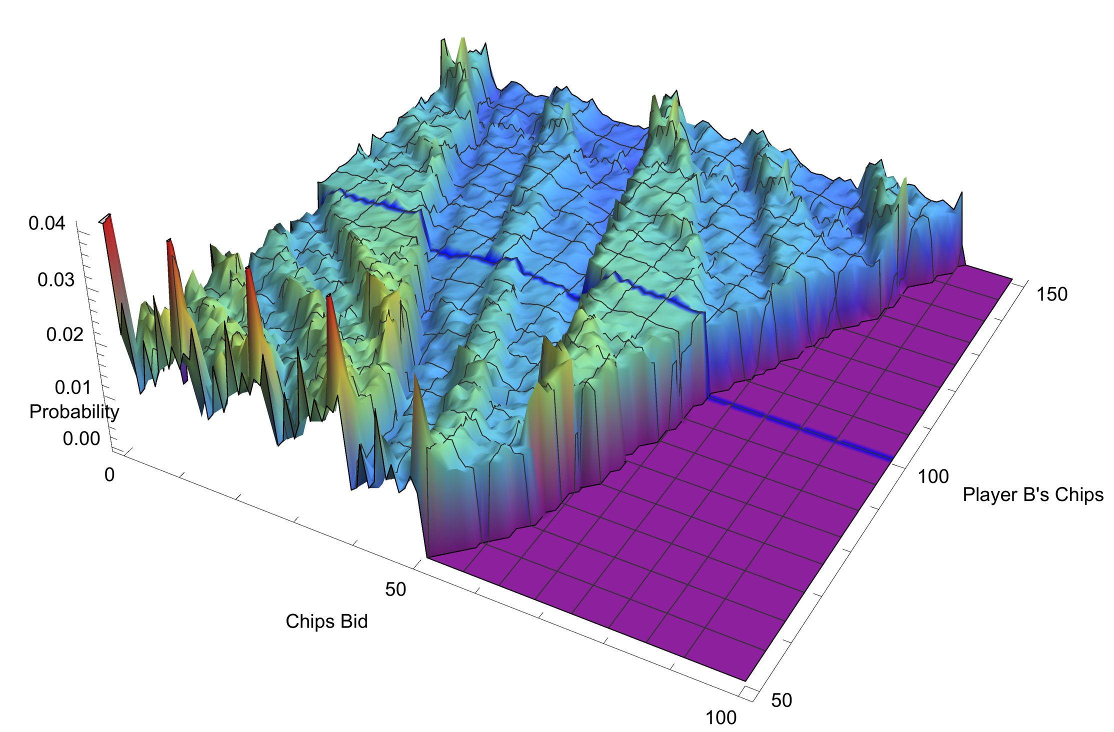
Definition A strategy is called a **Nash equilibrium strategy** if it guarantees a player the payoff he would recive in a Nash equilibrium. Von Neumann and Morganstern proved that in two-player zero-sum games a player’s Nash equilibrium strategies are interchangeable [1].

Our Results

Player A's Best of Three Strategies With 100 Chips



Player B's Best of Three Strategies Against 100 Chips



Lemma 1. *For a Nash equilibrium with opposing strategies S_A and S_B if $(S_A)_i \neq 0$ then $(M_B S_B)_i = v_B$.*

Assume, without loss of generality, that player A has the tie-breaking advantage at a given turn. Let the game in question, $G_{a,b}$, be precise.

Lemma 2. *Player A has a gap-free and grounded optimal strategy. Player B has a gap-free optimal strategy.*

Definition. The **reverse** of a length l strategy S is given by $\mathcal{R}(S) = \mathcal{R}((s_0, s_2, \ldots, s_{l-1}, 0, \ldots, 0)) = (s_{l-1}, s_{l-2}, \ldots, s_0, 0, \ldots, 0)$. where the number of trailing is zeroes will be clear from context.

Reverse Theorem

Let S be a Nash equilibrium strategy for player A. Then $\mathcal{R}(S)$ is a Nash equilibrium strategy for his opponent.

Corollary. *Player A and player B have optimal strategies of the same length.*

Lemma 3. *Let the length of A’s Nash equilibrium strategy of maximum length be l. Then there exists a valid strategy for player A that produces the same payoff against player B’s first k pure strategies if and only if $k \leq l$.*

Our Algorithm

By Lemmas 1 and 2, if S_A has length l, we have $M_A(l) \cdot S_A = v_a \cdot 1_l$. Thus, if we know what l is, we can compute S_A by evaluating $M_A(l)^{-1} \cdot 1_l$ and normalizing the result into a probability vector. Thus, the main problem is to find the length of the equilibrium strategy. Using Lemma 3 above, we can employ a binary search to find the length of a player’s strategy in $\mathcal{O}(\log(n) \cdot n^2)$ time, where n is the player’s chip count.

$$\begin{bmatrix} 1 & .90 & .80 & .67 & .50 & .50 & 0 & 0 & 0 & 0 & 0 \\ .20 & 1 & .90 & .80 & .67 & .50 & .50 & 0 & 0 & 0 & 0 \\ .33 & .20 & 1 & .90 & .80 & .67 & .50 & .50 & 0 & 0 & 0 \\ .50 & .33 & .20 & 1 & .90 & .80 & .67 & .50 & .50 & 0 & 0 \\ .50 & .50 & .33 & .20 & 1 & .90 & .80 & .67 & .50 & .50 & 0 \\ 1 & .50 & .50 & .33 & .20 & 1 & .90 & .80 & .67 & .50 & .50 \\ 1 & 1 & .50 & .50 & .33 & .20 & 1 & .90 & .80 & .67 & .50 \\ 1 & 1 & 1 & .50 & .50 & .33 & .20 & 1 & .90 & .80 & .67 \\ 1 & 1 & 1 & 1 & .50 & .50 & .33 & .20 & 1 & .90 & .80 \\ 1 & 1 & 1 & 1 & 1 & .50 & .50 & .33 & .20 & 1 & .90 \\ 1 & 1 & 1 & 1 & 1 & 1 & .50 & .50 & .33 & .20 & 1 \end{bmatrix}$$

In the payoff matrix above where the player has 10 chips, the algorithm would check $k = 6$, then $k = 9$, then $k = 7$, and finally $k = 8$ before settling on the correct length of 7.

A Sample Bidding Game

Consider the following “Best of Three” game in which the first player to make two moves wins. Each player begins with 100 chips.

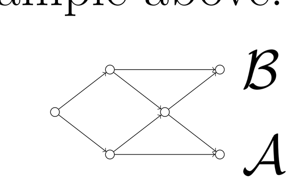
Table 1: A Game of Best of Three

Turn	A	B	A’s bid	B’s bid	Score
1	100	100	20	50	0-1
2	130	70	45	30	1-1
3	115	85	115	85	2-1

Note that updating each player’s chip count only depends on the difference between their bids, not the bid values themselves. In the event of tied bids, the player with more chips gets to move. One player is arbitrarily designated to win ties when chip counts are equal.

Formalizing Bidding Games

We represent games as directed graphs, where vertices are game states and edges are possible moves. The below graph represents the “Best of Three” game from the example above.



$\mathcal{A}$ and $\mathcal{B}$ correspond to victories for players A and B, respectively. We can use recursion starting at $\mathcal{A}$ and $\mathcal{B}$ and moving opposite the arrows to compute optimal strategies at any vertex of the graph. We simply fill the payoff matrix for a vertex by computing the possible payoffs at all of that vertices sucessors. Then we use our algorithm to derive an optimal strategy from that matrix.

References

[1] K. Binmore, Fighting it out. *Playing For Real: A Text on Game Theory*, Oxford University Press, Oxford, 2007.

[2] M. Develin, S. Payne, Discrete bidding games. *The Electronic Journal of Combinatorics*, **7**, 2010.