Strategy Document
Team 11

We initially planned our strategy under the assumption that we will compete against a single opponent simultaneously in an auction. However this assumption was clarified to be wrong one day before the deadline so we were unable to make modifications to adjust for this.

Strategy For Auctions Against A Single Opponent

The main idea is to use information gathered from other auctions played against some player, and use it to predict what their bids in other auctions will be.

Specifically, we had a `max_bid_history` array which stores the highest n^{th} bid by the opponent, for each n.
For each index `i`, we store `max_bid_history[i] =` $max_j\ b_{ij}$ where b_{ij} is the `i`th bid made by the player in the `j`th auction.

For example, if in 3 auctions our opponent makes the bids
[0.10, 0.21],
[0.11, 0.20, 0.30],
[0.09, 0.19],

Then the `max_bid_history` array would be [0.11, 0.21, 0.30]

We then bid (the opponent's expected bid + delta) where delta is a number uniformly sampled between 0.01 and 0.05. If the bid we receive is the maximum for a particular index, we react to the bid and update `max_bid_history.` Otherwise, if we have information about the opponent's next bid from other auctions in the `max_bid_history` array we assume that this will be the opponent's expected bid and outbid that by delta.

It is clear that this works well against trivial bots. For instance, the type of bot that was discussed in the tutorial i.e. bots which only increment their bid by a fixed amount after a fixed time.
We also ran this against every bot from the warmup submission, and it performed very well against them.

Simultaneous Auctions Against More Than One Opponent

This approach is not suitable when auctions against more than one opponent take place simultaneously. This is because the bot is given no information about which player a bid is coming from. Since one player's bids do not help us predict another player's bids, the bot performed much worse than it would have against a single opponent.

As a simple example, if one player always bids higher than another player in the auction, none of the bids from the lower-bidding player will ever be stored in the `max_bid_history` array, so all information about one player is discarded.

To show that our strategy does work well in practice, we ran our bot against all of the bots from the final submission with our original assumption enforced. The results are tabulated here:

Team Number	**Profit**
8	246.41816877002213
11	233.4284356988464
3	227.1496300460128
9	200.91996742451738
2	178.2770291616793
4	177.20147106082825
7	169.70781227041203
5	145.91456948078428
1	145.17717243771
12	142.76086889116837
10	129.75136920921238
16	109.69076963149442
13	109.26279978957542

6	92.426201989374
17	77.83237987395297
15	62.6193728223761
14	37.01623224434877

The bot performed very well finishing 2nd by a close margin to the winner. To adapt this to handle simultaneous auctions with more than one opponent, we would require some sort of identifier as to which player the bid came from. This is a reasonable requirement since we don’t require the identity of the player, only some sort of index to recognize whether two bids belong to the same player.