Designing an Agent for the AGT AdX Competition

Michael Fu, Ryan Lee
CSCI1440 - Algorithmic Game Theory (AGT)
Brown University

1 Introduction

The AGT AdX Competition is a 10-day competition in which participants submit agents to bid for user impressions on a virtual ad exchange. The competition consists of two components: campaign auctions, in which agents bid for new advertising campaigns to fulfill, and impression auctions, in which agents bid to show their ad to each user. For our agent, we used insights from both game theoretic and Walrasian equilibrium models.

2 Initial Intuition + Baseline Heuristic (JohnBot[1])

Our initial approach was to follow simple heuristics that followed our general intuition built up from prior labs.

In our baseline agent JohnBot, our ad bidding strategy was simple; for each campaign in our active campaigns, we first compute the reach factor $\delta \in \{0.3, 0.5, 0.7\}$ to estimate how difficult the campaign is to fulfill. Depending on the reach factor, we modulate the $item\_bid$ to bid more aggressively as the reach proportion grows. We also further adjust the campaign-wide spending limit (budget) based on campaign size and timing. Campaigns with lower reach are more likely to be completed fully, so we scale the budget up by 5–10% to allow for some margin. Campaigns that begin early or mid-game are slightly prioritized by incrementing the bid value and boosting the spending cap, under the intuition that early success builds reputation (via effective reach and quality score) and reduces regret from missed opportunities.

For our campaign bidding strategy, we built it around the intuition that overlapping user segments would lead to internal bidding conflicts and thus modulate our bidding based off of 3 levels of overlap: no overlap, partial overlap, or full overlap. We bid the lowest values (most aggressively) for campaigns with no overlap with our current campaigns, bid higher for those with partial overlap, and don't bid at all for those that fully overlap with our current campaigns.

This simple heuristic gave us a surprisingly solid baseline and was able to consistently profit off Tier 1 (T1) agents. However, JohnBot didn't perform well at all in any of the class auctions (sorry John). To build a more robust agent, we decided to look into the literature and slides attached to the handouts and looked into both Game-Theoretic and Market Equilibrium Models.

3 Game-Theoretic Model

For our basic game-theoretic model, we created a simplified version of a one-day AdX game with the following assumptions:

- *Supply and Demand Assumptions*: Out of the total 10000 users, we know exactly how many of them are from each market segment. Moreover, we know the specifics of other agents' campaigns.

[1] On the leaderboards, we were named "John", but for clarity, we will refer to this bot as JohnBot. Additionally, we'll have our bot names formatted in similar fashion throughout this writeup!

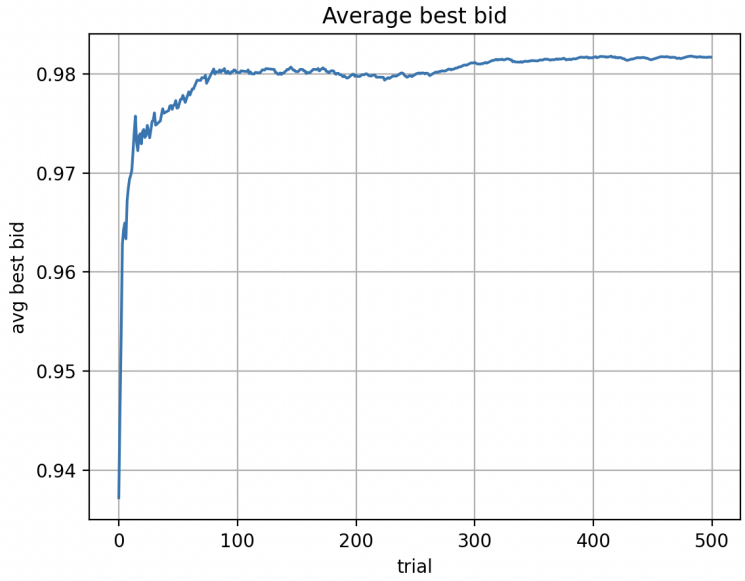


Figure 1: Average best-response bid for the one-day game-theoretic simulation.

- *One Campaign*: All agents get one campaign from a market segment chosen uniformly at random and reach chosen from $\{0.3, 0.5, 0.7\}$, and they bid equally on all subsets of that market segment.
- *Bidding Assumption*: All bot agents i bid a fraction of budget over reach: $\gamma_i \cdot b_i/r_i$, where γ_i is a shading factor chosen uniformly at random in [0.8, 1].
- *Fixed Budget*: All agents set their spending limit for each campaign equal to their budget, which is in turn equal to their campaign's reach.

Given this, we created a simulation of one user agent playing against multiple bot agents. With our simplifying assumptions, the bot bids would all be deterministic, so we could optimize our own agent's best response to these as follows:

- For all $k < n$, where n is the number of total agents, try to "slot in" our agent into kth place in the market segment its campaign needs. That is to say, if the current kth place bidder bids β, then run the simulation and bid $\beta + \epsilon$ for some small ϵ.
- Calculate effective reach ρ given each k, and pick the k and its corresponding bid that maximize ρ.

Using this, we ran the simulation over 500 sets of 100 iterations. For each set, we generated a list of the optimal bids for each of the 100 trials, then turned this into the distribution in which the bot agents pull their new bids from. We then generated new best-response bids, repeating iteratively. As we see in the graph below, the bids somewhat converge to around 0.982. Generally, the optimal bid tends to converge to close to the maximum of the original range of bids (so if bids were originally from 0.6 to 1.1, they converge at around 1.06). This implies that there is high competition among bids. We attempted to extend this game-theoretic model and apply it to an agent setting to submit bids, where at the beginning of each day, we run the simulation as described above to get an optimal bid, but found poor results (i.e. losing to T1 bots). This lack of transferability was likely due to us modeling an oversimplified version of the game.

4 ILP Model

We began our exploration of market equilibrium models with the goal of gaining insights into per-user pricing strategies. Our underlying intuition was that many competitors in the class would likely adopt game-theoretic or heuristic bidding models, so we aimed to investigate whether a welfare-maximizing

approach grounded in economics could offer a principled alternative or at least guide smarter bidding heuristics.

Our Market Equilibrium approach focused specifically on modeling the ad allocation problem as a centralized optimization problem, aiming to compute a Walrasian equilibrium. We modeled the allocation problem as an Integer Linear Program where agents act as buyers and users as goods. Formally, for each agent i and user subset $S \subseteq G$, we set:

$$v_i(S) = \text{campaign\_budget}_i \cdot \min\left(\frac{|S|}{\text{campaign\_reach}_i}, 1\right)$$

and solve the ILP that maximizes total social welfare (as described in the slides). However, we quickly ran into computational challenges, as the powerset of users grows exponentially. As a result, we configured our ILP to a drastically simplified version of the game: a single-day simulation with 10 agents, one market segment, and a segment population of only 10 users. Each agent was assigned a campaign targeting the same segment, and we varied the reach parameter across agents. Under this reduced setting, we were able to solve the ILP and recover equilibrium allocations and prices by solving the corresponding dual LP. Most of the insights we gathered from this oversimplified version lined up with our intuitions (ex. scarcity drove up prices, redundancy in target segments would drive out reach requirements, etc.); this version of market-equilibrium proved largely unfruitful.

5 ILP Model Part 2: Fisher Markets (B99)

To try and find a simplified model that could scale beyond toy instances, we turned to a *Fisher Market* formulation [1, 2]. Fisher markets differ such that each buyer now has a fixed budget and derives utility from consuming divisible goods. We make relaxations that our utility is linear and goods are divisible such that equilibrium can be found through the Eisenberg-Gale (EG) convex program. We assume each campaign as distinct (we later further support this by omitting bidding on overlapping campaign bids). Concretely, we index campaigns by $i = 1, ..., n$, segments by $j = 1, ..., m$, and let x_{ij} denote the number of impressions of segment j allocated to campaign i. Each campaign i has budget B_i, and derives utility $u_i = \sum_j x_{ij}$. The EG program is as follows:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} B_i \ln\left(\sum_{j=1}^{m} x_{ij} + \varepsilon\right) \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} \le s_j \quad \forall j \in \{1, \ldots, m\}, \\
& x_{ij} \ge 0 \quad \forall i, j
\end{aligned}$$

Along with an optimal allocation of impressions x_{ij}, EG produces a set of equilibrium dual variables p_j. In practice, we solve EG every day to translate associated duals into our agent's bids, where for each active campaign i and target segment j, we set the *unit bid* equal to p_j and the bid limit to $p_j * x_{ij}$. We were very surprised with the overall performance of our agent by implementing this bidding function, as we were worried that well-implemented game-theoretic bots would dominate a model-theoretic approach. This formulation gave us most of our gains (shooting our rankings in the group competition from close to last towards somewhere between the 4-8 range (depending on when you check results) and consistently scoring 2300+ profit against the T1 bots. Thus, we decided to use EG as our *ad_bids* and spent the rest of our time hoping to improve our campaign bids.

6 Campaign Bids

For improving our campaign bids, we drew inspiration/intuition from 2 papers, which describe strategies of the top 2 teams in the 2014 TAC-ADX competition (ANL and Giza) [3, 4]. While the TAC-ADX competition differs from our competition (60 day campaign periods, additional UCS auctions, etc.), we hoped to take many of their ideas and test it on our agent(s).

1. **General Heuristics (B99)**

   We began by formalizing a set of assumptions based on our testing with JohnBot and insights from prior TAC AdX papers. First, we chose to avoid bidding on campaigns with overlapping target segments, to minimize internal cannibalization and simplify impression allocation. We then take into account the segment size to estimate the user supply and value.

2. **Greed (Linsanity)**

   ANL introduces an additional parameter called greed, which is a factor describing how aggressive their agent is when bidding for a contract. We attempted to implement our own version of greed with CI. After each day's contract auction we update:

$$\mathrm{CI}_{t+1} = \begin{cases} G \cdot \mathrm{CI}_t, & \text{if we lost any contract on day } t, \\ \frac{\mathrm{CI}_t}{G}, & \text{if we won a contract at second-price on day } t, \\ \mathrm{CI}_t, & \text{if we won at full budget (tie) or no auctions occurred.} \end{cases}$$

   On each new auction day t, for a campaign with reach R and quality score q, we first compute a base bid

$$b_{\text{base}} = \begin{cases} 0.25\,R, & q < 0.9, \\ 0.50\,R, & q \geq 0.9, \end{cases}$$

   and then scale it by our prior multiplier:

$$b_{\text{raw}} = b_{\text{base}} \times \mathrm{CI}_t.$$

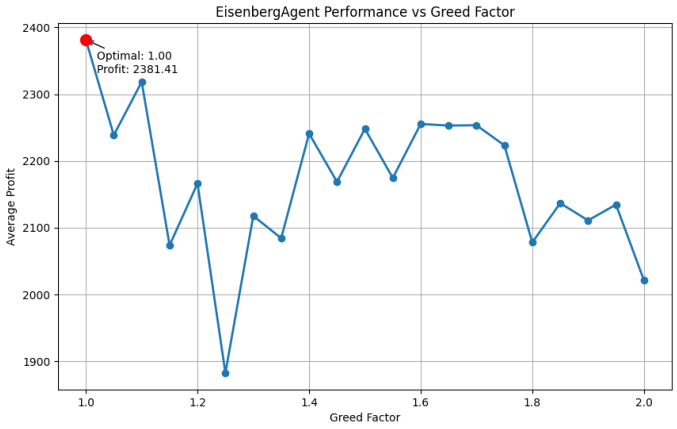
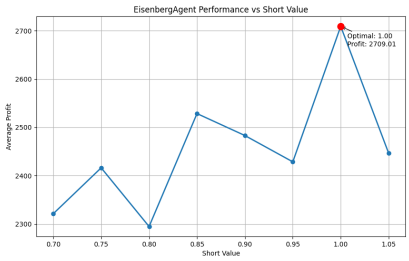


Figure 2: Experimental Results for Greed Factor Campaign Bids

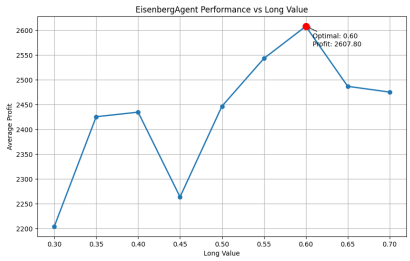
   To test our campaign bid heuristics, we ran experiments where each experimental point is the average profit across 5 experimental runs, where each experimental run is 50 simulations between our agent and 9 other T1 agents. We would follow this setup across future heuristic trials as well. Our experiments greedy experiments yielded poor results, and we decided to scrap the idea.

3. **Short Campaigns -> Long Campaigns (Long)**

   From Giza, we got the idea of minimum bidding on longer campaigns and prioritizing shorter campaigns. Thus, we introduce a *short_factor* that we multiply our *raw_bid* by if a campaign is 2 or less days. Interestingly, our results actually seem to support preferring longer campaigns, and we confirmed these results afterwards. Upon further thought, this intuition makes sense, as longer campaigns equates to more time to fill them, and thus should correspondingly be worth more.



(a) Experimental Results for Short Values



(b) Experimental Results for Long Values

Figure 3: Experimental results comparing short campaign prioritization and dynamic pacing strategies.

4. **Beta (Linsanity 2.0)**
   Although dynamic pacing is often discussed in the context of per-impression auctions, we apply the same principle directly to our campaign bids in attempt to smoothen our spending curve. We update our bid by

$$\mathrm{pm}_t = 1 + \beta\,(1 - \mathrm{pace}_t),$$

   where

$$\mathrm{pace} \;=\; \min\!\Big(1,\; \frac{R}{\mathrm{remaining\_days} \cdot \max\{1,\; \mathrm{daily\_users}\}}\Big)$$

   We observe the following results:

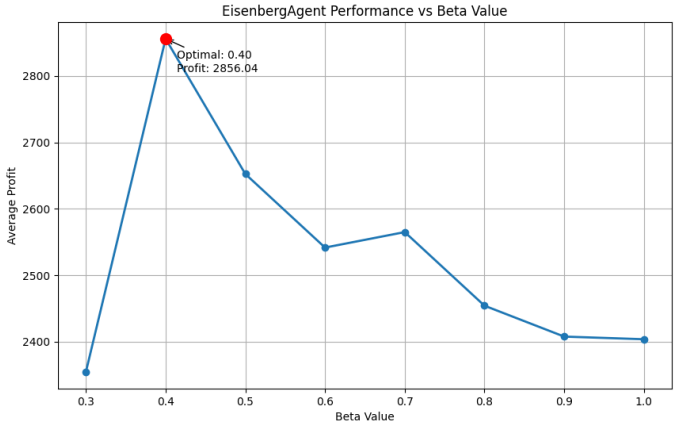


Figure 4: Experimental Results for Beta Factor Campaign Bids

   While we found that our Beta bot worked well against T1 agents, we it didn't work nearly as well as we expected in the class competition (LINSANITY 2.0). This is likely due to overbidding on campaigns or winning on too many campaigns that we aren't able to complete.

7 Concluding Thoughts

Overall, we had a lot of fun attempting to solve this problem through varying approaches and heuristics. One issue we faced (in part due to starting on the later side) was comparing our bot heuristics between each other. While the class competition served as a great way to test the overall robustness of our bots, we found it difficult to compare which heuristic worked better than the other.

In fact, we found a general trend that as we implemented heuristics that performed better against T1, they performed worse in the class competition (maybe due to some type of overfitting). We attempted to compare heuristics (and hybrid combinations of multiple heuristics) by following the similar method of running them against each other across multiple experiments and averaging results, but found that testing locally against bots with similar strategies made it difficult to test for robustness. If we were to do this again, we would focus more on experimentation, as our experimentation led to valuable insights. The Eisenberg-Gale convex program proved to be a surprisingly strong backbone for ad bidding. While some of our campaign bidding heuristics, such as greed and short campaign preference, did not work, they gave us a framework for systematic experimentation.

If we had more time, we would have focused on deeper exploration of hybrid strategies (ex. combining EG-based impression bidding with reinforcement-learned campaign bidding), better benchmarking infrastructure (ex. building out a visualizer), and dynamic adaptation based on campaign fulfillment status mid-game.

References

[1] Simina Brânzei et al. "The Fisher Market Game: Equilibrium and Welfare". In: *Proceedings of the 28th Conference on Artificial Intelligence (AAAI)*. Quebec City, Canada, July 2014. URL: https://projects.iq.harvard.edu/files/yiling/files/the-fisher-market-game-equilibrium-and-welfare.pdf.

[2] Richard Cole et al. *Convex Program Duality, Fisher Markets, and Nash Social Welfare*. arXiv:1609.06654. Sept. 2016. URL: https://arxiv.org/pdf/1609.06654.

[3] Tomer Greenwald. "The AdX Game". Master's thesis. Tel Aviv, Israel: Tel Aviv University, Feb. 2018. URL: https://www.tau.ac.il/~mansour/students/Tomer-Greenwald-MSc-thesis.pdf.

[4] Bingyang Tao, Fan Wu, and Guihai Chen. "TAC AdX'14: Autonomous Agents for Realtime Ad Exchange". In: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*. Ed. by R. H. Bordini et al. Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems, May 2015, pp. 1111–1118. URL: https://www.ifaamas.org/Proceedings/aamas2015/aamas/p1111.pdf.