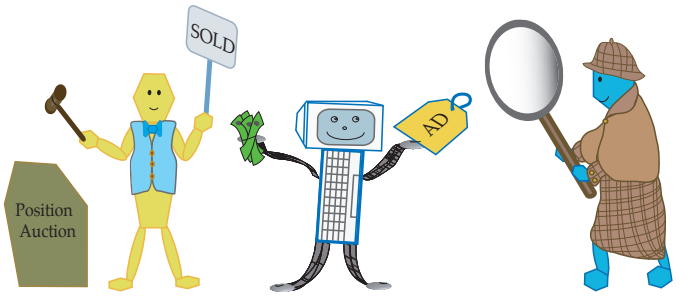

Trading Agent Competition

Ad Auctions

The Ad Auctions Game

for the 2010 Trading Agent Competition

[Specification for Server Version 10.1.0.0]

Patrick R. Jordan, Akshat Kaul, and Michael P. Wellman
Computer Science & Engineering
University of Michigan
Ann Arbor, MI 48109-2121 USA
{prjordan,akaul,wellman}@umich.edu

May 6, 2010

Abstract

We specify the Trading Agent Competition Ad Auction game (TAC/AA), a TAC market game in the domain of sponsored search. Agents play the role of search engine advertisers, who compete with each other on ad placement for search results. This report corresponds to the 2010 version of the game rules.

Contents

1 Background and Motivation

Internet advertising provides a substantial source of revenue for online publishers, amounting to billions of dollars annually. Sponsored search [17] is a popular form of targeted advertising, in which query-specific advertisements are placed alongside organic search-engine results. The placement (position) of an ad for a given query, along with the cost (to the advertiser) per click (CPC), is determined through an auction process. Under cost-per-click pricing, both the publisher and advertiser bear some of the risk associated with uncertain user behavior. The use of automated auctions addresses the combinatorial problem of quoting an appropriate price (CPC) for each display slot for each distinct query. Advertisers bid for the family of keywords of interest, and competition among them determines the going CPC for each of the available slots on a query-by-query basis.

Over the history of sponsored search, a variety of ad auction mechanisms, differing on pricing and ranking rules, have been employed [10]. For example, the CPC for a slot can be set at the price bid by the winner of that slot (called the *generalized first-price* rule[1]), or by the price bid by the winner of the next-best position (*generalized second-price* rule[1]). Similarly, mechanisms may rank by the CPC bid, or adjust these bids by a *quality score* taking into account such factors as the probability the ad will be clicked. For any such mechanism, advertisers face the problem of how to generate bids over time, considering their value for exposure, the competitive environment, and many other factors. Complicating matters, the value an advertiser assigns to an ad may change dynamically, forcing advertisers to balance their current knowledge of the ad markets with a need to explore the bid, keyword, and ad design space.

Given the salience of ad auction mechanisms, a growing body of researchers have started to investigate the *mechanism design* problem faced by search publishers, as well as the strategic problems faced by advertisers. Common to many of these approaches are stylistic restrictions on the scenario or the bidding strategies considered. For instance, analysis may focus on a single keyword auction or on a static, one-shot auction. For a single keyword auction, Varian [18], Edelman et al. [8], and Börgers et al. [4] characterize the equilibrium of the auction using a model based on a static game of complete information. Edelman et al. [8] go on to describe a dynamic model of ad auctions, the *generalized English auction*, where advertisers increase their bids until it is no longer profitable to do so. Aggarwal et al. [3] show that the *rank-by-bid* and *rank-by-revenue* auctions currently in use by publishers are not truthful. The authors construct a truthful variant called the *laddered auction*.

Cary et al. [5] introduce a dynamic single-keyword game where bidders can change their bids over time. The authors demonstrate convergence of heuristic strategies in this setting. Vorobeychik and Reeves [19] empirically analyze a similar set of heuristic strategies in a two-stage game where players condition their base strategy choice on their valuation, which is revealed in the first stage. The authors compare the revenue of the publisher and the advertisers under different strategies and investigate the gains from collusion.

The TAC Ad Auction (TAC/AA) game presents a sponsored-search scenario that,

[1] See Edelman et al. [8] for a discussion of the two pricing rules.

while still simplified in many respects, includes elements not commonly present in work on more stylized environments. The scenario employs a standard ad auction mechanism and a simple yet structured model of a population of search users within a simulated retail market. Developing advertiser strategies for bidding and ad selection in this domain is a challenging problem, beyond the reach of known analytical solutions. Our hope is that by tackling this problem competitively, researchers and practitioners participating in TAC/AA will produce new ideas about bidding strategy for advertising (and in general), as well as insights about sponsored-search mechanisms and ways to improve the model.

The remainder of this document provides a specification of the TAC/AA game. Section 2 presents the new scenario at a high level. Section 3 describes the market model underlying the environment, as well as the form of the queries and ad choices. Competition participants control the *advertiser* agents, the subject of Section 4. The *user* model, introduced in Section 5, drives the queries, clicks, and sales observed in TAC/AA. Section 6 discusses the *publisher*, who runs the ad auctions. Finally, Section 7 describes the basic game flow.

2 Overview of the Ad Auction Game

In designing the TAC/AA scenario, our goal was to create a realistic simulator in which participants can develop strategies that could apply to real sponsored-search auctions. One way to assure realism would be to have the participants develop software for actual sponsored-search interfaces, which was the approach taken by Brendan Kitts, developer of the *Pay Per Click Bidding Agent Competition*[2], held as part of the ACM EC-06 Sponsored Search Workshop. Participants in this competition managed a live Microsoft AdCenter campaign for a given set of keywords over a 24-hour period.

In an effort to support repeatable experimental evaluation of alternative designs, researchers at Yahoo! developed a sponsored-search framework named Cassini [1]. This system simulates low-level query and click behavior, publisher ranking and budget enforcement, and other aspects of the sponsored-search environment.

The TAC/AA scenario is designed to include many of the interesting strategic aspects of sponsored-search auctions, while being repeatable and computationally amenable to empirical analysis. Some important aspects of managing a campaign are left out, such as exploration of a large keyword space for profitable keywords or optimizing landing page content to improve the advertiser's quality score. These are sacrificed not for lack of interest or value, but rather because we lack useful models to represent them. In designing the scenario, we attempted to draw on the sponsored-search literature wherever possible. Nevertheless, we found three general questions not adequately resolved in published work:

- What drives query generation? (Section 5.1)
- How do advertisers derive value? (Section 4.4 and Section 5.4)
- How are keyword auctions independent?

[2] `http://www.biddingagentcompetition.com`

These questions are interrelated. For instance, any nonlinearity in an advertiser's value per click can account for keyword interdependence. Whether values are nonlinear may depend in part on how queries and clicks are generated by users.

In the TAC/AA scenario, advertisers representing retailers of home entertainment products compete for ad placement across a set of related keywords. A game instance represents a simulated ad campaign, comprising a fixed number D of bidding periods, called *days*. Each day, and for each keyword, the N advertisers select between targeted and generic ads, and decide how much to bid for ad placement. Search publishers collect the bids, place ads, and charge advertisers based on the family of ranking algorithms described by Lahaie and Pennock [16]. An evolving population of search users generates queries. The users observe the search results and take actions (click on ads, buy products from advertisers) according to their preferences. Advertisers derive sales profit from the user purchases and the publisher derives revenue from user clicks at the CPC rates determined at auction.

At the beginning of a game instance, advertiser agents connect to the game server and receive initializing information. The server simulates the publisher (Section 6) and the user population (Section 5). At the end of the D-day campaign, agents are evaluated based on their cumulative surplus: sales profit minus cost of advertising.

3 Scenario Elements

3.1 Agents

There are three types of agents in the TAC/AA scenario: *advertisers*, *publishers*, and *users*. The search users and publisher follow fixed (stochastic) policies built into the game environment. The advertiser agents, with the exception of *dummy agents* provided for testing, follow policies implemented by competition entrants. The interactions among agent types are summarized as follows:

Agent	Action
Advertiser	Bids for ad placement
	Selects ads for display
	Receives analytics reports
User	Queries search engine
	Clicks on ads
	Purchases products
Publisher	Runs auction for each user query
	Processes user queries and clicks
	Delivers daily query reports to advertisers

During each simulated day, the behaviors of users, advertisers, and publisher interact to produce search advertising events. A single interaction sequence follows a user as it queries, clicks, and purchases a product. Figure 1 highlights the information flow for such a sequence. Each advertising agent has an ad it selects to appear for a given query. The advertiser also sets its bid for the class of query. The publisher uses the bids and ads from the advertisers to determine slot placement through the use of

an ad auction. When a user submits the query, the auctioneer runs the auction and the results of the slot placement are returned to the user in the form of a ranked list of ads, called an impression. The user views the impression and determines whether or not to click on each of the ads. If the user clicks on an ad, the user is taken to the respective advertiser's landing page. In addition, the advertiser is charged a *cost per click* that was determined by the publisher when the ad auction was run. When a user clicks an advertiser's ad, it determines whether or not to purchase a product from that advertiser. If the user purchases, the event is termed a *conversion*, and the advertiser earns a profit. This process is repeated for each user during every day of the simulation.

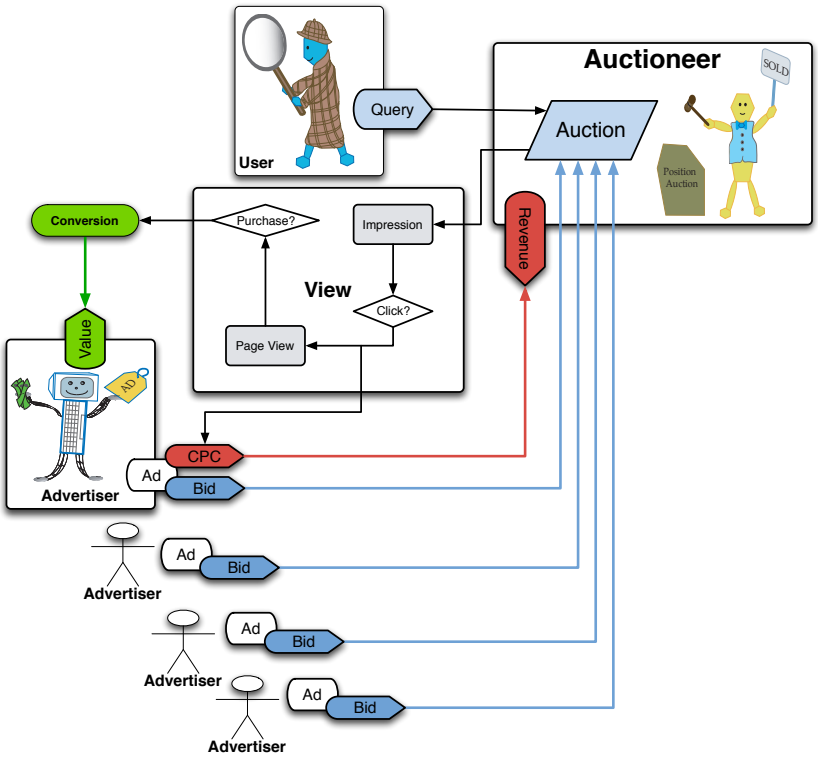

Figure 1: Illustration of the possible chain of activities surrounding a query: the advertisers bid on a keyword, the publisher ranks the ads, the user clicks an ad, views the landing page, and converts its interest to a sale.

3.2 Market Model: Home Entertainment Retail

In the TAC/AA scenario, users search for and potentially purchase components of a home entertainment system illustrated in Figure 2. There are three manufacturers in this market: *Flat*, *Lioneer*, and *PG*. Each of the manufacturers produce televisions *(TV)*, audio systems *(Audio)*, and DVD players *(DVD)*. There are therefore nine distinct

products, specified by *(manufacturer, component)* pair. Advertisers represent retailers who deal in these products. The advertisers use the ad auctions to attract user attention to their offerings, in an attempt to generate sales.

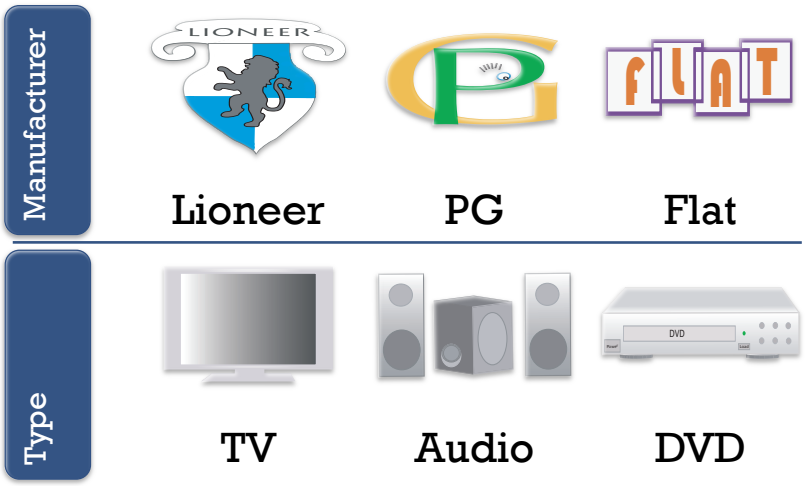

Figure 2: Simple home entertainment market.

3.3 Queries

Each user has an underlying preference for one of the nine products. A user's search behavior depends on its internal state, described in detail in Section 5. At any given time, the population of users is divided into three broad classes: *non-searching*, *searching*, and *transacted*. Non-searching users are currently inactive, generating no queries. The searching users are further divided into *informational* and *shopping* searchers. The informational searchers seek to gather information about their desired product but not to purchase. The shoppers navigate available ads and possibly transact. Shopping users are further divided by levels of search sophistication[3] (focus): low focus (level 0), intermediate (level 1), and high focus (level 2). The transacted users have satisfied their preferences and thus do not search.

A query consists of a collection of words. In our model, we consider only the six words corresponding to manufacturers and components in the home entertainment market. Each query contains at most two of these words: the user's desired manufacturer and component. For instance, a user with preference *(Lioneer, TV)* may generate a query mentioning:

- *Lioneer*
- *TV*

[3]We can also think of these levels as reflecting their degree of knowledge about their own preference.

- both *Lioneer* and *TV*
- neither

Mentioning neither a component nor manufacturer is denoted an F0 level query. Mentioning one or the other, but not both, is denoted an F1 level query. Mentioning both component and manufacturer is denoted an F2 level query. In total, there are 16 distinct queries:

- 1 F0 query
- 6 F1 queries
- 9 F2 queries

A user with given preference will generate one of four queries: two possible F1 queries, and one possibility each at F0 and F2.

3.4 Advertisements

Each advertiser selects an ad for display in each query class, choosing between a *generic* ad, or a *targeted* ad mentioning a particular product. If the user's underlying product preference is the ad shown in the targeted ad, the odds that the user will click on the ad are increased relative to the generic ad (Section 5.3). However, if user's product preference disagrees with the ad shown, the odds of a user click are decreased accordingly.

An advertiser selects the ad to be displayed and the bid amount for that ad at the beginning of each day for each possible query class (Section 4.1). Using those settings, the publisher handles incoming user queries by running ad auctions.

4 Advertisers

Advertisers in sponsored-search auctions typically manage bids over a portfolio of keywords [14]. Optimal bids depend on the advertisers' valuation for placement of their ads in various slots, as well their assessment of the competitive environment. Advertiser strategies are a function of both private and public information. Popular strategies often employ rule-based systems with large rule sets, machine learning, or other techniques from the artificial intelligence and operations research communities [15]. These techniques can respond to dynamic slot valuations and other dynamic inputs. The strategies may directly model competitor behavior, as in opponent modeling, or treat the market environment in aggregate terms.

In the TAC/AA game, entrants play the role of advertisers in the retail home entertainment domain described above. Although every advertiser sells every product, each specializes in a particular manufacturer and a particular component, assigned at the beginning of the game instance. Specialization affects sales profit and click rates as described below. Each day, and for each query class, the advertiser agent selects an ad for display, updates its bid price, and optionally sets spending limits. The agent must

act autonomously, that is, human intervention of any kind is prohibited during a game instance.

At the beginning of each day, each advertiser receives three reports based on events from the prior day:

Source	Report type	Description
Publisher	A daily query report	Section 4.1
Bank	A daily account status report	Section 4.3
Sales Analyst	A daily sales report	Section 4.4

4.1 Bidding

Each day, the advertiser submits a *bid bundle* specifying its CPC offer and ad choice for each query class, to be applied on the subsequent day.[4] For each (*query*,*bid*) pair in the bundle, the publisher updates the advertiser's CPC offer for the keyword *query* to *bid*. A bid price of zero is equivalent to no bid. If there is no (*query*,*bid*) pair for a given query, the advertiser's bid from the previous day persists. Similarly, for each (*query*,*ad*) pair, the publisher updates the ad choice as specified. If not updated explicitly, the previous-day ad is continued, or the generic is chosen if no ad was ever specified. Advertisers may also submit daily *spend limits* for individual queries, as well as a limit for the aggregate spend across queries. When a spend limit is reached, the affected ads will no longer be shown to users for the rest of the current day.

4.2 Query Reports

The daily query report from the publisher includes the following statistics for each query:

- **Ads**: the type of ads displayed by each advertiser for the query
- **Positions**: the average positions (over 10 randomly selected samples) of all of advertisers given by the ad auction for each of the 16 query types
- **CPC**: the advertiser's average cost per click assigned in the ad auction
- **Impressions**: the number of times users viewed a search results page that contained the advertiser's ad
- **Clicks**: the number of times users clicked on the ad

Note that the first two items reveal information about all advertisers, whereas the last three are specific to the advertiser receiving the report.

[4]Note that the bid for day d is due before the day starts, and so the advertiser has not yet received its daily reports for the day. The latest information available to the advertiser therefore reflects market activity on day $d-2$.

4.3 Scoring

The "Bank" keeps a running tally of scores for each advertiser agent. An agent's initial score s is zero, and on each day d it is updated to reflect sales profits and click costs incurred on that day:

$$s_{d+1} \leftarrow s_d + \text{sales\_profits}_d - \text{click\_costs}_d.$$

Each day, the Bank notifies each advertiser of its current score.

4.4 Sales Profits

When an advertiser receives a conversion, the bank credits it with the associated profit. For a sale of a product from a manufacturer that is not the advertiser's specialty, the standard unit sales profit (USP) applies. If the product's manufacturer is the advertiser's specialty, then the advertiser receives $USP(1 + MSB)$ where MSB is the manufacturer specialist bonus.

Every day, the Sales Analyst sends a *sales report* to each advertiser, enumerating sales for each query class.

5 Search Users

The searching, viewing, clicking, and purchasing activity in TAC/AA is generated by a population of M simulated users. Each user has a specific product preference and will only buy the product it has a preference for, thus the overall user population comprises sub-populations for each product. Users are further distinguished by their internal states (Section 5.1), which condition their behavior. We discuss the elements of user behavior in turn: search (Section 5.2), clicking (Section 5.3), and conversion (Section 5.4).

5.1 User State Process

As noted above, the TAC/AA game server maintains a population of search users for each of the nine product types. Within each product sub-population users can transition between the states shown in Figure 3:

- Non-searching (NS)
- Searching
  - Informational search (IS)
  - Shopping, focus level 0 (F0)
  - Shopping, focus level 1 (F1)
  - Shopping, focus level 2 (F2)
- Transacted (T)

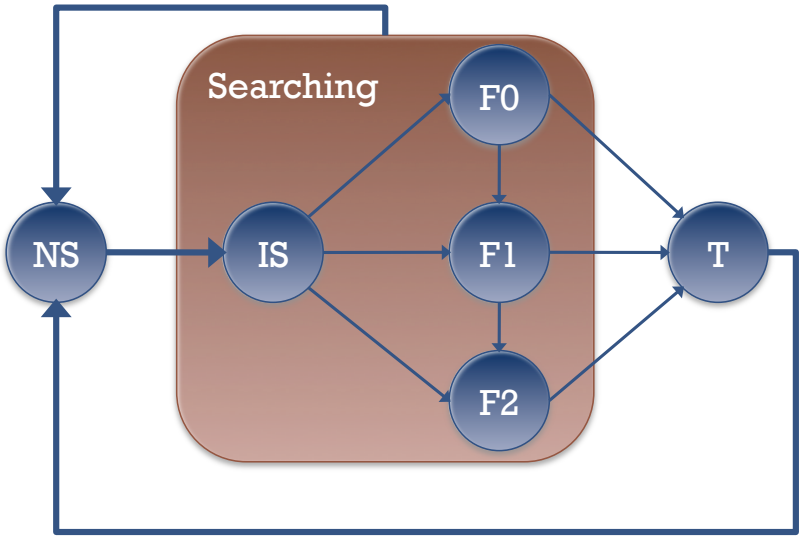

Figure 3: User state transition model. Each state also has an implicit self-loop (not shown).

Figure 3 shows the allowed transitions. The transitions are independently, identically distributed for each user in a specific state, on a given day. All users in a population are initialized to the NS state. The number of initial users for each product type is thus denoted NS_{init}. Users in the *searching* states generate queries by a process discussed in Section 5.2. From the IS state, a user may transition to any of the focused searching states (F0, F1, F2), or remain in the IS state. Users in the focus-level states may transact (transition to state T), stay at their current focus level, increase their focus, or return to the NS state. Once in the transacted state (T), users may transition back to NS or remain in their current state.

Each user sub-population is modeled as a Markov chain. Most transition probabilities are stationary, with the following exceptions. To model bursts of search behavior, we provide stochastic spikes in the $NS \rightarrow IS$ transition. The standard transition probability is given by the parameter $\Pr^{\text{standard}}_{NS \rightarrow IS}$ and the burst transition probability by $\Pr^{\text{burst}}_{NS \rightarrow IS}$. Each day, and independently for each preference type, the users of that type will be subject to the burst transition probability with probability $\Pr_{\text{burst}}$, and the standard transition otherwise. If a burst happens, the burst transition probability for the next BL days will be $\Pr_{\text{sburst}}$. A full specification of transition probabilities is given in Table 3. The transition probabilities from focused search states to state T are also non-stationary, governed by the click and conversion process discussed below.

Before day one of the TAC/AA game instance, the user population undergoes *virtual initialization*, where we simulate D_v days of user transitions. The virtual initialization is performed without advertisers, so there are no impressions, clicks, or conversions.

5.2 Generating Queries

Each user in a searching state generates a single query per day. An F0, F1, or F2 user submits a query pertaining to its level of focus as discussed in Section 3.3. An informational user selects among the three query types uniformly at random. If an F1 query is selected, the informational user selects between the manufacturer and component with equal probability.

5.3 Click Model

Many models have been proposed to model click behavior in users. The functional forms of the models vary, but in essence each model returns the probability that an ad at a given position will be clicked. The probability may be dependent not only on the position, but also on the other ads shown and the position they are in. Edelman et al. [8] use a simple model where each position has an ad-independent click-through effect. This *separability assumption* [3], used either implicitly or explicitly, has been a popular model for analysis [8, 4, 18]. Aggarwal et al. [2] and Kempe et al. [13] develop an alternative click model, called the *cascade model*, where users proceed down the ranked list of ads in a Markovian manner. This model accommodates some dependence between the probability of a user clicking an ad and the other ads on the page, a dependence with appears to be significant [6]. Das et al. [7] propose an extension of the separability model in which the user will convert from at most one of the advertisers. The click model we employ in TAC/AA is a hybrid of the cascade model and the model proposed by Das et al.

Specifically, the click behavior of searching users is modeled by the following parameters:

- an advertiser effect e_q^a for each combination of advertiser a and query class q,
- a targeting effect TE which modifies the probability clicking targeted ads depending on whether the user's preferences match the ad target,
- a promotion bonus modifying the click probability for promoted slots, and
- a continuation probability γ_q for query class q.

Given an impression page for query q, the user proceeds to sequentially view ads, starting from the first position. For a generic ad viewed from advertiser a, the baseline probability that the user clicks is given by e_q^a. This probability can be modified by two factors. First, the *targeting factor*, f_{target}, applies the targeting effect positively or negatively depending on whether the targeted ad selection matches user preference:

$$f_{\text{target}} = \begin{cases} 1 + TE & \text{if targeted ad, matches} \\ 1 & \text{if generic ad} \\ 1/(1 + TE) & \text{if targeted ad, does not match.} \end{cases}$$

Second, the *promotion factor* f_{pro} applies a *promotion slot bonus* PSB if the ad position is a promoted slot. Promoted slots are placed in a premium location on the page (see

Section 6.2), and therefore enjoy an enhanced click rate. For a regular slot, $f_{\text{pro}} = 1$, and for a promoted slot, $f_{\text{pro}} = 1 + PSB$.

The overall click probability starts with the baseline and gets adjusted based on these factors.

$$\Pr(\text{click}) = \eta(e_q^a, f_{\text{target}} f_{\text{pro}}),$$

where

$$\eta(p, x) = \frac{px}{px + (1 - p)}. \tag{1}$$

If the ad is not clicked, or clicked but no purchased is made, then the user will proceed to the next ad with continuation probability γ_q. The parameters e_q^a and γ_q are drawn uniformly other the respective range listed in Table 2, where the focus level of q determines the distribution.

5.4 Conversions

Once an ad has been clicked-through, the shopping users will convert at different rates according to their focus levels. The probability is a function of several parameters. The baseline conversion probability is given by π_l, for $l \in \{\text{F0}, \text{F1}, \text{F2}\}$. Higher focus level queries convert at higher rates: $\pi_{\text{F2}} > \pi_{\text{F1}} > \pi_{\text{F0}}$.

The second factor captures an effect of constrained distribution capacity. The assumption is that if the advertisers sell too much product in a short period, their inventories run short and they have to put items on backorder. As a result, shoppers will be less inclined to purchase, and conversions suffer.[5] All product sales contribute to the distribution constraint, thus rendering the queries interdependent. Let c_d be the total number of conversions over all products on day d, and W the aggregation window for distribution capacity. The distribution constraint effect is given by

$$I_d = \lambda \left(\left(\sum_{i=d}^{d-(W-1)} c_i \right) - C^{cap} \right)^+,$$

where C^{cap} is the critical distribution capacity, beyond which conversion rates decrease. Note that c_d for the current day is the total number of conversions having occurred so far in the day. In our scenario, advertisers are assigned one of three discrete capacity levels: $cap \in \{\text{HIGH}, \text{MED}, \text{LOW}\}$. The number of agents assigned to each level is given by N^{HIGH}, N^{MED}, and N^{LOW}, respectively.

Finally, we consider the effect of component specialization. For users with preference for a component matching the advertiser's specialization, the odds of converting are increased by a component specialization bonus (CSB), using the formula for odds adjustment (1). In sum, the overall expression for conversion probability becomes

$$\Pr(\text{conversion}) = \begin{cases} \eta(\pi_l I_d, 1 + CSB) & \text{if user matches component specialty} \\ \pi_l I_d & \text{otherwise.} \end{cases}$$

[5]The explanation in terms of inventories and backorder is meant to be suggestive. The overall effect is to impose a diminishing marginal value on clicks, and this is just one causal explanation for such an effect.

6 Publisher

For each query a user submits, the publisher must determine which ads are to be shown where and at what price, given the bids of each agent.

6.1 Ranking Ads

Various ranking mechanisms have been employed search publishers, and quite a few more have been proposed and studied by researchers. Analyses by Feng et al. [11] and Lahaie and Pennock [16], for example, use stylized simulations to compare sponsored-search auction rules. A particular question studied by Lahaie and Pennock is the choice between ranking CPC offers directly (*rank-by-bid*), or adjusting these offers by estimated click probabilities (*rank-by-revenue*). They propose a generalized ranking method that interpolates between these extremes using a *squashing parameter* χ. Specifically, if e_q is the estimated click probability for query q (taking into account all available information), and b_q the advertiser's bid, the bid is assigned a score of $(e_q)^\chi b_q$. A setting of $\chi = 0$ is equivalent to rank-by-bid and a setting of $\chi = 1$ is equivalent to rank-by-revenue.

We adopt this ranking mechanism in TAC/AA, using as estimated click probability the baseline value e_q^a. The squashing parameter χ is revealed to advertisers at the beginning of the game instance.

The publisher may impose reserve scores (minimum bid scores) for both regular and promoted slots for each query type (ie. F0, F1 or F2). The number K of slots and the number k of slots available for promotion is revealed to the advertiser agents at the start of the game instance. The winner of a slot eligible for promotion will in fact be promoted only if its score is at least the promotion reserve score $\rho_{\text{pro(FLevel)}}$. Scores below the regular reserve $\rho_{\text{reg(FLevel)}}$ are discarded prior to the ranking process. The instance-specific settings of these reserve scores are not revealed to advertisers.

The publisher determines a ranking of ads for each query each day, based on bids received and current values of click-model parameters. The ranking holds constant throughout the day, unless one of the advertisers reaches a spend limit (see Section 6.3).

6.2 Pricing Clicks

The publisher employs a generalized second-price pricing model. Let $e_{q,(p)}$ be the baseline click probability of the ad in the pth position, and $b_q^{(p)}$ the bid by the advertiser ranked in that position (we take $b_q^{(p)} = e_{q,(p)} = 0$ if there is no such bid). The cost-per-click for position p of the auction for query q is determined by the *effective score* of the next position, which we denote by

$$score_{\mathit{eff}}(p) = \begin{cases} \rho_{(p)} & \text{if } e_{q,(p)}^{\chi} b_q^{(p)} \geq \rho_{(p)} \geq e_{q,(p+1)}^{\chi} b_q^{(p+1)} \\ \\ e_{q,(p+1)}^{\chi} b_q^{(p+1)} & \text{otherwise,} \end{cases}$$

where $\rho_{(p)}$ is $\rho_{\text{pro(FLevel)}}$ if p is a promoted slot and $\rho_{\text{reg(FLevel)}}$ otherwise. The CPC price itself is then given by the minimum this advertiser would have had to bid to beat this

effective score,

$$\frac{score_{eff}(p)}{e^{\chi}_{q,(p)}}.$$

6.3 Enforcing Spend Limits

If an advertiser has specified spend limits for the current day, its ads are monitored and potentially excluded by the publisher. This will occur if the advertiser's click price for an ad on a given query class, when combined with the current spend for the day, exceeds either the corresponding query-specific spend limit or the aggregate spend limit. Upon this condition, the advertiser is removed from consideration for that query and the remaining bids are re-ranked and re-priced for subsequent users that day.

7 Game Interaction

A high level depiction of the game interaction is show in Figure 4. The game flow can be described by considering the game initialization phase and the daily tasks performed by the agents after initialization.

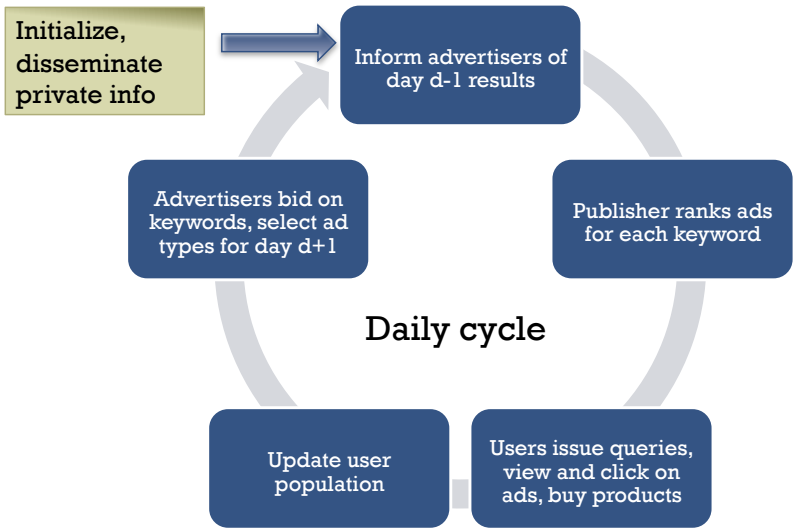

Figure 4: Cycle of activities for day d of a TAC/AA game instance.

Game Initialization At the beginning of a game instance, the instance-varying user, advertiser, and publisher parameter settings are drawn from their associated distributions. The baseline click probabilities e^a_q are set for every query class and advertiser, and the continuation probabilities γ_q for every query class. These are not revealed to the advertisers. All users are initialized to the NS state, and the server simulates D_v

virtual days of user activity without advertising, to spread the population across various states. Advertisers learn their product and manufacturer specialization as well as their distribution capacity parameter C^{cap} (they are not told the specialties and capacities of competitors). Finally, the publisher determines and reveals the squashing parameter χ, and generates hidden reserve scores $\rho_{\text{reg(FLevel)}}$ and $\rho_{\text{pro(FLevel)}}$ for each query type (F0, F1 and F2).

Daily Tasks At the beginning of each day d, the daily reports summarizing day $d-1$ activity are delivered to the advertisers. The publisher executes an ad auction for each query class to determine the ad rankings and click prices. Users then issue queries, receive results, consider clicking on ads and purchasing products. The publisher monitors spend limits and reruns ad auctions as necessary. After all searching users have acted, the server updates the population based on the results of the queries, ads, and purchases. Finally, the advertisers submit their bid and ad selection updates to the publisher, for the auctions determining placement on day $d+1$.

8 Conclusion

This document specifies the 2010 TAC/AA scenario. It reflects relatively incremental changes from the 2009 game rules [12], which can be summarized as follows:

- In 2009, a single global reserve score was applied to all query classes. This year, the reserve scores are determined independently for each query class. The interval from which the reserve scores are drawn has increased significantly.
- Manufacturer specialist bonus has decreased.
- Component specialist bonus has increased.
- Distribution capacity discounter (λ) has increased, leading to a slightly gentler effect on conversion rates.
- Distribution capacities have increased.
- Searching bursts in 2009 were i.i.d.; in 2010 they exhibit a small positive correlation across a short time window.
- The average positions given in the query reports for other advertisers is not exact, but rather based on a small sample average.

The game definition remains subject to change, though at this point we anticipate only incremental modifications and tuning of the parameters listed in Table 1. The game server implementing these rules has been released. The agentware library that exposes the advertiser API is unchanged from 2009. This software and additional documentation and updated versions of this specification can be found at the official TAC/AA web site (see Table 4). The table also identifies the e-mail lists for general discussion and technical support inquiries. We welcome comments, questions, and especially corrections to this document at the support e-mail address. To subscribe

to the general announcement and discussion e-mail list (we anticipate modest traffic), send a blank message with subject `subscribe` to `tac-aa-discuss-request@umich.edu`.

Acknowledgments

We benefited from feedback and suggestions at early design presentations at Yahoo! Research, Google, and Microsoft Research. We are happy to share credit with the broader sponsored-search research community for ideas incorporated, but we take responsibility for decisions about the final game design. Thanks in particular to Maziar Gomrokchi, Chris Kiekintveld, Satinder Singh, Eric Sodomka, Eugene Vorobeychik, and [*your name here*] for comments improving the clarity and correctness of this specification. Implementation of the TAC/AA game server benefited enormously from the platform developed by the Swedish Institute for Computer Science for the TAC Supply Chain Management game [9].

References

[1] Soam Acharya, Prabhakar Krishnamurthy, Ketan Deshpande, Tak Yan, and Chi-Chao Chang. A simulation framework for evaluating designs for sponsored search markets. *WWW-07 Workshop on Sponsored Search Auctions*, 2007.

[2] Gagan Aggarwal, Jon Feldman, S. Muthukrishnan, and Martin Pal. Sponsored search auctions with Markovian users. In *ACM EC-08 Workshop on Ad Auctions*, Chicago, 2008.

[3] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. Truthful auctions for pricing search keywords. In *Seventh ACM Conference on Electronic Commerce*, pages 1–7, Ann Arbor, 2006.

[4] Tilman Börgers, Ingemar J. Cox, Martin Pesendorfer, and Vaclav Petricek. Equilibrium bids in auctions of sponsored links: Theory and evidence. Working paper, September 2007.

[5] Matthew Cary, Aparna Das, Benjamin Edelman, Ioannis Goitis, Kurtis Heimerl, Anna R. Karlin, Claire Mathieu, and Michael Schwartz. Greedy bidding strategies for keyword auctions. In *Eighth ACM Conference on Electronic Commerce*, pages 262–271, San Diego, 2007.

[6] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *First ACM Conference on Web Search and Web Data Mining*, pages 87–94, Stanford, 2008.

[7] Aparna Das, Ioannis Goitis, Anna R. Karlin, and Claire Mathieu. On the effects of competing advertisements in keyword auctions. Working paper, 2008.

[8] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97:242–259, 2007.

[9] Joakim Eriksson, Niclas Finne, and Sverker Janson. Evolution of a supply chain management game for the trading agent competition. *AI Communications*, 19:1–12, 2006.

[10] Daniel C. Fain and Jan O. Pedersen. Sponsored search: A brief history. *Bulletin of the American Society for Information Science and Technology*, 32(2):12–13, 2006.

[11] Juan Feng, Hemant K. Bhargava, and David M. Pennock. Implementing sponsored search in web search engines: Computational evaluation of alternative mechanisms. *INFORMS Journal on Computing*, 19:137–148, 2007.

[12] Patrick R. Jordan, Ben Cassell, Lee F. Callender, and Michael P. Wellman. The ad auctions game for the 2009 trading agent competition. Technical report, University of Michigan, 2009.

[13] David Kempe and Mohammed Mahdian. A cascade model for externalities in sponsored search. In *ACM EC-08 Workshop on Ad Auctions*, Chicago, 2008.

[14] Brendan Kitts and Benjamin Leblanc. Optimal bidding on keyword auctions. *Electronic Markets*, 14(3):186–201, 2004.

[15] Brendan Kitts and Benjamin J. Leblanc. A trading agent and simulator for keyword auctions. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 228–235, 2004.

[16] Sébastien Lahaie and David M. Pennock. Revenue analysis of a family of ranking rules for keyword auctions. In *Eighth ACM Conference on Electronic Commerce*, pages 50–56, San Diego, 2007.

[17] Sébastien Lahaie, David M. Pennock, Amin Saberi, and Rakesh V. Vohra. Sponsored search auctions. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, pages 699–716. Cambridge University Press, 2007.

[18] Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 25:1163–1178, 2007.

[19] Yevgeniy Vorobeychik and Daniel M. Reeves. Equilibrium analysis of dynamic bidding in sponsored search auctions. *International Journal of Electronic Business*, 6(2):172–193, 2008.

Parameter	Symbol	Standard Game Setting
Length of game	D	60 days
Length of day (real time)		10 seconds
Number of advertising agents	N	8 agents
Number of simulated search users	M	90,000 users
Number of users of each preference type	NS_{init}	$M/9$
Probability of searching burst	Pr_{burst}	.10
Probability of successive searching bursts	$\text{Pr}_{\text{sburst}}$	.20
Number of successive days affected by a burst	BL	3
Target effect	TE	0.5
Promoted slot bonus	PSB	0.5
Virtual initialization days	D_v	10 days
Unit sales profit	USP	\$10
Manufacturer specialist bonus	MSB	0.4
Component specialist bonus	CSB	0.6
Distribution capacity discounter	λ	0.996
F0 conversion baseline	π_{F0}	0.11
F1 conversion baseline	π_{F1}	0.23
F2 conversion baseline	π_{F2}	0.36
High capacity threshold	C^{HIGH}	600 units
Medium capacity threshold	C^{MED}	450 units
Low capacity threshold	C^{LOW}	300 units
High capacity agents	N^{HIGH}	2 agents
Medium capacity agents	N^{MED}	4 agents
Low capacity agents	N^{LOW}	2 agents
Capacity window size	W	5 days
Squashing parameter	χ	$0 \leq \chi^4 \leq 1$
Ad slots	K	5 slots
Slots eligible for promotion	k	$0 \leq k \leq 2$
Regular slot reserve score (F0)	ρ_{regF0}	$0.08 \leq \rho_{\text{regF0}} \leq 0.29$
Regular slot reserve score (F1)	ρ_{regF1}	$0.29 \leq \rho_{\text{regF1}} \leq 0.46$
Regular slot reserve score (F2)	ρ_{regF2}	$0.46 \leq \rho_{\text{regF2}} \leq 0.6$
Promoted slot reserve score boost	$Boost$	0.5
Promoted slot reserve score	$\rho_{\text{pro(FLevel)}}$	$\rho_{\text{reg(FLevel)}} \leq \rho_{\text{pro(FLevel)}} \leq (\rho_{\text{reg(FLevel)}} + Boost)$

Table 1: Parameters of the TAC/AA game and their provisional settings. Parameter values listed as ranges are drawn uniformly from the given range.

Factor	Lower Bound	Upper Bound
e_{F0}	0.20	0.30
e_{F1}	0.30	0.40
e_{F2}	0.40	0.50
γ_{F0}	0.20	0.50
γ_{F1}	0.30	0.60
γ_{F2}	0.40	0.70

Table 2: Click Factor Distribution Range

From	To	$\Pr^{\text{standard}}_{\text{From}\rightarrow\text{To}}$	$\Pr^{\text{burst}}_{\text{From}\rightarrow\text{To}}$
NS	NS	0.99	0.80
NS	IS	0.01	0.20
IS	NS	0.05	Same
IS	IS	0.20	Same
IS	F0	0.60	Same
IS	F1	0.10	Same
IS	F2	0.05	Same
F0	NS	0.10	Same
F0	F0	0.70	Same
F0	F1	0.20	Same
F1	NS	0.10	Same
F1	F1	0.70	Same
F1	F2	0.20	Same
F2	NS	0.10	Same
F2	F2	0.90	Same
T	NS	0.80	Same
T	T	0.20	Same

Table 3: Transition Probabilities

Description	Resource
Official TAC/AA web site	`http://aa.tradingagents.org`
Server API	`http://aa.tradingagents.org/software`
Agentware API	`http://aa.tradingagents.org/software`
Discussion e-mail list	`tac-aa-discuss@umich.edu`
Support e-mail list	`tac-aa-support@umich.edu`

Table 4: TAC/AA web resources and mailing lists.