The Cost of Inexpressiveness in Advertisement Auctions

Michael Benisch
School of Computer Science
Carnegie Mellon University
mbenisch@cs.cmu.edu

Norman Sadeh
School of Computer Science
Carnegie Mellon University
sadeh@cs.cmu.edu

Tuomas Sandholm
Computer Science Dept.
Carnegie Mellon University
sandholm@cs.cmu.edu

ABSTRACT

A key trend in (electronic) commerce is a demand for higher levels of expressiveness in the mechanisms that mediate interactions. Recent results indicate the increase in expressiveness tends to lead to increased efficiency. Online advertisement (ad) auctions account for tens of billions of dollars in revenue annually and are some of the fastest growing mechanisms on the Internet. However, the most frequent variant of these mechanisms does not allow bidders (agents) to offer a separate bid for each ad position, and is thus inexpressive on a fundamental level. In this paper we attempt to characterize the cost of this inexpressiveness. We adapt a theoretical framework to show that the commonly used generalized second price (GSP) mechanism is arbitrarily inefficient for some distributions over agent preferences. We then describe a search technique that computes an upper bound on the expected efficiency of the GSP mechanism for a given distribution over agent preferences. We report the results of running our search technique on synthetic preference distributions. Our results demonstrate that the cost of inexpressiveness is most severe when agents have diverse preferences and relatively low profit margins. Our results also show that designating one or more positions as "premium" and soliciting an extra bid for these positions eliminates almost all of the inefficiency.

1. INTRODUCTION

A recent trend in the world, especially in electronic commerce, is a demand for higher levels of expressiveness in the mechanisms that mediate interactions such as the allocation of resources, matching of peers, or elicitation of privacy and security preferences. This trend has already manifested itself in combinatorial auctions, multi-attribute auctions, and generalizations thereof, which are used to trade tens of billions of dollars worth of items annually [9, 19]. It is also reflected in the richness of preference expression offered by businesses as diverse as matchmaking sites, sites like Amazon and Netflix, and services like Google's AdSense. In Web 2.0 parlance, this demand for increasingly diverse offerings is called the Long Tail [2].

Online advertisement (ad) auctions account for tens of billions of dollars in revenue annually and are some of the fastest growing mechanisms on the Internet. The most frequent variant of these auctions, the generalized second price (GSP) mechanism used by *Google*, *Yahoo!* and *MSN*, solicits a single bid from each advertiser (or agent) for a keyword and assigns them to positions according to their bids (with the first position going to the highest bidder, the second position to the second highest, etc.). However, since agents cannot offer a separate bid price for each ad position, the GSP mechanism is inexpressive on a fundamental level. In this paper we attempt to characterize the cost of this inexpressiveness, and to explore the conditions under which it is most severe.

Intuitively, one would think that more expressiveness would lead to higher efficiency (sum of the agents' utilities) of the mechanism's outcome (e.g., due to better matching of supply and demand). Efficiency improvements have indeed been reported from combinatorial and multi-attribute auctions (e.g., [18, 19, 20]). However, until recently we lacked a general way of characterizing the expressiveness of different mechanisms, the impact that it has on the agents' strategies, and thereby ultimately the outcome. Our initial work developed a theory that ties the expressiveness of mechanisms to their efficiency in a domain-independent manner [4]. In that work we derived an upper bound on the expected efficiency of a mechanism that strictly increases with any increase in expressiveness.

We begin this paper with a brief discussion of work related to increasing expressiveness in ad auctions and an overview of the theoretical framework we developed for studying the expressiveness of mechanisms. We then describe how this framework can be adapted to analyze the GSP mechanism. Using this adaptation we show theoretically that for some preference distributions the GSP mechanism is arbitrarily inefficient.

Next, we describe a search technique for computing (or approximating) our upper bound on the expected efficiency of the GSP mechanism for a known distribution over agent preferences. In the worst case our search algorithm takes time that is exponential in the number of agents and types, but it provides an anytime upper bound that is continually tightened.

We conclude with a series of experiments using our search technique on synthetic preference distributions, which illustrate the conditions under which the cost of inexpressiveness in the GSP mechanism is most severe. While we must be careful not to read too much into experiments on synthetic data, they suggest that this mechanism is most inefficient when some agents have a strong preference for top positions (e.g., due to branding concerns), while others prefer middle positions due to higher conversion rates (e.g., when customers coming from middle ranking positions are more likely to result in revenue). We find that the inefficiency is magnified when agents have relatively small profit margins or when the number of agents increases. We also show that in most of the synthetically generated instances the cost of inexpressiveness can be eliminated or significantly reduced by allowing agents to submit a single extra "premium" bid for the top ranking position.

2. RELATED WORK

One of the first applications to benefit from increased expressiveness was strategic sourcing. Sandholm [19] described how building more expressive mechanisms—that generalize both combinatorial auctions and multi-attribute auctions—for supply chains has saved billions of dollars through increased efficiency.

Some work showing similar efficiency benefits due to added expressiveness has also begun to appear in the context of ad auctions (aka sponsored search). Even-Dar *et al.* [10] demonstrated efficiency improvements for sponsored search auctions that allow agents to bid for ads on keywords associated with specific contexts (e.g., the geographical location of the searcher). Boutillier *et al.* [7] and Parkes and Sandholm [14] demonstrated efficiency improvements for more expressive ad auctions, such as those that allow bids on combinations of keywords or across advertising channels. This work suggests that increasing expressiveness can improve the efficiency of ad auctions on a macro scale. In this paper we examine the impact of increased expressiveness on a micro scale.

There has also been some recent work that serves as a counter point to our work and the work described above. A working paper by Paul Milgrom [12] describes how, in some cases, the inexpressiveness in the GSP mechanism serves to eliminate some inefficient equilibria. Our work, on the other hand, implies that in some cases the inexpressiveness eliminates the efficient equilibria as well. Another paper by Abrams *et al.* [1] shows that the GSP mechanism does have an efficient *ex post* equilibrium. However, our work in this paper and our recent theoretical results [4] suggest that this efficiency result *relies* on an assumption that agents have no private information.

Work on expressiveness in general dates back to Mount and Reiter [13] and Hurwicz [11] who asked the question: how many real-valued dimensions must a mechanism's message space have in order to accomplish some design goal? However, to get around Cantor's theorem that implies the equivalence of low and high dimensional expression spaces, they had to rely on certain technical assumptions that precluded a general mapping between $\mathbb{R}^n$ and $\mathbb{R}^m$.

Another thread of related work tries to characterize the equilibrium in inexpressive mechanisms in specific settings (e.g., [16]). The challenge here is that determining equilibrium behavior is usually prohibitively difficult even for the simplest non-trivial mechanisms. Furthermore, when a particular equilibrium is found to have certain properties, one often cannot rule out the possibility of additional equilibria that do not share those properties.

There has been some research related to expressiveness issues in dominant-strategy mechanisms. Blumrosen and Feldman [6] showed a tradeoff between the efficiency of the best possible mechanism and the number of discrete actions available to the designer. Similarly, Ronen [15] described methods for achieving near efficiency with limited bidding languages. The restriction to studying dominant-strategy mechanisms imposes severe limitations on which questions about expressiveness arise. In particular, uncertainty about others' private information becomes an issue only when considering mechanisms that do not have dominant strategies.

3. PRELIMINARIES

The setting we study is a single instance of an auction for a set of k advertising positions that are ranked from 1 to k (rank 1 is the highest rank). In the model there are n agents. Each agent i has some private information (not known by the mechanism or any other agent) denoted by a type, t_i, (e.g., a vector of valuations, one for each of the k positions) from the space of the agent's possible types, T_i.

Settings where each agent has a utility function, $u_i(t_i, O)$, that depends only on its own type and the outcome (matching of advertisers to positions), $O \in \mathcal{O}$ chosen by the mechanism are called *private values* settings. We also discuss more general *interdependent values* settings, where $u_i = u_i(t^n, O)$, i.e., an agent's utility depends on the others' private signals (for example, if one advertiser's value for a position depends on market estimates of the other advertisers). In both settings, agents report expressions to the mechanism, denoted θ_i, based only on their own types. In the GSP mechanism agents report a single real value indicating their bid. A mapping from types to expressions is called a *pure strategy.*

Definition 1 (pure strategy). *A* pure strategy *for an agent i is a mapping, $h_i : T_i \to \Theta_i$, that is, it selects an expression for each of i's types. A* pure strategy profile *is a list of pure strategies, one strategy per agent, i.e., $h_I \equiv \left[h_1, h_2, \ldots, h_{|I|}\right]$. For shorthand, we often refer to h_I as a mapping from types of the agents in I to an expression for each agent, $h_I(t_I) = \left[\theta_1, \theta_2, \ldots, \theta_{|I|}\right]$.*

Based on these expressions the mechanism computes the value of an outcome function, $f(\theta^n)$, which chooses an outcome or assignment from advertisers to positions. In the GSP mechanism the outcome function maps advertisers to positions based on the order of their bids (the highest bidder is assigned the first position, the second highest bidder is assigned the second, and so on). The mechanism may also compute the value of a payment function, $\pi(\theta^n)$, which determines how much each agent must pay or get paid. In the GSP mechanism agents must pay the bid of the advertiser assigned the position directly below them. In this paper we ignore the mechanism's payment function.[1]

For analysis purposes, we assume that the expression of each agent in the mechanism's most efficient Nash equilibrium (i.e., the equilibrium with the greatest sum of agent utilities) can be described by a function that takes as input its type, $m_i(t_i)$. We do not restrict these equilibrium reports to be deterministic pure strategies: we allow m_i to be

[1]Since the efficiency bound that we study does not directly depend on equilibrium behavior this is without loss of generality, as long as agents do not care about *each others'* payments.

a *mixed strategy*, i.e., a random variable specifying a probability distribution over possible reports (in the GSP mechanism this would amount to randomly choosing a bid from some distribution).

For convenience, we will let $W(t^n, o)$ denote the total social welfare of outcome o when agents have private types (or private signals) t^n, $W(t^n, o) = \sum_i u_i(t^n, o)$. Using this formalism we can describe the expected efficiency, $\mathcal{E}(f, \pi)$, of the mechanism for a particular type distribution under its most efficient equilibrium (expectation is taken over the types of the agents, and their randomized equilibrium expressions),

$$E\left[\mathcal{E}(f, \pi)\right] = \int_{t^n} P(t^n) \int_{\theta^n} P(m(t^n) = \theta^n)\ W(t^n, f(\theta^n)).$$

3.1 Our prior work on characterizing mechanism expressiveness

In this section we will discuss the relevant parts of the framework we developed in our prior work for characterizing the expressiveness of mechanisms in general [4].

If we consider mechanisms that allow expressions from the set of multi-dimensional real numbers, such as the GSP mechanism and the *Vickery-Clarke-Groves* (VCG) mechanism (the VCG allows agents to submit real valued bids on each different position individually), one seemingly natural way of characterizing their expressiveness is the dimensionality of the expressions they allow (e.g., this is one difference between the GSP mechanism and the fully expressive VCG for ad auctions). However, the following result illustrates that this notion does not adequately differentiate between expressive and inexpressive mechanisms.

Proposition 1. *For any mechanism that allows multi-dimensional real-valued expressions, (i.e., $\Theta_i \subseteq \mathbb{R}^d$), there exists an* equivalent *mechanism that only allows the expression of one real value (i.e., $\Theta_i = \mathbb{R}$). (This follows immediately from Cantor (1890): being able to losslessly map between the spaces $\mathbb{R}^d$ and $\mathbb{R}$.)*

Thus, it is not the number of real-valued questions that a mechanism can ask that truly characterizes expressiveness, it is how the answers are used!

In order to properly differentiate between expressive and inexpressive mechanisms, we proposed to measure the extent to which an agent can impact the outcome that is chosen.

We define an *impact vector* to capture the impact of a particular expression by an agent under the different possible types of the other agents. (Given a mechanism let the subscript $-i$ refer to to all the agents other than agent i.)

Definition 2 (impact vector). *An* impact vector *for agent i is a function, $g_i : T_{-i} \to \mathcal{O}$. To represent the function as a vector of outcomes, we order the joint types in T_{-i} from 1 to $|T_{-i}|$; then g_i can be represented as $\left[o_1, o_2, \ldots, o_{|T_{-i}|}\right]$.*

We say that agent i can *express* an impact vector if there is some pure strategy profile of the other agents such that one of i's expressions causes each of the outcomes in the impact vector to occur.

Definition 3 (express). *Agent i can* express *an impact vector, g_i, if $\exists h_{-i}, \exists \theta_i, \forall t_{-i}, f(\theta_i, h_{-i}(t_{-i})) = g_i(t_{-i})$.*

We say that agent i can *distinguish* among a set of impact vectors if it can express each of them against the same pure strategy profile of the other agents by changing only its own expression.

Definition 4 (distinguish). *Agent i can* distinguish *between a set of impact vectors, G_i, if*

$$\exists h_{-i},\ \forall g_i \in G_i,\quad \exists \theta_i,\ \forall t_{-i},\quad f(\theta_i, h_{-i}(t_{-i})) = g_i(t_{-i}),$$

when this is the case, we write $D_i(G_i) = \top$.

Figure 1 illustrates how an agent can distinguish between two different impact vectors against a pure strategy profile of the other agents.

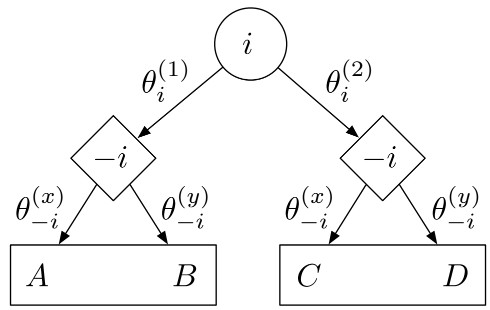

Figure 1: *By choosing between two expressions, $\theta_i^{(1)}$ and $\theta_i^{(2)}$, agent i can distinguish between the impact vectors $[A, B]$ and $[C, D]$ (enclosed in rectangles). The other agents are playing the pure strategy profile $\left[\theta_{-i}^{(x)}, \theta_{-i}^{(y)}\right]$.*

Intuitively, more expressive mechanisms allow agents to distinguish among larger sets of impact vectors. In this paper we will consider a notion of expressiveness, which we call *outcome shattering*, that captures this intuition. Outcome shattering is based on a notion called *shattering*, which we adapted from the field of computational learning theory [21, 5].

Our adaptation captures an agent's ability to distinguish among each of the $|\mathcal{O}'|^{|T_{-i}|}$ impact vectors that include outcomes from a given set $\mathcal{O}'$.

Definition 5 (outcome shattering). *A mechanism allows agent i to shatter a set of outcomes, $\mathcal{O}' \subseteq \mathcal{O}$, if $D_i(G_i^{\mathcal{O}'})$, where $G_i^{\mathcal{O}'} = \left\{g_i \middle| g_i = \left[o_1, o_2, \ldots, o_{|T_{-i}|}\right], o_j \in \mathcal{O}'\right\}$.*

We also use a slightly weaker adaptation of shattering for analyzing the more restricted setting where agents have private values. It captures an agent's ability to cause each of the $\binom{|\mathcal{O}'|+1}{2}$ unordered pairs of outcomes (with replacement) to be chosen for every pair of types of the other agents, but without being able to control the *order* of the outcomes (i.e., which outcome happens for which type). We call this *semi-shattering*.

Definition 6 (outcome semi-shattering). *A mechanism allows agent i to semi-shatter a set of outcomes, $\mathcal{O}'$, if i can distinguish among a set of impact vectors, $G_i^{\mathcal{O}'}$, such that*

$$\forall \{\{x, y\} | x, y \in T_{-i} \wedge x \neq y\},\ \forall o_1, o_2 \in \mathcal{O}',\ \exists g_i \in G_i^{\mathcal{O}'},$$

$$\left[g_i\left(x\right) = o_1\ \wedge\ g_i\left(y\right) = o_2\right]\ \vee\ \left[g_i\left(x\right) = o_2\ \wedge\ g_i\left(y\right) = o_1\right].$$

The following example illustrates the impact an agent must be able to have in order to semi-shatter four different outcomes.

EXAMPLE 1. *If agent i can distinguish among the following set of impact vectors, G_i, then it can semi-shatter a set of outcomes, $\{A, B, C, D\}$, over a set of two different joint types of the other agents, $t_{-i}^{(1)}$ and $t_{-i}^{(2)}$ (note that the order of the pairs that are included does not matter, for example AB could be replaced with BA):*

$$G_i = \left\{ \begin{array}{llll} [A,A], & & & \\ [A,B], & [B,B], & & \\ [A,C], & [B,C], & [C,C], & \\ [A,D], & [B,D], & [C,D], & [D,D] \end{array} \right\}$$

Finally, in our prior work we presented an upper bound on the expected efficiency of a mechanism's *most efficient* equilibrium. We also showed that the upper bound for an optimally designed mechanism is tied directly to its expressiveness.

We derived the bound by making the optimistic assumption that the agents play strategies which, taken together, attempt to maximize social welfare. This allows us to avoid the difficulty involved in calculating equilibrium strategies. It also implies that we can restrict our analysis to *pure* strategies because a pure strategy always exists that achieves at least as much expected efficiency as any mixed strategy.

PROPOSITION 2. *The following quantity, $E\left[\mathcal{E}(f)\right]^+$, is an upper bound on the expected efficiency of the most efficient equilibrium in any mechanism with outcome function f,*

$$E\left[\mathcal{E}(f)\right]^+ = \max_{\hat{h}(\cdot)} \int_{t^n} P(t^n)\ W\Big(t^n, f(\hat{h}(t^n))\Big). \qquad (1)$$

The maximum is taken over $\hat{h}(\cdot)$, a pure strategy profile that maps every joint type to an expression for each agent.[2]

We showed that the bound in Equation 1 strictly increases as the set of outcomes an agent can shatter increases. We also showed that the bound can be arbitrarily inefficient whenever an agent cannot shatter the mechanism's entire outcome space.

4. EXPRESSIVENESS IN AD AUCTIONS

In order to study the expressiveness properties of the GSP mechanism's outcome function we first derive a mathematical representation of the function. Let $R(i, o)$ be the rank of the position given to the i'th agent in the matching of advertisers to positions denoted by outcome o. For analysis purposes we will assume, without loss of generality, that each agent i's bid, θ_i, is restricted to be a real value between 0 and 1 (this is not a limiting assumption based on the same reasoning employed in Proposition 1). Under these assumptions the following is functionally equivalent to the GSP mechanism's outcome function,

$$f(\theta^n) = \arg\max_{o \in O} \sum_{i=1}^{n} \left(\theta_i \times 10^{-R(i,o)}\right) \qquad (2)$$

[2]Recall that an agent's strategy can only depend on its own private type, even if its utility depends on the private signals of others.

This function chooses the outcome that maximizes a weighted sum of the bids. Each bid in the sum is weighted by 10 raised to the negative power of the corresponding agent's rank under the chosen outcome. Thus, agents with higher bids will contribute significantly more to the overall sum when they are placed in the first position.

We will now show that the outcome function of the GSP mechanism is inexpressive according to the notion of outcome semi-shattering that we introduced in the previous section.

THEOREM 1. *Consider a set of outcomes, $\{A, B, C, D\}$, under which agent i is assigned different positions. In the GSP mechanism agent i* cannot semi-shatter *both pairs of outcomes $\{A, B\}$ and $\{C, D\}$ if agents other than i have more than one joint type and,*

$$R(i, A) < R(i, C) < R(i, D) < R(i, B)$$

PROOF. We will assume for contradiction that agent i can semi-shatter both pairs of outcomes, $\{A, B\}$ and $\{C, D\}$. First we introduce the following lemma.

LEMMA 1. *Agent i can (semi-)shatter an outcome space $\mathcal{O}'$ when the agents other than i have more than one joint type only if there exists at least one pair of expressions by the other agents, $\theta_{-i}^{(1)}$ and $\theta_{-i}^{(2)}$, which allows i to (semi-)shatter $\mathcal{O}'$.*

This lemma implies that there must be at least one pair of bids by the agents other than i, $\theta_{-i}^{(1)}$ and $\theta_{-i}^{(2)}$, such that agent i can cause all four outcomes to happen by changing its own bid alone (although we are dealing with semi-shattering so the order in which they happen does not matter).

Let the weighted sum of the bids of the agents other than i for the first (second) profile under outcome A be a_1 (a_2), under outcome B be b_1 (b_2), and so on. Also, let the weights on agent i's bid under outcomes A through D in the the GSP outcome function (Equation 2) be α_A through α_D. (Note that the predicate of our theorem implies that $\alpha_A > \alpha_C > \alpha_D > \alpha_B$.)

Let us assume (without loss of generality) that $b_1 - a_1 < b_2 - a_2$ and that A will happen against $\theta_{-i}^{(1)}$ and B will happen against $\theta_{-i}^{(2)}$ (if the inequality does not hold, we can reverse the labels on the θ_{-i}'s). In order to cause A to happen against the first opponent profile and B against the second the following inequalities must hold (we assume that ties are broken consistently so that an agent cannot use them to semi-shatter),

$$A \text{ happens against } 1 \begin{cases} \alpha_A\theta_i + a_1 > \alpha_B\theta_i + b_1 \\ \alpha_A\theta_i + a_1 > \alpha_C\theta_i + c_1 \\ \alpha_A\theta_i + a_1 > \alpha_D\theta_i + d_1 \end{cases}$$

$$B \text{ happens against } 2 \begin{cases} \alpha_B\theta_i + b_2 > \alpha_A\theta_i + a_2 \\ \alpha_B\theta_i + b_2 > \alpha_C\theta_i + c_2 \\ \alpha_B\theta_i + b_2 > \alpha_D\theta_i + d_2 \end{cases}$$

By simplifying the above equations we derive the following set of constraints.

$$\begin{array}{ccc} \dfrac{c_1 - a_1}{\alpha_A - \alpha_C} & < \theta_i < & \dfrac{b_2 - d_2}{\alpha_D - \alpha_B} \\ \dfrac{d_1 - a_1}{\alpha_A - \alpha_D} & < \theta_i < & \dfrac{b_2 - c_2}{\alpha_C - \alpha_B} \end{array}$$

In order to semi-shatter C and D with C happening against the first set of bids by the other agents and D against the second we have the following inequalities generated in the same fashion,

$$\frac{b_2 - d_2}{\alpha_D - \alpha_B} < \theta_i < \frac{c_1 - a_1}{\alpha_A - \alpha_C}$$

In order to semi-shatter over C and D in the opposite direction (with D first and C second) the constraints would change to the following,

$$\frac{b_2 - c_2}{\alpha_C - \alpha_B} < \theta_i < \frac{d_1 - a_1}{\alpha_A - \alpha_D}$$

Now we can see that our assumption that agent i could semi-shatter both sets of outcomes when the other agents have more than a single type leads to a contradiction. □

This result, in conjunction with our earlier results, implies that under some preference distributions the GSP mechanism is arbitrarily inefficient.

Corollary 1. *For any setting there exists a distribution over agent preferences such that the upper bound on expected efficiency (Equation 1) for the GSP mechanism's outcome function is arbitrarily less than fully efficient.*

Proof. This follows directly from the result above and Theorem 2 in our earlier work [4], which states that the bound in Equation 1 for any mechanism that does not allow agents to shatter (in an interdependent values setting) or semi-shatter (in a private values settings) the entire outcome space is arbitrarily inefficient for some preference distributions. □

5. COMPUTING THE EFFICIENCY BOUND FOR AD AUCTIONS

The results in the previous section prove that there exist distributions over agent preferences for which the GSP mechanism is arbitrarily inefficient. However, in order to measure the inefficiency in practice we must be able to compute (or approximate) the value of the efficiency bound for a particular distribution over agent preferences. In this section we describe a technique for doing just that. Our algorithm takes as input a distribution over agent preferences with a finite number of types (this distribution could be learned from data or approximated by a domain expert) and provides a continually tightening approximation of the upper bound on the mechanism's most efficient equilibrium.

5.1 Mathematical programming formulation

First we will describe the problem using a mathematical programming formulation. The program includes a binary decision variable, z_o^t, for each outcome and each joint type of the agents. A value of 1 for z_o^t denotes that outcome o will be chosen by the mechanism when the agents have the joint type t, a value of 0 indicates that the outcome will not be chosen under t. The program also includes continuous variables representing the agents' bids under each of their types, $\theta_i^{t_i}$ (as in the previous section, we limit these bids to be between 0 and 1 without loss of generality). The following objective function is used to maximize the expected efficiency of the mechanism.

$$\max_{z_o^t, \theta_i^{t_i}} \sum_{t \in T^n} P(t) \sum_{o \in O} z_t^o W(t, o) \tag{3}$$

The first set of constraints enforces that exactly one outcome is chosen for each joint type. There are $|T^n|$ such constraints.

$$\text{s.t.} \quad (\forall t \in T^n) \quad \sum_{o \in O} z_o^t = 1 \tag{4}$$

The next set of constraints ensures that for each z_o^t variable that is set to 1, the agents' bids under type t do indeed cause the outcome function of the GSP mechanism to choose outcome o. This set includes one constraint for each joint type and each pair of distinct outcomes. Thus there are $|T^n| \times (|O|^2 - |O|)$ such inequality constraints.[3] We use M to denote a sufficiently large number such that the sum of all the agents' bids cannot exceed it and $R(i, o)$ to denote the rank of agent i's position under outcome o.

$$\left(\forall t \in T^n, \forall o \in O, \forall o' \neq o \in O\right)$$

$$\sum_i \left(\theta_i^{t_i} 10^{-R(i,o)}\right) > \sum_i \left(\theta_i^{t_i} 10^{-R(i,o')}\right) - (1 - z_o^t) M \tag{5}$$

Finally, we have constraints on the decision variables.

$$(\forall t \in T^n, \forall o \in O)\, z_o^t \in \{0, 1\}, (\forall i, \forall t_i \in T_i)\, 0 \leq \theta_i^{t_i} \leq 1 \tag{6}$$

An ad auction with k positions and n agents with two types each has $\frac{n!}{(n-k)!}$ distinct outcomes and 2^n joint types. The mathematical program described above involves $|O| \times |T^n|$ binary decision variables, making it prohibitively large for general purpose integer program solvers, such as CPLEX, for mechanisms with more than 3 agents. However, these solvers do not explicitly take advantage of certain aspects of the problem structure, for example the fact that only one outcome can be chosen for each joint type. To address this problem we have developed our own search technique, which has so far been successfully used to find provable inefficiency in synthetic instances with up to five agents (although in this paper we report only results with up to four agents due to time constraints).

5.2 Our A* search technique

The skeleton of our technique is an A* search algorithm with each level of the search tree corresponding to a different joint type for all of the agents (for more details about the A* algorithm see Chapter 4 of Russel and Norvig [17]). Each branch in the tree corresponds to the assignment of an outcome to the joint type associated with its source node. The tree has a maximum depth of $|T^n|$ nodes and a branching factor of $|O|$. Figure 2 illustrates the search tree for a small example.

At any node j a partial assignment of outcomes to joint types can be constructed by traversing the edges from j to the root. We will denote the set of all joint types in the partial assignment at node j as T_j^n. For each type $t_j^n \in T_j^n$ we will denote the outcome it is assigned under the partial assignment at node j as o_{t_j}. In addition, for each joint type t^n we will denote the outcome that maximizes social welfare as o_t^* (i.e., $o_t^* = \arg\max_o W(t^n, o)$).

[3]In practice we ensure that these inequality constraints are strict by adding a small ϵ term to one side.

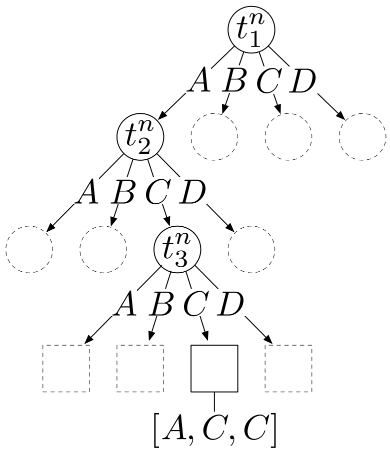

Figure 2: ***This diagram illustrates part of the search tree for a distribution with*** 3 ***types,*** $[t_1^n, t_2^n, t_3^n]$***, and*** 4 ***outcomes*** $[A, B, C, D]$***. Circles represent internal nodes in the tree and squares represent leaf nodes. The dashed nodes are not expanded in this diagram but they would be considered by the search algorithm. The expanded path corresponds to the outcome assignment*** $[A, C, C]$ ***to types*** t_1^n***,*** t_2^n***, and*** t_3^n ***respectively.***

Our A* search orders the nodes on its open queue according to an admissible (or optimistic) heuristic. The heuristic approximates the expected efficiency of the best assignment originating from a particular node based on the assumption that any unassigned types will be assigned optimally. The priority of a node j, $\tilde{f}(j)$, is given by the expected welfare of its current partial assignment plus the expected welfare of the optimal assignment for any unassigned types.

$$\tilde{f}(j) = \sum_{t_j \in T_j^n} P(t_j) W(t_j, o_{t_j}) + \sum_{t \notin T_j^n} P(t) W(t, o_t^*) \qquad (7)$$

The $\tilde{f}(j)$ approximation is guaranteed to be greater than or equal to the true optimal value of any feasible assignment that descends from node j. It may overestimate this value if the optimal assignment is not achievable due to inexpressiveness, but it has the benefit of serving as a valid upper bound on the expected efficiency achievable by the mechanism. The A* algorithm is designed such that any feasible node it visits has a lower $\tilde{f}$ value than any previously visited node. Thus, the $\tilde{f}$ value of the current node is a continually tightening upper bound on the mechanism's expected efficiency, and it can be provided at any time during the search. In our experiments we were occasionally forced to terminate the search early in order to evaluate a greater number of preference distributions. In these cases we reported the $\tilde{f}$ value of the last feasible node that was visited as our upper bound.

Whenever a node is popped off the front of the open queue, its feasibility is checked by solving a linear feasibility problem (LFP). If the node is not feasible its children are not placed on the open queue. The LFP involves a set of constraints similar to those described in Equation 5, however the assignment of outcomes to types is fixed and there are no binary decision variables. Specifically, at any node j we verify that there exist expressions for the agents conditioned on their types, $\theta_i^{t_i}$, which satisfy the following constraints.

$(\forall t_j \in T_j^n, \forall o' \neq o \in O)$

$$\sum_i \left(\theta_i^{t_i} 10^{-R(i, o_{t_j})} \right) > \sum_i \left(\theta_i^{t_i} 10^{-R(i, o')} \right) \qquad (8)$$

6. EMPIRICAL ANALYSIS

In this section we discuss the results of experiments using our A* search technique to compute or approximate the upper bound on expected efficiency for various synthetic preference distributions. Our experiments consist of collections of runs, each involving randomly generated types with different parameter settings. The parameters are chosen to investigate circumstances under which the inexpressiveness of the GSP mechanism is costly (i.e., when the upper bound is low) and when it is not. Each instance in one of our experiments represents a single auction for a single keyword with either three or four agents.[4]

6.1 Experimental setup

In our experiments we assume that each agent draws one of two different types. For each instance we randomly generate types for each agent according to a process described below. We report the average and standard deviation of the efficiency bound for runs of 50 instances each. On occasion our algorithm is unable to find the optimal solution for an instance before a hard-coded timeout of 20 minutes (this occurred around 25% of the time on the four agent instances). In these cases we report the lowest $\tilde{f}$ value discovered prior to termination, which also serves as a valid upper bound on the mechanism's expected efficiency.

6.1.1 *Random type generation*

When we generate types for our experiments we assume that agents only care about the position of their own ad and that an agent's valuation for being assigned a particular position is the expected value of having its ad displayed in that position. We will let "clk" denote the event that the ad was clicked, and "cnv" denote that the click resulted in a conversion (e.g., a sale at a commerce site, or a user registration at a service oriented site). Let C_i denote the amortized cost per click of running agent i's web site, and $V_i(\text{cnv})$ be the expected value of a conversion to agent i. The expected value to agent i of having an ad in position ranked R, $E[V_i(R)]$, is then given by the following equation.

$$E[V_i(R)] = P(\text{clk}|R, i)\, [P(\text{cnv}|\text{clk}, R, i) V_i(\text{cnv}) - C_i] \qquad (9)$$

In order to keep our synthetic data as simple as possible and to isolate factors which contribute to inefficiency in the GSP mechanism, we assume that all agents in the same instance are relatively similar. Many of the values involved in generating an agent's valuations for each position are fixed throughout our experiments. For example, we assume that the marginal cost of a click, $C_i = C = \$1$ for all agents. Unless otherwise specified, we assume that $V_i(\text{cnv}) = V(\text{cnv}) = \$50$ for all agents. We also assume that click-through rates conditional on the rank of an ad's position are the same for all agents. The specific rates are given

[4]We do not include results on instances with two agents because the GSP mechanism is not inefficient when there are only two outcomes.

Parameter	Value	Parameter	Value
$P(\text{clk}\|R=1)$	10%	$C(\text{clk})$	$1
$P(\text{clk}\|R=2)$	7.74%	Brand μ	$\sim$ Uniform[.8, 1]
$P(\text{clk}\|R=3)$	6.66%	Brand σ	25% of μ
$P(\text{clk}\|R=4)$	5.74%	Value μ	$\sim$ Uniform[.4, .6]
$P(\text{cnv}\|\text{clk})$	10%	Value σ	25% of μ
p_B	50%	$V_i(\text{cnv})$	$30 to $150

Table 1: *Default parameter settings for each parameter in our type generation model. Unless otherwise specified, the values of the parameters are those shown in this table.*

in Table 1, along with the default values for all parameters. The click-through rates were taken from an Atlas Institute Digital Marketing pamphlet [8], and are the same as those used by Even-Dar *et al.* in their experiments [10].

Rather than generating arbitrary values of $P(\text{cnv}|\text{clk}, R, i)$, we assume that the probability of a particular position achieving a conversion, $P(R|\text{cnv}, i)$, is normally distributed. The mean, μ, of this distribution is randomly chosen for each type from an interval between 0 and 1. (When working with this distribution, we also normalize the value of R to be between 0 and 1, so that, for example, the third position out of four has rank 0.25). Values of μ close to 1 indicate that the agent's conversion rate is higher in high ranked positions, and values close to 0 indicate that the rate is higher in low ranked positions. The distribution's standard deviation is assumed to be 25% of the mean.

We transform $P(R|\text{cnv}, i)$ into $P(\text{cnv}|\text{clk}, R, i)$ using Bayes' rule (and the observation that the cnv event implies the clk event). We also assume that $P(\text{cnv}|\text{clk}, i) = P(\text{cnv}|\text{clk}) = 10\%$ for all agents.

$$P(\text{cnv}|\text{clk}, R, i) \propto P(R|\text{cnv}, i)P(\text{cnv}|\text{clk}, i) \qquad (10)$$

Based on recent work examining different advertising attitudes on the Internet, we generate two different types for each advertiser in each instance, a *brand* type and a *value* type [3]. The advertisers take on the brand type with probability p_B and the value type with probability $1 - p_b$. The value of μ for the brand type is drawn uniformly between [0.8, 1]. The value of μ for the value type is drawn uniformly between [0.4, 0.6], unless otherwise specified. Figure 3 illustrates prototypical brand and value preferences over different positions based on their rank.

6.1.2 Premium GSP mechanism

We focus most of our attention on the traditional GSP mechanism, where advertisers are charged per click. However, we also report results for a slightly more expressive ad mechanism, which we call a premium generalized second price (PGSP) mechanism.

The PGSP mechanism solicits an additional bid from each agent that determines whether or not it will receive a "premium" ad position (in our experiments the only premium position is the top one, however this could easily be adjusted). The premium positions are assigned as if a traditional GSP mechanism was run on the premium bids (the top premium position goes to the agent with the highest premium bid, etc...). The standard positions are then assigned among the remaining agents according the traditional GSP mechanism run on their standard bids.

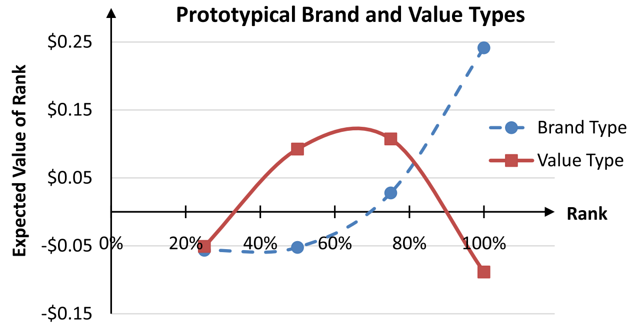

Figure 3: *Examples of valuations from a brand type and value type. The brand type shown has $\mu = 1$ and the value type has $\mu = 0.5$. Valuations are shown in expectation, not per-click.*

6.2 Experimental results

Our first set of runs investigates the impact of the expected value of a conversion on the cost of inexpressiveness in the GSP mechanism. We vary the expected value of a conversion, $V(\text{cnv})$, between $30 and $150 (or 30 to 150 times the cost per click of running the site). The average values and standard deviations of the upper bound on efficiency are shown in Figure 4 for instances with three and four agents. The values are reported in terms of the percentage of the optimal efficiency achievable for each instance.

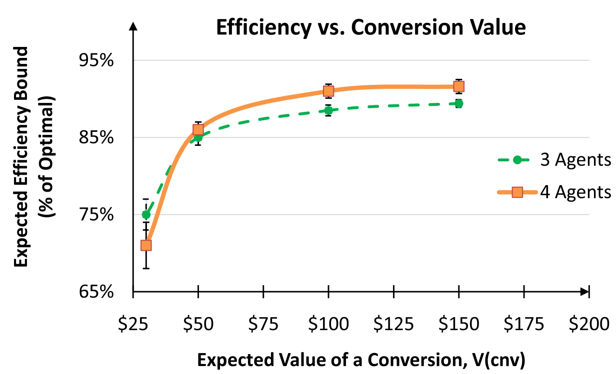

Figure 4: *The value of our upper bound on expected efficiency for the GSP mechanism. Results are averaged over 50 runs for different settings of the expected value of a conversion (all other parameters are assigned their default values).*

The results demonstrate that when conversions generate relatively low profits the cost of inexpressiveness in the GSP mechanism can be more than 30%. However, as the profit margin of the agents increases, the cost of inexpressiveness decreases to around 10%. The instances with three and four agents exhibit relatively similar values for the efficiency bound at each value of $V(\text{cnv})$ when all other parameters are held at their default values. (The slightly higher values of the bound for the four agent instances can be partially explained by the fact that around 25% of these instances were terminated early due to our 20 minute timeout).

Our second set of experiments examines how the cost of inexpressiveness is affected by the position that generates the most value for agents with the value type (the brand type remains unchanged throughout the experiments). In each run the the mean of $P(R|\text{cnv}, i)$ for the value type is drawn uniformly from an interval of size 0.2 (i.e., $\mu \sim \text{Uniform}[a, a+0.2]$). The results are shown in Figure 5. The x-axis indicates the mid-point of the interval used in each run, which is also the the expected value of μ for the value type.

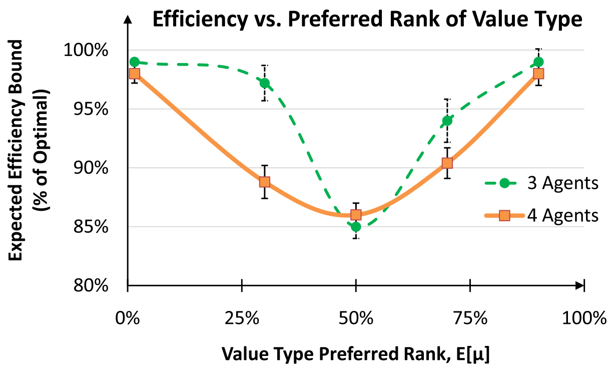

Figure 5: ***The value of our upper bound on expected efficiency for the GSP mechanism. The results are averaged over 50 runs for different settings of the value type's most preferred rank. Larger values of $E[\mu]$ correspond to runs in which higher ranking positions are more valuable for the value type and vice-versa.***

The results demonstrate that the cost of inexpressiveness in the GSP mechanism is highest when the most profitable positions for the value type are the middle ranking positions. The inexpressiveness is less costly when the value type becomes more like the brand type (i.e., as $E[\mu]$ increases) or when the value type is significantly different than the brand type (i.e., low values of $E[\mu]$). We also see that for most value type preferences, the cost of inexpressiveness tends to be more severe when the GSP mechanism is run with four agents than when it is run with three.

Our final set of experiments tests the impact of the added expressiveness in the PGSP mechanism, which solicits a single extra bid from each agent for the top position. For these results we calculate the upper bound of the PGSP mechanism on the same instances as those used to generate the results in Figure 4. The results for the PGSP mechanism are shown in Figure 6.

These results demonstrate that by adding slightly more expressiveness, the cost of inexpressiveness can be completely eliminated in the three agent instances and significantly reduced in the four agent instances. For example, the efficiency bound value for low profit four agent instances improves from 71% in the GSP to 91% in the PGSP.

7. CONCLUSIONS AND FUTURE WORK

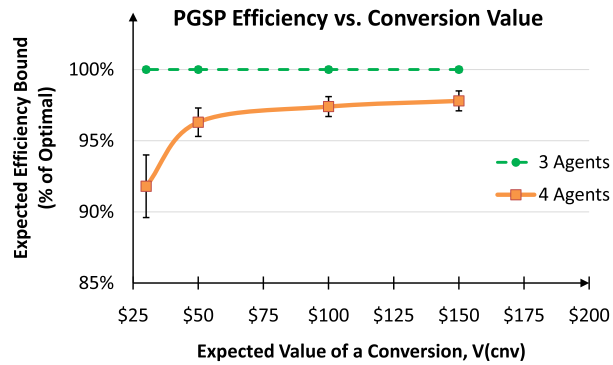

Figure 6: ***The value of our upper bound on expected efficiency for the* premium *GSP mechanism. Results are averaged over 50 runs for different settings of the expected value of a conversion.***

A recent trend in the world, especially in electronic commerce, is a demand for higher levels of expressiveness in mechanisms. Online ad auctions account for tens of billions of dollars in revenue annually and are some of the fastest growing mechanisms on the Internet. The most frequent variant of these auctions, the generalized second price (GSP) mechanism used by *Google*, *Yahoo!* and *MSN*, solicits a single bid from each advertiser (or agent) for a specific keyword and orders the ranking of their ads based on their bids (with the first position going to the highest bidder, the second position to the second highest, etc.). However, since agents cannot offer a separate bid price for each ad position, the GSP mechanism is inexpressive on a fundamental level. We characterized the cost of this inexpressiveness, and explored the conditions under which it is most severe.

We began with a brief discussion of work related to increasing expressiveness in ad auctions and an overview of the theoretical framework we developed in our previous work for studying the expressiveness of mechanisms. We then described how our framework could be adapted to analyze the GSP mechanism. Using this adaptation we were able to show theoretically that for some preference distributions the GSP mechanism is arbitrarily inefficient.

Next, we described a search technique for computing (or approximating) an upper bound on the expected efficiency of the GSP mechanism for a known distribution over agent preferences.

We used our search algorithm to perform a series of experiments on synthetic preference distributions. While we must be careful not to read too much into experiments on synthetic data, we were able to illustrate the conditions under which the cost of inexpressiveness in the GSP mechanism is most severe. Our experiments showed that the cost of inexpressiveness in the GSP mechanism is greatest when some agents have a strong preference for top ranking positions (e.g., due to branding concerns), while others prefer middle ranking positions due to higher conversion rates. Additionally, we found that the cost is magnified when agents have relatively small profit margins or when the number of agents increases. We also showed that in most of the synthetically generated instances the cost of inexpressiveness can be significantly reduced by allowing agents to submit a single extra “premium” bid for the top ranking position.

There are many opportunities to extend the work described in this paper. One future direction involves investigating the counter intuitive findings that increasing the number of agents in the mechanism leads to greater inefficiency (typically in economic systems the opposite is true). With more computing resources we can extend our current experiments to include instances with five agents. In addition, we believe that we may be able to improve the speed of our A* search technique, allowing it to handle instances with six or more agents.

Another obvious future direction involves applying our methodology to actual, rather than synthetic, preference data. We believe that this can be accomplished by learning conversion rate distributions conditioned on the rank of an ad’s position, $P(\text{cnv}|R)$, from real conversion data. The other parameters in our type generation model can be approximated or varied to get a sense of whether or not the inexpressiveness of the GSP mechanism has a high cost in practice.

8. ACKNOWLEDGEMENT

This work has been supported by the National Science Foundation under grants IGERT-9972762, ITR-0205435, IIS-0427858, and CNS-0627513.

9. REFERENCES

[1] Z. Abrams, A. Ghosh, and E. Vee. Cost of conciseness in sponsored search auctions. In *Proceedings of Workshop on Internet Economics (WINE)*, 2007.

[2] C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, July 2006.

[3] M. Baye and J. Morgan. Brand and price advertising in online markets. Technical Report 0504009, EconWPA, 2005.

[4] M. Benisch, N. Sadeh, and T. Sandholm. A theory of expressiveness in mechanisms. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2008. Longer version availabe as Technical Report CMU-CS-07-178, School of Computer Science, Carnegie Mellon University, 2007.

[5] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):865–929, 1989.

[6] L. Blumrosen and M. Feldman. Implementation with a bounded action space. In *Proceedings of ACM Conference on Electronic Commerece (ACM-EC)*, 2006.

[7] C. Boutilier, D. Parkes, T. Sandholm, and W. Walsh. Expressive banner ad auctions and model-based online optimization for clearing. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2008.

[8] N. Brooks. The atlas rank report: How search engine rank impacts traffic. Available at `http://atlassolutions.com/pdf/RankReport.pdf` in May 2008.

[9] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[10] E. Even-Dar, M. Kearns, and J. Wortman. Sponsored search with contexts. In *Proceedings of Workshop on Internet Economics (WINE)*, 2007.

[11] L. Hurwicz. On informationally decentralized systems. In C. McGuire and R. Radner, editors, *Decision and Organization*. Amsterdam: North Holland, 1972.

[12] P. Milgrom. Simplified mechanisms with applications to sponsored search and package auctions. Working paper, 2007.

[13] K. Mount and S. Reiter. The informational size of message spaces. *Journal of Economic Theory*, 8(2):161–192, 1974.

[14] D. Parkes and T. Sandholm. Optimize-and-dispatch architecture for expressive ad auctions. In *First Workshop on Sponsored Search Auctions, at the ACM Conference on Electronic Commerce*, Vancouver, BC, Canada, June 2005.

[15] A. Ronen. Mechanism design with incomplete languages. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 105–114, 2001.

[16] R. W. Rosenthal and R. Wang. Simultaneous auctions with synergies and common values. *Games and Economic Behavior*, 17:32–55, 1996.

[17] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition, 2003.

[18] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, Jan. 2002.

[19] T. Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Mag.*, 28(3):45–58, 2007.

[20] T. Sandholm and S. Suri. Side constraints and non-price attributes in markets. In *IJCAI-2001 Workshop on Distributed Constraint Reasoning*, 2001. Short version appeared in *Games and Economic Behavior* 55: 321-330, 2006.

[21] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.