On Expressing Value Externalities in Position Auctions

Florin Constantin
Georgia Institute of Technology
Atlanta, GA, USA
florin@cc.gatech.edu

Malvika Rao
SEAS, Harvard University
Cambridge, MA, USA
malvika@eecs.harvard.edu

Chien-Chung Huang
Max-Planck-Institut für Informatik
Saarbrücken, Germany
villars@mpi-inf.mpg.de

David C. Parkes
SEAS, Harvard University
Cambridge, MA, USA
parkes@eecs.harvard.edu

ABSTRACT

Externalities are recognized to exist in the sponsored search market, where two co-located ads compete for user attention. Existing work focuses on the effect of another ad on the *quantity* of clicks received. We focus instead on the negative effect of another ad on the *value* per click, and propose a general model of externalities, in which a bidder has no value for a slot under a set of certain conditions, each on one other bidder's allocated slot. We provide a generic greedy algorithm for the winner determination problem (WDP) in this model together with a pricing scheme that closely follow the Generalized Second Price (GSP) auction used in practice. For value externalities that satisfy a property of *downward-monotonicity*, these mechanisms provide no new opportunities for manipulation beyond the ones already available via untruthful claims about bid value in GSP under the standard slot auction model. Our main instantiation of downward-monotonic constraints is an *identity-specific* language, in which a bidder can require that it precedes some subset of other bidders. For this language's WDP, we establish worst-case complexity and inapproximability results. This motivates the choice of approximations, e.g. via the greedy algorithm. As another way of circumventing the hardness results, we present fixed-parameter algorithms for the WDPs of two sub-languages of the identity-specific model.

1. INTRODUCTION

Internet advertisers compete for ad slots on a search results webpage, e.g. triggered by queries related to sports shoes. The advertisers only pay for a click and are naturally interested in attracting user clicks that lead to profits, e.g. via a subsequent purchase by the user. It is well-known and backed by consumer data (e.g. [7]) that ads placed higher on a webpage are considered more seriously by users. It is also widely accepted that the distribution of clicks that an ad attracts is dependent on the other ads shown, e.g. via their number [27, 32], or via their relative position [15].

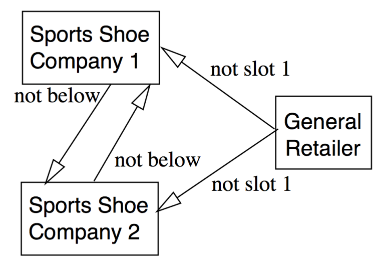

Figure 1: Each sports shoe company conditions its value on not being displayed below the other, fearing that only customers looking for bargains will click on the lower one as they perceive the higher placed one as better. The general retailer conditions its value on neither of the sports shoe companies being placed in slot 1, fearing that otherwise the clicks it receives will only be out of curiosity, without an intent of purchase.

We are especially interested in the effect of co-located ads on the *value* of an advertiser for a click (or impression), i.e. on the value conditioned on receiving a click. This effect is in addition to the effect that another ad can have on the *quantity* of clicks or impressions due to competition with other ads for user attention. That is, not only does an advertiser stand to lose user attention if appearing below a competitor's ad because people may be satisfied with the competitor's product and not visit the advertiser, but the attention that it does receive may be only from those users that were not interested in purchasing shoes in any case (see Fig. 1). Whereas a quantity effect, for example with the specific set of advertisers allocated affecting the number of clicks received by each ad, can be learned by a search engine because the data is observable, value externality is private to a bidder and needs to be expressed through a bidding language. It is natural to consider the implications of allowing an advertiser to submit along with its bid additional constraints to state how its value for impressions or clicks depends on the (relative or absolute) positions of the other ads with which it is allocated.

Our general framework for value externalities is provided by the *unit-bidder constraints* model, and allows a bidder to specify a set of constraints, each one forbidding the simultaneous allocation of a specific slot for itself and a specific slot for one competitor. Note that this is a constraint that binds only in the event that a bidder is allocated a slot: it does not force the auctioneer to select a particular allocation, but instead precludes allocations.

Our main instantiation of the unit-bidder constraints model is the *identity-specific* model, in which a bidder B can submit a bid to express a value (e.g., per impression or per click) and also, optionally, a set of (hard) directed constraints. Each constraint identifies another advertiser B' and specifies that B has full value if it receives a higher slot than B' (in particular if B' is not allocated) and 0 value otherwise. For example, each sports shoe company in Fig. 1 has such a directed constraint towards the other. We immediately see that at most one of the two companies can be allocated for the given query. The corresponding winner determination problem (WDP) is to allocate a set of ads (bidders) to slots such that no constraints are violated and the total value of the allocation is maximized. The WDP for the identity-specific model is NP-hard even for the special case of constraint graphs with maximum in-degree and out-degree of 2 (i.e., with at most two constraints per bidder and no bidder being targeted by three or more others). The problem is also computationally hard to approximate, under a plausible complexity assumption, by a reduction from an inapproximability result for IndependentSet on bounded-degree graphs.

In seeking approximation algorithms, a natural choice is a greedy algorithm, extending the current practice in slot auctions. We provide such a generic algorithm for unit-bidder constraints and establish its approximation ratio for the identity-specific model. We identify a subspace of unit-bidder constraints that insists on *downward-monotonic* constraints and contains identity-specific constraints. For this family of downward-monotonic constraints, we demonstrate that a payment rule analogous to the Generalized Second Price (GSP) rule from current practice precludes any new manipulations due to the ability to misreport constraints. That is, a bidder with downward-monotonic value externalities has no incentive to misreport its set of constraints by falsely claiming an additional constraint or not reporting an existing one. The downward-monotonic property requires that a bidder who is dissatisfied with a particular slot given an allocation to other bidders is also dissatisfied with any lower slot. Fig. 2 illustrates the inclusion relationships among the classes of constraints we consider. Another downward-monotonic class that we consider is the *slot-specific* model, generalizing the "bid-to-the-top" model of Aggarwal et al. [3]. A bidder may specify that it must be in the top k slots and that certain bidder(s) cannot be in the top k' slots.

The aforementioned hardness results alternatively suggest further narrowing the model in search of tractable algorithms. We offer two such sub-languages of the identity-specific model. In the *category-specific* sub-language, bidders are divided into different categories for a given query (e.g. "cleats", a certain type of sports shoe), for example into bidders that insist on an exact match on "cleats" and those who are willing to advertise against a broad query match to "sports shoes." Allocation constraints can be specified only within the same category, so that a bidder that places the constraint must receive a slot higher than any other bidder in the category. For one category only, this model is a special case of the exclusivity model of Ghosh and Sayedi [12]. We present an algorithm solving the WDP that scales exponentially with the number of distinct categories (thus polynomial for a fixed number of categories). A second sub-language that we consider is that of *symmetric identity-specific* constraints, i.e. bidders that exclude each other, like the two sports shoe companies in Fig. 1. We provide a dynamic programming algorithm for the WDP with complexity exponential in the tree-width of the constraint graph, a standard algorithmic measure of a problem's locality structure.

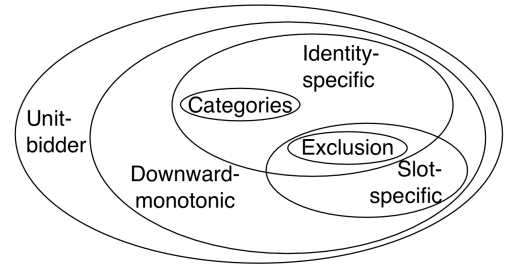

Figure 2: Classes of value-externality constraints.

In the Appendix we discuss a relationship between the identity-specific model and a classic problem of scheduling to minimize weighted, discounted completion time given precedence constraints, which provides tractable algorithms for a very restrictive special case of our problem. We hope this discussion will provide a springboard for future work.

1.1 Previous Work

In recent years ad auctions have formed an active area of research [8, 34, 24]. Relatively little studied, however, is the problem of externalities in auctions. Aggarwal *et al.* [1] and Kempe and Mahdian [21] describe cascade models of quantity externalities. These models associate with each ad a click-through-rate as well as a continuation probability representing the probability that a user continues the search after viewing the given ad. The authors solve the winner determination problem in their models. Giotis and Karlin [13] analyze the equilibria of the cascade model in GSP auctions. Gomes *et al.* [15] consider the role of information and position externalities in the ordered search model, where users are assumed to browse ads from top to bottom and take clicking decisions slot by slot. They examine user data to estimate their model and study the Nash equilibria of GSP under different scoring rules. Athey and Ellison [5] propose a model where users search in a top-down manner and clicking is costly. They compute and analyze the resulting equilibria. Aggarwal *et al.* [3] study *prefix position auctions* where an advertiser can specify that he is interested in only the top k positions. The authors present an allocation and pricing mechanism and show the existence of envy-free Nash equilibria.

Lately, a few papers have begun to include the effect of value externalities in their model. Muthukrishnan [27] considers an auction where advertisers bid on the eventual maximum number (called the *configuration*) of ads shown. The paper presents a WDP algorithm and a critical value pricing scheme. Ghosh and Sayedi [12] design extensions of VCG and GSP for sponsored search auctions where the user is allowed to submit two bids: one for being placed alone and another for being placed alongside multiple other ads. Equilibria of these mechanisms are investigated and the tradeoff between revenue and efficiency properties of the mechanisms are compared. Ghosh and Mahdian [11] study a model in which the value to an advertiser depends on the relative quality compared to other co-located ads, as induced by different probabilistic models of user types. The authors mention the direction of location-dependent externalities but study externalities that depend only on the set of ads allocated. Sponsored listings appearing just above the organic search results are referred to as "north" ads [32]. Reiley *et al.* [32] conduct experiments to measure the externalities imposed by additional north listings on existing ones. Surprisingly, these additional north listings appear to impose a positive, rather than a negative, externality on existing north listings. The authors propose interesting hypotheses to explain this phenomenon. Krysta *et al.* [23] present a formal model for combinatorial auctions with externalities and show that the winner determination problem can be solved in polynomial time with a small number of queries to an NP-oracle. Within economics, there is a literature on informational as well as allocative externalities [19, 20] but without a focus on computational or representation issues.

Paper structure. We set forth notation in Sec. 2. We introduce our general constraints model and present a generic greedy algorithm in Sec. 3. We present the resilience of this algorithm to downward-monotonic constraint manipulations and other incentive considerations in Sec. 4. Our main instantiation of the downward-monotonic constraints model, the identity-specific model, is defined in Sec. 5 which also provides computational hardness results for it. Sec. 6 presents fixed-parameter algorithms for two sub-classes of the identity-specific model.

2. PRELIMINARIES

We start by formalizing our model and introducing notation. Let N denote a set of bidders $\{1, \ldots, n\}$ in a position auction with m slots. Since each bidder is interested in exactly one slot, we assume $m \leq n$. We assume that the click-through rate falls off according to discount factor $\delta \in (0, 1)$, and to keep things simple we normalize the first slot's click-through rate to 1. Thus slots $1, 2, \ldots, m$ have click-through rates $1, \delta, \ldots, \delta^{m-1}$. Each bidder i is associated with a per-click value $v_i \geq 0$ and a constraint set $C_i \subseteq N \setminus \{i\}$.[1] Constraint set C_i imposes conditions on the slots allocated to bidder i and other bidders: bidder i has value v_i if all conditions in C_i hold, and value 0 otherwise. A bidder can make a claim about its bid value b_i and its constraint set C'_i. The semantics of a bid are that the bidder is willing to pay b_i per click as long as the allocation satisfies its reported constraint set C'_i, and zero otherwise. Given the discounted click-through rate model, bidder i's willingness-to-pay, in expectation, for an allocation to slot j is $b_i\delta^{j-1}$, given that constraints C_i are satisfied.

[1]Because the click-through rate in slot 1 is normalized to 1, value v_i is more precisely a normalized per-click value for the specific click-through rate for slot 1. Moreover, each bidder may also have a bidder-specific quality term, which can be easily introduced by adjusting the value v_i for bidder i.

As described, the model also immediately captures settings with banner ads in which a bidder has a per-impression value and makes a per-impression payment.

Given this externality-based framework we are interested in finding the optimal allocation of slots to bidders, i.e. in solving the winner determination problem (WDP):

DEFINITION 1. *Given bids b and constraints C,* WDP$_C$ *for discount factor $\delta \in (0, 1)$ is to find a winner set $W \subseteq N$ and an allocation A ($i \in W$ winning slot $A_i \leq m$) solving:*

$$\max_{(W,A)\in\mathbb{F}} \sum_{i\in W} b_i\delta^{A_i-1} \tag{1}$$

where $\mathbb{F}$ is the set of feasible solutions (W, A): $A_i \neq A_j$ for all $i \neq j$ (both in W), and A satisfies $C_1, \ldots, C_n$.

We model each bidder as self-interested, and seeking to maximize its expected utility, which is $(v_i - p_i)\delta^{A_i-1}$, where p_i is the per-click payment, given that i is allocated slot A_i and allocation A satisfies its constraint set C_i. For an allocation that does not satisfy its constraint set, i has value 0 and expected utility $-p_i\delta^{A_i-1}$. The bid b_i and constraints C'_i submitted by a bidder will, in general, affect both its allocation and its payments.

This model of value-externalities restricts a bidder's value to be zero if any of its constraints are not satisfied. In practice, it seems likely to be of interest to extend the model, for example to allow a bidder to state a smaller but non-zero value for an allocation that does not satisfy its constraints. We think this is an interesting direction for future work. For now, we note the worst-case complexity and inapproximability results extend to this more general model. Moreover, the incentive analysis will need to be extended to allow for a bidder that reports two values in addition to a constraint set. For now, we prefer to consider just a single extension to the standard slot auction model, with a notion of incentive-compatibility that is stated relative to the standard GSP model.

Unless otherwise specified, each bidder may submit an arbitrary number of constraints. However, each constraint can only impose a condition on the bidder itself and at most one other bidder (see Fig. 1 for examples). We formalize this restriction in the next section.

3. UNIT-BIDDER CONSTRAINTS

Let $K = \{1, \ldots, m\}$ denote the set of slots. In the *unit-bidder constraints* model each bidder $i \in N$ can submit a set of tuples,

$$C_i = \{(\alpha_i, \beta_j)^\ell \;:\; \ell \in \{1, \ldots, c_i\}\} \tag{2}$$

where $\alpha_i \in \{(i, k_i) \;:\; k_i \in K\} \bigcup \{\text{True}\}$ and $\beta_j \in \{(j, k_j) \;:\; j \neq i, k_j \in K\} \bigcup \{\text{True}\}$, and we cannot have both α_i and β_j set to True. These constraints are "no goods", with a semantics that, conditioned on being allocated, bidder i insists that no condition $(\alpha_i, \beta_j) \in C_i$ is true. If $\alpha_i = \text{True}$ then this requires that j is not in slot k_j for $(j, k_j) = \beta_j$. If $\beta_j = \text{True}$ then this requires that i is not in slot k_i for $(i, k_i) = \alpha_i$. Otherwise it requires that j is not allocated slot k_j when i is allocated slot k_i, where $(i, k_i) = \alpha_i$ and $(j, k_j) = \beta_j$. We think of these as "unit-bidder" constraints because each constraint (α_i, β_j) imposes a requirement on the allocation to a single, other bidder.

We now provide two brief instantiations of the unit-bidder constraints framework. Sec. 5 will discuss in detail another language within this framework.

For a first example of unit-bidder constraints we can encode in unit-bidder constraints the model of Muthukrishnan [27], where each bidder i specifies a maximum of $P_i \leq m$ positions that it wants to be displayed (to any bidder): $C_i = \{(i, k, \text{True}), (\text{True}, j, k) : j \neq i, k \in \{P_i, \ldots, m\}\}$.

Slot-specific constraints. In our second model, a bidder i can require its slot to be in the top $k_{ii} \in \{1, \ldots, m\}$ slots and, for an arbitrary set of other bidders $j \neq i$, that j is not in the top $k_{ij} \in \{0, \ldots, m\}$ slots. For $k_{ii} = m$, bidder i has no constraints regarding its own slot only. For $k_{ij} = m$, i completely excludes some other bidder j while for $k_{ij} = 0$ no constraint is imposed. This model generalizes the "bid-to-the-top" model (Aggarwal et al. [3]) by introducing the ability to restrict the position allocated to other bidders as a condition to winning.

Slot-specific constraints are a subset of unit-bidder constraints: e.g. a constraint $k_{ii} = 2$ and 4 slots can be encoded as $C_i = \{(i, 3, \text{True}), (i, 4, \text{True})\}$, and a constraint $k_{ij} = 2$ can be encoded as $C_i = \{(\text{True}, j, 1), (\text{True}, j, 2)\}$. They generate the following structure:

(S1) α_i or β_j is set to True,

(S2) $(\text{True}, j, k_j) \in C_i \Rightarrow (\text{True}, j, k_j - 1) \in C_i$ for all $k_j > 1$,

(S3) $(i, k_i, \text{True}) \in C_i \Rightarrow (i, k_i + 1, \text{True}) \in C_i$ for all $k_i < m$.

The WDP for slot-specific constraints is NP-hard since the special case of exclusion constraints in which bidder i selects $k_{ij} \in \{0, m\}$ for every $j \neq i$, so that j is either unrestricted by i or completely excluded, is equivalent to INDEPENDENTSET. We provide a fixed-parameter algorithm for this restriction in Sec. 6.

3.1 Generic Greedy Algorithm

Simple and efficient algorithms are desirable given the practical requirements of position auctions. We present such a greedy, sub-optimal, algorithm for WDP_C. Optimal algorithms for WDP_C are unlikely to be efficient as we show, even for strictly less general models, in Sec. 3 and Sec. 5.

A slot k is allocated to the bidder (if any) with maximal bid value across those that are *eligible*, i.e. not yet allocated and for whom an allocation to slot k is not precluded by a constraint of a bidder already allocated or by own constraints.

Greedy Algorithm for unit-bidder constraints

For slot $k = 1$ to m

Eligible $\leftarrow \{i : i \text{ can win } k \text{ given } C_{i_1} .. C_{i_{k-1}}, C_i\}$

Highest bidder (if any) $i_k \in$ *Eligible* wins slot k

End

We now analyze this algorithm's incentive properties and exhibit its robustness to constraint manipulations.

4. INCENTIVES IN GREEDY ALGORITHM

For optimal algorithms for the WDP (such as the ones in Sec. 6) it would be straightforward to adopt the VCG mechanism to determine payments, and achieve a truthful auction with respect to both bid value and constraints [18].[2] The more interesting incentive question is in regard to our greedy algorithm.

4.1 Value Truthfulness

In the greedy algorithm a higher bid b_i, fixing everything else, leads to a (weakly) higher slot.

LEMMA 1. *The greedy algorithm is monotonic (with respect to bids) given unit-bidder constraints.*

PROOF. Fix constraints $C = (C_1, \ldots, C_n)$ and bids b_{-i}. Suppose bidder i is allocated in slot k for bid b_i. Then the bidder is at least allocated in slot k when bidding $b'_i > b_i$ because if it remains unallocated when slot k is allocated, the state of the algorithm is unchanged from when bid b_i is submitted because earlier decisions are oblivious to the bid values of unallocated bidders. □

For known C_i, we could adopt the standard theory due to Myerson [28] and achieve truthfulness in bid value b_i by charging a winner,

$$p_i(b_i, C_i) = b_i f(b_i, C_i) - \int_{w=0}^{b_i} f(w, C_i) \mathrm{d}w \tag{3}$$

where others' bids and constraints b_{-i} and C_{-i} are fixed and suppressed in the notation, and $f(w, C_i) = \delta^{A_i(w, C_i) - 1}$ where $A_i(w, C_i)$ is the slot allocated to bidder i given bid value w.[3] This can be easily computed, by stepping through the discrete points at which the allocation to bidder i changes as its bid b_i is decreased towards 0.

EXAMPLE 1. *Consider 2 slots and three bidders with values 30, 20 and 10, and in the slot-specific model where bidder 1 reports* $k_{12} = 1$ *to preclude bidder 2 from appearing in slot 1 when bidder 1 is allocated. Discount* $\delta = 0.9$. 1*'s payment given this constraint is* $30 - [(30 - 20)] = 20$. *If bidder* 1 *did not have its constraint (i.e.* $k_{12} = 0$*), it would still win slot 1 but pay instead* $30 - [(30 - 20) + (20 - 10)(0.9)] = 11$.

This example immediately reveals a profitable manipulation with respect to *constraints*: bidder 1 can benefit by hiding its constraint on bidder 2. It is not surprising that we see a problem here– this is a setting with multidimensional types. In particular, $p_i(b_i, C_i)$ is not independent of C_i conditioned on receiving the same slot.

The basic problem is that a bidder's constraints can make the allocation it would receive for lower values look less

[2] We use "truthful" for dominant-strategy incentive compatible.

[3] See Aggarwal et al. [2] for a discussion of this payment rule and its relation to VCG in the context of position auctions.

promising, through the role in $f(w, C_i)$ when computed the payment in Eq. (3). A bidder can improve his payment by dropping constraints that are redundant given his true value but would not be for lower bids. From this, and given the essential uniqueness of the payment rule in providing value truthfulness (subject to linear translations) we see that it is impossible to achieve value and constraint truthfulness with a greedy algorithm.

4.2 Constraint Truthfulness

We consider now a GSP payment rule for our greedy algorithm, and show that this is in fact truthful with respect to constraints, while retaining local stability with respect to value misreports as is familiar from standard position auctions [8]. A winner in slot k pays according to a "next price" payment rule:

$$p_i(b_i, C_i) = \delta^{k-1}[\min_{b'_i} b'_i] \quad (4)$$

$$\text{s.t. } A_i(b'_i, C_i) = k, \quad (5)$$

determining the smallest bid, given its declared set of constraints, for which a bidder is allocated the same slot, and charging the effective value for a bidder with that value when allocated slot k. Thus in Example 1 the highest bidder (30) pays 20 *irrespective* of the constraints that it (or other bidders) declare.

As is well understood, adopting this payment rule does not provide truthfulness with respect to bid values: this is also the case for the standard position auction model [8]. But the payment rule is nevertheless interesting because it is the rule used in current practice, and it provides local stability so that a bidder can only improve the outcome by deviating enough to change the slot allocated.

In understanding this payment rule's incentive properties with respect to the reporting of constraints, we consider "downward-monotonic" constraints, a generalization of the slot-specific model. We will analyze in detail the algorithmic properties of identity-specific constraints, another subclass of downward-monotonic constraints, in Sec. 5 and Sec. 6.

DEFINITION 2 (DOWNWARD-MONOTONIC). *Unit-bidder constraints $C = (C_1, \ldots, C_n)$ are* downward-monotonic *if, for every i, we have*

$$(i, k_i, \beta_j) \in C_i \;\Rightarrow\; (i, k_i + 1, \beta_j) \in C_i, \text{for all } \beta_j. \quad (6)$$

This property of downward-monotonicity requires that, fixing the allocation to other bidders, if a bidder i is dissatisfied with an allocation to slot k_i then the bidder is also dissatisfied with an allocation to any lower slot. As a special case, when $\beta_j =$ True it requires that if a bidder has zero value for slot k_i whatever the allocation to other bidders then the bidder also has zero value for any lower slot.

THEOREM 1. *The greedy algorithm coupled with the second price payment rule in Eq. (5) is truthful for downward-monotonic constraints when bid values $b_i \leq v_i$.*

PROOF. Fix any $b_i \leq v_i$. Let k denote the slot allocated to i when reporting true C_i. Conditioned on report C'_i not changing the allocated slot k, the payment does not change because constraints have no effect on other bidders until a bidder is allocated and so the eligible set is unchanged. Moreover, i does not want to misreport $C'_i \neq C_i$ if this changes the allocated slot because (i) if i may be allocated to some slot $k' < k$ then there must be a true constraint that is violated upon allocation to slot k', since the allocation made to other bidders on slots $[1, \ldots, k' - 1]$ does not change, and otherwise (ii) any change to preclude i from being eligible for slot k will preclude i from being eligible for all subsequent slots $k' > k$ by the admissibility property. Thus i will not be allocated, which is not strictly better than an allocation since $b_i \leq v_i$ and the payment rule in Eq. (5) charges no more than the value of the allocation. □

The earlier examples illustrate the basic point: with this payment rule the constraints are used only to preclude the allocation of some slots to a bidder and do not affect the payments collected from a bidder, conditioned on such an allocation. This is different from the Myerson payment rule and provides robustness against the misreport of constraints.

Because slot-specific constraints are downward-monotonic by **(S1)** and **(S3)**, we have

COROLLARY 1. *The greedy algorithm is truthful for slot-specific constraints when coupled with the second price payment rule in Eq. (5) and bid values $b_i \leq v_i$.*

The total-position constraints [27], that set an upper bound on the total number of slots allocated (recall Sec. 3), are downward-monotonic as well.

This property of downward-monotonicity is required for truthfulness regarding constraints.

EXAMPLE 2. *Consider 3 slots and 3 bidders with values 60, 40 and 10, and discount $\delta = 0.9$. If bidder 1 is truthful then he wins slot 1 and pays 40 for payoff $60 - 40 = 20$. But by reporting constraint "I don't want slot 1", he wins slot 2 and pays 10 for payoff $(60 - 10)0.9 = 45 > 20$. This constraint is not downward-monotonic; in particular, it cannot be expressed within our slot-specific model.*

4.3 Complete-Information Nash Equilibrium

We can also explore whether the equivalence between a particular complete information Nash equilibrium (NE) and the VCG outcome, established for position auctions in the standard model without externalities [8, 34] holds in the current setting. This connection between NE and the VCG outcome has been used to understand the relative merits of GSP vs VCG for sponsored search. We show that this connection is not sustained with constraint-based value externalities.

EXAMPLE 3. *Consider 4 bidders with values 40, 30, 20 and 10, 3 slots, discount $\delta = 0.9$ and the slot-specific model, with $k_{23} = 3$ and $k_{32} = 3$ so that each bidder precludes the other bidder from being allocated in any slot. For the VCG payments, note that the optimal solution is to allocate $(1, 2, 4)$ to slots $(1, 2, 3)$, with total discounted value $V^* = 40 + 30(0.9) + 10(0.9^2)$. Considering the solutions without bidders 1 and 2, we have $V_{-1} = 30 + 10(0.9)$ and $V_{-2} = 40 + 20(0.9) + 10(0.9^2)$ for the total value of the allocation that would be selected without each bidder, respectively. From* *this, we see that the VCG payments would be* $p_{1,\text{vcg}} = 30 + 10(0.9) - 30(0.9) - 10(0.9^2) = 3.9$ *and* $p_{2,\text{vcg}} = 40 + 20(0.9) + 10(0.9^2) - 40 - 10(0.9^2) = 18$*. To sustain the VCG outcome in a NE we would require* $b_1 > b_2 > \max(b_3, b_4)$*. For the payment to bidder 1 to be correct, we need* $3.9 = b_2$*. But for the payment to bidder 2 to be correct, we need* $18 = (0.9)b_3$ *and* $b_3 = 20$ *but then* $b_3 > b_2$ *and the bidders would be allocated out of order.*

We see the same kind of reversal that is observed by Aggarwal [3] in their bid-to-the-top model: they achieved a correspondence between VCG and GSP for a simple model in which bidders require only prefix positions (e.g. one of slots {1,2,3}) but lost this property with more general requirements (e.g., one of slots $\{2, 4\}$.) The basic challenge is that the mapping from GSP to VCG relies on a monotonicity property, where each successively higher bidder imposes a larger marginal externality on the value of the allocation to the other bidders. This is not the case in our model of value externalities.

5. IDENTITY-SPECIFIC EXTERNALITIES

In this section we introduce identity-specific externalities, which fall within the unit-bidder constraints model and also satisfy downward-monotonicity.

In this new language, a bidder i can specify a set C_i of other bidders such that i's value for a slot is realized if and only if all bidders in C_i are either not allocated or allocated in slots below i. Each sports shoe company in Fig. 1 specifies such a constraint towards the other one. This precedence model is motivated by the inherent interpretation (by users) of the list of ad slots as a ranking. We denote the corresponding value maximization problem by $\text{WDP}_C^{\text{pre}}$.

Identity-specific constraints are a subset of unit-bidder constraints: for example a constraint $C_i = \{j\}$ and with 3 slots is encoded as $C_i = \{(i, 2, j, 1), (i, 3, j, 2), (i, 3, j, 1)\}$.

In fact, identity-specific constraints generate the following structure on C_i:

(I1) $(i, k_i, j, k_j) \in C_i \Rightarrow (i, k_i, j, k_j - 1) \in C_i$ for $k_j > 1$

(I2) $(i, k_i, j, k_j) \in C_i \Rightarrow (i, k_i + 1, j, k_j) \in C_i$ for $k_i < m$,

(I3) $(i, k_i, j, k_j) \in C_i \Rightarrow (k_j < k_i)$.

Thus, due to property **(I2)**, identity-specific constraints are downward-monotonic yielding, via Theorem 1,

Corollary 2. *The greedy algorithm is truthful with respect to identity-specific constraints when coupled with the second price payment rule in Eq. (5) and bid values* $b_i \le v_i$*.*

Example 3's constraints are easily expressible in the identity-specific model and its conclusions thus carry through.

The unit-bidder constraints model is strictly more general than the union of the identity-specific and slot-specific models. For example a downward-monotonic set of unit-bidder constraints that cannot be encoded with the earlier models is $(1, s, 2, m)$, for all slots $s \le m$, specifying that bidder 1 does not want bidder 2 in the last slot m, regardless of 1's slot. This may be of interest for example if bidder 2 is compelling for users that scroll down the page and click on 2 instead of considering the other ads again. In future work it will be interesting to understand the most general semantics consistent with the *downward-monotonic* property.

5.1 Algorithmic considerations

Let $G_c = (V_c, E_c)$ denote a *constraint graph*, which is a directed graph where there is one vertex $v_i \in V_c$ for each advertiser $i \in N$ and the edges represent constraints. The edge semantics are that there exists a directed edge from vertex i to j if and only if i wants to be placed above j.

We provide (proof deferred to the Appendix) a guarantee on the quality of approximation of our greedy algorithm. We say that an algorithm achieves a ρ-approximation if the value of the assignment it outputs is within a multiplicative factor of $\rho \le 1$ of the value of the optimal offline solution, for all possible instances.

Theorem 2. *Let d denote an upper bound on all vertices' in-degrees in G_c. The greedy algorithm for the* WDP_C^{pre} *problem achieves an* $\frac{1-\delta}{1-\delta^{d+2}}$ *approximation when* $\delta < 1$*, and an* $\frac{1}{d+2}$ *approximation when* $\delta = 1$ *in* $O(n \log n)$ *time.*

The bound on this algorithm's performance is almost tight: consider a setting with $d+1$ bidders where every bidder in $\{1, \ldots, d\}$ bids 1 and has $d+1$ as common enemy: $C_i = \{d+1\}$. Bidder $d+1$ bids 1.01 and has $C_i = \emptyset$. Then the greedy algorithm achieves value 1.01 while the optimal solution achieves value $1 + 1\delta + \ldots + 1\delta^{d-1} + 1.01\delta^d = \frac{1-\delta^{d+1}}{1-\delta} + 0.01\delta^d$. For a general graph (arbitrary d) and $\delta < 1$ we recover the trivial approximation bound $(1 - \delta)$.

A drawback of this solution is that the approximation ratio is stated in terms of a bound on the *indegree* of the problem. By Theorem 3 in Sec. 5.3, a certain dependence of approximation ratios on the constraint graph structure is however unavoidable. Whereas a bidding language can easily constrain the outdegree of a graph by limiting the number of constraints a bidder is allowed, it is more difficult to see how to control *a priori* the maximum indegree.

5.2 Structural Observations

We have the following easy lemma:

Lemma 2. *Given a constraint graph G_c and a subset S of bidders, an allocation that allocates every bidder in S is infeasible if and only if there exists a directed cycle in the subgraph induced by S on G_c.*

Consider the constraint graph induced for a fixed set of bidders $S \subseteq N$. For bidders S with an acyclic constraint graph the *optimal ordering* is defined as the sequence of bidders allocated to slots 1 through $\min\{m, |S|\}$ that maximizes the total discounted bid price. That is, the optimal ordering solves the $\text{WDP}_C^{\text{pre}}$ problem restricted to allocating all bidders in S. A contiguous sub-ordering refers to any contiguous subsequence of such an ordering. A sub-ordering is optimal if the ordering of the bidders in the sub-ordering would be maintained if the same restricted $\text{WDP}_C^{\text{pre}}$ problem was solved on just those bidders.

Lemma 3. *All contiguous sub-orderings of an optimal ordering are optimal. In particular, an optimal ordering cannot have bidder i placed immediately above bidder j where i and j have no constraints between them and i's bid is less than j's bid.*

To understand this, notice that any bidders outside of a subsequence are agnostic to a local reordering within a subsequence: if their constraints are satisfied beforehand then they are satisfied for any reordering. For a pair of adjacent bidders in any optimal ordering, a useful swap could be executed because it does not violate any new constraints with other bidders. The only concern is that a constraint between these two bidders themselves is not violated in the rearrangement.

From Lemma 3, it seems that dynamic programming (DP) is a possible technique for the $\text{WDP}_C^{\text{pre}}$ problem. We will highlight the challenge of applying DP on a very simple case, an *increasing path*: the constraint graph is a directed path with bids increasing along the path. We have $b_i < b_{i+1}$ and $C_i = \{i+1\}$ for all $i \in \{1, \ldots, n-1\}$, with $C_n = \emptyset$. In light of Lemma 2, we can consider paths as basic building blocks of a potential recursive algorithm for constructing a subset of bidders with an associated directed acyclic constraint graph.

A first approach is to apply DP on the number of slots, by obtaining the optimal solution with k slots from optimal solutions on $k-1$ slots. Such an approach fails, however, as illustrated by the following increasing path instance with 4 bidders and bids $30, 32, 36, 40$ respectively. Let $\delta = 0.45$. Then the optimal solution for $k = 2$ slots is to allocate 40 in the top slot followed by 32 $(40, 32)$. This solution has total value of $40 + 32 \cdot 0.45 = 54.4$ whereas allocating $36, 40$ yields $36 + 40 \cdot 0.45 = 54$. However the optimal solution for $k = 3$ slots is $36, 40, 30$, with a higher value (60.075) than that of any feasible ordering containing bidders 40 and 32 $(40, 30, 32; 32, 36, 40; 40, 32)$. We note that the optimal solution may not allocate all available slots, even if that is feasible: the only feasible allocation of 4 slots has value 55.335, inferior to that using just 3 slots.

An alternate approach is to apply DP on the number of bidders. Recall from our example that 40 is placed in the second slot in the optimal solution for $k = 3$ slots. Two issues that complicate DP appear upon considering the subproblem (fixing bidder 40 in slot 2) with the other three bidders and slots 1 and 3. First, the 36 bidder cannot placed in position 3. Second, there is now a gap in discount factors, which are 1 and δ^2 instead of decaying at a constant rate δ. Such issues render us skeptical about the effectiveness of DP on general problem structures.

For an increasing path, however, a successful DP approach exists. It exploits additional structure (via Lemma 3) on the optimal allocation, which must display bidders in decreasing order of value except for contiguous sub-paths (with bidders in increasing order). For example, for $k = 3$, the optimal solution $\{36, 40, 30\}$ has increasing sub-path $\{36, 40\}$, followed by $30 < 36$. By Lemma 3, sub-ordering $\{40, 30\}$ cannot be improved by a swap. The DP has one state for each pair of $i \le n$ and $k' \le m$, corresponding to the optimal allocation of a subset of bidders $\{1, \ldots, i\}$ on at most k' slots. Sec. 6 provides a different DP method for a special case of the slot-specific externalities model, in which bidders can only choose to completely exclude another bidder when allocated.

5.3 Complexity on Bounded-Degree Graphs

In this section, we provide computational hardness results for $\text{WDP}_C^{\text{pre}}$ with equal bids by a unified reduction from two well-studied graph-theoretical problems. For the reduction, we adopt the following construction of an instance of $\text{WDP}_C^{\text{pre}}$ from a given graph $G = (\{1, \ldots, n\}, E)$: construct an $\text{WDP}_C^{\text{pre}}$ instance where each vertex i is mapped to a bidder $i \in N$ with bid $b_i = 1$, and for each edge $a = (i, j) \in E$ we place $i \in C_j$ (note that if G is undirected we also place $j \in C_i$), that is, we add the corresponding edge(s) to the constraint graph G_c. Clearly, a set of winners W in $\text{WDP}_C^{\text{pre}}$ is feasible if and only if W is an acyclic set in G (if G is directed) and an independent set in G (if G is undirected). If so, then there is a feasible allocation of slots to $i \in W$: by Lemma 2 if G is directed and by the lack of any constraints within W if G is undirected. Since all bids are equal, the value-maximizing W is a maximal set with the corresponding property in G.

Thus, for any $\delta \in (0, 1]$, the $\text{WDP}_C^{\text{pre}}$ problem is NP-hard for general constraint graphs by the immediate reduction from Independent Set. Moreover, this provides an inapproximability lower bound for $\delta = 1$, namely $\min(n^{1-\epsilon}, \frac{c}{2}^{1/2-\epsilon})$ for any fixed $\epsilon > 0$, where c is the number of constraints [16], relying on the $NP \neq ZPP$ complexity assumption. We obtain instead two statements we regard as more informative for practical settings where the number of constraints expressed by a bidder is bounded. Let $\text{WDP}_d^{=}$ denote the restriction of $\text{WDP}_C^{\text{pre}}$ to constraint graphs with degree at most d and equal bids. Via an analogous hardness of approximation result [6] for Independent Set on bounded-degree graphs relying on the Unique Games conjecture [22] our construction yields:

Theorem 3. *Let $l(d) = \frac{\log^2 d}{d}$ and $\phi(d) = \frac{1-\delta^{1+(n-1)l(d)}}{1-\delta^n} = \frac{1+\delta+\cdots+\delta^{(n-1)l(d)}}{1+\delta+\cdots+\delta^{n-1}}$. Symmetric ($i \in C_{i'} \iff i' \in C_i$) $WDP_d^{=}$ is* Unique Games*-hard to approximate to within a $O(l(d))$ factor for $\delta = 1$ and to within a $O(\phi(d))$ factor for $\delta < 1$.*

Proof. The result for $\delta = 1$ follows from our construction and [6]. We prove that for $\delta < 1$, any algorithm A with a better approximation factor on any instance must have a better approximation factor than $l(d)$ for $\delta = 1$. Consider an instance with maximum independent set of size $s \le n$. We claim that A must output an independent set of size at least $s' \ge s \cdot l(d)$. Otherwise the approximation ratio of A is at most $\frac{1+\delta+\cdots+\delta^{sl(d)}}{1+\delta+\cdots+\delta^{s}}$ which in turn is at most $\frac{1+\delta+\cdots+\delta^{(n-1)l(d)}}{1+\delta+\cdots+\delta^{n-1}}$. □

In fact, the problem is computationally challenging even for a bound of 2 on the in-degree and out-degree of each vertex.

Theorem 4. *For all $\delta \in (0, 1]$, $WDP_2^{=}$ is NP-hard.*

Proof. We use the NP-hardness [10] of MinimumFeedbackVertexSet (MFVS) under the same in-degree and out-degree bound of 2. Recall that the MFVS problem is to determine a minimum set of vertices S whose removal makes a given graph (V, E) acyclic. Our construction provides a simple reduction from the MFVS problem to the

$\text{WDP}_C^{\text{pre}}$ problem: S is a MFVS in a *directed* graph G if and only if $V \setminus S$ is a maximal acyclic set in G, i.e. a bidder set whose allocation maximizes value given constraint sets C_i for $i \in \{1, \ldots, n\}$. □

In a sparse constraint graph, "high" bids will win.

LEMMA 4. *Let $\overline{d} \geq 1$ denote an upper bound*[4] *on any vertex's in-degree and out-degree. There is no optimal solution to the* WDP_C^{pre} *problem in which an advertiser whose bid ranks below the $(\overline{d}+1)m - 1^{th}$ highest ranking bid, for m slots, is assigned a slot.*

PROOF. We prove by contradiction. Let B denote the set of bidders with the $(\overline{d}+1)m-1$ highest ranking bids. Assume that in an optimal solution a bid $b(v)$ which is lower than all bid values in B is assigned a slot A_i. Then at least $\overline{d}m$ members in B are unallocated. We claim we can use at least one of them to replace v to get a higher-valued allocation without violating the constraints. To see this, note that for each of the unallocated $\overline{d}m$ members $u \in B$, there are two possible reasons why we cannot replace v with u: either there are at most $\overline{d}$ vertices $\{v'_1, ..., v'_{\overline{d}}\}$ having an incoming edge from u such that at least one vertex $v' \in \{v'_1, ..., v'_{\overline{d}}\}$ is assigned a higher slot $A_{i'}$ for $i' < i$, or there are at most $\overline{d}$ vertices $v'_1, ..., v'_{\overline{d}}$ having an outgoing edge toward u such that at least one vertex $v' \in \{v'_1, ..., v'_{\overline{d}}\}$ is assigned a lower slot A'_i for $i' > i$. However, there can be at most $m-1$ slots occupied by vertices other than v. By the Pigeonhole principle, at least one of the $\overline{d}m$ members in B who is unassigned does not have any out-neighbors or in-neighbors in the other $m-1$ slots (excluding A_i). We can safely use this member to replace v without violating any constraints. This contradicts the assumption that the given allocation is optimal. □

Lemma 4 allows us to preprocess input data and discard bidders ranked below the $(\overline{d}+1)m-1$ highest bids for any $\overline{d}$ and m. If, for a given instance of $\text{WDP}_C^{\text{pre}}$, the number of slots m is a constant then, for small $\overline{d}$, an enumerative WDP algorithm examining all feasible allocations becomes practical: its asymptotic runtime is dominated by that of finding the top $(\overline{d}+1)m-1$ bids, which is linear in n.

By considering restricted identity-specific models, we will circumvent the earlier computational hardness results.

6. FIXED-PARAMETER ALGORITHMS

In this section we introduce two instantiations of the identity-specific model and we provide fixed-parameter algorithms for the respective WDPs.

6.1 Category-Specific Externalities

The category-specific model is a special case of the identity-specific model in which every bidder is associated with a category and value externalities are limited to choosing to require placement above all bidders in the same category as a bidder. The category-specific model moves towards a more anonymous setting and provides additional structure to enable an optimal algorithm.

The algorithm presented below computes an optimal allocation in polynomial time in m and n if the number of categories g is a constant. To motivate this model, suppose the user query is "cleats" which is a specific type of sports shoes. For this query, there will be bidders for exact match (e.g. Nike and Adidas bidding precisely on cleats) and bidders for broad match (e.g. Amazon bidding on sports shoes). In this example bidders belong either to the exact match category or the broad match category and would be able to express externalities only in these terms.

Let $G = \{1, \ldots, g\}$ represent the categories, defining a partition of the bidders N. Let $c_i \in G$ denote the category of bidder $i \in N$. Each bidder is offered a binary choice when submitting a bid, of having constraints with respect to all other bidders that belong to the same category or having no constraints at all. Let $F_c \subseteq N$ denote the set of bidders in category $c \in G$ who have chosen to target all other bidders in category c through constraints. In a feasible allocation only one bidder in F_c can be allocated and clearly it is the maximum value bidder f^c_{max}. Let $F = \cup_{c \in G} f^c_{max}$. Let Q_c denote the set of bidders in category c who have no constraints. Let $Q = \cup_{c \in G} Q_c$. Finally let $Q_{free} = \cup_{c \in G} Q_c$ such that $F_c = \emptyset$. In other words Q_{free} is the set of *free* bidders who are nobody's "enemies". Let S be the list containing chosen slot positions. Note that we assume, in the algorithms described next, that once bidders are allocated they are removed from the sets to which they belong.

Algorithm *AllocateCategories(G, F, Q)*

> **For** each permutation $F' \subseteq F$.
>   **For** each slot combination S for F'.
>   Run *Subroutine(G, F', Q, S)*.
>   Store the resulting allocation.
>   **End**
> **End**
> Output the allocation of highest value.

Algorithm *Subroutine(G, F', Q, S)*

> Initialize $j = 0$. $c = 0$. $Q_0 = \emptyset$.
> Build and sort Q_{free} in decreasing bid order.
> Sort S to list slots in order of increasing index.
> **While** there remain available slots and unallocated bidders
>
> 1. Update $Q_{free} = Q_{free} \cup Q_c$.
> 2. Place next bidder $f \in F'$ in slot determined by the next unused slot position in S. Let c represent f's category.
> 3. Allocate top bidders $i \in Q_{free}$ in decreasing bid order in free slots above f.
>
> **End**

PROPOSITION 1. *Algorithm* AllocateCategories *outputs the optimal allocation.*

PROOF. The algorithm enumerates and compares all feasible allocations involving max value bidders $f^c_{max} \forall c \in G$ and bidders in $Q_c \forall c \in G$. The only allocations not considered are those involving lower value bidders in F_c who impose the same constraints as f^c_{max}. Other than due to constraints, no low bidder can be placed right above a higher bidder. This reduces the space of candidate optimal allocations to only the ones we consider. □

[4]The WDP for $\overline{d} = 0$, i.e. no constraints, is straightforward.

We introduce some notation to analyze the runtime of *AllocateCategories*. Let $P(z,t)$ denote the number of t-permutations of a set of z elements. $P(z,t) = \frac{z!}{(z-t)!}$. In order to enumerate all feasible allocations involving max value bidders $f^c_{max} \forall c \in G$, each permutation of bidders within a particular subset F' must be evaluated.

PROPOSITION 2. *If g, n, and m represent the number of categories, the number of bidders, and the number of slots respectively, then the runtime of* AllocateCategories *is* $O((n \log n + gn)(m^g)(g^g))$.

PROOF. Sorting Q_{free} and S takes $O(n \log n)$. Steps 1-3 in *Subroutine* take $O(n)$. The while loop is run at most g times. Hence *Subroutine* takes $O(n \log n + gn)$. The number of slot combinations for each subset F' is $\binom{m}{|F'|}$ where $|F'| \leq g$. $\binom{m}{g} \leq m^g$. Since the permutation of the subset F' is important in determining the optimal allocation, the total number of subsets (denoted TF) is $P(g,1) + P(g,2) + \ldots + P(g,g-1) + g! + 1$. Hence $TF = g! + \sum_{h=1}^{g-1} P(g,h) \leq g(g!) \leq g^g$. □

6.2 Symmetric Constraints

We restrict bidders to only specify *exclusion* constraints, i.e., a bidder i has zero value if any bidder i' in set C_i is also allocated, *regardless* of i''s slot. This model can be viewed as the symmetric restriction of the identity-specific model since i and i' can exclude each other by specifying $i \in C_{i'}$ and $i' \in C_i$. It can also be viewed as a restriction of the slot-specific model. We thus view the collection of constraint sets C as a set of (undirected) *forbidden* edges on $\{1,\ldots,n\}$: if $(i,i') \in C$ then i and i' cannot be both allocated: $|W \cap \{i,i'\}| \leq 1$.

By relabeling bidders, assume that bids are sorted decreasingly: $b_i \geq b_{i+1}, \forall i \in 1..n-1$. Clearly, winners are allocated in this order: otherwise the allocation could be improved, while still feasible, by swapping slots.

LEMMA 5. *If $i, i' \in W$, for $i < i'$ then $A_i < A_{i'}$.*

We provide a dynamic programming approach for the associated WDP based on a standard tree-width decomposition technique (see for example [29]). This approach has time and space complexity exponential in the tree-width of the graph, a quantity associated in our model with a locality measure of the constraints, to be defined shortly. Our algorithm is thus reasonably efficient for constraint graphs with a local structure.

Let L be a *constraint locality* measure such that if $(i,i') \in C$ then $|i-i'| \leq L-1$ (note that indices do not wrap around). L measures how far apart can two mutually excluded bidders be in the *sorted* order of bids. Lemma 5 is the critical property enabling DP in this context; in particular if the highest bidder 1 is allocated (necessarily in the first slot) then in all slots bidders in $2..L$ only need to be excluded because of 1. Lemma 5 (and this approach) fails if constraints are not symmetric (i.e. the $\mathrm{WDP}^{\mathrm{pre}}_C$ problem).

Let an (i,L)-*byte* be an L-digit binary word $B : \{i,\ldots,i+L-1\} \to \{0,1\}$, where $B(i+l)$ for $\ell \in \{0,\ldots,L-1\}$ indicates whether bidder $i+\ell$ is allocated. Let $\#^{<}_B(\ell)$ denote the number of 1s in $B(i..i+\ell-1)$. An (i,L)-byte B is *feasible* if no constraints are violated, i.e. if $(i_1,i_2) \notin C$ whenever $B(i_1) = B(i_2) = 1$.

1	1+1	$m^{<}_1$	$v_{\mathcal{X}_1}$
0	0	0	0
0	1	0	b_2
1	0	0	b_1
1	1	0	$-\infty$

table $\mathcal{T}_1$ for state $\mathcal{X}_1$

2	2+1	$m^{<}_2$	$v_{\mathcal{X}_2}$
0	0	0/1	$\max\{0, b_1\}$
0	1	0/1	$\max \{0+\delta^{0+0}b_3, b_1+\delta^{1+0}b_3\}$
1	0	0	$\max\{b_2, -\infty\}$
1	1	0	$-\infty$

table $\mathcal{T}_2$ for state $\mathcal{X}_2$ from table $\mathcal{T}_1$

Table 1: Exclusion-only dynamic programming example with $C = \{(1,2),(2,3)\}$ and mutually excluded bidders closer than $L = 2$.

We use $m^{<}_i \in \{1,\ldots,m\}$ for $i \geq 1$ storing how many bidders are allocated before i in the corresponding optimum. For each state $\mathcal{X}_i = \{i,\ldots,i+L-1\}$ we compute in a table called $\mathcal{T}_i$ via DP the value of each allocation given the optimum solutions in table $\mathcal{T}_{i-1}$. The value of the optimal allocation is found in $\mathcal{T}_{n-L+1}$ and this allocation can be traced back in lower-indexed states via standard DP techniques.

Initialization. For all $(1,L)$-bytes B, let $m^{<}_1 = 0$ and initialize in $\mathcal{X}_1$ the total value of allocating according to B

$$v_{\mathcal{X}_1}(B) = \begin{cases} \sum_{i=1}^{L} B(i) \cdot b_i \cdot \delta^{\#^{<}_B(i)}, & \text{if } B \text{ feasible} \\ -\infty, & \text{otherwise} \end{cases} \tag{7}$$

Dynamic programming. We describe how to populate $\mathcal{T}_i$ from $\mathcal{T}_{i-1}$ where $i \geq 2$. Table 1 illustrates $L = 2$ and $i = 2$.[5]

For all infeasible (i,L)-bytes B let $v_{\mathcal{X}_i}(B) = -\infty$. Fix now a feasible (i,L)-byte B. For $\beta \in \{0,1\}$, let $B_\beta = \beta B_{-(i+L)}$ be the $(i-1,L)$-byte obtained by prepending the bit β to B (i.e. $B_\beta(i-1) = \beta$) and deleting B's last bit.

B's value is the best value from either allocating $i-1$ or not (i.e. B_1's and B_0's values), plus the consequent value $V^+(\beta)$ resulting from whether $B(i+L-1) = 1$, i.e. whether $i+L-1$ is allocated. Critically for Eq. (8), bidders $i-1$ and $i+L-1$ cannot exclude each other. $V^+(\beta)$ depends on β only by how many bidders in $1..i-1$ are allocated in $v_{\mathcal{X}_{i-1}}(B_\beta)$. δ's exponent in Eq. (9) plus one equals $i+L-1$'s slot number A_{i+L-1}, i.e. one plus the number of allocated higher bidders: $m^{<}_i(B_\beta)$ in $1..i-1$ and $\#^{<}_B(L)$ in $i..i+L-2$.

$$v_{\mathcal{X}_i}(B) = \max_{\beta \in \{0,1\}} \{v_{\mathcal{X}_{i-1}}(B_\beta) + V^+(\beta)\} \text{ where} \tag{8}$$

$$V^+(\beta) = B(i+L-1) \cdot b_{i+L-1} \cdot \delta^{m^{<}_{i-1}(B_\beta) + \#^{<}_B(L)} \tag{9}$$

$m^{<}_i(B)$ is also updated according to the max in Eq. (8).

We get a DP time and space complexity exponential in L.

[5] $L = 1$ amounts to no constraints. The greedy algorithm allocating bidders in decreasing order of their bids and removing excluded bidders is always optimal for $L = 2$ but not for $L = 3$: consider $C' = \{(1,2),(1,3)\}$ with $\delta = 1$ and $b'_i = 1 - (i-1)\varepsilon$ for $i = 1,2,3$.

THEOREM 5. *For constraint locality L, the dynamic programming technique in Eqs.* (7) *and* (8) *correctly computes optimal allocation OPT in* $O(n2^L + m)$ *time and* $O(2^L + m)$ *space. OPT's value can be recovered from table* $\mathcal{T}_{n-L+1}$.

PROOF. Only table $\mathcal{T}_{i-1}$ is needed for updating $\mathcal{T}_i$. Each table has size 2^L and one entry in $\mathcal{T}_i$ is computed in Eq. (8) in constant time[6] from two entries in $\mathcal{T}_{i-1}$. □

7. DISCUSSION

The problem of externalities in auctions is relatively little studied. In particular, while the effect of quantity externalities in sponsored search auctions has received some attention, very few papers have addressed the issue of value externalities. Unlike quantity externalities, value externalities cannot be estimated by a search engine. Hence capturing bidders' value externalities must necessarily entail designing ways in which they can express this value. To this end we have developed a constraint-based model of the effect of co-location on a bidder's value, conditioned on receiving a click.

Although the basic identity-specific model is NP-hard we are encouraged that a simple greedy algorithm can be coupled with a GSP payment rule and provide for truthfulness for constraints while retaining the same kind of local incentive properties with respect to value as in standard position auctions. Looking at our slot-specific model, we provide a dynamic programming algorithm for a special case of "exclusion constraints" and establish that the same incentive properties hold for GSP together with a greedy algorithm. It is interesting that the semantics of these two models provide for truthfulness with respect to constraints under this natural payment rule and it seems that similar results could be identified for other forms of expressiveness.

Turning to complexity results, a special case that remains open is that of the $\text{WDP}_C^{\text{pre}}$ problem with at most one constraint per bidder so that the out-degree of any vertex is at most 1. We colloquially refer to this as the "one enemy" special case. We find this case interesting because it could be achieved through a simple restriction to a bidding language. Another compelling direction is to expand on the possibility of connections with scheduling under precedence constraints, which is equivalent to a version of our problem in which a subset of bidders are selected to be allocated but the order of allocation is to be determined.

The most intriguing extension is a "soft" constraints model, in which a bidder also has a non-zero value for an allocation violating its constraints. This will require new algorithmic and incentive analysis but seems of practical importance. In this paper we have established a foundation on which future work may build towards even more realistic settings.

Acknowledgments. We thank Yiling Chen, Lisa Fleischer, Nicole Immorlica, David Pennock, Ariel Procaccia and the anonymous reviewers for valuable feedback and discussions.

8. REFERENCES

[1] G. Aggarwal, J. Feldman, S. Muthukrishnan, and M. Pal. Sponsored search auctions with Markovian users. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, pages 621–628, 2008.

[2] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing search keywords. In *Proc. of ACM Conference on Electronic Commerce EC'06*, 2006.

[3] G. Aggarwal, S. Muthukrishnan, and J. Feldman. Bidding to the top: VCG and equilibria of position-based auctions. In *Proceeding of the 4th International Workshop on Approximation and Online Algorithms*, pages 15–28, 2006.

[4] C. Ambühl and M. Mastrolilli. Single machine precedence constrained scheduling is a vertex cover problem. In *Proc. of European Symposium on Algorithms*, pages 28–39, 2006.

[5] S. Athey and G. Ellison. Position auctions with consumer search. In *NBER Working Paper*, 2009.

[6] P. Austrin, S. Khot, and M. Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. In *CCC '09*, pages 74–80, 2009.

[7] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of ACM WSDM '08: the international conference on Web search and web data mining*, pages 87–94, 2008.

[8] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth in keywords. *American Economic Review*, 97(1):242–259, 2007.

[9] M. R. Garey. Optimal task sequencing with precedence constraints. *Discrete Math.*, 4:37–56, 1973.

[10] F. Gavril. Some NP-complete problems on graphs. In *Proceedings of the 11th Conference on Information Sciences and Systems*, pages 91–95, 1977.

[11] A. Ghosh and M. Mahdian. Externalities in online advertising. In *Proc. WWW'08*, pages 161–168, 2008.

[12] A. Ghosh and A. Sayedi. Expressive auctions for externalities in online advertising. In *Proc. of WWW*, 2010.

[13] I. Giotis and A. R. Karlin. On the equilibria and efficiency of the GSP mechanism in keyword auctions with externalities. In *Proc. of 4th WINE*, pages 629–638, 2008.

[14] K. D. Glazebrook and J. C. Gittins. On single-machine scheduling with precedence relations and linear or discounted costs. *Operations Research*, 29(1):161–173, 1981.

[15] R. Gomes, N. Immorlica, and E. Markakis. Externalities in keyword auctions: An empirical and theoretical assessment. In *Proc. of the 5th WINE*, pages 172–183, 2009.

[16] J. Håstad. Clique is hard to approximate to within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[17] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. *Journal of the ACM*, 48:761–777, 2001.

[18] M. Jackson. Mechanism theory. *The Encyclopedia of Life Support Systems*, 2000.

[19] P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69(5):1237–1259, 2001.

[20] P. Jehiel, B. Moldovanu, and E. Stacchetti. Multidimensional mechanism design for auctions with externalities. *Journal of Economic Theory*, 85(2):258–294, 1999.

[21] D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. In *Proc. of the 4th WINE*, pages 585–596, 2008.

[22] S. Khot. On the power of unique 2-prover 1-round games. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002.

[23] P. Krysta, T. Michalak, T. Sandholm, and M. Wooldridge. Combinatorial auctions with externalities. In *Proc. AAMAS*, 2010.

[24] S. Lahaie, D. Pennock, A. Saberi, and R. Vohra. Sponsored search auctions. In *N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, Algorithmic Game Theory.*

[6] δ's power can be looked up in a pre-computed vector $\delta^{2..m}$.

Cambridge Univ. Press, 2007.

[25] E. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Math.*, 2:75–90, 1978.

[26] C. L. Monma and J. B. Sidney. Sequencing with series-parallel precedence constraints. *Mathematics of Operations Research*, 4(3):215–224, 1979.

[27] S. Muthukrishnan. Bidding on configurations in Internet ad auctions. In *Proc. of the 15th Annual International Conference COCOON 2009*, pages 1–6, 2009.

[28] R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[29] R. Niedermeier. *Invitation to fixed-parameter algorithms*. Oxford Univ. Press, 2006.

[30] D. C. Parkes and T. Sandholm. Optimize-and-dispatch architecture for expressive ad auctions. In *Proc. of First Workshop on Sponsored Search Auctions*, 2005.

[31] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, 2008.

[32] D. Reiley, S. Li, and R. Lewis. Northern exposure: A field experiment measuring externalities between search advertisements. In *Proc. of ACM EC*, 2010.

[33] J. B. Sidney. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Research*, 23(2):283–298, 1975.

[34] H. Varian. Position auctions. *International Journal of Industrial Organization*, 25:1163–1178, 2007.

Appendix

Greedy algorithm for $\text{WDP}_C^{\text{pre}}$

We state the greedy algorithm for vertices and edges on a graph, noting that each bidder is associated with a vertex and the edges are such that an edge from i to j indicates that $j \in C_i$. Let $N(v)$ denote the set of in-neighbors of vertex v in the constraint graph G. The greedy algorithm proceeds as follows:

Algorithm

While there is vertex left in G
Choose the remaining vertex v with highest bid
Assign v to the highest available slot
Remove $\{v\} \cup N(v)$ from G
End

Proof of Theorem 2 . Let $Gre(v)$ and $Opt(v)$ be the value collected from vertex v in the greedy algorithm and in the optimal solution respectively. Let G' be the same constraint graph as G; we assume that it is "used" by the optimal solution in our discussion. We give an inductive proof showing that in every step of the greedy algorithm, the value of the chosen vertex v is at least $\frac{1-\delta}{1-\delta^{d+2}}$ or $1/(d+2)$ of the accumulated values of those vertices in $\{v\} \cup N(v)$ in the optimal solution (if they are assigned), and of possibly an additional vertex whose assigned slot position is at most as high as v in the greedy algorithm. In the following, we assume $\delta < 1$; the case $\delta = 1$ follows essentially the same argument.

In the base case, let the chosen vertex in the first step of the greedy algorithm be v_1^g. Let *opt* denote the optimal solution. It is obvious that $b(v_1^g) \geq b(u), \forall u \in V$. In the greedy algorithm, none of the vertices $N(v_1^g)$ can be assigned; however, they along with v_1^g, may all be assigned in *opt*. Moreover, it may be the case that none of the vertices $\{v_1^g\} \cup N(v_1^g)$ is assigned the highest slot in *opt* and that slot is occupied by another vertex $\tilde{v}_1^o \notin \{v_1^g\} \cup N(v_1^g)$. Now remove from G' all the vertices $\{v_1^g, \tilde{v}_1^o\} \cup N(v_1^g)$, and from G all the vertices $\{v_1^g\} \cup N(v_1^g)$ (so after this step, $G \supseteq G'$). We claim that

$$Gre(v_1^g) \geq \frac{1-\delta}{1-\delta^{d+2}} \sum_{u \in \{v_1^g, \tilde{v}_1^o\} \cup N(v_1^g)} Opt(u) \quad (10)$$

For this, assume that in *opt*, $u \in \{v_1^g, \tilde{v}_1^o\} \cup N(v_1^g)$ wins slot t and among all vertices in $\{v_1^g, \tilde{v}_1^o\} \cup N(v_1^g)$ that are assigned in *opt*, its position is i-th highest (thus $t \geq i$). Then

$$Gre(v_1^g) = b(v_1^g) \geq b(u) = \frac{Opt(u)}{\delta^{t-1}} \geq \frac{Opt(u)}{\delta^{i-1}} \quad (11)$$

Summing the above expression from $i = 1$ to $|\{v_1^g, \tilde{v}_1^o\} \cup N(v_1^g)|$ gives the RHS expression in equation (10). (If not all vertices $|\{v_1^g, \tilde{v}_1^o\} \cup N(v_1^g)|$ are assigned in *opt*, the RHS expression serves as an upper bound).

For the second step, we observe that the bid of vertex v_2^g in the greedy algorithm must be at least as high as all the remaining vertices in G'. This follows from the fact that at this point, $G \supseteq G'$. Moreover, the highest position of a vertex remaining in G' that is assigned in *opt* can be at most as high as 2. (It can be even lower, since it is possible that the vertex that is assigned the second slot in *opt* happens to be part of $\{v_1^g\} \cup N(v_1^g)$ and is already removed in our first step). Now we can proceed in the same way as before. We remove $\{v_2^g\} \cup N(v_2^g)$ from G and $\{v_2^g, \tilde{v}_2^o\} \cup N(v_2^g)$ from G', where $\tilde{v}_2^o$ is the vertex currently in G' that is assigned the highest position in *opt*. By essentially the same argument, we can show that

$$Gre(v_2^g) \geq \frac{1-\delta}{1-\delta^{d+2}} \sum_{u \in \{v_2^g, \tilde{v}_2^g\} \cup N(v_2^g)} Opt(u) \qquad (12)$$

Repeating the same argument, since $G \supseteq G'$, G' becomes empty before G; moreover *opt* is fully accounted for during induction. □

The "One Enemy" Special Case

A special case that remains open is that of the $\mathrm{WDP}_C^{\mathrm{pre}}$ problem with at most one constraint per bidder so that the out-degree of any vertex is at most 1. We colloquially refer to this as the "one enemy" special case. We find this case interesting because it could be achieved through a simple restriction to a bidding language. We have not been able to prove that this is NP-hard or identify a polynomial time algorithm.

In considering this problem, one initial observation is that without the directed externality constraints the problem is just that of the assignment problem. A standard linear programming (LP) formulation adopts x_{ij} to indicate whether bidder i is allocated in position j. The feasibility constraints then specify that each bidder (resp. slot) is assigned to at most one slot (resp. bidder). It is well-known that, for general values of each bidder in each slot but without additional constraints, this problem is totally unimodular and has an integral solution. Using the same encoding, a constraint $i' \in C_i$ for bidder i can be specified as a set of linear constraints of the form

$$x_{ij} + x_{i'j'} \leq 1, \text{ for all } 1 \leq j' < j \leq m$$

That is, if i is allocated in slot j, then i' cannot be allocated in a better slot j'. Let $LP2$ denote the linear program with these constraints. This linear program is no longer totally unimodular and its solutions can be fractional. Consider now an arbitrary LP, *LP3*, with a zero-one constraint matrix B. It is known that such an LP has an integral solution for *any* linear objective function if B is the clique-vertex incidence matrix of a perfect graph. Unfortunately, using the hole characterization of perfect graphs, one can show that even a reformulation *LP3* of *LP2*, strengthened by clique inequalities does not satisfy this condition and can still have fractional solutions.

Scheduling with Precedence Constraints

We note here an intriguing connection between the $\mathrm{WDP}_C^{\mathrm{pre}}$ problem and a classic problem of scheduling non-preemptive jobs on a single machine with precedence constraints to minimize total weighted, discounted completion time.

Definition 3. *Given a set $J = \{1, \ldots, n\}$ of n jobs on a single machine, where each job j is of length $p_j \geq 0$ and has weight $w_j \geq 0$, precedence constraints between jobs specified by a directed acyclic graph $G = (J, R)$ such that $(i, j) \in R$ implies that job i must be completed before job j can be started, and discount factor $r \in (0, 1)$, the* DiscountedScheduling *problem is to find a schedule that minimizes $\sum_{j=1}^n w_j(1 - e^{-rC_j})$ where C_j is the time at which job j completes. In the scheduling literature this problem is denoted* $\mathbf{1|}\boldsymbol{Prec}\mathbf{|}\sum \boldsymbol{w_j}\mathbf{(1 - }\boldsymbol{e^{-rC_j}}\mathbf{)}$ *[31].*

We establish an equivalence between a special case of our $\mathrm{WDP}_C^{\mathrm{pre}}$ problem and the DiscountedScheduling problem. In particular, we consider a scheduling problem in which jobs have unit processing times, and note that,

$$\arg\min \sum_j w_j(1-e^{-rC_j}) = -\arg\max(\sum_j w_j e^{-rC_j} - \sum_j w_j).$$

so that the optimal solution to $1|Prec|\sum w_j(1 - e^{-rC_j})$ also solves the scheduling problem with objective $\arg\max \sum_j w_j(e^{-rC_j})$. Substituting $\delta = e^{-r}$, where $0 < \delta < 1$, this is equivalent to the problem of scheduling jobs to solve $\arg\max \sum_j w_j \delta^{C_j}$, and dividing through by δ just $\arg\max \sum_j w_j \delta^{C_j - 1}$. But with processing time $p_j = 1$, then completion time C_j is equivalently the position of job j in the ordering. For the special problem of $\mathrm{WDP}_C^{\mathrm{pre}}$ *in which the constraints are acyclic and every bidder must be allocated*, we immediately see that this problem is equivalent to DiscountedScheduling for $p_j = 1$.

Scheduling: Tractable Special Cases

Some interesting special cases have been identified for which the DiscountedScheduling problem is tractable. One possible motivation to Internet advertising is to a *dispatch* problem, in which a set of winners has been determined (and all can be feasibly allocated simultaneously) but a dispatcher must determine which slot to which bidder. There is interest, for example, in using offline optimization to guide the allocation by a dispatcher of banner ads to content networks, in meeting campaign targets [30].

The *Sidney decomposition* algorithm [33] provides a framework by which to solve both the discounted and undiscounted version of the job scheduling problem. The undiscounted version seeks to allocate jobs to minimize the total weighted completion time while respecting precedence constraints. When jobs have unit processing time, this is equivalent to our WDP problem for the case of $\delta = 1$. Not only can the Sidney decomposition be used to identify polynomial time, optimal algorithms, for special structures on precedence graphs but it also provides the basis for most known 2-approximation algorithms for the undiscounted scheduling problem [4].

Given a problem instance (J, R), a Sidney decomposition partitions J into subsets $S_1, S_2, \ldots, S_n$ such that there exists an optimal schedule where jobs from S_i are processed before jobs from S_{i+1} for any $i \in \{1, \ldots, n-1\}$. The Sidney decomposition does not specify any ordering among the jobs within a set S_i. If $U \subset J$, and $V \subset J$ then U has precedence over V if there exist jobs $i \in U, j \in V$ such that $(i, j) \in R$. A set U is said to be *initial* in (J, R) if there are no jobs in $J - U$ that must be processed before jobs in U. If α is any permutation of J then α/U is the permutation induced by α on U. Accordingly U is initial if and only if there exists a feasible permutation α of the form $\alpha = (\alpha/U, \alpha/J - U)$. Let ρ be a real-valued function whose domain is the set of all subsets of J. $U \subset J$ is defined to be ρ-*minimal* for (J, R) if U is initial in (J, R), and $\rho(U) \leq \rho(V)$ for any V that is initial in (J, R). U is ρ^*-*minimal* for (J, R) if U is ρ-*minimal* for (J, R), and there is no $V \subset U$ that is ρ-minimal for (J, R).

For suitable ρ the following algorithm produces optimal permutations for deterministic problems with either linear or discounted costs.

Algorithm(ρ)

(N, R) represents the current network and α the current permutation in the algorithm. Initialize by setting $(N, R) = (J, R), \alpha = 0$.

1. Find any ρ^*-minimal subset S for (N, R).
2. Set β to be any feasible optimal ordering for (S, R).
3. Append β to the end of α.
4. Replace N by $N - S$.
5. If N is not empty then return to Step 1. If N is empty, stop. α is optimal.

For the undiscounted version of the problem, the ρ function is defined as $\rho(U) = \sum_{i \in U} p_i / \sum_{i \in U} w_i$. The main obstacle to applying this algorithm to arbitrary precedence structures lies in implementing steps 1 and 2. However Lawler [25] showed that the next subset in a Sidney decomposition (step 1) can be computed in polynomial time. The difficulty in running the algorithm suggested by Sidney is in step 2, which is NP-hard in general [25]. However for certain graph structures ("parallel chain networks, rooted trees, job-modules, and series-parallel networks") efficient algorithms exist [33]. We illustrate the workings of **Algorithm**(ρ) with the example shown in Figure 3, which is taken from [31]. This is an undiscounted weighted completion time problem. The nodes are labeled as follows: at the top is the number j of the task, on the right is the processing time p_j and on the left is the weight $w_j = 1$. The algorithm produces the optimal permutation, α, as follows:

$N = \{1, 2, \ldots, 7\}, \alpha = \emptyset$
$S = \{1, 3\}, \alpha = \{1, 3\}, N = \{2, 4, 5, 6, 7\}$
$S = \{2, 4, 5\}, \alpha = \{1, 3, 2, 5, 4\}, N = \{6, 7\}$
$S = \{6, 7\}, \alpha = \{1, 3, 2, 5, 4, 6, 7\}, N = \emptyset$

For the DiscountedScheduling problem, a more complicated ρ function is required. Glazebrook and Gittins [14] show how to define the ρ function for the discounted setting but it is not immediately apparent how to compute ρ in general problems. Monma and Sidney [26] show that series-parallel networks [7] with suitable preprocessing can be solved in $O(n \log n)$ time. Of interest to us are solvable instances where the input graph has the kind of structure that can be produced by a bidding language. Garey [9] proposes an $O(n^2)$ algorithm to solve problems with an acyclic precedence graph G in which each connected component has the property that either no task in that component has more than one immediate predecessor or no task in that component has more than one immediate successor. In particular a graph may contain some connected components that satisfy the "maximum one immediate predecessor" property while others may satisfy the "maximum one immediate successor" property. An acyclic instance of the $\text{WDP}_C^{\text{pre}}$ problem has this property as long as a bidder is allowed at most one constraint, and this observation of Garey [9] is relevant for such a instance of our problem, under the additional restriction that all bidders should be allocated.

[7] A series-parallel graph is a graph with a source and a sink that may be constructed by a sequence of series and parallel compositions.

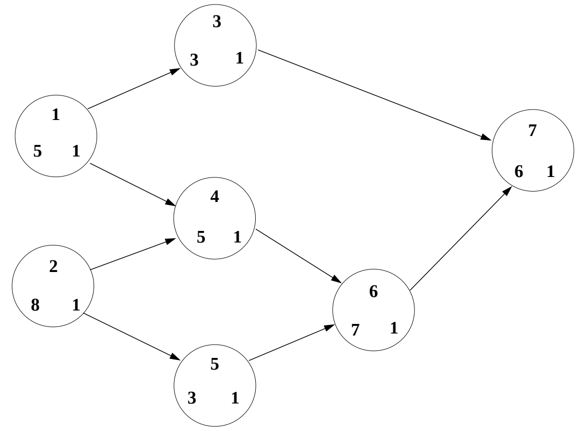

Figure 3: A precedence graph for task scheduling.

Scheduling: Discussion

From the above, we see that there is a rich literature on scheduling under precedence constraints that has strong parallels to the $\text{WDP}_C^{\text{pre}}$ problem. In particular, there are special constraint graphs for which there are fast algorithms to allocate *every* bidder to maximize discounted bid value (and thus, requiring an acyclic constraint graph for these bidders). Based on this, one natural approach for solving the $\text{WDP}_C^{\text{pre}}$ problem is to couple this with a search algorithm over sets of winners, for example by establishing a certain local property P of optimal allocations and employing algorithms tailored to P. A candidate property P that unfortunately fails is supermodularity of the optimal allocation *value*, which would couple well with existing efficient algorithms for supermodular maximization (or equivalently submodular minimization[17]). To see this failure of supermodularity, consider the example in Section 5.2: the marginal value to the optimal allocation of bidder 32 is less $(54.4 - 40)$ when added to the 40 bidder than 32, the marginal value when added to the empty set.