

Combinatorial Auctions with Conflict-Based Externalities*

Yun Kuen Cheung[†] Monika Henzinger[†] Martin Hoefer[‡]
Martin Starnberger[†]

Abstract

Combinatorial auctions (CA) are a well-studied area in algorithmic mechanism design. However, contrary to the standard model, empirical studies suggest that a bidder’s valuation often does not depend solely on the goods assigned to him. For instance, in adwords auctions an advertiser might not want his ads to be displayed next to his competitors’ ads. In this paper, we propose and analyze several natural graph-theoretic models that incorporate such negative externalities, in which bidders form a directed conflict graph with maximum out-degree Δ . We design algorithms and truthful mechanisms for social welfare maximization that attain approximation ratios depending on Δ .

For CA, our results are twofold: (1) A *lottery* that eliminates conflicts by discarding bidders/items independent of the bids. It allows to apply any truthful α -approximation mechanism for conflict-free valuations and yields an $\mathcal{O}(\alpha\Delta)$ -approximation mechanism. (2) For fractionally sub-additive valuations, we design a rounding algorithm via a novel combination of a semi-definite program and a linear program, resulting in a *cone program*; the approximation ratio is $\mathcal{O}((\Delta \log \log \Delta) / \log \Delta)$. The ratios are almost optimal given existing hardness results.

For the prominent application of adwords auctions, we present several algorithms for the most relevant scenario when the number of items is small. In particular, we design a truthful mechanism with approximation ratio $o(\Delta)$ when the number of items is only logarithmic in the number of bidders.

1 Introduction

Combinatorial auctions (CA) are an important area in algorithmic mechanism design due to wide-spread applications in resource allocation and e-commerce, e.g., spectrum

*This work was funded by the Vienna Science and Technology Fund (WWTF) through project ICT10-002. This work is supported by DFG through Cluster of Excellence MMCI.

[†]University of Vienna, Faculty of Computer Science.

[‡]Max-Planck-Institut für Informatik and Saarland University.

or adwords auctions [CSS06]. In the standard CA, a set of items is assigned to a set of bidders in order to maximize social welfare, which is given by the total valuations of bidders for their assigned items. This assumes that each bidder values exclusively the set of items assigned to him — his valuation is *independent of the assignment of other items to other bidders*. In many applications (see [JMS96, JMS99] for examples), however, such an assumption is not justified since bidder preferences have a significant dependence on how items are assigned to other bidders. Such a dependence is called *externality*.

Mechanism design for CA with externalities in the most general form is difficult, primarily due to the huge complexity of bidders’ preferences with externalities, which then also leads to the computational complexity issue for (approximately) maximizing social welfare. Prior work has studied more restricted scenarios, e.g., when there is only one item on sale, or when the bidders’ preferences are simple (e.g., unit-demand). In this work, we focus on a simple type of externalities called *conflict-based externality*, which is readily motivated by sponsored search auctions (SSA); in our model, there are multiple items on sale, and the bidders’ preferences might be more complex than unit-demand ones.

SSA are one of the most popular special cases of CA, where ad slots on a search result page are assigned to advertisers. *Negative externality* arises when, for example, a car-rental company has much smaller value for an ad slot if an ad of another prominent rental company is shown right next to it. More generally, for an advertiser there might be a number of competitors, and an assignment yields value to the bidder only if the ads of competitors are not displayed simultaneously. The existence of negative externalities in sponsored search has been confirmed empirically [GIM09]. Moreover, similar negative externalities also arise in other prominent applications of CA, e.g., in secondary spectrum auctions where interferences induce negative externalities; or when selling luxury goods, where the value of a buyer for items from an exclusive brand drops when other buyers also obtain items from the same brand. These examples give rise to a natural and simple graph-based model of externalities: each bidder is a node in a directed graph, and a directed edge indicates that a bidder sees another bidder as a competitor; assigning an item to a bidder yields value only if none of the competitors receives any item (or just any “similar” or “better” item).

Negative externalities in auctions have recently received attention, but — perhaps surprisingly — the natural and simple idea sketched above has not been analyzed in a rigorous and general fashion. We propose three graph-theoretic models that incorporate these conflict-based externalities. We study approximation algorithms and truthful mechanisms under the models. Formally, we assume there is a directed conflict graph on the set of bidders. Each edge (i, j) indicates a conflict: i has no value for any assignment in which j receives an item. More generally, we also consider cases where conflicts arise only among certain pairs of items, or different values for assignments that include or avoid certain conflicts. Our algorithms cope with externalities via new extensions of algorithmic techniques for independent set problems *in combination with* algorithms for conflict-free CA. We also provide additional results for the prominent special case of SSA. Before we state our results, we proceed with a formal introduction and discussion of the

models on conflict-based externalities treated in this paper.

1.1 Auctions with Conflict-Based Externalities

In all models, we have a bidder set B of n bidders and an item set I of m items. Each item can be given to at most one bidder. For each $i \in B$, there is a valuation function $v_i : 2^I \rightarrow \mathbb{R}^+$, where $v_i(S_i)$ represents the value for receiving item set $S_i \subseteq I$. In the SSA case, the items in I are ad slots. Each slot k has a click-through rate $\alpha_k \geq 0$. Each bidder i has a valuation per click of $v_i \geq 0$ in one slot. Then $v_i(S_i) = \max_{k \in S_i} v_i \cdot \alpha_k$, a unit demand valuation function with free disposal.

The valuation $v_i(S_i)$ will be extended, due to externalities, to $v_i^c(S)$, a valuation that depends on the complete allocation $S = (S_1, \dots, S_n)$. The goal is to find an allocation S that maximizes social welfare $SW(S) = \sum_{i \in B} v_i^c(S)$.

CA with Bidder Conflicts. The set of bidders B is the vertex set of a (*bidder*) *conflict graph* $G = (B, E)$, which is a directed graph. Each bidder i has a valuation function $v_i : 2^I \rightarrow \mathbb{R}^+$. Given a complete allocation $S = (S_1, \dots, S_n)$,

$$v_i^c(S) = \begin{cases} v_i(S_i) & \text{if } \bigcup_{j:(i,j) \in E} S_j = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

This models the situation that advertiser i is not interested in showing its ad together with an ad from a competitor j , represented by an edge $(i, j) \in E$.

The introduction of conflicts turns social welfare maximization NP-hard; in the special case SSA with all $v_i = 1$, all $\alpha_k = 1$, and $m = n$, it reduces to the maximum independent set problem.

CA with Bidder and Item Conflicts. There are two conflict structures in this model, each represented by a directed graph. The bidder set B is the vertex set of a *bidder conflict graph* $G = (B, E)$. The item set I is the vertex set of an *item conflict graph* $G_I = (I, E_I)$. Both graphs are directed. Intuitively speaking, if $(i, j) \in E$ and $(k, \ell) \in E_I$, then bidder i has no use for item k if j receives item ℓ . Formally, for any allocation S , bidder i has a set D_i of *useless items*, defined as $D_i := \{k \in S_i \mid \exists \ell \in S_j : (i, j) \in E \text{ and } (k, \ell) \in E_I\}$, and $v_i^c(S) := v_i(S_i \setminus D_i)$.

An intuitive example is *ordered conflicts*, where ad slots are ordered on a page top-down, and a bidder has a conflict only if a competitor receives a slot *above* him. This can be modelled by numbering slots top-down and $E_I = \{(k, \ell) \mid k, \ell \in I, \ell < k\}$. Another intuitive example is *neighbor conflicts*, where ad slots are arranged horizontally, and a bidder has conflict only if a competitor receives a slot *right next to* him. This can be modelled by numbering slots from left to right and $E_I = \{(k, \ell) \mid k, \ell \in I, |k - \ell| = 1\}$.

Note that CA with bidder conflicts is a sub-case of this model, when G_I is the complete digraph.

The results in this paper depend on two parameters Δ and Δ_I of the conflict graphs, which are the maximum out-degrees of the graphs G and G_I respectively.

CA with Bidder Conflicts and Conflict Value. In CA with bidder conflicts, we assume that the valuation of a bidder drops to $v_i(S) = 0$ as soon as a competitor receives any item. We can generalize this assumption to a second valuation function $w_i(S_i)$: if $\bigcup_{j:(i,j) \in E} S_j = \emptyset$, then $v_i^c(S) = v_i(S_i)$; otherwise, $v_i^c(S) = w_i(S_i)$.

This model can be reduced to the model with bidder-conflicts only. Given an instance of CA with bidder conflicts and conflict value, we build an instance without conflict value as follows: for each bidder i , we add an auxiliary bidder i_c , where $v_{i_c}(S_i) = w_i(S_i)$. In the bidder conflict graph, we add the edges (i, i_c) and (i_c, i) . This increases Δ by exactly 1. Now if bidder i is conflicted, we can take all items assigned to it and assign them to bidder i_c instead. In this way, we can transform any allocation into the instance without conflict value and obtain the same social welfare. It is straightforward to observe that social welfare maximization in both instances is equivalent. This, however, does not directly apply to truthfulness.

There are numerous further ways to extend our models, e.g., to combinations of item conflicts and conflict values, weighted conflicts, etc. Studying their properties are interesting avenues for future work.

1.2 Our Contribution

For CA with conflict-based externalities, we design and analyze poly-time approximation algorithms and truthful mechanisms which provide almost best possible approximation guarantees of maximizing social welfare. To state our results, we first define the class of *fractionally sub-additive* valuations, which is known to strictly contain the more well-known unit-demand valuations, linear valuations, gross substitute valuations and submodular valuations.

Definition 1 (See [FV10]). *A fractionally sub-additive valuation is a valuation function $v : 2^I \rightarrow \mathbb{R}$ that satisfies the following property for any $S, T_1, T_2, \dots, T_k \in 2^I$ and $0 \leq \alpha_1, \alpha_2, \dots, \alpha_k \leq 1$: if for all $j \in S$, $\sum_{\ell: j \in T_\ell} \alpha_\ell \geq 1$, then $v(S) \leq \sum_{\ell=1}^k \alpha_\ell \cdot v(T_\ell)$.*

For CA with bidder conflicts, we use well-known techniques for independent set problem to eliminate conflicts, which is in a spirit similar to *lottery*, to give a reduction to conflict-free CA. Given any α -approximation algorithm for the unconflicted problem, we obtain an $\mathcal{O}(\alpha\Delta)$ -approximation algorithm for CA with bidder conflicts (Theorem 1). If the original algorithm is a truthful mechanism, our reduction preserves the truthfulness. Moreover, our reduction preserves the use of randomization (deterministic, universally truthful, truthful in expectation). If the bidders have fractionally sub-additive valuations, our results extend to CA with bidder and item conflicts (Theorem 6).

The next natural question to ask is whether one can improve the approximation ratio to $o(\Delta)$. Since our problem generalizes the weighted independent set (WIS) problem¹, the ratio must be $\Omega(\Delta/\log^4 \Delta)$ [Cha13], even for unit-demand valuations. We answer the

¹Given any graph $G = (V, E)$, $J \subset V$ is an independent set if no two vertices in J are connected by an edge in E . The WIS problem is: suppose each vertex $v \in V$ has a positive weight w_v , find an independent set J which maximizes $\sum_{v \in J} w_v$.

question positively: if the bidders have fractionally sub-additive valuations and if there is a *demand oracle* for each bidder, we design an $\mathcal{O}((\Delta \log \log \Delta) / \log \Delta)$ -approximation algorithm (Theorem 7). This implies, for example, ratios of $\mathcal{O}((\Delta \log \log \Delta) / \log \Delta)$ for sponsored search, unit-demand, or more general gross-substitute valuations. The dependence on Δ mirrors the best-known approximation ratio for WIS. Our algorithm combines an approach for WIS based on semi-definite programming (SDP) with the standard approach for CA based on linear programming (LP) to design a cone program relaxation and a rounding scheme. To the best of our knowledge, we are the first to combine an SDP with an LP in this fashion, and to show how to analyze it. We believe this technique might be of independent interest in other applications. It is an interesting open problem if this approach can be turned into a truthful mechanism, or be generalized to CA with bidder and item conflicts.

We then focus on SSA with bidder conflicts. Even in this special case, the hardness bound of $\Omega(\Delta / \log^4 \Delta)$ applies. We consider a restriction to a small number of slots that is natural in the context of sponsored search. For the case of $m = \mathcal{O}(\log n)$ slots, we present a truthful mechanism based on SDP that obtains an $\mathcal{O}(\Delta \cdot \sqrt{(\log \log \Delta) / (\log \Delta)})$ -approximation (Theorem 17). To obtain the desired truthfulness property, the first step of our mechanism is to gather a statistic from a sampling of bidders who will not be allocated any item, which is similar the first two steps in the framework of Dobzinski et al. [DNS12] for designing truthful mechanisms. However, the subsequent steps of our algorithm will be different from theirs.

Also, we get an $\mathcal{O}(\log m)$ -approximation algorithm based on partial enumeration that runs in time $\mathcal{O}((m\Delta)^m)$ (Theorem 18); the algorithm can be turned into truthful-in-expectation mechanisms with the same approximation guarantee, and it extends to CA with bidder and item conflicts.

1.3 Related Work

The study of auctions with externalities was initiated by seminal work of Jehiel et al. [JMS96, JMS99] in the single-item setting. The externality in this work is *identity-dependent*, i.e., each bidder can have a different valuation when different bidders obtain the item. The preference of each bidder can thus be represented by a low dimensional \mathbb{R}^{n+1} vector, which reflects the bidder's valuation on the $(n+1)$ possible outcomes. In our model, a bidder is indifferent between the bidders who he conflicts with, but our model allows multiple items in an auction.

Gomes et al. [GIM09] gave empirical evidence that externalities exist in real-life SSA. Externalities in online advertising were investigated by [GM08] using a probabilistic model. CA with externalities were presented in [KMSW10, CS12, HIMM13], and maximizing social welfare was shown to be NP-hard. In [GS10] a sponsored search setting was treated where each advertiser has two valuations, one if his ad is shown exclusively and one if it is shown together with other ads. This is a special case of our model for CA with bidder conflicts and conflict values. A different line of work considered bidder-independent externalities in the click-through rates of SSA [AFMP08, KM08, RT12]. All

this work considered only the unit-demand setting.

Our model of SSA with bidder conflicts has been proposed and studied before by Papadimitriou and Garcia-Molina [PGM12]. They consider an approach based on exact optimization algorithms using ILP, implement truthfulness using VCG, and experimentally evaluate their approach with respect to running time and revenue on a dataset from the Yahoo! Webscope. However, they do not consider polynomial-time algorithms, provable approximation ratios, or extensions to CA with more general valuations.

Our work is related to approximation algorithms for weighted independent set problem, a central problem in the study of approximation algorithms and computational hardness over the past four decades. For a survey on some of the work on approximation algorithms, see, e.g., [Hal98]; here, we just mention a number of directly related results. The problem is known to be NP-hard to approximate within a ratio of $n^{1-\epsilon}$ [Hås99], and even in undirected Δ -regular graphs it remains hard for a ratio of $\mathcal{O}(\Delta/\log^4 \Delta)$ [Cha13]. A trivial greedy algorithm obtains an approximation ratio of $(\Delta + 1)$ in undirected graph with maximum degree Δ . For directed graphs, which arise in our application, a simple randomized (4Δ) -approximation algorithm exists. The best-known approximation algorithms for undirected graphs with maximum degree Δ attain ratios of $\mathcal{O}((\Delta \log \log \Delta)/\log \Delta)$ [Hal00, Hal02]. They are based on rounding suitable SDP relaxations, and below we build on these techniques and their analysis to provide algorithms for our cases, which involve directed graphs.

More recently, the study of asymmetric and edge-weighted versions of independent set has found interest, especially in the context of secondary spectrum auctions [ZGSZ08, HK12, HK13, HKV14], where bidders are wireless devices that strive to obtain channel access under interference constraints. In these scenarios, bidders become vertices in a conflict graph. Each channel is an item that can be given to any subset of bidders representing an independent set in the graph.

2 CA with Bidder and Item Conflicts via Lottery

In this section, we present results for CA with bidder and item conflicts. We assume that either (i) Δ is bounded and G_I is arbitrary, or (ii) Δ_I is bounded, G is arbitrary and bidders have fractionally sub-additive valuations.

2.1 Bounded Out-degree in the Bidder Conflict Graph

For case (i), we prove the following result:

Theorem 1. *Given a (maximal-in-range) deterministic α -approximation algorithm f for CA without conflicts, there exists a (truthful maximal-in-range) deterministic $(16\Delta\alpha/3)$ -approximation algorithm f^c for CA with bidder and item conflicts satisfying condition (i).*

The main idea of Theorem 1 is to first generate a “good” conflict-free bidder set B^c , and then apply the blackbox algorithm f w.r.t. the bidders in B^c . Initially, each bidder

is in B^c with probability $1/(2\Delta)$. Then, if there are still bidder conflicts within B^c , we remove those bidders having conflicts. Overall, each bidder is in B^c with probability of at least $1/(4\Delta)$, and this will translate into a randomized $(4\Delta\alpha)$ -approximation algorithm f^c .

We derandomize the above algorithm using the standard technique of pairwise independent distributions [LW05]; this will lead to an increase of the approximation ratio from $4\Delta\alpha$ to $16\Delta\alpha/3$. Since no bidder can alter B^c by changing his valuation, truthfulness is preserved from f to f^c . The details of derandomization is given in Appendix A.1.

Next, we discuss the details of designing the randomized $(4\Delta\alpha)$ -approximation algorithm. The results apply to arbitrary restrictions on the valuations (e.g., submodular valuations). Given an allocation $S = (S_1, \dots, S_n)$ of items to bidders, we show how to compute a random set $B^c \subseteq B$ such that (1) if $(i, j) \in E$ then $i \notin B^c$ or $j \notin B^c$, (2) $\sum_{i \in B} v_i(S) \leq (4\Delta) \mathbf{E}_{B^c} [\sum_{i \in B^c} v_i(S)]$, and (3) the selection of B^c does not depend on the valuations.

We will use pairwise independent distributions; such distributions always exist as one can pick the elements in B independently with probability q .

Definition 2. We call a distribution \mathcal{D} over subsets of a set B “pairwise independent with probability q ” if for $B^R \sim \mathcal{D}$ and $i \neq j \in B$ holds that $\Pr [i \in B^R] = q$ and $\Pr [\{i, j\} \subseteq B^R] = \Pr [i \in B^R] \cdot \Pr [j \in B^R]$.

Algorithm 1: Conflict-free random set

- 1 Pick a random subset B^R from a distribution over subsets of B that is pairwise independent with probability $1/(2\Delta)$;
 - 2 $B^c \leftarrow B^R$;
 - 3 **For each** $i \in B^R$: **if** $\exists j \in B^R$ with $(i, j) \in E$ **then** delete i from B^c ;
 - 4 **return** B^c
-

The random set B^c computed by Algorithm 1 is constructed in the following way. First, in line 1 the algorithm picks a random subset from a pairwise independent distribution with probability $1/(2\Delta)$. Next, in line 3 the algorithm resolves all the remaining conflicts between the bidders. The proof of Lemma 3 exploits that every bidder is in B^c with probability at least $1/(4\Delta)$.

Lemma 2. In Algorithm 1, every bidder is in B^c with a probability of at least $1/(4\Delta)$.

Proof: For all $i \in B$ let Q_i be the event that $i \in B^R$ in line 1. Thus, the probability for this event is $\Pr [Q_i] = 1/(2\Delta)$. The probability that a bidder $i \in B^R$ gets deleted in line 3, conditioned on that it was selected in B^R , is $\Pr \left[\bigcup_{j \in N(i)} Q_j \mid Q_i \right]$, where $N(i)$

denotes the set of out-neighbors of bidder $i \in B$.

$$\begin{aligned} \Pr \left[\bigcup_{j \in N(i)} Q_j \mid Q_i \right] &= \frac{\Pr \left[\left(\bigcup_{j \in N(i)} Q_j \right) \cap Q_i \right]}{\Pr [Q_i]} = \frac{\Pr \left[\bigcup_{j \in N(i)} (Q_j \cap Q_i) \right]}{\Pr [Q_i]} \\ &\stackrel{(*)}{\leq} \frac{\sum_{j \in N(i)} \Pr [Q_j \cap Q_i]}{\Pr [Q_i]} \stackrel{(**)}{=} \frac{\sum_{j \in N(i)} \Pr [Q_j] \Pr [Q_i]}{\Pr [Q_i]} = \sum_{j \in N(i)} \Pr [Q_j] \leq \frac{1}{2}. \end{aligned}$$

In the above (in)equalities, (*) follows from Boole's inequality and (**) follows from pairwise independence.

Thus, the probability that $i \in B^c$ at the end of the loop is

$$\Pr [Q_i] \Pr \left[\left(\bigcup_{j \in N(i)} Q_j \right)^c \mid Q_i \right] = \Pr [Q_i] \left(1 - \Pr \left[\bigcup_{j \in N(i)} Q_j \mid Q_i \right] \right) \geq \frac{1}{4\Delta}.$$

□

Lemma 3. *A (randomized) α -approximation algorithm f for CA without conflicts can be turned into a randomized $(4\Delta\alpha)$ -approximation algorithm f^c for CA with bidder and item conflicts, in which Δ bounded and G_I arbitrary.*

Proof: We define our $(4\Delta\alpha)$ -approximation algorithm f^c as follows: f^c first calls Algorithm 1 to compute a random subset B^c and then calls f for the bidders in B^c .

Assume that the α -approximation algorithm f returns the allocation $(S_1(B'), \dots, S_n(B'))$ if we use it on the set of bidders $B' \subseteq B$ and if we set $S_i(B') = \emptyset$ for all $i \notin B'$. Furthermore, assume that the optimal allocation is $(\text{OPT}_1(B'), \dots, \text{OPT}_n(B'))$ given the constraints $\text{OPT}_i(B') = \emptyset$ if $i \notin B'$. Then

$$\begin{aligned} \mathbf{E}_{B^c} \left[\sum_{i \in B^c} v_i(S_i(B^c)) \right] &\geq \mathbf{E}_{B^c} \left[\frac{1}{\alpha} \sum_{i \in B^c} v_i(\text{OPT}_i(B^c)) \right] \\ &\stackrel{(*)}{\geq} \frac{1}{\alpha} \mathbf{E}_{B^c} \left[\sum_{i \in B^c} v_i(\text{OPT}_i(B)) \right] = \frac{1}{\alpha} \sum_{i \in B} v_i(\text{OPT}_i(B)) \cdot \Pr [i \in B^c] \\ &\geq \frac{1}{4\Delta\alpha} \sum_{i \in B} v_i(\text{OPT}_i(B)). \end{aligned}$$

(*) holds because $\text{OPT}(B^c)$ gives the bidders in B^c the maximal social welfare. □

Since no bidder can alter B^c by changing his valuation, it also holds that a universally truthful (resp. truthful in expectation) α -approximation mechanism (f, p) for combinatorial auctions without conflicts can be turned by the same approach into a universally truthful (resp. truthful in expectation) $(4\Delta\alpha)$ -approximation mechanism (f^c, p^c) for combinatorial auctions with conflicts. That is, we can first call Algorithm 1 to compute a random subset B^c and can then use (f, p) only for the bidders in B^c . Note that bidders not in B^c cannot change their utility by changing their bid; it is always zero. Furthermore, bidders in B^c behave like in (f, p) , i.e. they bid truthfully so as to maximize their own utilities for each realization of B^c . The approximation guarantee follows from Lemma 3.

Corollary 4. *A universally truthful (resp. truthful in expectation) α -approximation mechanism for CA without conflicts can be turned into a universally truthful (resp. truthful in expectation) $(4\Delta\alpha)$ -approximation mechanism for CA with Δ bounded and G_I arbitrary.*

2.2 Bounded Out-degree in the Item Conflict Graph

For case (ii), the ideas are similar, but here we generate a conflict-free item set I^c using the same technique as in case (i). We require that bidders have fractionally sub-additive valuations to avoid the existence of complementary goods, which should not be deleted independently as the following example illuminates: Suppose there are two items which are *total complements* and they have item conflicts. If we remove either of the two, the social welfare drops to zero, while the optimal allocation attains a positive social welfare by allocating both items to the same bidder.

The algorithm for case (ii) and its analysis are similar to those for case (i), except that we need the following lemma about fractionally sub-additive valuations, which will imply that if each item is not discarded with probability at least $1/k$, then social welfare retained is at least $1/k$ of the original. For comparison, for case (i), we have shown that if each bidder is not discarded with probability at least $1/k$, then the eventual social welfare retained is at least $1/k$ of the original.

Lemma 5 ([Fei09, Proposition 2.3]). *Let $k \geq 1$ and let w be an arbitrary fractionally sub-additive utility function. For a set S , consider a distribution over subsets $S' \subset S$ such that each item of S is included in S' with probability at least $1/k$. Then $\mathbf{E}[w(S')] \geq w(S)/k$.*

We obtain the following theorem; we defer its proof to Appendix A.2.

Theorem 6. *For CA with bidder and item conflicts satisfying conditions (i) and (ii), given a (maximal-in-range) deterministic α -approximation algorithm f for CA without conflicts, there exists a (truthful maximal-in-range) deterministic $(16\alpha/3) \cdot \min\{\Delta, \Delta_I\}$ -approximation algorithm f^c .*

3 CA with Bidder Conflicts via Cone Program Relaxation

We design an approximation algorithm via a combination of (i) an SDP for WIS problem and (ii) an LP for conflict-free CA. This yields a *cone program*, which we round its solution to yield a good allocation. Cone program (CP) is a generalization of the more well-known LP and SDP. Briefly speaking, a CP is a program that optimizes a linear function in the intersection of some hyperspaces and a *proper cone*. More about CP will be given in Section 3.2.

To the best of our knowledge, we are the first to combine this SDP and a LP in this fashion. Also, our analysis for the cone program rounding algorithm is novel, combining

of the analyses for (i) and (ii) so as to get the benefits of both. We prove the following result:

Theorem 7. *For CA with bidder conflicts, suppose that the bidders have fractionally sub-additive (FSA) valuations. If there is a demand oracle for each bidder (we shall define this in Section 3.1), then there exists an $\mathcal{O}((\Delta \log \log \Delta) / \log \Delta)$ -approximation algorithm of social welfare that runs in $\text{poly}(m, n)$ -time.*

The rest of this section is organized as follows. We include the relevant standard facts about CA without conflicts in Section 3.1. We include the formal definition of cone program and the relevant standard facts about cone programs in Section 3.2. Then we describe our algorithm in Section 3.3, provide some intuitions of the algorithm in Section 3.4, and give its analysis in Section 3.5. We have a few remarks and discussion in Section 3.6. We slightly abuse the notation and use S to denote a subset of I .

3.1 CA with No Conflicts

The optimal social welfare of a CA without conflicts can be represented by the program ILP-NC: maximizing $\sum_{i \in B} \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S}$ subject to three sets of constraints: (i) $\forall i \in B$, $\sum_{S \neq \emptyset} x_{i,S} \leq 1$; (ii) $\forall k \in I$, $\sum_{S \ni k} \sum_{i \in B} x_{i,S} \leq 1$; (iii) $\forall i \in B$ and $\forall S \subseteq I$, $x_{i,S} \in \{0, 1\}$.

In general, solving ILP-NC is NP-hard. The usual remedy is to solve its LP relaxation LPR-NC, i.e., relaxing (iii) from $x_{i,S} \in \{0, 1\}$ to $x_{i,S} \in [0, 1]$, to obtain a *fractional* solution, and *round* it to an integral solution.

There are $\Omega(2^m n)$ variables in LPR-NC, but it can be solved in $\text{poly}(m, n)$ -time if there is a *demand oracle* for each bidder: given the prices of the items p_1, p_2, \dots, p_m , the demand oracle of bidder i returns a set $S \subseteq I$ that maximizes $v_i(S) - \sum_{k \in S} p_k$. The demand oracles serve as *separation oracles* for the dual of LPR-NC, thus allow solving LPR-NC efficiently using the ellipsoid algorithm [NS06].

For CA without conflicts where bidders have FSA valuations, a rounding algorithm called *fair contention resolution algorithm* (FCRA) [FV10, Section 1.2] attains approximation ratio $1 - \frac{1}{e}$. In Lemma 8 below, we state the precise result on FCRA, which will be useful for our conflict setting; we need the following notation: $\forall B' \subseteq B$, let $\text{LPR}(B')$ denote the program LPR-NC with the item set I and the bidder set restricted to B' . Given any feasible point $\{x_{i,S}\}_{i \in B', S \subseteq I}$ of $\text{LPR}(B')$, $\forall i \in B'$, let $L_i(\{x_{i,S}\}_{i \in B', S \subseteq I}) := \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S}$.

Lemma 8 ([FV10]). *Suppose the bidders in B have FSA valuations. Given any feasible point $\{x_{i,S}\}_{i \in B, S \subseteq I}$ of $\text{LPR}(B)$, FCRA outputs a randomized allocation in which each $i \in B$ obtains expected welfare of at least $(1 - \frac{1}{e}) \cdot L_i(\{x_{i,S}\}_{i \in B, S \subseteq I})$.*

Let $\text{FCRA}(B, \{x_{i,S}\}_{i \in B, S \subseteq I})$ denote the randomized allocation in Lemma 8. For any $B' \subseteq B$, let $\hat{x}(B')$ denote the optimal solution to $\text{LPR}(B')$.

3.2 Cone Programs in a Nutshell

Cone programs (CP) are generalization of the more familiar linear programs (LP) and semi-definite programs (SDP). We list the relevant definitions and properties of CP here. They are extracted from Chapters 2 and 5 in [BV04].

A closed set $K \subseteq \mathbb{R}^q$ is a *proper cone* if (a) for any real numbers $a_1, a_2 \geq 0$ and for any $k_1, k_2 \in K$, $a_1 k_1 + a_2 k_2 \in K$; (b) it has nonempty interior; and (c) it is pointed, i.e., if $x, -x \in K$, then $x = 0$. Two examples of proper cones are the non-negative orthant (the set of points with non-negative coordinates), and the set of symmetric positive semi-definite (SPSD) matrices.

There is a natural partial ordering on \mathbb{R}^q associated with any proper cone K , which is denoted by \succeq_K : For any $x_1, x_2 \in \mathbb{R}^q$, $x_1 \succeq_K x_2$ if and only if $(x_1 - x_2) \in K$. The corresponding *strict* partial ordering, \succ_K , is defined as follows: For any $x_1, x_2 \in \mathbb{R}^q$, $x_1 \succ_K x_2$ if and only if $(x_1 - x_2)$ is an interior point of K .

The *dual cone* of a proper cone K , is the set $K^* := \{z \mid \forall k \in K, k \cdot z \geq 0\}$, where $k \cdot z$ is the inner product of k and z . The dual cone of the non-negative orthant is itself, and the dual cone of the set of SPSP matrices is again itself [BV04, Examples 2.23 and 2.24]. The two proper cones are said to be *self-dual*.

There are various forms of CP, but they can be shown to be equivalent. In this paper, we will use two of the forms of CP. The *standard form* of CP is as follows; note that $c, x \in \mathbb{R}^q$, A is an $\ell \times q$ real matrix, $b \in \mathbb{R}^\ell$ and K is a proper cone in \mathbb{R}^q :

$$\min c \cdot x \quad \text{s.t.} \quad Ax = b \quad \text{and} \quad x \succeq_K 0. \quad (\text{CP-STD})$$

The above CP has two constraints. The first one, $Ax = b$, is called *non-conic constraint*. The second one, $x \succeq_K 0$ or equivalently $x \in K$, is called *conic constraint*.

LP is a special case of CP, in which K is the non-negative orthant. SDP is a special case of CP, in which K is the set of SPSP matrices.

As in the cases of LP and SDP, there is a dual for CP too, which is also a CP. Before describing the dual, we note that the non-conic constraint $Ax = b$ can be broken into ℓ equality constraints $A_h x = b_h$, where A_h is the h -th row of the matrix A and b_h is the h -th entry of the vector b . Each such equality constraint in the primal will associate to one distinct real variable in the dual, but the conic constraint will not associate to any dual variable. This is important since for our problem, we will introduce a CP with exponentially many variables but only $\text{poly}(m, n)$ equality constraints. Then its dual will have only $\text{poly}(m, n)$ dual variables, which is a necessary feature for using ellipsoid algorithm to solve it in $\text{poly}(m, n)$ -time.

The dual of CP-STD, in an *inequality form* of CP, is (see [BV04, Example 5.12])

$$\max b \cdot y \quad \text{s.t.} \quad c \succeq_{K^*} A^\top y. \quad (\text{CP-DUAL-INEQ})$$

We may solve CP-STD by solving its dual CP-DUAL-INEQ if strong duality holds between them. While strong duality always holds for LP, it may not hold for SDP and CP. The standard method to determine strong duality of CP is to check that *Slater's condition* holds, i.e., there exists an x such that $Ax = b$ and $x \succ_K 0$.

3.3 Algorithm

Halperin [Hal02] designed an SDP and a rounding scheme for WIS with approximation guarantee $\mathcal{O}((\Delta \log \log \Delta) / \log \Delta)$. We conglomerate his SDP with LPR-NC for our problem, which is equivalent to solving the discrete program ICP-C below.

As the constraint (1) involves a product of variables, an LP relaxation is not admissible. As LP is a subclass of SDP, one might think that an SDP relaxation suffices. However, this is not true. If we use a “fully” SDP relaxation, each constraint $x_{i,S} \leq 1$ will be converted to a non-conic constraint in the SDP relaxation. This will introduce exponentially many dual variables, prohibiting an ellipsoid algorithm on its dual to run in poly-time.

Thus, we relax to CPR-C, a “mixture” of LP and SDP; note that in CPR-C, $w_0, w_i \in \mathbb{R}^{n+1}$. CPR-C is a CP. In Appendix B, we show that strong duality holds between CPR-C and its dual, and we can solve the dual in $\text{poly}(m, n)$ -time using the ellipsoid algorithm, assuming that we have a demand oracle for each bidder. We then round the fractional solution of CPR-C as in Algorithm 2.

<p>(ICP-C)</p> $\max \sum_{i \in B} \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S}$ <p>subject to</p> $\sum_{S \neq \emptyset} x_{i,S} \leq 1, \quad \forall i \in B$ $\sum_{S \ni k} \sum_{i \in B} x_{i,S} \leq 1, \quad \forall k \in I$ $\frac{1 + w_i}{2} = \sum_{S \neq \emptyset} x_{i,S}, \quad \forall i \in B$ $(1 + w_i)(1 + w_j) = 0, \quad \forall (i, j) \in E \quad (1)$ $w_i \in \pm 1, \quad \forall i \in B$ $x_{i,S} \in \{0, 1\}, \quad \forall i \in B, S \subseteq I.$	<p>(CPR-C)</p> $\max Z := \sum_{i \in B} \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S}$ <p>subject to</p> $\sum_{S \neq \emptyset} x_{i,S} \leq 1$ $\sum_{S \ni k} \sum_{i \in B} x_{i,S} \leq 1$ $\frac{1 + w_0 \cdot w_i}{2} = \sum_{S \neq \emptyset} x_{i,S} \quad (2)$ $(w_0 + w_i) \cdot (w_0 + w_j) = 0 \quad (3)$ $\ w_0\ = \ w_i\ = 1. \quad (4)$ $x_{i,S} \geq 0.$
--	---

3.4 Intuitions on the Algorithm

Let $(Z^*, \{x^*\}, \{w^*\})$ be the solution to CPR-C. For any $B' \subseteq B$, let $Z^*(B') := \sum_{i \in B'} \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S}^*$.

We partition the bidders according to the values of $1 + w_0^* \cdot w_i^*$ into three sets B_0 , B_1 and B_2 . Items are allocated to one of the sets; the best one is chosen. The methods of allocating items to B_2 and B_1 (Steps 3 and 4) are well motivated by Halperin’s algorithm – first selecting a “good” independent subset of bidders from them, and then apply FCRA for conflict-free CA; we call this “IS-then-FCRA”. In Section 3.5, we prove that \mathcal{A}_2 and \mathcal{A}_1 attain expected social welfares of at least $(1 - \frac{1}{e}) Z^*(B_2)$ and $\Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*(B_1)$ respectively.

Algorithm 2: Approximation Algorithm via Cone Program Relaxation.

- 1 Solve CPR-C to obtain the solution $(Z^*, \{x^*\}, \{w^*\})$.
 - 2 Set $\tau \leftarrow \frac{3 \log \log \Delta}{4 \log \Delta}$, which is less than $1/2$. Partition the bidders into three sets B_0, B_1, B_2 : $B_0 = \{i \mid 0 \leq 1 + w_0^* \cdot w_i^* \leq 2\tau\}$, $B_1 = \{i \mid 2\tau < 1 + w_0^* \cdot w_i^* \leq 1\}$, $B_2 = \{i \mid 1 < 1 + w_0^* \cdot w_i^* \leq 2\}$.
 - 3 Let $J_2 = B_2$. $\mathcal{A}_2 \leftarrow \text{FCRA}(J_2, \hat{x}(J_2))$.
 - 4 For the bidders in B_1 , do as follows:
 - Project all vectors in $\{w_i^* \mid i \in B_1\}$ to $(w_0^*)^\perp$, the space orthogonal to w_0^* , then normalize them. Let $\{w'_i\}$ denote the projected normal vectors. Note that $(w_0^*)^\perp$ has dimension n , so we can treat each w'_i as an n -dimensional vector.
 - Choose a random n dimensional vector $r = (r_1, r_2, \dots, r_n)$, where each r_i follows the standard normal distribution with density function $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$.
 - Let $\gamma := (1 - 2\tau)/(2 - 2\tau)$. Let $B'_1 := \left\{i \in B_1 \mid w'_i \cdot r \geq \sqrt{\frac{2\gamma}{1-\gamma} \log \Delta}\right\}$.
 - Let $J_1 := B'_1 \setminus \{i \in B'_1 \mid \exists j \in B'_1 \text{ such that } (i, j) \in E\}$. $\mathcal{A}_1 \leftarrow \text{FCRA}(J_1, \hat{x}(J_1))$.
 - 5 For the bidders in B_0 , do as follows:
 - Let $\{q_{i,S}\}_{i \in B_0, S \subseteq I}$ denote the following distribution: $\forall S \neq \emptyset, q_{i,S} = \frac{x_{i,S}^*}{2\tau\Delta}$, and $q_{i,\emptyset} = 1 - \sum_{S \neq \emptyset} q_{i,S}$.
 - $\{T_i\}_{i \in B_0} \leftarrow \text{FCRA}(B_0, \{q_{i,S}\}_{i \in B_0, S \subseteq I})$.
 - (Conflict handling.) Let \mathcal{A}_0 denote the following allocation: for each bidder $i \in B_0$, if there exists another bidder j such that $(i, j) \in E$ and $T_j \neq \emptyset$, bidder i gets nothing in \mathcal{A}_0 ; otherwise bidder i gets T_i in \mathcal{A}_0 .
 - 6 Return the best allocation among $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$.
-

For B_0 , we face two difficulties which force us to use an approach quite different from Halperin's. Firstly, Halperin's algorithm is for undirected graph while in our application the graph is directed. Secondly, we notice that the "IS-then-FCRA" approach will not work for B_0 , and we ought to do the opposite – first apply FCRA by ignoring conflicts (see the next paragraph), and then resolve any remaining conflicts. These force us to have an analysis for B_0 quite different from Halperin's one.

We provide more intuitions for B_0 . The bidders in B_0 have low values of $\frac{1+w_0^* \cdot w_i^*}{2} = \sum_{S \neq \emptyset} x_{i,S}^*$. The values $x_{i,S}^*$ are typically viewed as probability densities. Low values of $\frac{1+w_0^* \cdot w_i^*}{2}$ allow room to "expand" these densities by a factor of $1/\tau$, where $\tau < \frac{1}{2}$. However, to handle conflicts, we ought to "dwell" these densities by a factor of $1/(2\Delta)$ afterwards. Then we apply FCRA with the "expanded then dwelled" densities to obtain a sufficiently good allocation to B_0 . These will allow us to show that \mathcal{A}_0 attains an expected social welfare of $\Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*(B_0)$.

Finally, note that $Z^*(B_0) + Z^*(B_1) + Z^*(B_2) = Z^*$, so the best among $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$

attains an expected social welfare of $\Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*$.

3.5 Analysis

We need the following notation. Recall that for any set $B' \subseteq B$, $Z^*(B') := \sum_{i \in B'} \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S}^*$. Note that $Z^*(B_0) + Z^*(B_1) + Z^*(B_2) = Z^*$; let $L^*(B')$ denote the optimal objective value of $\text{LPR}(B')$; let $x^*(B')$ denote the vector x^* restricted to bidders in B' .

Analysis on Step 3. Constraints (3) guarantee that $J_2 = B_2$ is an independent set. Since $x^*(J_2)$ is a feasible point of $\text{LPR}(J_2)$, $L^*(J_2) \geq Z^*(J_2) = Z^*(B_2)$. FCRA gives an allocation which is at least $(1 - \frac{1}{e}) L^*(J_2) \geq (1 - \frac{1}{e}) Z^*(B_2)$ in expectation.

Analysis on Step 4. We note that the selection of independent set J_1 is identical to the corresponding part in Halperin's algorithm, modulo that since we are dealing with a directed graph, we can remove fewer bidders from B'_1 .² Thus, we can follow closely to Halperin's analysis to show that $\mathbf{E}[Z^*(J_1)] = \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*(B_1)$.

Since $x^*(J_1)$ is a feasible point of $\text{LPR}(J_1)$, $L^*(J_1) \geq Z^*(J_1)$. FCRA gives an allocation which is at least $(1 - \frac{1}{e}) L^*(J_1) = \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*(B_1)$ in expectation.

Analysis on Step 5. Observe that $2\tau\Delta > 1$ for sufficiently large Δ , so the vector q , which collects $\{q_{i,S}\}_{i \in B_0, S \subseteq I}$, is a feasible point of $\text{LPR}(B_0)$.

For the analysis of this step, we need to unwind FCRA. Taking the feasible point q as input, the algorithm first selects a random set S_i for each bidder i as follows: a non-empty set S is selected with probability $q_{i,S}$, and the empty set is selected with probability $1 - \sum_{S \neq \emptyset} q_{i,S}$. Note that the random sets S_1, S_2, \dots, S_n may not be disjoint, so the algorithm next carries on a *resolution scheme* to randomly generate disjoint sets T_1, T_2, \dots, T_n , which are the sets stated in Step 5, while for all i , $T_i \subseteq S_i$.

By Lemma 8, $\mathbf{E}[v_i(T_i)]$, the expected welfare of bidder i (modulo conflicts), is at least $(1 - \frac{1}{e}) \frac{Z^*(\{i\})}{2\tau\Delta}$.

To handle conflicts, the algorithm resets the allocation of some bidders to the empty set. We will show that for each bidder $i \in B_0$, at least half of his expected welfare (modulo conflicts) is retained after conflict handling.

For every $i \in B_0$, let F_i be the event: $\forall j$ with $(i, j) \in E$, $S_j = \emptyset$. Then \overline{F}_i is the event: $\exists j$ such that $(i, j) \in E$ and $S_j \neq \emptyset$. We note that before conflict handling, for all $i \in B_0$, $\mathbf{E}[v_i(T_i) | F_i] \geq \mathbf{E}[v_i(T_i) | \overline{F}_i]$. We will prove the above inequality formally in Lemma 9 below, but it is indeed intuitive in the following sense: S_i is the set of items the bidder i competes for during the resolution scheme, thus the above inequality depicts that bidder i gets more when facing less competition from bidders he conflicts with.

Note that $\mathbf{E}[v_i(T_i)] = \mathbf{E}[v_i(T_i) | F_i] \cdot \Pr[F_i] + \mathbf{E}[v_i(T_i) | \overline{F}_i] \cdot \Pr[\overline{F}_i]$, i.e., $\mathbf{E}[v_i(T_i)]$ is a weighted average of the two conditional expectations. Since the first conditional

²Halperin's analysis is for undirected graphs, but his proof can be reused for directed graphs with little modification. In [Hal02, Lemma 5.2], if there is an edge between vertices in B'_1 , both vertices of the edge are removed from B'_1 . For directed graphs it suffices to remove the outgoing vertex only, so the bound provided in the lemma is also applicable.

expectation is larger than the second one,

$$\mathbf{E}[v_i(T_i) | F_i] \geq \mathbf{E}[v_i(T_i)] \geq \left(1 - \frac{1}{e}\right) \frac{Z^*({i})}{2\tau\Delta}.$$

Next, note that $\mathbf{Pr}[F_i] = 1 - \mathbf{Pr}[\overline{F_i}] = 1 - \mathbf{Pr}[\cup_{j:(i,j) \in E} (S_j \neq \emptyset)]$ is at least

$$\begin{aligned} 1 - \sum_{j:(i,j) \in E} \mathbf{Pr}[S_j \neq \emptyset] &= 1 - \sum_{j:(i,j) \in E} \frac{1}{2\tau\Delta} \sum_{S \neq \emptyset} x_{j,S}^* \geq 1 - \sum_{j:(i,j) \in E} \frac{1}{2\tau\Delta} \cdot \tau \quad (j \in B_0) \\ &\geq 1/2. \quad (i \text{ conflicts with at most } \Delta \text{ bidders}) \end{aligned}$$

Bidder i 's allocation is reset during conflict handling only if $\overline{F_i}$ holds. By the last two paragraphs, the expected welfare of bidder i after conflict handling is at least

$$\mathbf{E}[v_i(T_i) | F_i] \cdot \mathbf{Pr}[F_i] \geq \left(1 - \frac{1}{e}\right) \frac{Z^*({i})}{2\tau\Delta} \cdot \frac{1}{2} = \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*({i}).$$

Then the expected social welfare is at least $\sum_{i \in B_0} \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*({i}) = \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*(B_0)$.

Analysis on Step 6. The final step is to choose the best allocation among $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$, which, by the analyses of the previous three steps, is at least

$$\frac{1}{3} \left[\left(1 - \frac{1}{e}\right) Z^*(B_2) + \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot (Z^*(B_1) + Z^*(B_0)) \right] = \Omega\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right) \cdot Z^*.$$

This concludes the analysis of the algorithm.

Lemma 9. *Let S_i, S_j, T_i be as defined above. For every $i \in B_0$, let F_i be the event: $\forall j$ with $(i, j) \in E, S_j = \emptyset$. Then $\mathbf{E}[v_i(T_i) | F_i] \geq \mathbf{E}[v_i(T_i) | \overline{F_i}]$.*

Proof: Let C_i be the set of bidders j with $(i, j) \in E$. Fix S_j of bidders $j \in B_0 \setminus C_i$. For an item $k \in S_i$ and every $i' \in B_0$, let $p_{i'}(k) := \sum_{S \ni k} x_{i',S}^*$. Let $A(k) := \{i' | S_{i'} \ni k\}$.

In FCRA (see [FV10, Section 1.2]), if $|A(k)| = 1$, i.e. $A(k) = \{i\}$, then $\mathbf{Pr}[k \in T_i] = 1$; if $|A(k)| > 1$, then

$$\mathbf{Pr}[k \in T_i] = \frac{1}{\sum_{i' \in B_0} p_{i'}(k)} \left(\sum_{i' \in A(k) \setminus \{i\}} \frac{p_{i'}(k)}{|A(k)| - 1} + \sum_{i' \notin A(k)} \frac{p_{i'}(k)}{|A(k)|} \right).$$

Recall that we are fixing S_j of $j \in B_0 \setminus C_i$. If F_i holds, then $\forall j \in C_i, j \notin A(k)$. However, if $\overline{F_i}$ holds, some bidders in C_i may get into $A(k)$, i.e.,

$$(|A(k)| \text{ when } F_i \text{ holds}) \leq (|A(k)| \text{ when } \overline{F_i} \text{ holds}),$$

no matter what S_j the bidders j in C_i choose. Then it is easy to see that

$$\mathbf{Pr}[k \in T_i | F_i] \geq \mathbf{Pr}[k \in T_i | \overline{F_i}].$$

As each item is allocated independently, and each item in S_i is allocated to bidder i with higher probability when F_i holds, the lemma follows. \square

3.6 Discussion

We obtain the following proposition as a notable special case.

Proposition 10. *There is a poly-time $\mathcal{O}((\Delta \log \log \Delta) / \log \Delta)$ -approximation algorithm for the WIS problem in a directed graph G with out-degree at most Δ .*

Its proof is via a simple reduction from WIS to our problem. Consider a CA in which the bidder conflicts are represented by the directed graph G . Each bidder wants one distinct item, and his valuation of the item is the weight of his corresponding vertex in the WIS problem.

By Theorem 6, we have an $\mathcal{O}(\min\{\Delta, \Delta_I\})$ -approximation algorithm for CA with bidder and item conflicts, in which bidders have FSA valuations. An interesting open problem is whether it is possible to improve the approximation guarantee to $o(\min\{\Delta, \Delta_I\})$. We note that if each bidder has linear valuation, the problem reduces to the WIS problem in the *tensor product* of the graphs G and G_I , which might be a problem of independent interest.

Motivated by our CP relaxation CPR-C for CA with bidder conflicts, a valid CP relaxation for CA with bidder and item conflicts can be obtained by replacing (2)–(4) in CPR-C with

$$\frac{1 + w_0 \cdot w_{ik}}{2} = \sum_{S \ni k} x_{i,S}, \quad \forall i \in B, k \in I \quad (2')$$

$$(w_0 + w_{i_1 k_1}) \cdot (w_0 + w_{i_2 k_2}) = 0, \quad \forall (i_1, i_2) \in E, (k_1, k_2) \in E_I \quad (3')$$

$$\|w_0\| = \|w_{ik}\| = 1, \quad \forall i \in B, k \in I. \quad (4')$$

However, we do not see a good rounding algorithm for this relaxation.

4 Sponsored Search with Limited Number of Slots

In this section we consider sponsored search with bidder conflicts. Some of our results extend to ordered conflicts and more general graph-based slot conflicts. In light of the application, we concentrate on the case with a small number m of slots. Note that a trivial enumeration solves the problem in time $\mathcal{O}(n^m)$. Moreover, it is unlikely that significantly faster algorithms exist that solve the problem exactly, even for $m \leq \log n$; it is W[1]-hard to decide Log-Independent-Set, i.e., given $k \leq \log n$, deciding if G has an independent set of size at least k cannot be done in time $f(k) \cdot n^c$ for constant c unless $\text{FPT} = \text{W}[1]$ [DF99]. Thus, we present two approximation algorithms. The first one uses semi-definite programming and has polynomial running time for $m \in \mathcal{O}(\log n)$. The second one is a partial enumeration approach and runs in polynomial time if $m \in \mathcal{O}((\log n) / (\log \max(\Delta + 1, \log n)))$.

Algorithm 3: Sponsored search auction with conflicts

- 1 Assign all bidders in B independently with probability $1/2$ to set B_1 and set $B_2 \leftarrow B \setminus B_1$. With $v_1 \geq v_2 \geq \dots \geq v_n$ and $h = |B_1|$ define the functions $\phi : [h] \rightarrow [n]$ and $\chi : [n - h] \rightarrow [n]$ such that $B_1 = \{\phi(1), \dots, \phi(h)\}$ and $\phi(j) < \phi(j + 1)$ for $j \in [h - 1]$ and $B_2 = \{\chi(1), \dots, \chi(n - h)\}$ and $\chi(j) < \chi(j + 1)$ for $j \in [n - h - 1]$;
 - 2 Set $q \leftarrow 1$ with probability $\frac{1}{2}$ and set $q \leftarrow 2$ otherwise;
 - 3 **if** $n - h \geq \lceil \frac{m}{4} \rceil + 1$ **then** $t \leftarrow \chi(\lceil \frac{m}{4} \rceil + 1)$ **else** $t \leftarrow \infty$ and $v_t \leftarrow 0$;
 - 4 ;
 - 5 **if** $q = 1$ **then**
 - 6 Set $r_1 \leftarrow v_t$; Set $B_1^1 \leftarrow \{\phi(j) | j \in [h] \text{ and } \phi(j) < t\}$;
 - 7 **if** $t \leq m + 1$ **then** set \mathcal{A} to the set of all subsets of B_1^1 **else** $\mathcal{A} \leftarrow \emptyset$;
 - 8 **else**
 - 9 Set $r_2 \leftarrow v_t \cdot \frac{1}{8} R(\Delta)$; Set $B_1^2 \leftarrow \{\phi(j) | j \in [h] \text{ and } v_{\phi(j)} \geq r_2\}$;
 - 10 Set $J \leftarrow$ (unweighted) independent set in B_1^2 computed by using the WIS algorithm (Proposition 10) giving bidders in B_1^2 in random order and with equal weights; $\mathcal{A} \leftarrow \{J\}$;
 - 11 Add m bidders without conflicts and with valuation r_q to B and each set in \mathcal{A} ;
 - 12 For each set $A \in \mathcal{A}$ let $\mathcal{M}(A)$ define all the conflict-free matchings of bidders in A to slots; define $\mathcal{M} = \bigcup_{A \in \mathcal{A}} \mathcal{M}(A)$;
 - 13 Select allocation $M' \in \arg \max_{M \in \mathcal{M}} \sum_{i \in B} v_i(M)$;
 - 14 Every real-bidder a in B pays $p_a \leftarrow \max_{M \in \mathcal{M}} \sum_{i \in B \setminus \{a\}} (v_i(M) - v_i(M'))$;
-

4.1 Sponsored Search via Semidefinite Programming

We study sponsored search with bidder conflicts and $m \in \mathcal{O}(\log n)$. We assume for simplicity that $n \geq m \geq 6$. If $m > n$, we could add $(m - n)$ dummy bidders with valuation zero. We assume consistent tie-breaking among bidders with the same valuation. Recall that in this setting bidders have unit demands, and thus we can represent an allocation S of slots to bidders by a matching M_S in a bipartite bidder-slot-graph. We define $v_i(M_S) = v_i(S)$ for all $i \in B$. We call a matching M_S *conflict-free* if $D_i \cap S_i = \emptyset$ for all $i \in B$. Note that for every matching there exists a conflict-free matching with the same social welfare; we simply unassign all the slots in $\bigcup_{i \in B} D_i \cap S_i$. Furthermore, we define the expected social welfare $SW(M) := \mathbf{E}[\sum_{i \in B} v_i(M)]$ for a (randomized) matching M . In the following, we also use the notation $R(\Delta) := \sqrt{\log \log \Delta / \log \Delta}$.

The mechanism is presented in Algorithm 3 and its approximation guarantee is analyzed in Lemma 11. Let t , r_1 , r_2 , B_1^1 , and B_1^2 be defined as in Algorithm 3. We show that if the optimal conflict-free assignment of bidders to slots OPT was restricted to a random subset OPT'' of the $t - 1$ most valuable edges, where each of those edges is picked with probability $1/2$, then $SW(\text{OPT}'') \geq SW(\text{OPT})/16$. Thus, it suffices to compare the performance of a mechanism with OPT''. We run two different mechanisms, ALG_1 and ALG_2 , each with probability $1/2$, and receive at least $1/2$ of the maximum of their social

welfares SW_1 and SW_2 , respectively.

If ALG_1 performs very well, i.e., if $SW_1 > SW(\text{OPT}'')/(\Delta R(\Delta))$, we achieve the result promised in Lemma 11. Mechanism ALG_1 tries out all possibilities to find the best non-conflicting matching for bidders in B_1^1 . If ALG_1 does not perform very well, we can show that OPT'' must get at least a quarter of its social welfare from bidders in $B_1^2 \setminus B_1^1$. In this case, we build an (unweighted) independent set J of all bidders in B_1^2 using the WIS algorithm described in Proposition 10, which guarantees that the number of bidders in J is at least an $\mathcal{O}(1/(\Delta R(\Delta)^2))$ -fraction of the optimal number for bidders in $B_1^2 \setminus B_1^1$. As in OPT'' every bidder in $B_1^2 \setminus B_1^1$ contributes at most with valuation r_1 to $SW(\text{OPT}'')$ and in ALG_2 every bidder in J contributes at least with valuation r_2 to SW_2 , the overall approximation ratio of ALG_2 is $\mathcal{O}(\Delta R(\Delta)^2 \cdot r_1/r_2) = \mathcal{O}(\Delta R(\Delta))$.

Lemma 11. *The matching M' computed in Algorithm 3 is in expectation an $\mathcal{O}(\Delta \sqrt{\log \log \Delta / \log \Delta})$ -approximation of the optimal social-welfare.*

To proceed, we need the lemma below.

Lemma 12. *It holds that $\Pr[t \leq m + 1] \geq 3/4$.*

Proof: Note that $t \leq m + 1$ if and only if $|B_2 \cap [m + 1]| \geq \lceil \frac{m}{4} \rceil + 1$. This happens with probability $1 - \frac{1}{2^{m+1}} \sum_{\ell=0}^{\lceil \frac{m}{4} \rceil} \binom{m+1}{\ell}$, which is at least $3/4$ when $m \geq 6$. \square

Proof of Lemma 11: Assume that the social-welfare-maximizing conflict-free matching of bidders in B to slots is given by OPT . The valuation of the dummy bidders will not be considered in the social welfare as they were only included to guarantee truthfulness.

We first analyze the random partition of B into B_1 and B_2 by the mechanism. Let m^* be the number of edges in OPT and let us denote those edges by $(i(1), j(1)), \dots, (i(m^*), j(m^*))$ such that they are ordered by their value, i.e., $v_{i(1)} \cdot \alpha_{j(1)} \geq \dots \geq v_{i(m^*)} \cdot \alpha_{j(m^*)}$. Let OPT' be the random subset of OPT where all the edges but the $t - 1$ most valuable ones are discarded, i.e., $\text{OPT}' = \{(i(1), j(1)), \dots, (i(t-1), j(t-1))\}$. Furthermore, let OPT'' be the random subset of OPT' where (1) all the edges that contain bidders in B_2 are discarded and (2) if $t > m + 1$ all edges are discarded. We will show that $SW(\text{OPT}) \leq 16 \cdot SW(\text{OPT}'')$. Since, $B_2 \subseteq B$ it holds that $t \geq \lceil m/4 \rceil + 1 \geq m/4 + 1$, and thus, it follows by $m^* \leq m$ that

$$\frac{SW(\text{OPT})}{SW(\text{OPT}')} = \frac{\sum_{s=1}^{t-1} v_{i(s)} \cdot \alpha_{j(s)} + \sum_{s=t}^{m^*} v_{i(s)} \cdot \alpha_{j(s)}}{\sum_{s=1}^{t-1} v_{i(s)} \cdot \alpha_{j(s)}} \leq 1 + \frac{\sum_{s=t}^{m^*} v_{i(s)} \cdot \alpha_{j(s)}}{\sum_{s=1}^{t-1} v_{i(s)} \cdot \alpha_{j(s)}} = \frac{m^*}{t-1} \leq 4.$$

Now, for all $i \in B$ let E_i be the event that bidder i is not in B_1 and let T be the event that $t > m + 1$. By Lemma 12, it holds for each bidder $i \in B$ that $\Pr[E_i \cup T] \leq \Pr[E_i] + \Pr[T] \leq 1/2 + 1/4 = 3/4$. Thus, $SW(\text{OPT}'') = \sum_{s=1}^{t-1} v_{i(s)} \cdot \alpha_{j(s)} \cdot (1 - \Pr[E_{i(s)} \cup T]) \geq (1/4) \cdot SW(\text{OPT}')$. It follows that $SW(\text{OPT}'') \geq (1/16) \cdot SW(\text{OPT})$.

We will now compare the outcome M' of the mechanism with OPT'' . Let M_1 or M_2 be the matching computed by the mechanism under the condition $q = 1$ or $q = 2$, respectively. It holds that $2 \cdot SW(M') \geq \max\{SW(M_1), SW(M_2)\}$. Then the following claim completes the proof.

Claim 13. For some constant $c > 1$ it holds that

$$c \cdot \Delta R(\Delta) \cdot \max\{SW(M_1), SW(M_2)\} \geq SW(\text{OPT}''). \quad (5)$$

Proof: Notice that if $SW(\text{OPT}'') < \max\{4, \Delta R(\Delta)\} \cdot SW(M_1)$ then (5) is satisfied. Thus, we assume that $SW(\text{OPT}'') \geq \max\{4, \Delta R(\Delta)\} \cdot SW(M_1)$. Moreover, we assume that $t \leq m + 1$, as otherwise, $SW(\text{OPT}'') = 0$.

Next, we define by SW_u the optimal social welfare for bidders in B_1^1 when bidder conflicts are ignored. Furthermore, note that Theorem 3 implies that $4\Delta \cdot SW(M_1) \geq SW_u$. Thus,

$$SW(\text{OPT}'') \geq \Delta R(\Delta) \cdot SW(M_1) \geq \Delta R(\Delta) \cdot SW_u / (4\Delta) \geq (R(\Delta)/4) \cdot SW_u \quad (6)$$

Let us now partition the matching OPT'' into OPT_1 that contains the edges to bidders in B_1^1 , OPT_2 that contains the edges to bidders in $B_1^2 \setminus B_1^1$, and $\text{OPT}_3 := \text{OPT}'' \setminus (\text{OPT}_1 \cup \text{OPT}_2)$. Thus, $SW(\text{OPT}'') = SW(\text{OPT}_1) + SW(\text{OPT}_2) + SW(\text{OPT}_3)$.

As the matching OPT_1 is considered when computing M_1 , $SW(M_1) \geq SW(\text{OPT}_1)$, and thus, by the assumption taken above holds that

$$SW(\text{OPT}_1) \leq SW(M_1) \leq SW(\text{OPT}'') / \max\{4, \Delta R(\Delta)\} \leq SW(\text{OPT}'') / 4.$$

Furthermore, $SW(\text{OPT}_3) \leq SW(\text{OPT}'') / 2$, as otherwise,

$$SW(\text{OPT}'') < 2 \cdot SW(\text{OPT}_3) < 2 \cdot r_2 \cdot \sum_{j=1}^{t-1} \alpha_j = 2 \cdot \frac{1}{8} \cdot R(\Delta) \cdot v_t \cdot \sum_{j=1}^{t-1} \alpha_j \leq \frac{R(\Delta)}{4} SW_u,$$

which contradicts (6). Hence, $SW(\text{OPT}_2) \geq SW(\text{OPT}'') / 4$. It follows that

$$\begin{aligned} \frac{SW(\text{OPT}_2)}{SW(M_2)} &= \frac{\sum_{(i,j) \in \text{OPT}_2} v_i \cdot \alpha_j}{\sum_{(i,j) \in M_2} v_i \cdot \alpha_j} \leq \frac{r_1 \cdot \sum_{j=1}^{|\text{OPT}_2|} \alpha_j}{r_2 \cdot \sum_{j=1}^{|M_2|} \alpha_j} \\ &\leq \frac{8}{R(\Delta)} \cdot \left(1 + \frac{\sum_{j=|M_2|+1}^{|\text{OPT}_2|} \alpha_{|M_2|}}{\sum_{j=1}^{|M_2|} \alpha_{|M_2|}} \right) = \frac{8}{R(\Delta)} \cdot \frac{|\text{OPT}_2|}{|M_2|} \\ &\leq \frac{8}{R(\Delta)} \cdot c' \cdot (\Delta \cdot R(\Delta))^2 = 8c' \cdot \Delta \cdot R(\Delta), \quad (7) \end{aligned}$$

where $c' \Delta R(\Delta)^2$ is the approximation factor of Proposition 10. \square \square

We show that the mechanism runs in $\text{poly}(n, \Delta)$ time for certain restrictions on the number of slots m , and it is universally truthful. The crucial idea for showing truthfulness is to prove that no bidder has an incentive to alter the set of matchings \mathcal{M} . Thus, even though the range of allocations \mathcal{M} depends on the valuations of the bidders, no bidder has an incentive to change it.

Proposition 14. If $m \in \mathcal{O}(\log n)$ the mechanism takes time $\text{poly}(n, \Delta)$.

Proof: We have to show that line 13 and 14 can be computed in polynomial time in n and Δ .

We first argue that $|\mathcal{A}|$ is polynomial in n . Consider the case where $q = 1$. If $t > m+1$ then $\mathcal{A} = \emptyset$; otherwise, $\mathcal{A} = \mathcal{P}(B_1^1)$ and $|B_1^1| < t \leq m+1$. Moreover, if $q = 2$ then $|\mathcal{A}| = 1$. Thus, $|\mathcal{A}|$ is bounded by 2^m which is polynomial in n .

Next, assume that we are given some $A \in \mathcal{A}$. Given $q = 1$, we can ignore A if bidders in A have conflicts, because we know that there exists a conflict-free set of bidders in \mathcal{A} that is optimal. Moreover, given $q = 2$ we know that the bidders in all sets in \mathcal{A} have no conflicts. Thus, we can assume that the bidders in A are conflict-free. It follows that computing $\arg \max_{M \in \mathcal{M}(A)} \sum_{i \in C} v_i(M)$ can be done in polynomial time in n for all $C \subseteq B$; for all $i \in [m]$ the bidder in A with the i -th largest index has to be matched to the i -th slot. \square

Lemma 15. *No bidder has an incentive to report a non-truthful bid that alters \mathcal{M} .*

Proof: We can restrict the proof to bidders in B_1 as the other bidders always have utility zero. Assume that all bidders bid truthful. In both cases, $q = 1$ and $q = 2$, bidders not in B_1^q are in no matching in \mathcal{M} , and thus, they are not in M' and their utility is zero. However, they have no incentive to increase their bid because there are m competing dummy-bidders that have a valuation that is at least the same as theirs. Thus, if they increase their valuation, their utility cannot increase because they have to pay their externality. Furthermore, in both cases, $q = 1$ and $q = 2$, bidders in B_1^q have two possibilities: (i) Bidding high enough to stay in B_1^q and (ii) bidding below the value that is necessary for staying in B_1^q . In (i), if a bidder bids high enough to stay in B_1^q , he cannot affect B_1^q . Moreover, he cannot affect the outcome of the WIS algorithm by his bid because we randomized the order of the bidders. Thus, he cannot influence whether he belongs to a subset in \mathcal{A} and, in turn, he cannot influence \mathcal{M} . In (ii), if a bidder bids below the value that is necessary for staying in B_1^q , he will receive nothing and has utility zero. Thus, no bidder has an incentive to change his bid if this alters \mathcal{M} . \square

Lemma 16. *The mechanism is universally truthful.*

Proof: We can assume that all random decision are taken before the bidders report their bids. We first fix a bidder a . Since by Lemma 15 no bidder has an incentive to report a non-truthful bid that changes \mathcal{M} , we can restrict the analysis to bidder a 's non-truthful bids that do not change \mathcal{M} . Thus, we can consider \mathcal{M} as fixed. The utility of a bidder a for a matching M' is given by $u_a(M') = v_a(M') - (\max_{M \in \mathcal{M}} \sum_{i \in B \setminus \{a\}} (v_i(M) - v_i(M'))) = \sum_{i \in B} v_i(M') - \max_{M \in \mathcal{M}} \sum_{i \in B \setminus \{a\}} v_i(M)$ which is maximized when a bids truthful. \square

The following theorem follows from Lemma 11, Proposition 14 and Lemma 16.

Theorem 17. *For sponsored search with bidder conflicts and $m \in \mathcal{O}(\log n)$, Algorithm 3 is a universally-truthful mechanism that attains approximation guarantee of $\mathcal{O}(\Delta \sqrt{\log \log \Delta / \log \Delta})$. It runs in time $\text{poly}(n, \Delta)$.*

4.2 Sponsored Search via Partial Enumeration

We treat a slightly more general *small-supply* case with $m \leq n/(\Delta + 1)$. For this case we observe that the problem can be solved optimally in linear time when all bidders i have uniform values $v_i = v$. For non-uniform values v_i , we will strive for a truthful mechanism that solves the problem approximately but much faster than the trivial enumeration that solves the problem exactly in $\mathcal{O}(n^m)$ time. Note that there is an m -approximation algorithm that assigns slot 1 to the highest bidder, obtains value $\max_{k,i} \alpha_k \cdot v_i$, and runs in time $\mathcal{O}(n)$. Thus, we obtain the following trade-off.

Theorem 18. *In sponsored search with bidder and slot conflicts, there is a universally-truthful mechanism that yields an $\mathcal{O}(\log m)$ -approximation of social welfare and runs in time $\mathcal{O}(n + (m(\Delta + 1))^m)$.*

We first prove the existence of an approximation algorithm achieving the claimed approximation ratio.

Lemma 19. *In sponsored search with bidder conflicts, there is an $\mathcal{O}(\log m)$ -approximation algorithm that runs in time $\mathcal{O}(n + (m(\Delta + 1))^m)$.*

Proof: The algorithm is extremely simple for uniform values $v_i = v$ for all $i \in B$ if $m \leq n/(\Delta + 1)$. Initially, every bidder is active. We assign slot 1 to the bidder i with smallest out-degree, label i and its all out-neighbors to be inactive. We repeat this procedure with slots $2, 3, \dots, m$. Since $m \leq n/(\Delta + 1)$, we will be able to assign all slots in this way. This yields an optimum solution and takes time $\mathcal{O}(n)$. If the v_i are different, we apply logarithmic scaling. Let $v_{\max} = \max_{i \in B} v_i$. We consider $\lceil \log_2(2m) \rceil$ classes, where class k contains bidders i with value $v_i \in (v_{\max}/2^k, v_{\max}/2^{k-1}]$. The unclassified bidders have a value which is at most $v_i \leq v_{\max}/(2m)$. Thus, by discarding this set of bidders, we discard at most $1/2$ of the optimum value.

For the remaining bidders, we pick $k \in \{1, 2, \dots, \lceil \log_2(2m) \rceil\}$ uniformly at random and consider $V_k = \{i \in B \mid v_i > v_{\max}/2^k\}$, the union of all bidders in classes $1, \dots, k$. Let $n_k = |V_k|$. If $n_k/(\Delta + 1) \geq m$, then we can apply the above algorithm for identical values to V_k . Otherwise, if $n_k/(\Delta + 1) \leq m$, then $n_k \leq (\Delta + 1)m$, and a complete enumeration takes time at most $\mathcal{O}((m(\Delta + 1))^m)$. In either case, we obtain the optimum for V_k under the assumption that every bidder has value $v_{\max}/2^k$, and hence at least half of the value that the optimum gets from bidders in class k . In expectation over the random choice of k , this shows that we recover an $\mathcal{O}(\log m)$ -fraction of the optimum.

The highest valuation can be found in time $\mathcal{O}(n)$. Computing the threshold and reducing the set of considered bidders can be done in time $\mathcal{O}(n)$. Applying the previous algorithm can be done in time $\mathcal{O}(n)$, enumeration takes time $\mathcal{O}((m(\Delta + 1))^m)$. \square

Note that for a particular choice of k , the algorithm described in the proof of Lemma 19 is applied in the induced subgraph of V_k and produces an optimum solution under the assumption that all nodes have the same valuation. If this results from the greedy algorithm for the independent set of bidders, it also remains an optimum solution with arbitrary additional slot conflicts. If this results from enumeration, we can apply the enumeration also for additional slot conflicts in the same asymptotic running time. Thus, we obtain

the same running time and approximation ratio also for sponsored search with bidder and slot conflicts.

By the sampling arguments in [DNS12, HK13] we can turn the algorithm into a universally truthful mechanism with the same asymptotic running time and approximation ratio. The idea is as follows. First, choose a random bit q . If $q = 0$, partition B into B_1 and B_2 randomly and set v_{\max} be the highest valuation in B_1 . However, we run the algorithm in Proposition 19 on B_2 only; if bidder $i \in B_2$ gets assigned slot ℓ he has to pay $\alpha_\ell \cdot v_{\max}/2^k$. If $q = 1$, we keep the best slot and remove all others, and run a second price auction among all bidders in B . This ensures that the claimed approximation ratio even if there is a dominant bidder, i.e., a bidder who contributes at least a constant fraction of the optimal social welfare. This completes the proof of Theorem 18.

References

- [AFMP08] Gagan Aggarwal, Jon Feldman, S. Muthukrishnan, and Martin Pál. Sponsored search auctions with markovian users. In *4th WINE*, pages 621–628, 2008.
- [BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [Cha13] Siu On Chan. Approximation resistance from pairwise independent subgroups. In *45th STOC*, pages 447–456, 2013.
- [CS12] Vincent Conitzer and Tuomas Sandholm. Computing optimal outcomes under an expressive representation of settings with externalities. *J. Comput. Syst. Sci.*, 78(1):2–14, 2012.
- [CSS06] Peter Cramton, Yoav Shoham, and Richard Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [DD13] Shahar Dobzinski and Shaddin Dughmi. On the power of randomization in algorithmic mechanism design. *SIAM J. Comput.*, 42(6):2287–2304, 2013.
- [DF99] Rod Downey and Michael Fellows. *Parametrized Complexity*. Springer Verlag, 1999.
- [DNS12] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. *J. Comput. Syst. Sci.*, 78(1):15–25, 2012.
- [dOF14] Fernando Mário de Oliveira Filho. The ellipsoid method. <http://www.ime.usp.br/~fmario/sdp/ell.pdf>, 2014.
- [Fei09] Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM J. Comput.*, 39(1):122–142, 2009.

- [FV10] Uriel Feige and Jan Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6(1):247–290, 2010.
- [GIM09] Renato Gomes, Nicole Immorlica, and Evangelos Markakis. Externalities in keyword auctions: An empirical and theoretical assessment. In *5th WINE*, pages 172–183, 2009.
- [GLS88] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer, 1988.
- [GM08] Arpita Ghosh and Mohammad Mahdian. Externalities in online advertising. In *17th WWW*, pages 161–168, 2008.
- [GS10] Arpita Ghosh and Amin Sayedi. Expressive auctions for externalities in online advertising. In *19th WWW*, pages 371–380, 2010.
- [Hal98] Magnús Halldórsson. A survey on independent set approximations. In *1st APPROX*, pages 1–14, 1998.
- [Hal00] Magnús M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Alg. Appl.*, 4(1):1–16, 2000.
- [Hal02] Eran Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM J. Comput.*, 31(5):1608–1623, 2002.
- [Hås99] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.
- [HIMM13] Nima Haghpanah, Nicole Immorlica, Vahab Mirrokni, and Kamesh Mungala. Optimal auctions with positive network externalities. *ACM Trans. Econ. Comput.*, 1(2):13:1–13:24, 2013.
- [HK12] Martin Hoefer and Thomas Kesselheim. Secondary spectrum auctions for symmetric and submodular bidders. In *13th EC*, pages 657–671, 2012.
- [HK13] Martin Hoefer and Thomas Kesselheim. Brief announcement: Universally truthful secondary spectrum auctions. In *25th SPAA*, pages 99–101, 2013.
- [HKV14] Martin Hoefer, Thomas Kesselheim, and Berthold Vöcking. Approximation algorithms for secondary spectrum auctions. *ACM Trans. Internet Techn.*, 14(2–3):16, 2014.
- [JMS96] Philippe Jehiel, Benny Moldovanu, and Ennio Stacchetti. How (not) to sell nuclear weapons. *The American Economic Review*, 86:814–829, 1996.
- [JMS99] Philippe Jehiel, Benny Moldovanu, and Ennio Stacchetti. Multidimensional mechanism design for auctions with externalities. *J. Econom. Theory*, 85:258–293, 1999.

- [KM08] David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *4th WINE*, pages 585–596, 2008.
- [KMSW10] Piotr Krysta, Tomasz P. Michalak, Tuomas Sandholm, and Michael Wooldridge. Combinatorial auctions with externalities. In *9th AAMAS*, pages 1471–1472, 2010.
- [LW05] Michael Luby and Avi Wigderson. Pairwise independence and derandomization. *Foundations and Trends in Theoretical Computer Science*, 1(4):237–301, 2005.
- [Nis07] Noam Nisan. Introduction to mechanism design (for computer scientists). In *Algorithmic game theory*, pages 209–241. Cambridge University Press, 2007.
- [NS06] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *J. Econom. Theory*, 129(1):192–224, 2006.
- [PGM12] Panagiotis Papadimitriou and Hector Garcia-Molina. Sponsored search auctions with conflict constraints. In *5th WSDM*, pages 283–292, 2012.
- [RT12] Tim Roughgarden and Éva Tardos. Do externalities degrade gsp efficiency. In *8th Ad-Auctions Workshop*, 2012.
- [ZGSZ08] Xia Zhou, Sorabh Gandhi, Subhash Suri, and Haitao Zheng. eBay in the Sky: Strategy-proof wireless spectrum auctions. In *14th MobiCom*, pages 2–13, 2008.

A Proofs and Results Omitted in Section 2

A.1 Derandomization

It is crucial for the derandomization to show that there exists a pairwise independent distribution of subsets of B with probability $1/2^{\lceil \log_2 \Delta \rceil + 1}$ that has a domain with a cardinality polynomial in n , and this follows from [LW05, Section 1.2]. The idea is to represent the distribution by a randomization over a family of 2-universal hash functions that assigns each bidder u.a.r. values from the set $\{t_1, \dots, t_{2\Delta}\}$ and to consider the subset of bidders with value t_1 . Furthermore, this implies that also a distribution over the random sets B^c computed by Algorithm 1 exists that has a domain with a cardinality polynomial in n .

Lemma 20 ([LW05]). *Given a set B with $|B| = n$, for any integer $1 \leq \Delta \leq n$, there exists a pairwise independent distribution over subsets of B with probability $1/2^{\lceil \log_2 \Delta \rceil + 1}$ and a domain with a cardinality in $\mathcal{O}(n^2)$.*

Thus, instead of picking a subset from a pairwise independent distribution with probability $1/2^{\lceil \log_2 \Delta \rceil + 1}$ in Algorithm 1, we can iterate over the domain of the distribution in polynomial time. This gives us at least the same social welfare. Moreover, note that $1/(2\Delta) \geq 1/2^{\lceil \log_2 \Delta \rceil + 1} > 1/(4\Delta)$. By slightly modifying the proof of Lemma 2, we can show that every bidder is in B^c with a probability of at least $3/(16\Delta)$.

Corollary 21. *A deterministic α -approximation algorithm f for combinatorial auctions without conflicts can be turned into a deterministic $(16\Delta\alpha/3)$ -approx. algorithm f^c for combinatorial auctions with Δ bounded and G_I arbitrary.*

Furthermore, we can extend the results to *maximal-in-range* algorithms (see [DD13]) which are important for the design of truthful approximation mechanisms. In fact, most deterministic truthful approximation mechanisms for combinatorial auctions are maximal-in-range mechanism.

Definition 3. *An algorithm f is called “maximal-in-range” if there exists a subset A' of the set of allocations A for which $f(v_1, \dots, v_n) \in \arg \max_{S \in A'} (\sum_{i \in B} v_i(S))$.*

Given a maximal-in-range algorithm f for CA without conflicts that is a deterministic α -approximation algorithm, we show in Algorithm 4 how to construct a maximal-in-range algorithm f^c for CA with conflicts that is a deterministic $(16\Delta\alpha/3)$ -approximation algorithm. Hence, Theorem 1 follows. Note that Algorithm 4 calls f always for the same set of bidders and set of items, only the valuations of the bidders change; thus, the target set A' of f is the same in each call.

Proof of Theorem 1: Let us assume that OPT was set to its final value when $B' = B^*$. Furthermore, assume that B^c is a random subset computed by Algorithm 1. It follows

Algorithm 4: Maximal-in-range algorithm f^c over range A'_D .

```

1 Let  $D$  be the domain of a distribution over the random set  $B^c$  computed by
  Algorithm 1 satisfying  $|D| \in \mathcal{O}(n^2)$ ;
2 Let  $A' \subseteq A$  be the target set of  $f$ ;
3 Set  $\text{OPT} \leftarrow (\emptyset, \dots, \emptyset)$ ;
4 foreach  $B' \in D$  do
5   forall the  $i \in B$  do set  $v_i^{B'} \leftarrow v_i$  if  $i \in B'$  and  $v_i^{B'} \leftarrow 0$  else;
6   ;
7   Set  $\text{OPT}(B') \leftarrow f(v^{B'})$ ;
8   if  $\sum_{i \in B} v_i^{B'}(\text{OPT}_i(B')) \geq \sum_{i \in B} v_i(\text{OPT}_i)$  then
9     Set  $\text{OPT}$  to the following assignment:
10    (1) all  $i \in B'$  get the same items as in  $\text{OPT}(B')$ ; and (2) others get no
11    items;
12  end
13 end
14 return  $\text{OPT}$ 

```

that

$$\begin{aligned}
\sum_{i \in B} v_i(\text{OPT}_i) &= \sum_{i \in B} v_i^{B^*}(\text{OPT}_i) = \sum_{i \in B} v_i^{B^*}(\text{OPT}_i(B^*)) \\
&\geq \mathbf{E}_{B^c} \left[\sum_{i \in B} v_i^{B^c}(\text{OPT}_i(B^c)) \right] \stackrel{(*)}{\geq} \mathbf{E}_{B^c} \left[\sum_{i \in B} v_i^{B^c}(\text{OPT}_i(B)) \right] \\
&= \sum_{i \in B} v_i(\text{OPT}_i(B)) \cdot \Pr[i \in B^c] \geq \frac{3}{16\Delta} \cdot \sum_{i \in B} v_i(\text{OPT}_i(B)).
\end{aligned}$$

Inequality $(*)$ holds because $\text{OPT}(B^c) \in \arg \max_{S \in A'} \sum_{i \in B} v_i^{B^c}(S_i)$ and $\text{OPT}(B) \in A'$. Since the maximum social welfare is at most $\alpha \cdot \sum_{i \in B} v_i(\text{OPT}_i(B))$ the claimed approximation factor follows.

We still have to show that the algorithm is maximal-in-range. For each $S \in A'$ and $B' \in D$ let $S^{B'}$ be the assignment when all bidders in $i \in B'$ get the same set of items as in S and all other bidders get no items. Define $A'_D := \{S^{B'} \mid (S, B') \in A' \times D\}$. We show next that f^c is maximal on the subset $A'_D \subseteq A$. It holds that

$$\sum_{i \in B} v_i(\text{OPT}_i) = \max_{B' \in D} \sum_{i \in B} v_i^{B'}(\text{OPT}_i(B')) = \max_{B' \in D} \max_{S \in A'} \sum_{i \in B} v_i^{B'}(S_i) = \max_{S^{B'} \in A'_D} \sum_{i \in B} v_i(S_i^{B'}).$$

The pricing scheme for a truthful mechanism is given in [Nis07, Proposition 9.31]. \square

A.2 Bounded Out-degree in the Item Conflict Graph

Here we provide the proof of Theorem 6, which follows from Theorem 22 below. Recall that we here handle the case when Δ_I is bounded, G is arbitrary, and valuations are frac-

tionally sub-additive. Again, the results apply to arbitrary restrictions on the valuations (e.g., submodular valuations).

Theorem 22. *Suppose that valuations are fractionally sub-additive, and that the original approximation guarantees of f^c in Theorem 3, Corollary 4, Corollary 21, and Theorem 1 are in the form of $C\Delta\alpha$, where C is either 4 or $16/3$. Then f^c can be modified so that the approximation guarantees are changed to $C\Delta_I\alpha$.*

Proof of Theorem 22 (for Lemma 3 with bounded Δ_I): The idea is to restrict the item set I to a random set I^c that is independent of the valuations of the bidders and then to call the α -approximation algorithm for the restricted item set I^c . Note that we can use Algorithm 1 also for items. Thus, by the same arguments as for bidders we obtain a random set of items I^c where items have no conflicts and where every item is in I^c with probability at least $1/(4\Delta_I)$. Let $(S_1(I'), \dots, S_n(I'))$ be the allocation that f returns when we restrict the set of items to $I' \subseteq I$. Furthermore, let $(\text{OPT}_1(I'), \dots, \text{OPT}_n(I'))$ be the optimal allocation of the item set I' . It holds that

$$\begin{aligned} \mathbf{E}_{I^c} \left[\sum_{i \in B} v_i(S_i(I^c)) \right] &\geq \mathbf{E}_{I^c} \left[\frac{1}{\alpha} \sum_{i \in B} v_i(\text{OPT}_i(I^c)) \right] \\ &\stackrel{(*)}{\geq} \mathbf{E}_{I^c} \left[\frac{1}{\alpha} \sum_{i \in B} v_i(\text{OPT}_i(I) \cap I^c) \right] \geq \frac{1}{\alpha} \sum_{i \in B} \mathbf{E}_{I^c} [v_i(\text{OPT}_i(I) \cap I^c)] \\ &\stackrel{(**)}{\geq} \frac{1}{4\Delta_I\alpha} \sum_{i \in B} \mathbf{E}_{I^c} [v_i(\text{OPT}_i(I))] . \end{aligned}$$

Above, inequality $(*)$ follows because $\text{OPT}_i(I^c)$ is optimal for item set I^c , and inequality $(**)$ follows by Proposition 5. \square

Proof of Theorem 22 (for Corollary 4 with bounded Δ_I): Again, we restrict the item set I to a random set I^c as in Theorem 3 and then to call mechanism (f, p) for the restricted item set. The approximation guarantee follows from Theorem 3 and universally truthfulness (resp. truthfulness in expectation) follows since truthful bidding is a dominant strategy for each realization of I^c . \square

Proof of Theorem 22 (for Lemma 21 with bounded Δ_I): Note that in Theorem 3 we use the same randomization technique for B^c and for I^c . Thus, we can apply our derandomization technique for B^c also to I^c . \square

Proof of Theorem 22 (for Theorem 1 with bounded Δ_I): Given a maximal-in-range algorithm f for combinatorial auctions without conflicts that is a deterministic α -approximation algorithm we show in Algorithm 5 how to construct a maximal-in-range algorithm f^c for combinatorial auctions with conflicts that is a deterministic $(16\Delta_I\alpha/3)$ -approximation algorithm. As in Algorithm 4, Algorithm 5 always calls f for the same set of bidders and set of items; thus, the target set A' of f is the same in each call.

Algorithm 5: Maximal-in-range algorithm f^c over range A'_D when Δ_I is bounded.

```

1 Let  $D$  be the domain of a distribution over the random set  $I^c$  computed by
  Algorithm 1 satisfying  $|D| \in \mathcal{O}(m^2)$ ;
2 Let  $A' \subseteq A$  be the target set of  $f$ ;
3 Set  $\text{OPT} \leftarrow (\emptyset, \dots, \emptyset)$ ;
4 foreach  $I' \in D$  do
5   | Define  $v_i^{I'}(S) := v_i(S \cap I')$  for all  $S \subseteq I$  and  $i \in B$ ;
6   | Set  $\text{OPT}(I') \leftarrow f(v^{I'})$ ;
7   | if  $\sum_{i \in B} v_i^{I'}(\text{OPT}_i(I')) \geq \sum_{i \in B} v_i(\text{OPT}_i)$  then  $\forall i \in B : \text{OPT}_i \leftarrow \text{OPT}_i(I') \cap I'$ ;
8   | ;
9 end
10 return  $\text{OPT}$ 

```

Let us assume that OPT was set to its final value when $I' = I^*$. Furthermore, assume that I^c is a random subset computed by Algorithm 1. It follows that

$$\begin{aligned}
\sum_{i \in B} v_i(\text{OPT}_i) &= \sum_{i \in B} v_i^{I^*}(\text{OPT}_i(I^*)) \geq \mathbf{E}_{I^c} \left[\sum_{i \in B} v_i^{I^c}(\text{OPT}_i(I^c)) \right] \\
&\stackrel{(*)}{\geq} \mathbf{E}_{I^c} \left[\sum_{i \in B} v_i^{I^c}(\text{OPT}_i(I)) \right] = \sum_{i \in B} \mathbf{E}_{I^c} [v_i(\text{OPT}_i(I) \cap I^c)] \\
&\stackrel{(**)}{\geq} \frac{3}{16\Delta_I} \sum_{i \in B} v_i(\text{OPT}_i(I)).
\end{aligned}$$

Inequality (*) holds because $\text{OPT}(I^c) \in \arg \max_{S \in A'} \sum_{i \in B} v_i^{I^c}(S_i)$ and $\text{OPT}(I) \in A'$, and Inequality (**) follows from Lemma 5. Since the maximum social welfare it at most $\alpha \cdot \sum_{i \in B} v_i(\text{OPT}_i(I))$, the claimed approximation guarantee follows.

We still have to show that the algorithm is maximal-in-range. For each $S \in A'$ and $I' \in D$ let $S^{I'} := (S_1 \cap I', \dots, S_n \cap I')$. Define $A'_D := \{S^{I'} \mid (S, I') \in A' \times D\}$. We show that f^c is maximal on the subset $A'_D \subseteq A$. It holds that

$$\sum_{i \in B} v_i(\text{OPT}_i) = \max_{I' \in D} \sum_{i \in B} v_i^{I'}(\text{OPT}_i(I')) = \max_{I' \in D} \max_{S \in A'} \sum_{i \in B} v_i^{I'}(S_i) = \max_{S^{I'} \in A'_D} \sum_{i \in B} v_i(S_i^{I'}).$$

Again, the pricing scheme for a truthful mechanism is given in [Nis07, Proposition 9.31]. \square

B Proofs and Results Omitted in Section 3

B.1 Solving Dual of CPR-C in Poly-time using Ellipsoid Algorithm

First, we refer the readers to [GLS88, Chapters 2–4] for details of ellipsoid algorithm. We will use the following result from fundamental linear algebra:

Lemma 23 ([GLS88, Section 0.1]). *Let M be a symmetric $q \times q$ real matrix. M is positive semi-definite if and only if there exists $w_1, w_2, \dots, w_q \in \mathbb{R}^q$ such that for $1 \leq k, \ell \leq q$, $M_{k\ell} = w_k \cdot w_\ell$. Furthermore, M is positive definite if and only if the vectors w_1, w_2, \dots, w_q are linearly independent.*

For our problem, we define the following proper cone K : K consists of all points $(\{x_{i,S}\}, \{\alpha_i\}, \{\beta_k\}, M)$, where $\{x_{i,S}\}, \{\alpha_i\}, \{\beta_k\}$ are vectors of dimension $(2^m - 1)n$, n , m respectively, and all of them have non-negative entries; M is symmetric positive semi-definite $(n + 1) \times (n + 1)$ real matrix. It is easy to verify that K is a proper cone. By following the arguments in [BV04, Examples 2.23 and 2.24], it is easy to show that $K^* = K$, i.e., K is self-dual.

Using Lemma 23 and introducing slack variables α 's and β 's, we can rewrite CPR-C in the standard CP form CP-STD:

$$\begin{aligned}
\min \quad & - \sum_{i \in B} \sum_{S \neq \emptyset} v_i(S) \cdot x_{i,S} && \text{(CPR-C')} \\
\text{s.t.} \quad & - \sum_{S \neq \emptyset} x_{i,S} - \alpha_i &= -1, & \forall i \in B \quad (\mathbf{u}_i) \\
& - \sum_{S \ni k} \sum_{i \in B} x_{i,S} - \beta_k &= -1, & \forall k \in I \quad (\mathbf{p}_k) \\
& 2 \sum_{S \neq \emptyset} x_{i,S} - M_{0i} &= 1, & \forall i \in B \quad (\mathbf{z}_i) \\
& M_{0i} + M_{0j} + M_{\min\{i,j\} \max\{i,j\}} &= -1, & \forall (i,j) \in E \quad (\mathbf{y}_{ij}) \\
& M_{00} &= 1, & \quad \quad \quad (\mathbf{q}_0) \\
& M_{ii} &= 1, & \forall i \in B \quad (\mathbf{q}_i) \\
& (\{x_{i,S}\}, \{\alpha_i\}, \{\beta_k\}, M) &\succeq_K 0.
\end{aligned}$$

Instead of solving CPR-C' directly, we will solve its dual. Each equality constraint in CPR-C' will associate to a variable in the dual. We have written the variables down on the right of their corresponding constraints.

To ensure that solving the dual of CPR-C' is equivalent to solving CPR-C', we need to check that strong duality holds between them. We do this soon later in this appendix.

The dual of CPR-C' is

$$\begin{aligned}
\max \quad & - \sum_{i \in B} u_i - \sum_{k \in I} p_k + \sum_{i \in B} z_i - \sum_{(i,j) \in E} y_{ij} + q_0 + \sum_{i \in B} q_i && \text{(CPR-C-DUAL)} \\
\text{s.t.} \quad & v_i(S) - u_i - \sum_{k \in S} p_k + 2z_i \leq 0, && \forall i \in B, S \subseteq I \\
& u_i \geq 0, && \forall i \in B \\
& p_k \geq 0, && \forall k \in I \\
& -Q \text{ is SPSD,}
\end{aligned}$$

where Q is the symmetric $(n+1) \times (n+1)$ -matrix determined as follows:

$$\begin{aligned}
Q_{ii} &= q_i, & \forall i \in B \cup \{0\} \\
Q_{0i} = Q_{i0} &= -z_i + \sum_{j:(i,j) \in E} y_{ij} + \sum_{j:(j,i) \in E} y_{ji}, & \forall i \in B \\
Q_{ij} = Q_{ji} &= \begin{cases} 0 & \text{if } (i,j) \notin E \text{ and } (j,i) \notin E \\ y_{ij} & \text{if } (i,j) \in E \text{ and } (j,i) \notin E \\ y_{ji} & \text{if } (i,j) \notin E \text{ and } (j,i) \in E \\ y_{ij} + y_{ji} & \text{if } (i,j) \in E \text{ and } (j,i) \in E \end{cases}, & \forall \text{ distinct } i, j \in B.
\end{aligned}$$

The final step is to design a poly-time separation oracle:

- If $u_i < 0$ for some $i \in B$ or $p_k < 0$ for some $k \in I$, we have an obvious separation hyperplane.
- Since the dimension of $-Q$ is $\text{poly}(n)$, we can use a standard algorithm to check whether it is SPSD in $\text{poly}(n)$ time, and obtain a separation hyperplane if $-Q$ is not SPSD. See [dOF14, Example 2] for details.
- If $v_i(S) - u_i - \sum_{k \in S} p_k + 2z_i > 0$ for some $i \in B$ and $S \subseteq I$, then we can use the demand oracle of bidder i to find $S = S^*$ that maximizes $v_i(S) - \sum_{k \in S} p_k$. Then $v_i(S^*) - u_i - \sum_{k \in S^*} p_k + 2z_i > 0$, which provides us a separation hyperplane. This is almost identical to the separation oracle used in the ellipsoid algorithm for solving LPR-NC in [NS06].

B.2 Strong Duality of CPR-C

While strong duality always holds for LP, it does not always hold for CP. To check strong duality, we verify that the primal program CPR-C' satisfies *Slater's condition*, i.e., find a feasible point which satisfies all equality constraints, and *strictly satisfy* the conic constraint. In other words, we need to find a feasible point $(\{x_{i,S}\}, \{\alpha_i\}, \{\beta_k\}, M)$ which satisfies all equality constraints, and such that $x_{i,S} > 0$ for all $i \in B, S \subseteq I$, $\alpha_i > 0$ for all $i \in B$, $\beta_k > 0$ for all $k \in I$, and M is positive *definite*.

Here, we only consider the cases $n \geq 2$; the auction with $n = 1$ bidder is trivial.

Consider the point with $x_{i,S} = \frac{1}{4(2^m-1)n^2}$ for all $i \in B$ and $S \subseteq I$. Then $\alpha_i = 1 - (2^m - 1) \cdot \frac{1}{4(2^m-1)n^2} > 0$ and $\beta_k = 1 - 2^{m-1}n \cdot \frac{1}{4(2^m-1)n^2} > 0$. Also, $\forall i \in B$, $M_{0i} = 2(2^m - 1) \cdot \frac{1}{4(2^m-1)n^2} - 1 = \frac{1}{2n^2} - 1$, and $\forall i, j \in B$ where $i \neq j$, we choose $M_{ij} = -1 - M_{0i} - M_{0j} = 1 - \frac{1}{n^2}$. Recall that $\forall i \in B$, $M_{00} = M_{ii} = 1$.

For notational convenience, let $\epsilon = \frac{1}{2n^2}$, i.e., $\forall i \in B$, $M_{0i} = \epsilon - 1$; $\forall i, j \in B$ where $i \neq j$, $M_{ij} = 1 - 2\epsilon$. By Lemma 23, to check that M is positive definite, equivalently, we find linearly independent $w_0, w_1, \dots, w_n \in \mathbb{R}^{n+1}$ such that $\forall i, j \in B \cup \{0\}$, $w_i \cdot w_j = M_{ij}$.

Let

$$\begin{aligned}
w_0 &= (1, 0, 0, \dots, 0) \\
w_1 &= (\epsilon - 1, a_1, 0, 0, \dots, 0) \\
w_2 &= (\epsilon - 1, b_1, a_2, 0, 0, \dots, 0) \\
w_3 &= (\epsilon - 1, b_1, b_2, a_3, 0, 0, \dots, 0) \\
&\vdots \\
w_n &= (\epsilon - 1, b_1, b_2, b_3, \dots, b_{n-1}, a_n).
\end{aligned}$$

Note that $\forall i \in B \cup \{0\}$, w_i has $n - i$ trailing zeroes. Also, $\forall i \in B$, the first entry of w_i is $\epsilon - 1$, followed by b_1, b_2, \dots, b_{i-1} , and then followed by a_i and the trailing zeroes. These ensure that $w_0 \cdot w_0 = 1 = M_{00}$ and $w_0 \cdot w_i = \epsilon - 1 = M_{0i}$ for all $i \in B$.

We will determine $a_1, b_1, a_2, b_2, a_3, b_3, \dots, a_{n-1}, b_{n-1}, a_n$ in this order. We will show that for every $i \in B$, $a_i \geq \frac{\sqrt{133}}{12n}$, and b_i 's are negative with $|b_i| \leq \frac{1}{3n^3}$.

Since $w_1 \cdot w_1 = 1$, $(\epsilon - 1)^2 + (a_1)^2 = 1$ and thus

$$a_1 = \sqrt{2\epsilon - \epsilon^2} = \sqrt{\frac{1}{n^2} - \frac{1}{4n^4}} = \sqrt{\frac{1}{n^2} \left(1 - \frac{1}{4n^2}\right)} \geq \sqrt{\frac{15}{16n^2}} > \frac{\sqrt{133}}{12n}.$$

Since for $i > 1$, $w_1 \cdot w_i = 1 - 2\epsilon$, $(\epsilon - 1)^2 + a_1 b_1 = 1 - 2\epsilon$. Hence $a_1 b_1 = -\epsilon^2$ and thus b_1 is negative with

$$|b_1| = \epsilon^2 / a_1 = \frac{1}{4n^4 a_1} < \frac{3}{\sqrt{133} n^3} < \frac{1}{3n^3}.$$

Next, we proceed by induction. Suppose that for some $q \geq 1$, $a_1, a_2, \dots, a_q \geq \frac{\sqrt{133}}{12n}$ and b_1, b_2, \dots, b_q are negative with $|b_1|, |b_2|, \dots, |b_q| \leq \frac{1}{3n^3}$.

Since $w_{q+1} \cdot w_{q+1} = 1$, $(\epsilon - 1)^2 + \sum_{\ell=1}^q (b_\ell)^2 + (a_{q+1})^2 = 1$ and thus

$$\begin{aligned}
a_{q+1} &= \sqrt{2\epsilon - \epsilon^2 - \sum_{\ell=1}^q (b_\ell)^2} \geq \sqrt{\frac{1}{n^2} - \frac{1}{4n^4} - n \cdot \frac{1}{9n^6}} \\
&\geq \sqrt{\frac{1}{n^2} - \frac{1}{4n^4} - \frac{1}{18n^4}} = \sqrt{\frac{1}{n^2} \left(1 - \frac{11}{36n^2}\right)} \geq \frac{\sqrt{133}}{12n}.
\end{aligned}$$

Since for $i > q + 1$, $w_{q+1} \cdot w_i = 1 - 2\epsilon$, $(\epsilon - 1)^2 + \sum_{\ell=1}^q (b_\ell)^2 + a_{q+1} b_{q+1} = 1 - 2\epsilon$. Hence

$$0 > a_{q+1} b_{q+1} = -\epsilon^2 - \sum_{\ell=1}^q (b_\ell)^2 \geq -\frac{1}{4n^4} - n \cdot \frac{1}{9n^6} \geq -\frac{1}{4n^4} - \frac{1}{18n^4} = -\frac{11}{36n^4},$$

and thus b_{q+1} is negative with

$$|b_{q+1}| \leq \frac{11}{36n^4 a_{q+1}} \leq \frac{11}{36n^4} \cdot \frac{12n}{\sqrt{133}} = \frac{11}{3\sqrt{133} n^3} < \frac{1}{3n^3}.$$

These complete the induction.

Since all a_i 's are strictly positive, $w_0, w_1, w_2, \dots, w_n$ are linearly independent.