Approximating bounded occurrence ordering CSPs

VENKATESAN GURUSWAMI* YUAN ZHOU*

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213.

Abstract

A theorem of Håstad shows that for every constraint satisfaction problem (CSP) over a fixed size domain, instances where each variable appears in at most $O(1)$ constraints admit a non-trivial approximation algorithm, in the sense that one can beat (by an additive constant) the approximation ratio achieved by the naive algorithm that simply picks a random assignment. We consider the analogous question for ordering CSPs, where the goal is to find a linear ordering of the variables to maximize the number of satisfied constraints, each of which stipulates some restriction on the local order of the involved variables. It was shown recently that without the bounded occurrence restriction, for *every* ordering CSP it is Unique Games-hard to beat the naive random ordering algorithm.

In this work, we prove that the CSP with monotone ordering constraints $x_{i_1} < x_{i_1} < \cdots < x_{i_k}$ of arbitrary arity k can be approximated beyond the random ordering threshold $1/k!$ on bounded occurrence instances. We prove a similar result for all ordering CSPs, with arbitrary payoff functions, whose constraints have arity at most 3. Our method is based on working with a carefully defined Boolean CSP that serves as a proxy for the ordering CSP. One of the main technical ingredients is to establish that certain Fourier coefficients of this proxy constraint have substantial mass. These are then used to guarantee a good ordering via an algorithm that finds a good Boolean assignment to the variables of a low-degree bounded occurrence multilinear polynomial. Our algorithm for the latter task is similar to Håstad's earlier method but is based on a greedy choice that achieves a better performance guarantee.

*This research was supported in part by a Packard Fellowship and US-Israel BSF grant 2008293. Email: guruswami@cmu.edu, yuanzhou@cs.cmu.edu.

1 Introduction

Constraint satisfaction. Constraint satisfaction problems (CSPs) are an important class of optimization problems. A CSP is specified by a finite set Π of relations, each of arity k, over a domain $\{0, 1, \ldots, D-1\}$, where k, D are some fixed constants. An instance of such a CSP consists of a set of variables V and a collection of constraints (possibly with weights) each of which is a relation from Π applied to some k-tuple of variables from V. The goal is to find an assignment $\sigma : V \to D$ that maximizes the total weight of satisfied constraints. For example in the Max Cut problem, $k = D = 2$ and Π consists of the single relation $\mathrm{CUT}(a, b) = \mathbf{1}(a \neq b)$. More generally, one can also allow real-valued payoff functions $f : \{0, 1, \ldots, D-1\}^k \to \mathbb{R}^+$ in Π (instead of just $\{0, 1\}$-valued functions), with the goal being to find an assignment maximizing the total payoff.

Most Max CSP problems are NP-hard, and there is by now a rich body of work on approximation algorithms and hardness of approximation results for CSPs. Algorithmically, semidefinite programming (SDP) has been the principal tool to obtain good approximation ratios. In fact, SDP is universal for CSPs in the sense that under the Unique Games conjecture a certain canonical SDP achieves the optimal approximation ratio [13]. However, many CSPs, including Max 3SAT, Max 3LIN, Max NAE-4-SAT, etc., are *approximation resistant*, meaning that for any $\epsilon > 0$, even when given a $(1-\epsilon)$-satisfiable instance, it is hard to find an assignment that satisfies more than a fraction $r + \epsilon$ of the constraints, where r, the *random assignment threshold*, is the expected fraction of constraints satisfied by a random assignment [10, 1]. In other words, it is hard to improve upon the naive algorithm that simply picks a random assignment without even looking at the structure of the instance.

Let us call a CSP that is not approximation resistant as *non-trivially approximable*. Inspite of a rich body of powerful algorithmic and hardness results, we are quite far from a complete classification of all CSPs into approximation resistant or non-trivially approximable. Several partial results are known; for example, the classification is known for Boolean predicates of arity 3. It is known that every binary CSP (i.e., whose constraints have arity 2), regardless of domains size (as long as it is fixed), is non-trivially approximable via a SDP-based algorithm [7, 6, 11]. In a different vein, Håstad [9] showed that for *every* Boolean CSP, when restricted to sparse instances where each variable participates in a bounded number B of constraints, one can beat the random assignment threshold (by an amount that is at least $\Omega(1/B)$). Trevisan showed that for Max 3SAT beating the random assignment threshold by more than $O(1/\sqrt{B})$ is NP-hard, so some degradation of the performance ratio with the bound B is necessary [14].

Ordering CSPs. With this context, we now turn to ordering CSPs, which are the focus of this paper. The simplest ordering CSP is the well-known Maximum Acyclic Subgraph (MAS) problem, where we are given a directed graph and the goal is to order the vertices $V = \{x_1, \ldots, x_n\}$ of the graph so that a maximum number of edges go forward in the ordering. This can be viewed as a "CSP" with variables V and constraints $x_i < x_j$ for each directed edge (x_i, x_j) in the graph; the difference from usual CSPs is that the variables are to be ordered, i.e., assigned values from $\{1, 2, \ldots, n\}$, instead of being assigned values from a fixed domain (of size independent of n).

An ordering CSP of arity k is specified of a constraint $\Pi : S_k \to \{0, 1\}$ where S_k is the set of permutations of $\{1, 2, \ldots, k\}$. An instance of such a CSP consists of a set of variables V and a collection of constraints which are (ordered) k-tuples. The constraint tuple $e = (x_{i_1}, x_{i_2}, \cdots, x_{i_k})$ is satisfied by an ordering of V if the local ordering of the variables $x_{i_1}, x_{i_2}, \cdots, x_{i_k}$, viewed as an element of S_k, belongs to the subset Π. The goal is to find an ordering that maximizes the number of satisfied constraint tuples. An example of an arity 3 ordering CSP is the *Betweenness* problem with constraints of the form x_{i_2} occurs between x_{i_1} and x_{i_3} (this corresponds to the subset $\Pi = \{123, 321\}$ of S_3). More generally, one can allow more than one kind of ordering constraint, or even a payoff function $\omega_e : S_k \to \mathbb{R}^+$ for each constraint tuple e. The goal in this case is to find an ordering $\mathcal{O}$ that maximizes $\sum_e \omega_e(\mathcal{O}_{|e})$ where $\mathcal{O}_{|e}$ is the relative ordering of vertices in e induced by $\mathcal{O}$.

Despite much algorithmic progress on CSPs, even for MAS there was no efficient algorithm known to beat the factor $1/2$ achieved by picking a random ordering. This was explained by the recent work [8] which showed that such an algorithm does not exist under the Unique Games conjecture, or in other words, MAS is approximation resistant. This hardness result was generalized to all ordering CSPs of arity 3 [4], and later to higher arities, showing that every ordering CSP is approximation resistant (under the UGC) [3]![1]

In light of this pervasive hardness of approximating ordering CSPs, in this work we ask the natural question raised by Håstad's algorithm for bounded occurrence CSPs [9], namely whether bounded occurrence instances of ordering CSPs admit a non-trivial approximation. For the case of MAS, Berger and Shor [2] gave an efficient algorithm that given any directed graph of total degree D, finds an ordering in which at least a fraction $(1/2 + \Omega(1/\sqrt{D}))$ of the edges go forward. This shows that bounded occurrence MAS is non-trivially approximable. The algorithm is quite simple, though its analysis is subtle. The approach is to order the vertices randomly, and process vertices in this order. When a vertex is processed, if it has more incoming edges than outgoing edges (in the graph at that stage), all outgoing edges are removed, and otherwise all its incoming edges are removed. The graph remaining after all the vertices are processed is returned as the acyclic subgraph.

Evidently, this algorithm is tailored to the MAS problem, and heavily exploits its underlying graph-theoretic structure. It therefore does not seem amenable for extensions to give non-trivial approximations to other ordering CSPs.

Our results. In this work, we prove that important special cases of ordering CSPs do admit non-trivial approximation on bounded occurrence instances. In particular, we prove this for the following classes of ordering CSPs:

1. The monotone ordering k-CSP for arbitrary k with constraints of the form $x_{i_1} < x_{i_2} < \cdots < x_{i_k}$ (i.e., the CSP defined by the constraint subset $\{123 \ldots k\} \subseteq S_k$ consisting of the identity permutation). This can be viewed as the arity k generalization of the MAS problem. (Note that we allow multiple constraint tuples on the same *set* of k variables, just as one would allow 2-cycles in a MAS instance given by a directed graph.)

2. All ordering CSPs of arity 3, even allowing for arbitrary payoff functions as constraints.

Our proofs show that these ordering CSPs admit an ordering into "4 slots" that beats the random ordering threshold. We remark that CSP instances which are satisfiable for orderings into n slots but do not admit good "c slot" solutions for any fixed constant c are the basis of the Unique Games hardness results for ordering CSPs [8, 3]. Our results show that for arity 3 CSPs and monotone ordering k-ary CSPs such gap instances cannot be bounded occurrence.

Our methods. As mentioned above, the combinatorial approach of the Berger-Shor algorithm for MAS on degree-bounded graphs seems to have no obvious analog for more complicated ordering constraints. We prove our results by applying (after some adaptations) Håstad's algorithm [9] to certain "proxy" Boolean CSPs that correspond to solutions to the ordering CSP that map the variables into a domain of size 4.

For the case of monotone ordering constraints (of arbitrary arity k), we prove that for this proxy payoff function on the Boolean hypercube, a specific portion of the Fourier spectrum carries non-negligible mass. This is the technical core of our argument. Once we establish this, the task becomes finding a Boolean

[1] This does not rule out non-trivial approximations for *satisfiable* instances. Of course for satisfiable instances of MAS, which correspond to DAGs, topological sorting satisfies all the constraints. For Betweenness, a factor $1/2$ approximation for satisfiable instances is known [5, 12].

assignment to the variables of a bounded-occurrence low-degree multilinear polynomial (namely the sum of the Fourier representations of all the constraints) that evaluates to a real number that is non-negligibly larger than the constant term (which is the random assignment threshold). We present a greedy algorithm for this latter task which is similar to Håstad's algorithm [9], but yields somewhat better quantitative bounds.

Our result on general ordering 3-CSPs faces an additional complication since it can happen that the concerned part of Fourier spectrum is in fact zero for certain kinds of constraints. We identify all the cases when this troublesome phenomenon occurs, proving that in such cases the pay-off function can be expressed as a linear combination of arity 2 pay-off functions (accordingly, we call these cases as "binary representable" pay-off functions). If the binary representable portion of the pay-offs is bounded away from 1, then the remaining pay-offs (called them "truly 3-ary") contribute a substantial amount to the Fourier spectrum. Fortunately, the binary representable portion of pay-offs can be handled by our argument for monotone ordering constraints (specialized to arity two). So in the case when they comprise most of the constraints, we prove that their contribution to the Fourier spectrum is significant and cannot be canceled by the contribution from the truly 3-ary pay-offs.

1.1 Outline for the rest of the paper

In Section 2, we formally define the ordering CSPs with bounded occurrence, and the proxy problems (the t-ordering version). We also introduce the notation and analytic tools we will need in the remainder of the paper. In Section 3, we present an algorithm which is a variant of Håstad's algorithm in [9], and is used to solve the proxy problems. In Section 4 and Section 5, we prove the two main theorems (Theorems 4.1 and 5.1) of the paper.

2 Preliminaries

2.1 Ordering CSPs, bounded occurrence ordering CSPs

An *ordering* over vertex set V is an *injective* mapping $\mathcal{O} : V \to \mathbb{Z}^+$. An instance of k-ary monotone ordering problem $G = (V, E, \omega)$ consists of vertex set V, set E of k-tuples of distinct vertices, and weight function $\omega : E \to \mathbb{R}^+$. The weight satisfied by ordering $\mathcal{O}$ is

$$\mathsf{Val}^{\mathcal{O}}(G) \stackrel{\text{def}}{=} \sum_{e=(v_{i_1}, v_{i_2}, \cdots, v_{i_k}) \in E} \omega(e) \cdot \mathbf{1}_{\mathcal{O}(v_{i_1}) < \mathcal{O}(v_{i_2}) < \cdots < \mathcal{O}(v_{i_k})}.$$

We also denote the value of the optimal solution by

$$\mathsf{Val}(G) \stackrel{\text{def}}{=} \max_{\text{injective } \mathcal{O}: V \to \mathbb{Z}} \{\mathsf{Val}^{\mathcal{O}}(G)\}.$$

We can extend the definition of the monotone ordering problem to ordering CSPs $\mathcal{I} = (V, E, \Omega)$ with general pay-off functions, where V and E are similarly defined. For each k-tuple $e = (v_1, v_2, \cdots, v_k) \in E$, a general pay-off function $\omega_e \in \Omega$, mapping from all $k!$ possible orderings among $\mathcal{O}(v_1), \mathcal{O}(v_2), \cdots, \mathcal{O}(v_k)$ to $\mathbb{R}^{\geqslant 0}$, is introduced. That is, for an ordering $\mathcal{O}$, its pay-off $\omega_e(\mathcal{O})$ for constraint tuple e only depends on $\mathcal{O}_{|e}$, the relative ordering on vertices of e induced by $\mathcal{O}$. The overall pay-off achieved by an ordering) is defined as $\mathsf{Val}^{\mathcal{O}}(\mathcal{I}) \stackrel{\text{def}}{=} \sum_{e \in E} \omega_e(\mathcal{O})$. The optimal pay-off for the instance is then given by

$$\mathsf{Val}(\mathcal{I}) \stackrel{\text{def}}{=} \max_{\text{injective } \mathcal{O}: V \to \mathbb{Z}} \{\mathsf{Val}^{\mathcal{O}}(\mathcal{I})\} .$$

An ordering CSP problem $\mathcal{I} = (V, E, \Omega)$ (or a monotone ordering problem $G = (V, E, \omega)$) is called *B-occurrence bounded* if each vertex $v \in V$ occurs in at most B tuples in E.

2.2 The t-ordering version of ordering CSPs

We start this section with several definitions. Two orderings $\mathcal{O}$ and $\mathcal{O}'$ are *essentially the same* if $\forall u, v \in V, \mathcal{O}(u) < \mathcal{O}(v) \Leftrightarrow \mathcal{O}'(u) < \mathcal{O}'(v)$, otherwise we call them *essentially different*. For a positive integer m, denote $[m] = \{1, 2, \ldots, m\}$. For integer $t > 0$, a *t-ordering* on V is a mapping $\mathcal{O}_t : V \to [t]$, not necessarily injective. An ordering $\mathcal{O}$ is *consistent* with a t-ordering $\mathcal{O}_t$, denoted by $\mathcal{O} \sim \mathcal{O}_t$, when $\forall u, v \in V, \mathcal{O}_t(u) < \mathcal{O}_t(v) \Rightarrow \mathcal{O}(u) < \mathcal{O}(v)$.

The monotone ordering problem G can be naturally extended to its t-ordering version, which is a regular CSP problem over domain $[t]$ defined as follows. For each constraint $e = (v_1, v_2, \cdots, v_k) \in E$, we introduce a pay-off function

$$\pi_e(\mathcal{O}_t) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{\mathcal{O} \sim \mathcal{O}_t}[\mathbf{1}_{\mathcal{O}(v_1) < \mathcal{O}(v_2) < \cdots < \mathcal{O}(v_k)}],$$

where the expectation is uniformly taken over all the essentially different orderings $\mathcal{O}$ that are consistent with $\mathcal{O}_t$. (In this paper, when $\mathcal{O}$ becomes a random variable for total ordering without further explanation, it is always uniformly taken over all essentially different orderings (that satisfy certain criteria).) Note that although π_e receives an n-dimensional vector as parameter in the equation above, its value depends only on the k values to $v_1, v_2, \cdots, v_k$. Then the k-ary CSP problem (t-ordering version of G) is to find the t-ordering $\mathcal{O}_t$ to maximize the objective function

$$\mathsf{Val}_t^{\mathcal{O}_t}(G) \stackrel{\text{def}}{=} \sum_{e \in E} \omega(e) \cdot \pi_e(\mathcal{O}_t).$$

We denote the value of the optimal solution by

$$\mathsf{Val}_t(G) \stackrel{\text{def}}{=} \max_{\mathcal{O}_t \in [t]^n} \{\mathsf{Val}_t^{\mathcal{O}_t}(G)\}.$$

We can also extend the ordering CSP problem $\mathcal{I}$ with general pay-off functions to its t-ordering version. For each constraint $e \in E$, the pay-off function in the t-ordering version is defined as $\pi_e(\mathcal{O}_t) \stackrel{\text{def}}{=} \mathbf{E}_{\mathcal{O} \sim \mathcal{O}_t}[\omega_e(\mathcal{O})]$. The pay-off achieved by a particular t-ordering $\mathcal{O}_t$ is given by $\mathsf{Val}_t^{\mathcal{O}_t}(\mathcal{I}) \stackrel{\text{def}}{=} \sum_{e \in E} \pi_e(\mathcal{O}_t)$, and the value of the optimal t-ordering solution is $\mathsf{Val}_t(\mathcal{I}) \stackrel{\text{def}}{=} \max_{\mathcal{O}_t \in [t]^n} \{\mathsf{Val}_t^{\mathcal{O}_t}(\mathcal{I})\}$.

Our approach to getting a good solution for (occurrence bounded) ordering CSPs is based on the following fact.

Fact 2.1. *For all positive integers t, $\mathsf{Val}(\mathcal{I}) \geqslant \mathsf{Val}_t(\mathcal{I})$.*

Note that for $t = 1$, $\mathsf{Val}_1(\mathcal{I})$ equals the expected pay-off of a random ordering. Since the monotone ordering problem is a special case of ordering CSP with general pay-off functions, Fact 2.1 is also true for the monotone ordering problem. By fact 2.1, it is enough to find a good solution for t-ordering version of $\mathcal{I}$ (or G) to show that $\mathsf{Val}(\mathcal{I})$ (or $\mathsf{Val}(G)$) is large.

2.3 Fourier transform of Boolean functions

For every $f : \{-1, 1\}^d \to \mathbb{R}$, we write the Fourier expansion of f as

$$f(x) = \sum_{S \subseteq [d]} \hat{f}(S) \chi_S(x),$$

where $\hat{f}(S)$ is the Fourier coefficient of f on S, and $\chi_S(x) = \prod_{i \in S} x_i$.

The Fourier coefficients can be computed by the inverse Fourier transform, i.e., for every $S \subseteq [d]$,

$$\hat{f}(S) = \mathop{\mathbf{E}}_{x \in \{-1,1\}^d}[f(x) \chi_S(x)].$$

3 Finding good assignments for bounded occurrence polynomials

Let f be a polynomial in n variables $x_1, x_2, \ldots, x_n$ containing only multilinear terms of degree at most k with coefficients $\hat{f}(S)$. In other words, let $f(x) = \sum_{S\subseteq[n],|S|\leqslant k} \hat{f}(S)\chi_S(x)$. We say that f is D-occurrence bounded if for each coordinate $i \in [n]$, we have $|\{S \ni i : \hat{f}(S) \neq 0\}| \leqslant D$. We also define

$$|f| \stackrel{\text{def}}{=} \sum_{\emptyset\neq S\subseteq[n]} |\hat{f}(S)|.$$

Then, the following proposition shows us how to find a good assignment for f.

Proposition 3.1. *Given a D-occurrence bounded polynomial f of degree at most k, it is possible, in* $\mathrm{poly}(n, 2^k)$ *time, to find $x \in \{-1, 1\}^n$ such that*

$$f(x) \geqslant \hat{f}(\emptyset) + |f|/(2kD).$$

Proof. We use the following algorithm to construct x.

Algorithm. As long as $|f| > 0$, the algorithm finds a non-empty set T that maximizes $|\hat{f}(T)|$, and let $\gamma = \hat{f}(T)$. We want to make sure we get $|\gamma|$ for credit while not losing too much other terms in $|f|$.

Note that for all $\emptyset \neq U \subsetneq T$, we have

$$\mathop{\mathbf{E}}_{z\in\{-1,1\}^T:\chi_T(z)=\mathrm{sgn}(\hat{f}(T))}[\chi_U(z)] = 0,$$

and therefore

$$\mathop{\mathbf{E}}_{z\in\{-1,1\}^T:\chi_T(z)=\mathrm{sgn}(\hat{f}(T))}\Big[\sum_{U\subseteq T} \hat{f}(U)\chi_U(z)\Big] = \hat{f}(\emptyset) + |\hat{f}(T)| = \hat{f}(\emptyset) + |\gamma|.$$

We can enumerate all the $z \in \{-1, 1\}^T$ such that $\chi_T(z_{|T}) = \mathrm{sgn}(\hat{f}(T))$ to find a particular z^*, with

$$\sum_{U\subseteq T} \hat{f}(U)\chi_U(z^*) \geqslant \hat{f}(\emptyset) + |\gamma|.$$

We fix $x_{|T} = z^*$. For the rest of the coordinates, let $g : \{-1, 1\}^{[n]\setminus T} \to \mathbb{R}$ be defined as,

$$g(y) \stackrel{\text{def}}{=} f(y, z^*), \forall y \in \{-1, 1\}^{[n]\setminus T}.$$

We note that g is also a D-occurrence bounded polynomial f of degree at most k, and by fixing all the variables in T, we have

$$\hat{g}(\emptyset) = \sum_{U\subseteq T} \hat{f}(U)\chi_U(z^*) \geqslant \hat{f}(\emptyset) + |\gamma|.$$

On the other hand, observing that $|T| \leqslant k$ and $|\gamma|$ is an upper bound of all $|\hat{f}(S)|$ with $S \neq \emptyset$, we have

$$\begin{aligned}
|g| &= \sum_{\emptyset\neq S\subseteq[n]\setminus T} |\hat{g}(S)| = \sum_{\emptyset\neq S\subseteq[n]\setminus T} \Big|\sum_{U\subseteq T} \hat{f}(S\cup U)\chi_U(z^*)\Big| \\
&\geqslant \sum_{\emptyset\neq S\subseteq[n]\setminus T} |\hat{f}(S)| - \sum_{\emptyset\neq S\subseteq[n]\setminus T}\sum_{\emptyset\neq U\subseteq T} |\hat{f}(S\cup U)| \\
&\geqslant |f| - 2\sum_{S:S\cap T\neq\emptyset} |\hat{f}(S)| \geqslant |f| - 2\sum_{i\in T}\sum_{S\ni i} |\hat{f}(S)| \geqslant |f| - 2|T|D|\gamma| \geqslant |f| - 2kD|\gamma|.
\end{aligned}$$

Then we can use the two inequalities above to establish

$$\hat{g}(\emptyset) + |g|/(2kD) \geqslant \hat{f}(\emptyset) + |f|/(2kD).$$

By recursively applying this algorithm on g, we can eventually fix all the coordinates in x, and get a constant function whose value is at least $\hat{f}(\emptyset) + |f|/(2kD)$. □

Remark 1. The algorithm is similar to Håstad's algorithm in [9] but we make a greedy choice of the term $\chi_T(x)$ to satisfy (the one with the largest coefficient $|\hat{f}(T)|$) at each stage. Our analysis of the loss in $|g|$ is more direct and leads to a better quantitative bound, avoiding the loss of a "scale" factor (which divides all non-zero coefficients of the polynomial) in the advantage over $\hat{f}(\emptyset)$.

4 Bounded occurrence monotone ordering problem

Our main result in this section is the following.

Theorem 4.1. *For any constant $k > 1$, given a B-occurrence bounded k-ary monotone ordering problem $G = (V, E, \omega)$, it is possible, in polynomial time, to find a solution satisfying at least $\mathsf{Val}(G)(1/k! + \Omega_k(1/B))$ weight (in expectation).*

To prove the above theorem, we will show the following lemma.

Lemma 4.2. *For any constant $k > 1$, given a B-occurrence bounded k-ary monotone ordering problem $G = (V, E, \omega)$ with total weight W. Then it is possible, in polynomial time, to find a 4-ordering solution $\mathcal{O}_4$ with $\mathsf{Val}(G)(1/k! + \Omega_k(1/B))$ weight.*

Note that given Lemma 4.2, the randomized algorithm that samples ordering $\mathcal{O} \sim \mathcal{O}_4$ fulfills the task promised in the theorem.

Lemma 4.2 also implies the following.

Corollary 4.3. *For any B-occurrence bounded k-ary monotone ordering problem G, we have $\mathsf{Val}_4(G) \geqslant \mathsf{Val}(G)(1/k! + \Omega_k(1/B))$.*

Proof of Lemma 4.2. We begin the proof with the analysis of the pay-off function $\pi_e : [4]^{\{v_1, v_2, \cdots, v_k\}} \to \mathbb{R}$ for some $e = (v_1, v_2, \cdots, v_k) \in E$. We can also view π_e as a real-valued function defined on Boolean cube $\{-1, 1\}^{2k}$, so that

$$\pi_e(x_1, x_2, \cdots, x_{2k}) = \pi_e\Big((1 - x_1) + \frac{(1 - x_2)}{2} + 1, \cdots, (1 - x_{2k-1}) + \frac{(1 - x_{2k})}{2} + 1\Big).$$

If we let $\Gamma(e)$ be the set of all $k!$ permutations of e, then

$$\begin{aligned}\sum_{e' \in \Gamma(e)} \mathop{\mathbf{E}}_{\mathcal{O}_4 \in [4]^k} [\pi_{e'}(\mathcal{O}_4)] &= \sum_{e' = (v_{i_1}, v_{i_2}, \cdots v_{i_k}) \in \Gamma(e)} \mathop{\mathbf{E}}_{\mathcal{O}_4 \in [4]^k} \Big[\mathop{\mathbf{E}}_{\mathcal{O} \sim \mathcal{O}_4} [\mathbf{1}_{\mathcal{O}(v_{i_1}) < \mathcal{O}(v_{i_2}) < \cdots < \mathcal{O}(v_{i_k})}]\Big] \\ &= \mathop{\mathbf{E}}_{\mathcal{O}_4 \in [4]^k} \Big[\mathop{\mathbf{E}}_{\mathcal{O} \sim \mathcal{O}_4} \Big[\sum_{e' = (v_{i_1}, v_{i_2}, \cdots v_{i_k}) \in \Gamma(e)} \mathbf{1}_{\mathcal{O}(v_{i_1}) < \mathcal{O}(v_{i_2}) < \cdots < \mathcal{O}(v_{i_k})}\Big]\Big] = 1.\end{aligned}$$

Since $\mathbf{E}_{\mathcal{O}_4 \in [4]^k}[\pi_{e'}(\mathcal{O}_4)]$ is the same for all $e' \in \Gamma(e)$, we know that $\mathbf{E}_{\mathcal{O}_4 \in [4]^k}[\pi_e(\mathcal{O}_4)] = 1/k!$. Hence we have the following fact.

Fact 4.4. $\widehat{\pi_e}(\emptyset) = \mathop{\mathbf{E}}_{x \in \{-1,1\}^{2k}} [\pi_e(x)] = \mathop{\mathbf{E}}_{\mathcal{O}_4 \in [4]^k} [\pi_e(\mathcal{O}_4)] = \dfrac{1}{k!}$.

By Fact 4.4, if we apply the algorithm in Proposition 3.1, to the objective function $f(x) = \sum_{e \in E} \omega(e)\pi_e(x)$ of the 4-ordering version, we are guaranteed to have a solution that is no worse than the random threshold ($1/k!$). Then, we only need to identify some non-negligible weights on the rest of the Fourier spectrum of f.

Let $S_{\text{odd}} = \{2i-1 | i \in [k]\}$, and $S_{\text{odd}}^{+} = S_{\text{odd}} \cup \{2k\}$. We make the following claim which we will prove at the end of this section.

Claim 4.5. $\widehat{\pi_e}(S_{\text{odd}}^{+}) = \dfrac{-2 + 2^{2-k}}{k!}$.

The above claim makes sure there is indeed non-negligible mass on non-empty-set Fourier coefficients for each constraint. Then we prove that, when summing up these constraints, either of the following two cases happens.

- Some weights shown in Claim 4.5 are not canceled by others, and finally appears in the non-empty-set Fourier coefficients for the final objective function f.
- Some weights are canceled by others, but in this case, the guarantee by $\hat{f}(\emptyset)$ itself beats $1/k!$ in terms of approximation ratio.

We define

$$\hat{\|}\pi_e\hat{\|} = \sum_{S \subseteq [2k] : \forall i \in [k], S \cap \{2i-1, 2i\} \neq \emptyset} |\widehat{\pi_e}(S)| \ .$$

Now Claim 4.5 implies $\hat{\|}\pi_e\hat{\|} = \Omega_k(1)$ for all $k \geqslant 2$. Let $\Gamma = \{e_i\}$ be a set of constraints sharing the same $\Gamma(e)$, and let us define, by abusing notation slightly

$$\omega(\Gamma) = \sum_{e \in \Gamma} \omega(e), \ \ \omega_{\max}(\Gamma) = \max_{e \in \Gamma}\{\omega(e)\}, \ \text{ and } \ \pi_\Gamma(x) = \sum_{e \in \Gamma} \omega(e)\pi_e(x) \ .$$

We treat $\pi_\Gamma : \{-1, 1\}^{2k} \to \mathbb{R}$ as a real-valued function defined on a Boolean cube.

The idea of defining $\hat{\|}\pi_e\hat{\|}$ and Γ is as follows. The Fourier mass identified in Claim 4.5 could be canceled within Γ, but once the mass goes into $\hat{\|}\pi_\Gamma\hat{\|}$, it cannot be canceled by $\hat{\|}\pi_{\Gamma'}\hat{\|}$ for a different Γ', and will finally go into $|f|$. Then the following lemma shows that either $\hat{\|}\pi_\Gamma\hat{\|}$, or $\widehat{\pi_\Gamma}(\emptyset)$ alone, beats $\omega_{\max}(\Gamma)/k!$, where $\omega_{\max}(\Gamma)$ is an upperbound on the optimal solution's performance on the constraints in Γ.

Lemma 4.6. *For all α, $0 < \alpha < 1$, we have* $\widehat{\pi_\Gamma}(\emptyset) + \alpha\hat{\|}\pi_\Gamma\hat{\|} \geqslant \omega_{\max}(\Gamma)\Big(\dfrac{1}{k!} + \alpha \cdot \Omega_k(1)\Big)$.

Proof. First, by Fact 4.4, we know that $\widehat{\pi_\Gamma}(\emptyset) = \sum_{e \in \Gamma} \widehat{\pi_e}(\emptyset) = \omega(\Gamma) \cdot \dfrac{1}{k!}$. Let $e^* \in \Gamma$ be the constraint with the most weight. If $\omega(e^*) = \omega_{\max}(\Gamma) \geqslant 2/3 \cdot \omega(\Gamma)$, we have

$$\begin{aligned}
\widehat{\pi_\Gamma}(\emptyset) + \alpha\hat{\|}\pi_\Gamma\hat{\|} &= \widehat{\pi_\Gamma}(\emptyset) + \alpha\hat{\|}\sum_{e \in \Gamma}\omega(e)\pi_e\hat{\|} \\
&\geqslant \widehat{\pi_\Gamma}(\emptyset) + \alpha\Big(\hat{\|}\omega(e^*)\pi_{e^*}\hat{\|} - \sum_{e \in \Gamma \setminus \{e^*\}} \hat{\|}\omega(e)\pi_e\hat{\|}\Big) \\
&= \omega(\Gamma) \cdot \frac{1}{k!} + \alpha\Big(\omega(e^*) - \sum_{e \in \Gamma \setminus \{e^*\}} \omega(e)\Big)\hat{\|}\pi_{e^*}\hat{\|} \\
&\geqslant \omega_{\max}(\Gamma)\Big(\frac{1}{k!} + \frac{\alpha}{2}\hat{\|}\pi_{e^*}\hat{\|}\Big) = \omega_{\max}(\Gamma)\Big(\frac{1}{k!} + \alpha \cdot \Omega_k(1)\Big).
\end{aligned}$$

where the last step follows from Claim 4.5. On the other hand, when $\omega(e^*) = \omega_{\max}(\Gamma) < 2/3 \cdot \omega(\Gamma)$,

$$\widehat{\pi_\Gamma}(\emptyset) + \alpha\hat{\|}\pi_\Gamma\hat{\|} \geqslant \widehat{\pi_\Gamma}(\emptyset) = \omega(\Gamma) \cdot \frac{1}{k!} > \omega_{\max}(\Gamma)\Big(\frac{1}{k!} + \frac{1}{2} \cdot \frac{1}{k!}\Big) = \omega_{\max}(\Gamma)\Big(\frac{1}{k!} + \Omega_k(1)\Big).$$

□

Given a k-ary monotone ordering problem $G = (V, E, \omega)$, we partition $E = \Gamma_1 \cup \Gamma_2 \cup \cdots \cup \Gamma_m$ into m disjoint groups, so that constraints in each group $\Gamma_i = \{e_j\}$ share a distinct $\Gamma(e)$ value. Then we write the objective function of its 4-ordering version as

$$f(x) = \sum_{e \in E} \omega(e)\pi_e(x) = \sum_{i=1}^{m} \sum_{e \in \Gamma_i} \pi_e(x) = \sum_{i=1}^{m} \pi_{\Gamma_i}(x),$$

where $f : \{-1, 1\}^{2n} \to \mathbb{R}$ is defined on Boolean cube. For each $1 \leqslant i \leqslant m$, let $\{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ be the k vertices participating in Γ_i, then we note that for each $S \in \{2i_t - 1, 2i_t : t \in [k]\}$ that intersects with $\{2i_t - 1, 2i_t\}$ for each $t \in k$, we have $\hat{f}(S) = \widehat{\pi_{\Gamma_i}}(S)$, since all other constraints will have 0 as its Fourier coefficient over S. Then, for $\alpha \in (0, 1)$, we have

$$\hat{f}(\emptyset) + \alpha \cdot |f| \geqslant \sum_{i=1}^{m} \Big(\widehat{\pi_{\Gamma_i}}(\emptyset) + \alpha \cdot \hat{\|}\pi_{\Gamma_i}\hat{\|}\Big) \geqslant \sum_{i=1}^{m} \omega_{\max}(\Gamma_i)\Big(\frac{1}{k!} + \alpha \cdot \Omega_k(1)\Big), \tag{1}$$

where the last inequality is because of Lemma 4.6.

For each $\Gamma_i (1 \leqslant i \leqslant m)$, a total ordering $\mathcal{O}$ will satisfy at most $\omega_{\max}(\Gamma_i)$ weight of constraints. This give an upper bound of the optimal solution

$$\mathsf{Val}(G) \leqslant \sum_{i=1}^{m} \omega_{\max}(\Gamma_i). \tag{2}$$

Fix a coordinate $i \in [2k]$, each constraint π_e contributes at most 2^{k-1} non-zero Fourier coefficients containing i. Since $G = (V, E, \omega)$ is B-occurrence bounded, there are at most $B2^{k-1}$ non-zero Fourier coefficients of f containing i, therefore f is $B2^{k-1}$-occurrence bounded.

Applying Proposition 3.1 to f, the polynomial time algorithm gets a vector $x \in \{-1, 1\}^{2k}$, which corresponds to a 4-ordering $\mathcal{O}_4$, such that

$$\mathsf{Val}_4^{\mathcal{O}_4}(G) = f(x) \geqslant \hat{f}(\emptyset) + \frac{1}{k2^kB}|f| \geqslant \sum_{i=1}^{m} \omega_{\max}(\Gamma_i)\Big(\frac{1}{k!} + \Omega_k(1/B)\Big) \geqslant \mathsf{Val}(G)\Big(\frac{1}{k!} + \Omega_k(1/B)\Big),$$

where the last two inequalities use (1) and (2) separately. □

Now all that is left to do is to prove Claim 4.5.

Proof of Claim 4.5. Since the coordinates of π_e has the same order as they are shown in e, we can write the value of π_e explicitly as

$$\pi_e(x) = \begin{cases} \dfrac{1}{n_1!n_2!n_3!n_4!} & \text{when } x = (1,1)^{n_1} \circ (1,-1)^{n_2} \circ (-1,1)^{n_3} \circ (-1,-1)^{n_4} \\ 0 & \text{otherwise} \end{cases}.$$

Therefore

$$\begin{aligned}\widehat{\pi_e}(S^+_{\text{odd}}) &= \mathop{\mathbf{E}}_{x\in\{-1,1\}^{2k}}[\pi_e(x)\chi_{S^+_{\text{odd}}}(x)] = \frac{1}{4^k}\sum_{n_1+n_2+n_3+n_4=k}\frac{(-1)^{n_3+n_4}}{n_1!n_2!n_3!n_4!}\cdot(-1)^{\mathbf{1}_{n_4>0\vee(n_3=n_4=0\wedge n_2>0)}}\\ &= \frac{1}{4^k}\Big(-\sum_{n_1+n_2+n_3+n_4=k,n_4>0}\frac{(-1)^{n_3+n_4}}{n_1!n_2!n_3!n_4!}\\ &\quad+\sum_{n_1+n_2+n_3=k,n_3>0}\frac{(-1)^{n_3}}{n_1!n_2!n_3!}-\sum_{n_1+n_2=k,n_2>0}\frac{1}{n_1!n_2!}+\frac{1}{k!}\Big).\end{aligned}$$

This is just the k-th coefficient of the polynomial

$$\frac{1}{4^k}\Big(-e^{2x}e^{-x}(e^{-x}-1)+e^{2x}(e^{-x}-1)-e^{x}(e^{x}-1)+e^{x}\Big)=\frac{-2e^{2x}+4e^{x}-1}{4^k}.$$

Thus, for $k>0$, we have $\widehat{\pi_e}(S^+_{\text{odd}})=\dfrac{-2+2^{2-k}}{k!}$. □

5 Bounded occurrence 3-ary ordering CSP with general pay-off functions

For a ordering CSP problem $\mathcal{I}=(V,E,\Omega)$ with general pay-off functions, we define

$$\mathsf{Rand}(\mathcal{I})\stackrel{\text{def}}{=}\mathop{\mathbf{E}}_{\text{injective }\mathcal{O}:V\to\mathbb{Z}}[\mathsf{Val}^{\mathcal{O}}(\mathcal{I})],$$

as the performance of random ordering. Then we prove our main result for 3-ary ordering CSPs:

Theorem 5.1. *Given a B-occurrence bounded 3-ary ordering CSP problem $\mathcal{I}=(V,E,\Omega)$ with general pay-off functions, it is possible, in polynomial time, to find a solution satisfying at least* $\mathsf{Rand}(\mathcal{I})+(\mathsf{Val}(\mathcal{I})-\mathsf{Rand}(\mathcal{I}))\cdot\Omega(1/B)$ *weight (in expectation).*

To prove Theorem 5.1, it is enough to prove the following lemma.

Lemma 5.2. *Given a B-occurrence bounded 3-ary ordering CSP problem $\mathcal{I}=(V,E,\Omega)$ with general pay-off functions, it is possible, in polynomial time, to find a 4-ordering solution $\mathcal{O}_4$ with* $\mathsf{Rand}(\mathcal{I})+(\mathsf{Val}(\mathcal{I})-\mathsf{Rand}(\mathcal{I}))\cdot\Omega(1/B)$ *weight.*

We call a set E of constraints *simple set* if there are no two constraints $e_1,e_2\in E$ with $\Gamma(e_1)=\Gamma(e_2)$. We can assume in the proof that E is simple, or we can combine the two constraints sharing the same $\Gamma(\cdot)$ into a new constraint (the new pay-off function is just an addition of two old pay-off functions, perhaps with some permutations), and this does not increase the occurrence bound B.

Ideal proof sketch of Lemma 5.2. Similarly as we did with monotone ordering problems, our ideal goal is to first show that for each constraint $e\in E$, the 4-ordering pay-off function π_e ensures that $\hat{\|}\pi_e\hat{\|}$ is proportional to the maximum possible value of ω_e, specifically to argue that there exists some constant $c>0$, such that $\hat{\|}\pi_e\hat{\|}\geqslant c(\max_{\mathcal{O}}\{\omega_e(\mathcal{O})\}-\mathbf{E}_{\mathcal{O}}[\omega_e(\mathcal{O})])$. Then, because this part of the Fourier spectrum cannot be canceled with coefficients of other constraints, they will appear in the objective function $f(x)=\sum_{e\in E}\pi_e(x)$. This will give a good lower bound on $|f|$, as follows:

$$\begin{aligned}|f|\geqslant\sum_{e\in E}\hat{\|}\pi_e\hat{\|} &\geqslant \sum_{e\in E}c\big(\max_{\mathcal{O}}\{\omega_e(\mathcal{O})\}-\mathop{\mathbf{E}}_{\mathcal{O}}[\omega_e(\mathcal{O})]\big)\\ &= c\Big(\sum_{e\in E}\max_{\mathcal{O}}\{\omega_e(\mathcal{O})\}-\mathop{\mathbf{E}}_{\mathcal{O}}[\sum_e\omega_e(\mathcal{O})]\Big)\geqslant c(\mathsf{Val}(\mathcal{I})-\mathsf{Rand}(\mathcal{I}))\,.\end{aligned}$$

At this point, we can use Proposition 3.1 to get a non-trivial gain over random solution.
But unfortunately, sometimes $\hat{\|}\pi_e\hat{\|}$ can be 0 even when there is a large gap between $\max_{\mathcal{O}}\{\omega_e(\mathcal{O})\}$ and $\mathbf{E}_{\mathcal{O}}[\omega_e(\mathcal{O})]$.

Fact 5.3. *Let $e = (v_i, v_j, v_k)$, the following pay-off functions are such kind of examples (for the statement above).*

- $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_i)<\mathcal{O}(v_j)}$, *and* $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_i)>\mathcal{O}(v_j)}$*;*
- $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_j)<\mathcal{O}(v_k)}$, *and* $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_k)>\mathcal{O}(v_j)}$*;*
- $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_k)<\mathcal{O}(v_i)}$, *and* $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_i)>\mathcal{O}(v_k)}$*;*
- $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_i)<\mathcal{O}(v_j)<\mathcal{O}(v_k)} + \mathbf{1}_{\mathcal{O}(v_j)<\mathcal{O}(v_k)<\mathcal{O}(v_i)} + \mathbf{1}_{\mathcal{O}(v_k)<\mathcal{O}(v_i)<\mathcal{O}(v_j)}$, *and* $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_k)<\mathcal{O}(v_j)<\mathcal{O}(v_i)} + \mathbf{1}_{\mathcal{O}(v_i)<\mathcal{O}(v_k)<\mathcal{O}(v_j)} + \mathbf{1}_{\mathcal{O}(v_j)<\mathcal{O}(v_i)<\mathcal{O}(v_k)}$ *.*

It is easy to see that first three pairs of pay-off functions have $\hat{\|}\pi_e\hat{\|} = 0$ since they only depend on two out of three coordinates. For the last pair, note we can rewrite them as summation of pay-off functions that depend on no more than two coordinates : $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_i)<\mathcal{O}(v_j)} + \mathbf{1}_{\mathcal{O}(v_j)<\mathcal{O}(v_k)} + \mathbf{1}_{\mathcal{O}(v_k)<\mathcal{O}(v_i)} - 1$ and $\omega_e(\mathcal{O}) = \mathbf{1}_{\mathcal{O}(v_k)<\mathcal{O}(v_j)} + \mathbf{1}_{\mathcal{O}(v_j)<\mathcal{O}(v_i)} + \mathbf{1}_{\mathcal{O}(v_i)<\mathcal{O}(v_k)} - 1$, respectively. We call the pay-off functions shown in Fact 5.3 as *binary representable* pay-off functions.

Proof of Lemma 5.2. In view of the above, we claim the following crucial lemma which shows that the above binary-representable pay-offs are the only obstacles to our plan. The proof of the lemma is deferred to the end of this section.

Lemma 5.4. *There exists a constant $c > 0$, such that for every 3-ary pay-off function ω_e on some triple $e = (v_i, v_j, v_k)$ the following holds. If for every binary representable pay-off function ω, there exists some ordering $\mathcal{O}$ of v_i, v_j, v_k such that $\omega(\mathcal{O}) > 0$ and $\omega_e(\mathcal{O}) = 0$ (i.e., ω_e does not "contain" any binary representable pay-off functions), then*

$$\hat{\|}\pi_e\hat{\|} \geqslant c \cdot \max_{\mathcal{O}}\{\omega_e(\mathcal{O})\} .$$

Using Lemma 5.4, it is not hard to see that we can rewrite the set E of constraints (together with their pay-off functions) as $E_3 \cup E_2$ together with a constant $b \in \mathbb{R}$, such that

$$\sum_{e\in E} \omega_e = \sum_{e_3\in E_3} \omega_{e_3} + \sum_{e_2\in E_2} \omega_{e_2} + b ,$$

where E_3 is a simple set of 3-ary pay-off functions none of which contain binary representable pay-off functions, and E_2 is a simple set of (weighted) monotone 2-ordering constraints.

The constraints in E_2 can be handled by virtue of Claim 4.5, since they just correspond to monotone k-ordering constraints for $k = 2$.

Corollary 5.5 (of Claim 4.5)**.** *For every $e_2 \in E_2$, $\hat{\|}\pi_{e_2}\hat{\|} \geqslant 1/2 \cdot \max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}$.*

We emphasize that pay-off functions are always non-negative, and therefore

Fact 5.6. $\mathsf{Rand}(\mathcal{I}) = \mathop{\mathbf{E}}_{\mathcal{O}}\Big[\sum_{e\in E}\omega_e(\mathcal{O})\Big] = \mathop{\mathbf{E}}_{\mathcal{O}}\Big[\sum_{e_3\in E_3}\omega_{e_3}(\mathcal{O}) + \sum_{e_2\in E_2}\omega_{e_2}(\mathcal{O}) + b\Big] \geqslant b\,.$

Remember our 4-ary objective function $f : [4]^n \to \mathbb{R}$ is

$$f(\mathcal{O}_4) = \mathop{\mathbf{E}}_{\mathcal{O}\sim\mathcal{O}_4}\Big[\sum_{e\in E}\omega_e(\mathcal{O})\Big] = \sum_{e_3\in E_3}\pi_{e_3}(\mathcal{O}_4) + \sum_{e_2\in E_2}\pi_{e_2}(\mathcal{O}_4) + b,$$

by a similar argument as Fact 4.4, we know that, viewing f as a function defined on Boolean cube, we have

Fact 5.7. $\hat{f}(\emptyset) = \mathop{\mathbf{E}}_{x}[f(x)] = \mathop{\mathbf{E}}_{\mathcal{O}}\Big[\sum_{e\in E}\omega_e(\mathcal{O})\Big] = \mathsf{Rand}(\mathcal{I})\,.$

Since f is a $O(B)$-occurrence bounded polynomial, in order to apply Proposition 3.1 to finish this proof, we only need to prove that

$$\Big|\sum_{e_3\in E_3}\pi_{e_3} + \sum_{e_2\in E_2}\pi_{e_2}\Big| \geqslant \Big(\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\} + \sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}\Big)\cdot\Omega(1), \qquad (3)$$

as this would imply that

$$|f| = \Big|\sum_{e_3\in E_3}\pi_{e_3} + \sum_{e_2\in E_2}\pi_{e_2}\Big| \geqslant (\mathsf{Val}(\mathcal{I}) - b)\cdot\Omega(1) \geqslant (\mathsf{Val}(\mathcal{I}) - \mathsf{Rand}(\mathcal{I}))\cdot\Omega(1)$$

where the last step used Fact 5.6.

To prove (3), we first establish the following upper bound.

Fact 5.8. *For each $e_3 \in E_3$, we have*

$$|\pi_{e_3}| \leqslant \sum_{S\subseteq[6]}|\widehat{\pi_{e_3}}(S)| \leqslant \sqrt{2^6\Big(\sum_{S\subseteq[6]}\widehat{\pi_{e_3}}(S)^2\Big)}$$
$$= 8\sqrt{\mathop{\mathbf{E}}_{x\in\{-1,1\}^6}[\pi_{e_3}(x)^2]} \leqslant 8\max_{x\in\{-1,1\}^6}\{\pi_{e_3}(x)\} \leqslant 8\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\}.$$

We discuss the following two cases to establish (3).

When $\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\} \geqslant \frac{1}{32}\cdot\sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}$, by Lemma 5.4, we have

$$\Big|\sum_{e_3\in E_3}\pi_{e_3} + \sum_{e_2\in E_2}\pi_{e_2}\Big| \geqslant \sum_{e_3\in E_3}\hat{\|}\pi_{e_3}\hat{\|} \geqslant c\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\}$$
$$\geqslant \frac{c}{33}\cdot\Big(\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\} + \sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}\Big).$$

On the other hand, when $\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\} < 1/32\cdot\sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}$, by Corollary 5.5 and Fact 5.8,

$$\Big|\sum_{e_3\in E_3}\pi_{e_3} + \sum_{e_2\in E_2}\pi_{e_2}\Big| \geqslant \sum_{e_2\in E_2}\hat{\|}\pi_{e_2}\hat{\|} - \sum_{e_3\in E_3}|\pi_{e_3}|$$
$$\geqslant \frac{1}{2}\sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\} - 8\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\} \geqslant \Big(\frac{1}{2}-\frac{1}{4}\Big)\sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}$$
$$\geqslant \frac{1}{4}\cdot\frac{32}{33}\cdot\Big(\sum_{e_3\in E_3}\max_{\mathcal{O}}\{\omega_{e_3}(\mathcal{O})\} + \sum_{e_2\in E_2}\max_{\mathcal{O}}\{\omega_{e_2}(\mathcal{O})\}\Big).$$

□ (Lemma 5.2)

Now the only thing left is to prove Lemma 5.4 .

Proof of Lemma 5.4. W.l.o.g. suppose that the 3-ary constraint $e = (v_1, v_2, v_3)$, and its pay-off function

$$\begin{aligned}\omega_e(\mathcal{O}) &= a_1\cdot\mathbf{1}_{\mathcal{O}(v_1)<\mathcal{O}(v_2)<\mathcal{O}(v_3)} + a_2\cdot\mathbf{1}_{\mathcal{O}(v_1)<\mathcal{O}(v_3)<\mathcal{O}(v_2)} + a_3\cdot\mathbf{1}_{\mathcal{O}(v_2)<\mathcal{O}(v_1)<\mathcal{O}(v_3)}\\ &\quad + a_4\cdot\mathbf{1}_{\mathcal{O}(v_2)<\mathcal{O}(v_3)<\mathcal{O}(v_1)} + a_5\cdot\mathbf{1}_{\mathcal{O}(v_3)<\mathcal{O}(v_1)<\mathcal{O}(v_2)} + a_6\cdot\mathbf{1}_{\mathcal{O}(v_3)<\mathcal{O}(v_2)<\mathcal{O}(v_1)}\,.\end{aligned}$$

We can check by definition that

$$\begin{aligned}\widehat{\pi_e}(\{1,2,4,6\}) &= 12/(2^6\cdot 3!)\cdot(-a_1-a_2+2a_3-a_4+2a_5-a_6),\\ \widehat{\pi_e}(\{2,3,4,6\}) &= 12/(2^6\cdot 3!)\cdot(2a_1-a_2-a_3-a_4-a_5+2a_6),\\ \widehat{\pi_e}(\{2,4,5,6\}) &= 12/(2^6\cdot 3!)\cdot(-a_1+2a_2-a_3+2a_4-a_5-a_6).\end{aligned}$$

Let $\boldsymbol{a}=(a_1,a_2,a_3,a_4,a_5,a_6)\in(\mathbb{R}^{\geqslant 0})^6$, and $\boldsymbol{u_1}=1/2\sqrt{3}\cdot(-1,-1,2,-1,2,-1), \boldsymbol{u_2}=1/2\sqrt{3}\cdot(2,-1,-1,-1,-1,2), \boldsymbol{u_3}=1/2\sqrt{3}\cdot(-1,2,-1,2,-1,-1)$ be unit vectors. Let $\mathrm{span}(\boldsymbol{v})$ be the linear space spanned by vector $\boldsymbol{v}$. For any linear space $D\subseteq\mathbb{R}^6$, denote by $D^\perp$ the orthogonal complement of D (in $\mathbb{R}^6$). Use $\mathrm{dist}(\boldsymbol{v},D)$ denote the Euclidean distance from vector $\boldsymbol{v}$ to space D. By the above identities

$$\begin{aligned}\hat{\|}\pi_e\hat{\|} &\geqslant |\widehat{\pi_e}(\{1,2,4,6\})|+|\widehat{\pi_e}(\{2,3,4,6\})|+|\widehat{\pi_e}(\{2,4,5,6\})|\\ &= 1/32\cdot\left(\mathrm{dist}(\boldsymbol{a},\mathrm{span}(\boldsymbol{u_1})^\perp)+\mathrm{dist}(\boldsymbol{a},\mathrm{span}(\boldsymbol{u_2})^\perp)+\mathrm{dist}(\boldsymbol{a},\mathrm{span}(\boldsymbol{u_3})^\perp)\right).\end{aligned}\tag{4}$$

Since ω_e does not include binary representable pay-off functions listed in Fact 5.3, we know that $\boldsymbol{a}$ belongs to the set A defined by

$$\begin{aligned}A=\{\boldsymbol{x}=(x_1,x_2,x_3,x_4,x_5,x_6)\in(\mathbb{R}^{\geqslant 0})^6 : x_1x_2x_5=x_3x_4x_6=0,\\ x_1x_2x_3=x_4x_5x_6=0, x_1x_3x_4=x_2x_5x_6=0, x_1x_4x_5=x_2x_3x_6=0\}.\end{aligned}$$

One can check that

$$\begin{aligned}A &= \{(x_1,x_2,0,x_4,0,x_6) : x_1,x_2,x_4,x_6\geqslant 0\}\cup\{(x_1,0,x_3,0,x_5,x_6) : x_1,x_3,x_5,x_6\geqslant 0\}\\ &\quad\cup\{(0,x_2,x_3,x_4,x_5,0) : x_2,x_3,x_4,x_5\geqslant 0\}.\end{aligned}\tag{5}$$

We now prove the following fact which will help us lower bound (4).

Lemma 5.9. *Given a linear space $K\subseteq\mathbb{R}^d$ (we only use the case $d=6$ in the proof of Lemma 5.4), and $T\subseteq\mathbb{R}^d$, a intersection of finite closed half-spaces (which pass the origin). If $T\cap K=\{\mathbf{0}\}$, then there exists a constant $c_0>0$, such that $\forall\boldsymbol{x}\in T, \mathrm{dist}(\boldsymbol{x},K)\geqslant c_0\cdot\|\boldsymbol{x}\|_2$.*

Proof. By the property of T, we know that 1) T is closed, and 2) $\forall\boldsymbol{x}\neq\mathbf{0},\lambda>0,\boldsymbol{x}\in T\Leftrightarrow\lambda\boldsymbol{x}\in T$. By the second property, we only need to prove $\forall\boldsymbol{x}\in T, ||\boldsymbol{x}||_2=1, \mathrm{dist}(\boldsymbol{x},K)\geqslant c_0$. Then by the first property, $f(\boldsymbol{x})=\mathrm{dist}(\boldsymbol{x},K)$, being a continuous mapping defined on closed set $T\cap\mathbb{S}^{d-1}$, has its image set $I=\{\mathrm{dist}(\boldsymbol{x},K):\boldsymbol{x}\in T\cap\mathbb{S}^{d-1}\}$ closed (where $\mathbb{S}^{d-1}$ is the unit sphere in $\mathbb{R}^d$). Since $K\cap(T\cap\mathbb{S}^{d-1})=\emptyset$, we can set $c_0=\inf I>0$. □

We can view each of the three components of A in (5) as intersection of closed half-spaces (passing through the origin). Then we check that the i-th ($i=1,2,3$) component only intersects $\mathrm{span}(\boldsymbol{u_i})^\perp$ at $\mathbf{0}$. Therefore by Lemma 5.9 we conclude that there exists $c>0$, such that for all $\boldsymbol{a}\in A$,

$$\hat{\|}\pi_e\hat{\|}\geqslant\sum_{i=1,2,3}\mathrm{dist}(\boldsymbol{a},\mathrm{span}(\boldsymbol{u_i})^\perp)\geqslant c\cdot||\boldsymbol{a}||_2\geqslant c\cdot\max_i|a_i|=c\cdot\max_{\mathcal{O}}\{\omega_e(\mathcal{O})\}.$$

□

References

[1] P. Austrin and E. Mossel. Approximation resistant predicates from pairwise independence. In *23rd Annual IEEE Conference on Computational Complexity*, pages 249–258. IEEE Computer Society, 2008.

[2] B. Berger and P. W. Shor. Tight bounds for the Maximum Acyclic Subgraph problem. *J. Algorithms*, 25(1):1–18, 1997.

[3] M. Charikar, V. Guruswami, J. Håstad, R. Manokaran, and P. Raghavendra. Beating the random ordering is hard: All ordering CSPs are approximation resistant. In preparation, 2010.

[4] M. Charikar, V. Guruswami, and R. Manokaran. Every permutation CSP of arity 3 is approximation resistant. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 62–73, July 2009.

[5] B. Chor and M. Sudan. A geometric approach to betweenness. *SIAM J. Discrete Math.*, 11(4):511–523, 1998.

[6] L. Engebretsen and V. Guruswami. Is constraint satisfaction over two variables always easy? *Random Structures and Algorithms*, 25(2):150–178, 2004.

[7] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[8] V. Guruswami, R. Manokaran, and P. Raghavendra. Beating the random ordering hard: Inapproximability of maximum acyclic subgraph. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, pages 573–582, 2008.

[9] J. Håstad. On bounded occurrence constraint satisfaction. *Inf. Process. Lett.*, 74(1-2):1–6, 2000.

[10] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[11] J. Håstad. Every 2-CSP allows nontrivial approximation. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 740–746, 2005.

[12] Y. Makarychev. Simple linear time approximation algorithm for betweenness. *Microsoft Research Technical Report MSR-TR-2009-74*, 2009.

[13] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 245–254, 2008.

[14] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 453–461, 2001.