ON LP-BASED APPROXIMABILITY FOR STRICT CSPS

Amit Kumar* Rajsekar Manokaran† Madhur Tulsiani‡ Nisheeth K. Vishnoi §

Abstract

In a beautiful result, Raghavendra established optimal Unique Games Conjecture (UGC)-based inapproximability for a large class of constraint satisfaction problems (CSPs). In the class of CSPs he considers, of which MAXIMUM CUT is a prominent example, the goal is to find an assignment which maximizes a weighted fraction of constraints satisfied. He gave a generic semi-definite program (SDP) for this class of problems and showed how the approximability of each problem is determined by the corresponding SDP (upto an arbitrarily small additive error) assuming the UGC. He noted that his techniques do no apply to CSPs with *strict* constraints (all of which *must* be satisfied) such as VERTEX COVER.

In this paper we address the approximability of these strict-CSPs. In the class of CSPs we consider, one is given a set of constraints over a set of variables, and a cost function over the assignments, the goal is to find an assignment to the variables of minimum cost which satisfies *all* the constraints. We present a generic linear program (LP) for a large class of strict-CSPs and give a systematic way to convert integrality gaps for this LP into UGC-based inapproximability results. Some important problems whose approximability our framework captures are VERTEX COVER, HYPERGRAPH VERTEX COVER, k-partite-HYPERGRAPH VERTEX COVER, INDEPENDENT SET and other covering and packing problems over q-ary alphabets, and a scheduling problem. For the covering and packing problems, which occur quite commonly in practice as well, we provide a matching rounding algorithm, thus settling their approximability upto an arbitrarily small additive error.

*Dept. of Computer Science and Engineering, IIT Delhi, New Delhi, email: amitk@cse.iitd.ac.in. This work was done while the author was visiting Microsoft Research India.

†Department of Computer Science, Princeton University, email: rajsekar@cs.princeton.edu. Supported by NSF Grants CCF-0832797, 0830673, and 0528414.

‡Institute for Advanced Study, Princeton, email: madhurt@ias.edu. Supported by NSF grant 0832797 and IAS sub-contract no. 00001583.

§Microsoft Research India, Bangalore, email: nisheeth.vishnoi@gmail.com

1 Introduction

In this paper we address the approximability of strict-Constraint Satisfaction Problems (CSPs). An instance of such a problem is specified by positive integers k, q and n, a collection of ordered k-tuples of $\{1,\ldots,n\}$ denoted by E and a collection of subsets of $[q]^k$, one for every $e \in E$, denoted by $\{A_e\}_{e\in E}$. It is customary to call the set $[n]$ the vertex set (denoted by V) and the elements of E, edges or hyper-edges. The collection of subsets are called the constraints and the constraint A_e in particular is said to be the constraint on the edge e. The goal is to assign labels $x_1, x_2, \ldots x_n$ from $[q]$ to the vertices such that for *every* edge e, the corresponding k-tuple of labels is an element of A_e (or "satisfy" the constraint on every edge) while minimzing $\sum_i x_i$, also called the objective function. In a more general setting, a set of weights $w_1, \ldots w_n$ is also specified and the objective is to minimize $\sum_i w_i x_i$. Fixing k and restricting the choice of the constraints A_e allowed in the specification (as opposed to allowing arbitrary subsets of $[q]^k$) gives raise to particular classes of strict-CSPs. Many important optimization problems are captured by this specification: VERTEX COVER, HYPERGRAPH VERTEX COVER, INDEPENDENT SET, covering and packing problems to name a few.

Note that strict-CSPs are different from the CSPs considered by Raghavendra [Rag08] where the goal, given a set of constraints is to find an assignment which maximizes a payoff function associated with whether a constraint is satisfied or not and, in particular, assignments which satisfy only part of the constraints are feasible, e.g., MAXIMUM CUT. We refer to them as strict-CSPs precisely for this reason. Even though optimal inapproximability and approximability for several problems such as MAXIMUM CUT which fell in Raghavendra's framework were known before (see [Rag08]), the main feature of his result was the use of semi-definite programming (SDP)-integrality gaps to come up with Unique Games Conjecture (UGC)-based hardness reductions, complementing the result of Khot and Vishnoi [KV05] who show how to use UGC-based hardness reductions to come up with SDP-integrality gaps. He gave a generic SDP for this class of CSPs and showed how the approximability of each problem is determined by the corresponding SDP up-to an arbitrarily small additive error assuming the UGC. He noted in his paper that his techniques do no apply to strict-CSPs such as Vertex Cover and Graph-3-Coloring.

In this paper we present a framework similar to the one in [Rag08] which applies to a large class of strict-CSPs. In particular, we show that a natural linear program (LP) captures precisely (up-to arbitrarily small additive error) the approximability of strict-CSPs such as covering-packing problems, which include Vertex Cover, Hypergraph Vertex Cover and Independent Set, as observed by Guruswami and Saket [GS10] - the k-partite-k-uniform-Hypergraph Vertex Cover problem, and the concurrent open shop problem in scheduling [MQS+09], [BK09a]. We show how to convert integrality gap for the LP for these problems to a Unique Games-based hardness of approximation result in a principled way. Thus, the above results are obtained by invoking known integrality gaps for the above-mentioned problems. In addition, for covering-packing problems we give a simple rounding algorithm which achieves the integrality gap, again up-to an arbitrarily small additive constant. The rounding result is an analogue in the strict-CSP world of that obtained by Raghavendra and Steurer [RS09b]. Note that apart from the natural packing and covering problems studied in the theory community and mentioned above, covering-packing problems also appear quite commonly in the operations research community as models for real world problems and we hope that our algorithmic results will be useful there.

We do not attempt to list all the corollaries in this paper and, rather, focus on providing a systematic framework to compose LP integrality gap instances for strict-CSPs with Unique Games instances and to demonstrate how the rounding algorithm comes out as a natural by-product of the soundness analysis.

Before we describe our results, it would be useful to introduce some notation. We keep the discussion here in the $\{0,1\}$ setting for the ease of presentation. The results extend to the q-ary world in a straightforward manner and we present the details in Section C.

1.1 Preliminaries

Strict-CSPs. A problem Π is said to be a k-strict-CSP if it consists of a set of vertices (variables) V with weights $\{w_v\}_{v\in V}$ on them, a set of hyper-edges of size at-most k and for every hyper-edge $e \in E$, a constraint $A_e \subseteq \{0,1\}^{|e|}$.[1] We will assume that $\sum_{v\in V}|w_v| = 1$. The objective is to find a $\{0,1\}$ assignment to the vertices so as to satisfy *all* the hyper-edge constraints and minimize $\sum_v w_v x_v$. This requirement, that in a feasible assignment all the constraints be satisfied, is why we refer to these CSPs as strict. If each constraint A_e is upward-monotone, i.e., given a feasible solution (a subset of vertices), adding more vertices to the solution keeps it feasible, and each $w_v \geq 0$, we refer to the problem as k-strict$^{\uparrow}$-CSP. If each constraint is downward-monotone, i.e., given a feasible solution (a subset of vertices), deleting vertices from the solution keeps it feasible, and each $w_v \leq 0$, we refer to the problem as k-strict$^{\downarrow}$-CSP. strict$^{\uparrow}$-CSP is also referred to as a covering problem while strict$^{\downarrow}$-CSP as a packing problem. Observe that Vertex Cover is a 2-strict$^{\uparrow}$-CSP and Independent Set a 2-strict$^{\downarrow}$-CSP. Sometimes, we may also be interested in a k-strict-CSP Π where the input hyper-graph has some pre-specified structure, e.g., the hyper-graph could be k-partite and k-uniform.

The LP for a k-strict-CSP problem. One can define the following generic LP relaxation for any k-strict-CSP. This relaxation is inspired by the Sherali-Adams [SA90] relaxation and plays a crucial role in our results.

(1.1)
$$\mathsf{lp}(\mathcal{I}) \stackrel{\text{def}}{=} \quad \text{minimize} \quad \sum_{v\in V} w_v x_v$$

(1.2)
$$\text{subject to} \quad \forall_{e=(v_1,v_2,\ldots,v_l)\in E} \quad (x_{v_1}, x_{v_2}, \ldots, x_{v_l}) \in \texttt{ConvexH}$$

(1.3)
$$\forall_{v\in V} \quad 0 \leq x_v \leq 1$$

Figure 1: LP for k-strict-CSP

Here, for a hyper-edge $e = (v_1,\ldots,v_l)$, $\texttt{ConvexHull}(A_e)$ denotes the convex hull of all assignments $\sigma \in \{0,1\}^l$ which satisfy the constraint A_e. For an instance $\mathcal{I}$, let $\mathsf{lp}(\mathcal{I})$ denote the optimum of the LP of Figure 1 for $\mathcal{I}$. Let $\mathsf{val}(\mathcal{I}, x)$ denote the value of $\mathsf{LP}(\mathcal{I})$ for a feasible x to it. Also, let $\mathsf{opt}(\mathcal{I})$ denote the value of the optimal integral solution for $\mathcal{I}$. For the sake of readability, we will assume that all the hyper-edges are exactly of size k.

Connected LP-solutions. Mossel [Mos08] introduced a notion of connectedness which we recall here. Two points $(x_1,\ldots,x_k), (y_1,\ldots,y_k) \in \{0,1\}^k$ are said to be connected by an edge if they differ in at-most one coordinate. A subset $S \subseteq \{0,1\}^k$ is said to be connected if the subgraph induced by the vertices of S along with the edges is connected. For an instance $\mathcal{I}$ of a k-strict-CSP, given a solution x to $\mathsf{LP}(\mathcal{I})$, x-is said to be connected if for every edge $e = (v_1,\ldots,v_l)$, $(x_{v_1},\ldots,x_{v_l})$ can be written as a convex combination of points in A_e such that the support of this convex combination is connected.

1.2 Results

[1] If A_e is not symmetric then the hyper-edge e is best thought of as an ordered tuple.

Theorem 1.1. (LP-integrality gap based Inapproximability) *Let Π be a k-strict-CSP for $k = O(1)$, and $\mathcal{J}$ be a constant-sized instance of Π. Let x be a feasible-connected solution for $\mathsf{LP}(\mathcal{J})$. Then for every $\delta > 0$, it is* Unique Games*-hard to distinguish between the following instances $\mathcal{I}$ of Π*

- **YES.** $\mathsf{opt}(\mathcal{I}) \leq \mathsf{val}(\mathcal{J}, x) + \delta$
- **NO.** $\mathsf{opt}(\mathcal{I}) \geq \mathsf{opt}(\mathcal{J}) - \delta$.

Hence, if it is the case that x is also an optimal solution to $\mathsf{LP}(\mathcal{J})$, then, assuming the UGC, the LP captures the approximability of the problem Π. In general, it is not clear whether the LP solution achieving the integrality gap is connected. Hence, the inapproximability obtained using connected-LP solutions may be weaker than the integrality gap. For k-strict$^{\uparrow}$-CSP and k-strict$^{\downarrow}$-CSP we can easily convert any optimal LP solution to a connected one with at-most a δ loss in the LP value, for arbitrarily small constant δ. Hence, we get the following important corollary which proves that the LP of Figure 1 captures precisely the approximability of all covering and packing problems with $k = O(1)$.

Corollary 1.1. (Optimal Inapproximability for Covering and Packing Problems) *Let Π be a k-strict$^{\uparrow}$-CSP or a k-strict$^{\downarrow}$-CSP for $k = O(1)$, and $\mathcal{J}$ be a constant-sized instance of Π. Then for every $\delta > 0$, it is* Unique Games*-hard to distinguish between the following instances $\mathcal{I}$ of Π*

- **YES.** $\mathsf{opt}(\mathcal{I}) \leq \mathsf{lp}(\mathcal{J}) + \delta$
- **NO.** $\mathsf{opt}(\mathcal{I}) \geq \mathsf{opt}(\mathcal{J}) - \delta$.

We will, henceforth, keep the discussion just to covering problems. All results can be directly translated in the packing world and we omit the details.

Rounding for covering-packing problems. For a k-strict$^{\uparrow}$-CSP Π we give a rounding algorithm called ROUND (see Figure 2) for the LP of Figure 1. For an instance $\mathcal{I}$ of Π, a solution x to $\mathsf{LP}(\mathcal{I})$, and a parameter $\varepsilon > 0$, which should be ignored for this discussion, let $\mathsf{round}(\mathcal{I}, x, \varepsilon)$ denote the value of the integral solution that ROUND produces for $\mathcal{I}$ starting from the LP solution x. We show that ROUND (unconditionally) achieves an approximation ratio equal to the integrality gap, up to an arbitrarily small additive constant, of the LP relaxation. This can be seen as an analogue of the result of Raghavendra and Steurer [RS09a] for the class of CSPs considered by Raghavendra [Rag08].

Theorem 1.2. (Rounding achieves Integrality Gap) *Let $\gamma^*(\Pi)$ be the worst-case approximation ratio (integrality gap) achieved by the LP relaxation for a k-strict$^{\uparrow}$-CSP Π, i.e., $\gamma^*(\Pi) \stackrel{\text{def}}{=} \sup_{\mathcal{J}}(\mathsf{opt}(\mathcal{J})/\mathsf{lp}(\mathcal{J}))$, where the supremum is taken over all instances $\mathcal{J}$ of Π. Then, for any given instance $\mathcal{I}$, an optimal LP solution $x^\star$ and $\varepsilon > 0$, $\mathsf{round}(\mathcal{I}, x^\star, \varepsilon) \leq \gamma^*(\Pi) \cdot (\mathsf{opt}(\mathcal{I}) + \varepsilon)$.*

For covering and packing problems, we show how to start with an instance $\mathcal{J}$ of Π and a solution x to $\mathsf{LP}(\mathcal{J})$, , and give a Unique Games-based reduction for Π whose soundness and completeness are roughly $\mathsf{val}(\mathcal{J}, x)$ and $\mathsf{round}(\mathcal{J}, x, \varepsilon)$ respectively. The reduction in this theorem is slightly different from that in the corollary. This theorem is more useful in the case when it is easier to come up with a LP-rounding gap rather than an integrality gap.

Corollary 1.2. (LP-rounding gap based Inapproximability) *Let Π be a k-strict$^{\uparrow}$-CSP for $k = O(1)$, and $\mathcal{J}$ be a constant-sized instance of Π, and x a solution to $\mathsf{LP}(\mathcal{J})$, Then for every $\delta > 0$, it is* Unique Games*-hard to distinguish instances $\mathcal{I}$ of Π with optimum less than $\mathsf{val}(\mathcal{J}, x) + 2\delta$ from those with optimum more than $\mathsf{round}(\mathcal{J}, x, \delta) - \delta$.*

1.3 Applications, Comparisons and Discussions

Comparison to previous hardness results on Vertex Cover **and** Hypergraph Vertex Cover**.** The k-Hypergraph Vertex Cover problem is the following: given a hyper-graph with each edge of cardinality at most k, the goal is to pick the smallest set of vertices such that every hyper-edge contains at-least one vertex in the picked set. The Vertex Cover problem is the 2-Hypergraph Vertex Cover problem. Vertex Cover and k-Hypergraph Vertex Cover have been extensively studied: while there is a simple factor k-approximation algorithm for it, on the hardness side, there is a series of results based on standard complexity assumptions [DS02, Hås97, Tre01, Gol01, Hol02, DGKR03]. They all fall short of coming arbitrarily close to the upper bound of k. Khot and Regev [KR08] proved that, assuming the UGC, k-Hypergraph Vertex Cover cannot be approximated to within a factor better than $k - \varepsilon$ for any $k \geq 2$ and any constant $\varepsilon > 0$. The $2-\varepsilon$ hardness for Vertex Cover has been reproved in [AKS09, BK09b, BK09a]. The analysis of Austrin, Khot and Safra [AKS09] also depends on Mossel's Invariance Principle and they were motivated by the problem of proving hardness of approximating Vertex Cover and Independent Set on bounded degree graphs.

Since k-Hypergraph Vertex Cover falls in the class k-strict$^{\uparrow}$-CSP, the existence of a $k - \varepsilon$ factor LP-integrality gap for these problems re-establishes these Unique Games-hardness results using Corollary 1.1. Note that our LP for the k-Hypergraph Vertex Cover problem is equivalent to the standard one in the literature. The advantage of our approach is that it converts any integrality gap into an inapproximability result. Moreover, since the reduction inherits the structure of the integrality-gap, our result has been used to derive new optimal inapproximability result for the k-partite-k-uniform-HYPERGRAPH VERTEX COVER problem by Guruswami and Saket [GS10] discussed later in this section.

Interestingly, we can also derive the $k-\varepsilon$ hardness result for k-HYPERGRAPH VERTEX COVER using a LP-rounding gap and appealing to Corollary 1.2. Consider the following instance $\mathcal{I}$ of k-HYPERGRAPH VERTEX COVER– we are given a set V of size k, and there is only one hyper-edge in E, namely, the set of all vertices in V. The weight of every vertex is $1/k$. Consider the solution x which assigns value $1/k$ to all variables x_u. It is easy to check that it is feasible to our LP relaxation. The value of the solution x is $1/k$. . Let us now see how the algorithm $\mathsf{ROUND}(\mathcal{I}, x, \varepsilon)$, where $\varepsilon < 1/k$, rounds the solution x. All entries in x^ε will still be same. Hence, the rounding algorithm will consider only two options – either pick all vertices in V, or do not pick any vertex. Since the latter case yields an infeasible solution, it will output the set V, which has value 1. Corollary 1.2 now implies that assuming UGC, it is NP-hard to distinguish between instances of k-HYPERGRAPH VERTEX COVER where the optimal value is at-most $1/k - 2\varepsilon$ from those where the optimal value is more than $1-\varepsilon$. Note that the integrality gap of $\mathsf{LP}(\mathcal{I})$ is 1. Still we are able to argue hardness of k-HYPERGRAPH VERTEX COVER problem starting from such an instance because the algorithm ROUND performs poorly on this instance. In this sense, the statement of Corollary 1.2 is stronger than that of Corollary 1.1.

Inheritance of structure from the starting instance: k-Partite-k-Uniform Hypergraph Vertex Cover. A nice feature about composing integrality gaps with UNIQUE GAMES-instances is that some structure of the integrality gap shows up in the final instance. Guruswami and Saket [GS10] considered the problem of k-partite-k-uniform-HYPERGRAPH VERTEX COVER, where, in addition to the vertices and the hyper-edges, one is also given a k-partition of the vertex set and each hyper-edge contains exactly one vertex from a partition. As proved by Lovasz [Lov75], this problem has a $k/2$-approximation algorithm. Guruswami and Saket [GS10] show that this problem is NP-hard to approximate to a factor better than $k/16-\varepsilon$ for all $\varepsilon > 0$. Moreover, using a slight modification of the main result from the initial version of this paper (and Corollary 1.1 from this version of this paper), they observe how the $k/2$ integrality gap of Aharoni, Holzman and Krivelevich [AHK96] implies $k/2-\varepsilon$ UNIQUE GAMES-hardness for this problem for any $\varepsilon > 0$ and settles the approximability of this problem. Their result applies for the more general SPLIT-HYPERGRAPH VERTEX COVER problem and we refer the reader to their paper. The key point is that this k-partition is preserved by the reduction if one starts from a k-partite integrality gap. This result demonstrates another interesting feature of our framework.

Application in Scheduling: Concurrent Open Shop. In an initial version of this paper, we restricted our attention to packing/covering problems. Bansal and Khot [BK09a] independently prove a $2-\varepsilon$ for any $\varepsilon > 0$ hardness for the Concurrent Open Shop scheduling problem. Upon reading their paper, we noticed that their result can be proven in our framework using an integrality gap for a related linear programming relaxation constructed by [MQS+09]. In Section A, we show how the gap instance of [MQS+09] can be used with Theorem 1.1 to obtain a $2-\varepsilon$ inapproximability assuming the UGC. We believe that our framework is more general and should help prove more inapproximability for similar problems and that the exposition in Section A would be helpful.

Comparison to Raghavendra's result. As noted, we are partially able to address the problems left open by Raghavendra [Rag08]. While he gives a systematic way to compose SDP-integrality gaps for his CSPs with UNIQUE GAMES to settle their approximability, we do the same for covering and packing problems, except that we just rely on LP-integrality gaps. As in his paper, the rounding algorithm for covering and packing problems comes out as a natural but important by-product. The *strict*-ness is critical in our results while, as Theorem 1.1 demonstrates, monotonicity does not seem to be. Both his and our result appeals to Mossel's Invariance Principle [Mos08] but the analysis differs and we end up needing some additional Gaussian estimates as in Austrin, Khot and Safra [AKS09]. We give more details of how our reduction differs from his in Section 1.4.

Computing approximation ratios. Similar to Raghavendra's result, our results do not imply any explicit inapproximability ratios. However, like [Rag08], for any constant $\delta > 0$ we can compute the best approximation ratio to within additive δ error in constant time for covering and packing problems. The proofs are identical to those in [Rag08] and we omit the details.

On monotonicity in the $\{0,1\}$-world. For k-$\mathsf{strict}^{\uparrow}$-$\mathsf{CSP}$ (and k-$\mathsf{strict}^{\downarrow}$-$\mathsf{CSP}$) over the alphabet $\{0,1\}$ one can reduce any problem Π to a l-HYPERGRAPH VERTEX COVER for some $l \le k$ in an approximation preserving sense. This does not happen in in the q-ary world when $q \ge 2$: consider the problem of minimizing $\sum_i x_i$ subject to constraints of the form $x_i + x_j \ge q+1$ and $x_i \in \{0,1,\ldots,q\}$. Also, as we note earlier, [KR08] does not apply when trying to prove inapproximability for families of graph with certain structure (as in the result of Guruswami and Saket [GS10]). Thus, we believe the right approach to understanding the approximability of these problems is not by reduction to a canonical problem in the class, but instead to study where the linear programming relaxation fails for the problem.

Future directions. We anticipate that our results will lead to a better understanding on the power of linear programming in approximability. In particular, we leave as an open problem of proving LP-based inapproximability results for hard-ordering CSPs as in Guruswami, Manokaran and Raghavendra [GMR08]. Our rounding algorithm do not seem to generalize for all strict-CSPs for which we can prove inapproximability results. It would be interesting to study rounding schemes for our LP for these problems It is a very interesting question to characterize strict-CSPs for which our LP captures optimality. Is existence of connected integrality gaps sufficient?

Previous LP inspired hardness results. There are several problems for which the best known inapproximability results have been obtained as follows: first construct integrality gap instances for the standard LP relaxations for these problems and then use these instances as guides for constructing hardness reductions based on standard complexity theoretic assumptions. These reductions yield inapproximability ratios quite close to the actual integrality gaps. Examples include Asymmetric k-center [CGH$^+$04], Group Steiner Tree [HK03] and Average Flow-time on Parallel Machines [GK07]. Assuming UGC, our result proves hardness of a large class of problems in a similar spirit. However, instead of explicitly constructing integrality gap examples for such problems, we give a more direct and intuitive proof that the integrality gap is close to the actual hardness of such problems. We note that the only other result for LPs similar in flavor as ours, though unrelated, is that of [MNRS08] for Multi-Way Cut and Metric Labeling problems.

Unique Games Conjecture. We refer the reader to the survey by Khot [Kho10] on this conjecture and its implications.

1.4 Overview of Techniques

In this section we outline the proof of Theorem 1.1 and how it implies Corollary 1.1.

Recall that we need to establish an inapproximability result for a k-strict-CSP Π, for which we start with a constant-sized instance $\mathcal{J} = (V, E, \{A_e\}_{e\in E}, \{w_v\}_{v\in V})$ of Π itself and a feasible-connected solution x to $\mathsf{LP}(\mathcal{J})$. As is common in basing most hardness results on UGC, we will first construct, for an integer $r \geq 1$, a bigger instance (*dictatorship test gadget*) $\mathcal{D}^r_{\mathcal{J},x}$ of Π and then compose it in a standard way with a Unique Games instance. For this discussion, we restrict ourselves to the dictatorship test gadget. The instance $\mathcal{D}^r_{\mathcal{J},x}$ will have the following components:

– **Vertex Set.** The vertex set of $\mathcal{D}^r_{\mathcal{J},x}$ is $V \times \{0,1\}^r$, i.e., for every vertex $v \in V$, there is an r-dimensional hyper-cube.

– **Vertex Weights.** The weight of a vertex $(v,(a_1,\ldots,a_r))$ will be w_v times the x_v-biased measure of $(a_1,\ldots,a_r)$. x_v is the LP value for the vertex v given by x. The p-biased measure of a point $a \in \{0,1\}^r$ is the probability of getting a if we pick a random point from $\{0,1\}^r$ where each coordinate is i.i.d. with probability of 1 being p.

– **Edges and Constraints.** Recall that for every hyper-edge $e = (v_1,\ldots,v_k)$ in $E(\mathcal{J})$, from the solution x, we can read off a probability distribution P_e on $\{0,1\}^k$. Moreover the constraint in the LP requires that this distribution is supported on A_e, and the hypothesis requires that this support is connected. For every $e = (v_1,\ldots,v_k) \in E(\mathcal{J})$ and every $a^{(1)},\ldots,a^{(k)} \in \{0,1\}^r$, there will be an hyper-edge in $\mathcal{D}^r_{\mathcal{J},x}$ between the vertices $((v_1,a^{(1)}),\ldots,(v_k,a^{(k)}))$ with the constraint A_e. We will also associate a *weight* with this edge which is $\prod_{i=1}^r P_e(a_i^{(1)},\ldots,a_i^{(k)})$. We will not keep any hyper-edges with 0 weight. These weights will be useful for the analysis and are irrelevant to the actual instance since every constraint has to be satisfied.

This completes the description of the dictatorship test gadget. Note that the main difference from what is constructed by Raghavendra is that we have a different hyper-cube for each $v \in V(\mathcal{J})$ whereas he has just one hyper-cube. This is so because we set the weights associated with the vertices based on the LP solution; thus a single hyper-cube might not suffice. Now we state the two class of assignments which we want to understand for this instance $\mathcal{D}^r_{\mathcal{J},x}$.

– **Dictator Assignments.** There are special *dictator* assignments $\{\Lambda_i\}_{i=1}^r$ to vertices of $\mathcal{D}^r_{\mathcal{J},x}$ which satisfy all its constraints and has cost $\mathsf{val}(\mathcal{J},x)$. Namely $\Lambda_i(v,(a_1,\ldots,a_r)) = a_i$.

  – **Feasibility.** It is easy check that Λ_i satisfies all the constraints as for the hyper-edge

$$((v_1,a^{(1)}),\ldots,(v_k,a^{(k)}))$$

  the assignment obtained from Λ_i is $(a_i^{(1)},\ldots,a_i^{(k)})$ which is in the support of P_e (as we threw away hyper-edges with zero weight) which is contained in A_e and, hence, satisfies this hyper-edge.

  – **Cost.** The cost of this assignment is precisely $\sum_{v\in V(\mathcal{J})} w_v x_v = \mathsf{val}(\mathcal{J},x)$. This is because the x_v-biased measure of the set selected by Λ_i in the hyper-cube of v is exactly x_v.

– **Pseudo-random Assignments.** We argue that every assignment to vertices of $\mathcal{D}^r_{\mathcal{J},x}$ which is far from a dictator (which we refer to as *pseudo-random*) and satisfies all the constraints has cost at-least $\mathsf{opt}(\mathcal{J})$ up-to a small additive error. We do this by decoding an assignment λ to $\mathcal{J}$ given a pseudo-random assignment Λ to the gadget. An assignment to the dictatorship test gadget is simply a function $\Lambda : V \times \{0,1\}^r \to \{0,1\}$.

– **Decoding assignment to $\mathcal{J}$.** Let δ be the additional cost we can incur. For every $v \in V$, define $S_v \stackrel{\text{def}}{=} \{b \in \{0,1\} | \ \mathbf{E}_a[\Lambda(v,a) = b] \geq \delta\}$ (the expectation is over picking $a \in \{0,1\}^r$ from the x_v biased measure). Set $\lambda(v)$ to be the element in S_v with minimum cost. (0 has less cost than 1.)
– **Relating cost of λ to Λ.** By definition of S_v, for every $v \in V$, at most a δ mass of the corresponding hyper-cube was assigned a value not in S_v. Since $\lambda(v)$ is the minimum cost element from S_v, we pay at most a δw_v additional cost in λ for the vertex v. Thus,

$$\mathsf{opt}(\mathcal{J}) \leq \mathsf{val}(\mathcal{J},\lambda) \leq \mathsf{val}(\mathcal{D}^r_{\mathcal{J},x},\Lambda)+\delta\sum_v w_v \leq \mathsf{val}(\mathcal{D}^r_{\mathcal{J},x},\Lambda)+\delta$$

– **Feasibility of λ.** We now just have to prove that λ is a feasible assignment. For every constraint hyper-edge $e = (v_1, v_2, \ldots, v_k) \in E$ in $\mathcal{J}$, we will in fact show that $S_{v_1} \times S_{v_2} \times \cdots \times S_{v_k} \subseteq A_e$. This is where we appeal to Mossel's Invariance Principle which in turn requires that x was a feasible-connected solution to $\mathsf{LP}(\mathcal{J})$. This last part is also where we differ from Raghavendra [Rag08]. We crucially rely on the fact that the assignment satisfies all the constraints. Fix an assignment $(s_1, \ldots, s_k) \in S_{v_1} \times S_{v_2} \times \cdots \times S_{v_k}$. If Λ is sufficiently pseudo-random, the probability that we sample $(a^{(1)}, \ldots, a^{(k)})$ such that the event $\Lambda(v_i, a^{(i)}) = s_i$ for every i; can be bounded up-to an ε error for arbitrarily $\varepsilon > 0$ in terms of the equivalent probability in the gaussian world where it can be shown to be a function dependent only on δ, k and P_e. When P_e is connected, this can be shown to be a strictly positive quantity. Choosing ε smaller than the estimate implies that there is a constraint $((v_1, a^{(1)}), \ldots, (v_k, a^{(k)}))$ such that $\Lambda(v_i, a^{(i)}) = s_i$. Since every constraint was satisfied by Λ, $(s_1, \ldots s_k) \in A_e$.

Hence, informally we have the following

1. The cost of any dictator assignment is at-most $\mathsf{val}(\mathcal{J}, x) \leq \mathsf{lp}(\mathcal{J})$.
2. The cost of any pseudo-random assignment is at-least $\mathsf{opt}(\mathcal{J}) - \delta$ for any small enough constant δ.

In Section B we show how to compose the dictatorship test gadget with UNIQUE GAMES-instances in a standard way to prove Theorem 1.1. Before that, we can quickly deduce Corollary 1.1.

Deducing Corollary 1.1 from Theorem 1.1. Let Π be a k-$\mathsf{strict}^{\uparrow}$-$\mathsf{CSP}$, $\mathcal{J}$ be an instance of Π and x any feasible solution to $\mathsf{LP}(\mathcal{J})$. Let $\delta > 0$ be the parameter in Corollary 1.1. Consider $y = (1-\delta)\cdot x + \delta \cdot (1, \ldots, 1)$. For a hyper-edge $e = (v_1, \ldots, v_k) \in E(\mathcal{J})$, let P_e be any probability distribution on $\{0,1\}$ such that $\mathbf{E}_{\sigma \leftarrow P_e}[\sigma] = (x_{v_1}, \ldots, x_{v_k})$. Let Q_e be the probability distribution on $\{0,1\}^k$ obtained from P_e in the following way:

– Pick σ from P_e.
– For each v_i, if $\sigma_{v_i} = 0$, let $\tilde{\sigma}_{v_i} = 1$ with probability δ and $\tilde{\sigma}_{v_i} = 0$ with probability $1-\delta$, else if $\sigma_{v_i} = 1$, let $\tilde{\sigma}_{v_i} = 1$ with probability 1.

It follows that $\mathbf{E}_{\tilde{\sigma} \leftarrow Q_e}[\tilde{\sigma}_{v_i}] = (1-\delta)\cdot x_{v_i} + \delta$. Moreover, the support of Q_e can be easily seen to upward closure of the support of P_e and, hence, connected. Hence, $y = (1-\delta)\cdot x + \delta$ is a feasible and connected solution for $\mathsf{LP}(\mathcal{J})$. $\mathsf{val}(\mathcal{J}, y) = (1-\delta)\cdot \mathsf{val}(\mathcal{J}, x) + \delta \leq \mathsf{val}(\mathcal{J}, x) + \delta$ as $\sum_{v \in V(\mathcal{J})} w_v = 1$. If x is an optimal solution to $\mathsf{LP}(\mathcal{J})$, then $\mathsf{val}(\mathcal{J}, x) = \mathsf{lp}(\mathcal{J})$.

1.5 Rest of the paper. In Section 2 we present the algorithm ROUND and prove Theorem 1.2 and Corollary 1.2. This part should be easy to read. In Section 3 we give a formal proof of the properties of the dictatorship test gadget described in Section 1.4. In Section B we give the details of composing our dictatorship test gadget with UNIQUE GAMES. In Section C we give the relevant statements and details of our results in the q-ary world. In Section A we show how our result applies to the Concurrent Open Shop Problem.

2 The Rounding Algorithm, its Optimality and LP-Rounding Gap based Inapproximability

In this section we describe our rounding algorithm ROUND and prove that it achieves the integrality gap unconditionally for covering and packing problems. We prove Theorem 1.2 and Corollary 1.2. We keep the discussion here to covering problems. Completely analogous results hold for packing problems.

The algorithm. Let $\mathcal{I}$ be an instance of a k-$\mathsf{strict}^{\uparrow}$-$\mathsf{CSP}$ Π. The algorithm will use a parameter ε. We assume without loss of generality that $1/\varepsilon$ is an integer. We first define a way of perturbing a solution x to $\mathsf{LP}(\mathcal{I})$ (Figure 1) such that the number of distinct values the variables of x take is at-most $1/\varepsilon + 1$.

Definition 2.1. *Given an x such that $0 \leq x_u \leq 1$ for all $u \in V$, and a parameter $\varepsilon > 0$, define x^ε as follows – for each $u \in V$, let k_u be the integer satisfying $k_u\varepsilon < x_u \leq (k_u+1)\varepsilon$, then $x^\varepsilon_u \stackrel{\text{def}}{=} (k_u+1)\varepsilon$ (if $x_u = 0$, we define x^ε_u to be 0 as well).*

In other words, x^ε is obtained from x by rounding up each coordinate to the nearest integral multiple of ε (note that this value will not exceed 1 because $1/\varepsilon$ is an integer). First we observe the following simple fact.

Fact 2.1. *Let x be a feasible solution to $\mathsf{LP}(\mathcal{I})$. Then*

1. *x^ε is feasible for $\mathsf{LP}(\mathcal{I})$.*
2. *$\mathsf{val}(\mathcal{I}, x^\varepsilon) \leq \mathsf{val}(\mathcal{I}, x) + \varepsilon$.*

Proof. We first prove the first statement. It is enough to prove this for x' where x' differs from x on only one coordinate u. Fix an edge $e = (u_1, \ldots, u_k)$ and without loss of generality assume that $u = u_1$. Let λ_σ for $\sigma \in A_e$ be the coefficients in the convex combination of vectors in A_e which yield $(x_{u_1}, \ldots, x_{u_k})$. Let A'_e be the set of σ for which $\sigma_1 = 0$. For each $\sigma \in A'_e$, define $m(\sigma)$ as vector which is same as σ except that $\sigma'_1 = 1$. Clearly, $m(\sigma) \in A_e$ as well. Now consider the vector $\sum_{\sigma \notin A'_e} \lambda_\sigma \sigma + \sum_{\sigma \in A'_e} \lambda_\sigma m(\sigma)$. This is equal to $(1, x_{u_2}, \ldots, x_{u_k})$. Thus, we have shown that the vector x'' which is identical to x except that $x''_u = 1$ is feasible to $\mathsf{LP}(\mathcal{I})$. Now note that x' is a convex combination of x and x''. Hence, the claim follows. We now prove the second statement. Since $x^\varepsilon_u \leq x_u + \varepsilon$, we get that

$$\mathsf{val}(\mathcal{I}, x^\varepsilon) = \sum_u w_u x^\varepsilon_u \leq \sum_u w_u x_u + \varepsilon \sum_u w_u = \mathsf{val}(\mathcal{I}, x) + \varepsilon.$$

The algorithm ROUND is described in Figure 2. This algorithm takes as input an instance $\mathcal{I}$, a feasible solution x to $\mathsf{LP}(\mathcal{I})$ and a parameter $\varepsilon > 0$. We denote $\mathsf{round}(\mathcal{I}, x, \varepsilon)$ as the value of the integral solution returned by $\mathsf{ROUND}(\mathcal{I}, x, \varepsilon)$. First, the algorithm perturbs x to x^ε to make sure that the number of distinct values taken by the variables in x^ε is at-most $m = O(1/\varepsilon)$, which is to be thought of as a (large) constant. Thus, the variables fall into m buckets and now, the rounding algorithm goes over all possible assignments to these constantly many buckets and outputs the assignment with the least cost.

The optimality of the rounding algorithm. We now prove Theorem 1.2. The proof is quite straightforward.

Definition 2.2. *Consider an input $(\mathcal{I}, x, \varepsilon)$ to the algorithm* ROUND. *We define a new instance $\mathcal{I}^\varepsilon$ of Π as follows: the set of variables $V^\varepsilon \stackrel{\text{def}}{=} \{0, \ldots, 1/\varepsilon + 1\}$ and hyper-edge set $E^\varepsilon \stackrel{\text{def}}{=} \{(i_1, \ldots, i_k) \mid (v_1, \ldots, v_k) \in E$ and $x^\varepsilon_{u_j} \stackrel{\text{def}}{=} i_j \cdot \varepsilon$ for all $j \in [k]\}$. We take the weight w_i of $i \in V^\varepsilon$ to be $\sum_{x^\varepsilon_v = i\varepsilon} w_v$ and take constraint $A_{e'}$ for an edge $e' \in E^\varepsilon$ to be the same as A_e for the corresponding edge in $e \in E$. Note that it follows from Fact 2.1-(1) that x^ε is also a feasible solution for $\mathsf{LP}(\mathcal{I}^\varepsilon)$.*

Proof. Consider an input $(\mathcal{I}, x, \varepsilon)$ to the algorithm ROUND. Let $\mathcal{I}^\varepsilon$ and x^ε be as in the definition above. Then, since $\mathsf{ROUND}(\mathcal{I}, x, \varepsilon)$ searches over all feasible assignments to variables in $\mathcal{I}^\varepsilon$, we get that $\mathsf{round}(\mathcal{I}, x, \varepsilon) = \mathsf{opt}(\mathcal{I}^\varepsilon)$. Hence, we get

$$\mathsf{round}(\mathcal{I}, x, \varepsilon) \;=\; \mathsf{opt}(\mathcal{I}^\varepsilon) \;\leq\; \gamma^*(\Pi) \cdot \mathsf{lp}(\mathcal{I}^\varepsilon) \leq \gamma^*(\Pi) \cdot \mathsf{val}(\mathcal{I}^\varepsilon, x^\varepsilon) \quad \text{Fact}$$

LP-Rounding Gap based Inapproximability. Now we see how Corollary 1.2 follows from Corollary 1.1 and the discussion on the rounding algorithm above for k-strict$^\uparrow$-CSPs. Let $\mathcal{J}$ be the constant-sized instance, and x a solution to $\mathsf{LP}(\mathcal{J})$ on which we would like to base the reduction of a k-strict$^\uparrow$-CSP Π and δ be a parameter. We convert $(\mathcal{J}, x)$ to $(\mathcal{J}^\delta, x^\delta)$ as in Definitions 2.1 and 2.2 with δ instead of ε. We know from the description of ROUND that $\mathsf{opt}(\mathcal{J}^\delta) = \mathsf{round}(\mathcal{J}, x, \delta)$. Moreover, from Fact 2.1-(2), we get that $\mathsf{val}(\mathcal{J}^\delta, x^\delta) \leq \mathsf{val}(\mathcal{J}, x) + \delta$. Moreover if x^δ is not connected for $\mathsf{LP}(\mathcal{J}^\delta)$, we can connect it at an additional additive δ loss to get y as in the proof of Corollary 1.1. Now we base our reduction on $(\mathcal{J}^\delta, y)$ rather than $(\mathcal{J}, x)$ to obtain that it is Unique Games-hard to distinguish between instances of Π with value at-most $\mathsf{val}(\mathcal{J}, x) + 2\delta$ form those with value at-least $\mathsf{opt}(\mathcal{J}^\delta) - \delta = \mathsf{round}(\mathcal{J}, x, \delta) - \delta$.

3 Dictatorship Gadget

3.1 Preliminaries We will be interested in functions on $\Omega^r \stackrel{\text{def}}{=} \{0,1\}^r$ along with a product probability measure. For $r = 1$, there are functions $(\chi_0 = 1, \chi_1)$ that form an orthonormal basis for all functions $f : \{0,1\} \to [0,1]$. Tensoring these gives a natural orthonormal basis $\{\chi_S\}_{\{S \subseteq [r]\}}$ where each χ_S is a product of χ_1 on the coordinates $i \in S$. Thus, every function $f : \{0,1\}^r \to [0,1]$ can be written in a multilinear representation:

$$f = \sum_{S \subseteq [r]} \hat{f}(S) \chi_S.$$

Definition 3.1. (Low Degree Influence) *The d-degree influence of the ith coordinate of f is given by:*

$$\mathrm{Inf}_i^{\{<d\}}(f) \stackrel{\text{def}}{=} \sum_{|S|<d} \hat{f}^2(S).$$

Note that the definition of influence implicitly depends on the probability measure on $\Omega^r = \{0,1\}^r$. In our setting, the measure will be clear from the function we measure the influence of.

Input: An instance $\mathcal{I} = (V, E, \{A_e\}_{e\in E}, \{w_v\}_{v\in V})$ of a problem in k-$\mathsf{strict}^{\uparrow}$-$\mathsf{CSP}$, a feasible solution x to $\mathsf{LP}(\mathcal{I})$ and a parameter $\varepsilon > 0$. Let $m := 1/\varepsilon$.

Output: A labeling $\Lambda : V \mapsto \{0, 1\}$.

1. Construct the solution x^{ε}.
2. For every $z \in \{0,1\}^m$, construct a 0-1 solution Λ^z as follows : $\Lambda^z_u \stackrel{\text{def}}{=} z_j$ if $x^{\varepsilon}_u = j\varepsilon$.
3. Output the solution $\Lambda^{z^\star}$ which has the smallest objective value among all feasible solutions in $\{\Lambda^z | z \in \{0,1\}^m\}$.

Figure 2: Algorithm ROUND

Definition 3.2. (τ-pseudo-random function) *A function, $f : \{0,1\}^r \to [0,1]$, is said to be τ-pseudo-random if for $d = \lceil 1/\tau \rceil$ and every i, $\mathrm{Inf}_i^{\{<d\}}(f) \le \tau$.*

Note that we have relaxed the range of f to $[0,1]$ (from $\{0,1\}$). This is necessary for finally composing with the Unique Games instance as we will average the function defined on multiple hyper-cubes. We will need the following well-known lemma in our composition with a Unique Games instance. We refer the reader to [KKMO07] for a proof of the lemma.

Lemma 3.1. *For any $f : \{0,1\}^r \to [0,1]$, there are at most $O(\frac{d}{\tau})$ coordinates such that $\mathrm{Inf}_i^{\{<d\}}(f) \ge \tau$.*

Invariance Principle. The space $\Omega^k = \{0,1\}^k$ along with a probability measure P is called a correlated space. Such a space is said to be connected if every point in the support can be reached from every other point by a path in the support such that adjacent points in the path change in exactly one coordinate of $\{0,1\}^k$.

Given a connected correlated space P on $\{0,1\}^k$, we can naturally extend it to a correlated space, $P^{(r)}$ on $\{0,1\}^r \times \{0,1\}^r \times \cdots \times \{0,1\}^r$ by sampling independently r times from P. Given pseudo-random functions, $f_1, \ldots, f_k : \{0,1\}^r \to [0,1]$, the invariance principle of Mossel stated below gives tight bounds on

$$\mathop{\mathbf{E}}_{(a^{(1)},\ldots,a^{(k)})\leftarrow P^{(r)}}[f_1(a^{(1)}) \cdots f_k(a^{(k)})]$$

in terms of properties of P and $\mathbf{E}[f_i]$ alone (thus independent of r).

Theorem 3.3. (Invariance Principle, Mossel [Mos08]) *For every integer k, and numbers $0 < \delta, \alpha < 1/2$, there exists a $\Gamma = \Gamma(k, \delta, \alpha) > 0$ such that, for every $\varepsilon > 0$, there exists a $\tau > 0$, such that for all connected correlated space P on $\{0,1\}^k$ such that the minimum non-zero probability of any event is α, and τ-pseudo-random functions $f_1, \ldots, f_k : \{0,1\}^r \to [0,1]$ such that $\mathbf{E}[f_i] \ge \delta$,*

$$\mathop{\mathbf{E}}_{(a^{(1)},a^{(2)},\ldots,a^{(k)})\leftarrow P^{(r)}}[\Pi_i f_i(a^{(i)})] + \varepsilon \ge \Gamma(k, \delta, \alpha) > 0.$$

Note that the pseudo-randomness and the non-triviality of measure of the functions are defined with respect to the corresponding marginal measures induced by $P^{(r)}$.

In our setting, the correlated space P will be obtained from a connected LP solution x for a (finite sized) instance $\mathcal{J}$ and hence α is a constant bounded away from zero. k will be the arity of the constraints in the $\mathsf{strict} - \mathsf{CSP}$. Thus, setting $\varepsilon < \Gamma(k, \delta, \alpha)/2$, we have the following corollary that we will use.

Corollary 3.1. *For every integer k, and numbers $0 < \delta, \alpha < 1/2$, there exists a $\tau > 0$ such that, given a connected correlated space P on $\{0,1\}^k$ such that the minimum probability of any event is α, and τ-pseudo-random functions $f_1, \ldots, f_k : \{0,1\}^r \to [0,1]$ such that $\mathbf{E}[f_i] \ge \delta$, $\mathbf{E}_{(a^{(1)},a^{(2)},\ldots,a^{(k)})\leftarrow P^{(r)}}[\Pi_i f_i(a^{(i)})] > 0$.*

3.2 Dictatorship Gadget We quickly recall the dictator gadget $\mathcal{D}^r_{\mathcal{J},x}$. Given a connected LP solution x to $\mathcal{J} = (V, E)$, the gadget is on $V \times \{0,1\}^r$. The weight of a vertex (v, a) is w_v times the x_v-biased measure of a. For every hyper-edge $e = (v_1, \ldots, v_k)$ in $E(\mathcal{J})$, the solution x gives a probability distribution connected P_e whose support is in A_e. For every $(a^{(1)}, a^{(2)}, \ldots, a^{(k)})$ with positive probability in $P_e^{(r)}$, add a constraint $((v_1, a^{(1)}), \ldots, (v_k, a^{(k)}))$ with accepting set A_e to $\mathcal{D}^r_{\mathcal{J},x}$.

Lemma 3.2. (Completeness) *The dictator assignments $\{\Lambda_i\}_{i=1}^r$, where $\Lambda_i(v, (a_1, \ldots, a_r)) = a_i$. satisfy every constraint in $\mathcal{D}^r_{\mathcal{J},x}$and costs exactly $\mathsf{val}(x, \mathcal{J})$.*

Proof. For any edge e, the distribution P_e is supported on the accepting set A_e. Thus, for any constraint $((v_1, a^{(1)}), \ldots, (v_k, a^{(k)}))$ added using edge e,

$(a_j^{(1)}, \ldots, a_j^{(k)}) \in A_e$ for any $j \in [r]$. Thus, the dictator assignments satisfy every constraint. Since we weight the hyper-cube corresponding to v by the x_v-biased measure, the cost of a hyper-cube is exactly $w_v x_v$. Summing the cost shows that the total cost is exactly $\mathsf{val}(x, \mathcal{J})$.

Now, we delve into the proof of the harder part. Let δ be the additional cost we can incur. Fix an assignment to the dictatorship gadget, $\Lambda : V \times \{0,1\}^r \to \{0,1\}$ that satisfies every constraint in $\mathcal{D}^r_{\mathcal{J},x}$. Denote by Λ_v the restriction of Λ to the hyper-cube corresponding to vertex $v \in V$. We will use the "shortform" Λ^1_v for Λ_v and Λ^0_v for the function $1 - \Lambda_v$. We call an assignment Λ τ-pseudo-random if for every $v \in V$ and $b \in \{0,1\}$, the function Λ^b_v is τ-pseudo-random.

Decoding assignment to $\mathcal{J}$. For every $v \in V$, define $S_v \stackrel{\text{def}}{=} \{b \in \{0,1\} | \ \mathbf{P}[\Lambda_v(a) = b] \geq \delta$ (the expectation is over the corresponding biased measure). Set $\lambda(v)$ to be the element in S_v with minimum value. In the binary world, this just means we set $\lambda(v) = 0$ if $0 \in S_V$ and 1 otherwise.

THEOREM 3.4. (COST OF λ) *For λ, Λ, δ as above,* $\mathsf{val}(\lambda, \mathcal{J}) \leq \mathsf{val}(\Lambda, \mathcal{D}^r_{\mathcal{J},x}) + \delta$.

Proof. For every $v \in V$, at most a δ fraction of the corresponding hyper-cube was assigned a value not in S_v. Since $\lambda(v)$ is the minimum value element from S_v, we pay at most a δw_v additional cost in λ for the vertex v. Thus, $\mathsf{opt}(\mathcal{J}) \leq \mathsf{val}(\mathcal{J}, \lambda) \leq \mathsf{val}(\mathcal{D}^r_{\mathcal{J},x}, \Lambda) + \delta \sum_v w_v \leq \mathsf{val}(\mathcal{D}^r_{\mathcal{J},x}, \Lambda) + \delta$.

THEOREM 3.5. (FEASIBILITY OF λ) *For every $\delta > 0$, there exists $\tau > 0$ such that if the assignment Λ is τ-pseudo-random, then λ is feasible for $\mathcal{J}$.*

Proof. Let τ be the minimum value stipulated by Corollary 3.1 over all the edges $e \in E(\mathcal{J})$. Note that for every $s \in S_v$, $\mathbf{E}[\Lambda^s_v] \geq \delta$ by the definition of S_v.

For every constraint hyper-edge $e = (v_1, v_2, \ldots, v_k) \in E$ in $\mathcal{J}$, we will in fact show that $S_{v_1} \times S_{v_2} \times \cdots \times S_{v_k} \subseteq A_e$. Fix an assignment $(s_1, \ldots, s_k) \in S_{v_1} \times S_{v_2} \times \cdots \times S_{v_k}$. Applying Corollary 3.1 to the functions $\{\Lambda^{s_{v_i}}_{v_i}\}_{\{1 \leq i \leq k\}}$ says that there is a constraint in $\mathcal{D}^r_{\mathcal{J},x}$with acceptance set A_e that was satisfied by the assignment $(s_1, \ldots, s_k)$. Thus, $(s_1, \ldots, s_k) \in A_e$.

Acknowledgments. The authors would like to thank Oded Regev for bringing the paper [AKS09] to our notice and also observing that every problem in the class k-strict$^{\uparrow}$-CSP over the alphabet $\{0,1\}$ can be reduced to a HYPERGRAPH VERTEX COVER problem in the approximation preserving sense.

References

[AHK96] Ron Aharoni, Ron Holzman, and Michael Krivelevich. On a theorem of lovász on covers in tau-partite hypergraphs. *Combinatorica*, 16(2):149–174, 1996. 4

[AKS09] P. Austrin, S. Khot, and S. Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, 2009. 3, 4, 9

[BK09a] Nikhil Bansal and Subhash Khot. Inapproximability of hypergraph vertex cover and applications to scheduling problems. In *Manuscript*, 2009. 2, 3, 4

[BK09b] Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *FOCS*, pages 453–462, 2009. 3

[CGH+04] J. Chuzhoy, S. Guha, E. Halperin, S. Khanna, G. Korsatz, and J. Naor. Asymmetic k-center is $\log^* n$ hard to approximate. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004. 5

[DGKR03] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered pcp and the hardness of hypergraph vertex cover. In *STOC*, pages 595–601, 2003. 3

[DS02] Irit Dinur and Shmuel Safra. The importance of being biased. In *STOC*, pages 33–42, 2002. 3

[GK07] N. Garg and A. Kumar. Minimizing average flowtime : Upper and lower bounds. In *Proceedings of the 48th Annual IEEE Symposium of Foundations of Computer Science*, 2007. 5

[GMR08] V. Guruswami, R. Manokaran, and P. Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. 5

[Gol01] Oded Goldreich. Using the fglss-reduction to prove inapproximability results for minimum vertex cover in hypergraphs. *Electronic Colloquium on Computational Complexity (ECCC)*, (102), 2001. 3

[GS10] Venkatesan Guruswami and Rishi Saket. On the inapproximability of vertex cover on k-partite k-uniform hypergraphs. In *Manuscript*, 2010. 2, 4

[Hås97] Johan Håstad. Some optimal inapproximability results. In *STOC*, pages 1–10, 1997. 3

[HK03] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003. 5

[Hol02] Jonas Holmerin. Improved inapproximability results for vertex cover on k -uniform hypergraphs. In *ICALP*, pages 1005–1016, 2002. 3

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity*, page 25, 2002. 11

[Kho10] Subhash Khot. Inapproximability of np-complete problems, discrete fourier analysis, and geometry. In *International Congress of Mathematics*, 2010. 5

[KKMO07] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. 8

[KR08] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. 3, 4, 11

[KV05] S. Khot and N. K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l_1. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2005. 1

[Lov75] L. Lovasz. On minimax theorems of combinatorics. doctoral thesis. *Mathematiki Lapok*, 26:209–264, 1975. 4

[MNRS08] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension and metric labelling. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 2008. 5

[Mos08] E. Mossel. Gaussian bounds for noise correlation of functions and tight analysis of long codes. In *Proceedings of the 49th Annual IEEE Conference on Foundations of Computer Science*, pages 156–165, 2008. 2, 4, 8, 15

[MQS+09] Monaldo Mastrolilli, Maurice Queyranne, Andreas S. Schulz, Ola Svensson, and Nelson A. Uhan. Minimizing the sum of weighted completion times in a concurrent open shop. In *Manuscript*, 2009. 2, 4, 10

[Rag08] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 2008. 1, 2, 3, 4, 6

[RS09a] P. Raghavendra and D. Steurer. Integrality gaps for strong sdp relaxations of unique games. In *Proceedings of the 49th Annual Symposium on Foundations of Computer Science*, 2009. 3

[RS09b] Prasad Raghavendra and David Steurer. How to round any csp. In *FOCS*, pages 586–594, 2009. 2

[SA90] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990. 2

[Tre01] Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *STOC*, pages 453–461, 2001. 3

A Concurrent Open Shop Scheduling

In the concurrent open shop model, we have a set of machines $M = \{1, \ldots, m\}$, each for processing one component and a set of jobs $N = \{1, \ldots n\}$. Each job needs a specific processing time on each of the M machines specified by a matrix $P = \{p_{ij}\}_{\{i \in M, j \in N\}}$. The processing on the machines can be done in any order. The machines can process one job at a time and the objective is to minimize the sum of the completion times of the jobs.

[MQS+09] obtain a 2-approximation for this problem via a linear programming relaxation and a rounding procedure. They also show a simple $2-\varepsilon$ integrality gap for their relaxation. Here, we will show a $2-\varepsilon$ inapproximability assuming the Unique Games Conjecture.

strict-CSP formulation. We first formulate the problem as a strict-CSP. For simplicity, let us restrict our attention to the case where p_{ij}s are all 0 or 1; the integrality gap of [MQS+09] has this property. Then, the maximum completion time of any job is m. We have a vertex for every job that takes an assignment between 1 and m denoting its completion time. For every machine, we have a constraint on all the vertices that restricts the assignment to set of acceptable configuration of completion times.

Remarks. As formulated, the arity of the constraints and the label set depend on the size of the instance. However, this is not an issue as we will apply the reduction to a finite sized instance (the size will depend on ε). In the instance produced by the reduction, each constraint will be on a finite (n) vertices and each vertex will take a finite (m) set of values. The important fact is that the strict-CSP produced by the reduction can be reformulated as a concurrent open shop problem (by setting the irrelevant entries of P to zero). As stated earlier, we believe the framework should be useful in proving inapproximability for many other problems as this.

Integrality Gap The gap instance, $\mathcal{J}$, (constructed by [MQS+09]) is simply the r-uniform complete hyper-graph on n vertices. Each hyper-edge is a machine which takes takes one unit of time to process jobs corresponding to the vertices it contains and zero otherwise. [MQS+09] show that $\mathsf{opt}(\mathcal{J})$ is at least $r(n-r+1)$.

We will now show a solution $x = \{x_v\}_{\{v \in \mathcal{J}\}}$ to the linear programming relaxation for the q-ary case (Figure 3 on page 13). Suppose, each x_v is a point in Δ_m such that the first r coordinates are $1/r$ and the rest are 0. Every machine has non-zero processing time for exactly r machines and hence a distribution over random permutations of those r vertices is a convex combination of accepting configurations. Thus, $x = \{x_v\}$ is a feasible solution with cost $\frac{n(r+1)}{2}$.

Connectedness Note that the support of the distribution is not connected as is (since every two permutations change in at least two places). To get around this, we "perturb" the distribution as follows: instead of distribution over all permutations, we take the uniform distribution over r tuples of $\{1, \ldots, r+1\}^r$ such that no two elements are the same. It is easy to check that this distribution has a connected support. The marginals induce an x_v whose first $r+1$ coordinates are $1/r+1$ and the rest are zero. The new cost, $\mathsf{val}(x, \mathcal{J}) = \frac{n(r+2)}{2}$. Thus, choosing r, say $\sqrt{n}$ and n large enough (depending on ε) gives a $2-\varepsilon$ gap for any $\varepsilon > 0$. This immediately gives a $2-2\varepsilon$ inapproximability for any $\varepsilon > 0$ using our main theorem.

B Composing the Dictatorship Test Gadget with Unique Games

In this section, we give the reduction from UNIQUE GAMES to a problem Π in the class k-strict-CSP. The proof is standard and uses the dictatorship test gadget in Section 3. Here, we highlight the important steps in the proof. We first state the version of UGC on which our results rely.

DEFINITION B.1. (UNIQUE GAMES) *An instance* $\mathcal{U} = (G(U,A), [r], \{\pi_e\}_{e\in A}, \mathrm{wt})$ *of* UNIQUE GAMES *is defined as follows:* $G = (U,A)$ *is a bipartite graph with set of vertices* $U = U_{\text{left}} \cup U_{\text{right}}$ *and a set of edges* A*. For every* $e = (v,w) \in E$ *with* $v \in U_{\text{left}}, w \in U_{\text{right}}$*, there is a bijection* $\pi_e : [r] \mapsto [r]$*, and a weight* $\mathrm{wt}(e) \in \mathbb{R}_{\geq 0}$*. We assume that* $\sum_{e\in E} \mathrm{wt}(e) = 1$*. The goal is to assign one* label *to every vertex of the graph from the set* $[r]$ *which maximizes the weight of the edges satisfied. A labeling* $\Lambda : U \mapsto [r]$ satisfies *an edge* $e = (v,w)$*, if* $\Lambda(w) = \pi_e(\Lambda(v))$*.*

The following notations will be used in the hardness reduction and we state them here.

Notations.

1. For a vertex $v \in U$, $\Gamma(v)$ is the set of edges incident to v.
2. For a vertex $v \in U$, define $p_v \stackrel{\text{def}}{=} \sum_{e\in\Gamma(v)} \mathrm{wt}(e)$. This gives a probability distribution over the vertices in U_{left} (or U_{right}).

We now state the Strong UGC which was shown by Khot and Regev [KR08] to be equivalent to the UGC [Kho02].

CONJECTURE B.2. (STRONG UGC) *For every pair of constants* $\eta, \zeta > 0$*, there exists a sufficiently large constant* $r := r(\eta,\zeta)$*, such that it is NP-hard to distinguish between the following cases for an instance* $\mathcal{U} = (G(U,A), [r], \{\pi_e\}_{e\in A}, \mathrm{wt})$ *of* UNIQUE GAMES*:*

– **YES***: There is a labeling* Λ *and a set* $U_0 \subseteq U_{\text{left}}$ *of vertices,* $\sum_{u\in U_0} p_u \geq (1-\eta)$*, such that* Λ *satisfies all edges incident to* U_0*.*

– **NO***: There is no labeling which satisfies a set of edges of total weight value more than* ζ*.*

Now we describe the reduction from UNIQUE GAMES instance to our problem. The reduction shall use the instance dictatorship test gadget $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{D}^r_{\mathcal{J},x}$ of Π described in Section 3.

Input Instance : The input to the reduction is an instance $\mathcal{U} = (G(U,A), [r], \{\pi_e\}_{e\in A}, \mathrm{wt})$ of UNIQUE GAMES problem as defined in Definition B.1. Recall that G is a bipartite graph with $U = U_{\text{left}} \cup U_{\text{right}}$, and the edge weights wt induce probability distribution p_v over vertices in U_{left}.

Output Instance : The output instance $\mathcal{F}$ of Π is as follows :

1 Vertex Set $V(\mathcal{F}) = U_{\text{left}} \times V(\mathcal{D})$, i.e., we place a copy of $V(\mathcal{D})$ at each vertex of U_{left}. We shall index a vertex by (u,b,y) where $u \in U_{\text{left}}$ and $(b,y) \in V(\mathcal{D})$.

2 Vertex Weights The weight of a vertex (u,b,y) is

$$w_{\mathcal{F}}((u,b,y)) = p_u \cdot w_{\mathcal{D}}((b,y)).$$

3 Hyper-edges For every hyper-edge $e = \big((b^1,y^1),(b^2,y^2),\ldots,(b^k,y^k)\big)$ in $\mathcal{D}$, we add the following edges to $\mathcal{F}$ – for each vertex $u \in U_{\text{right}}$ and all sets of k neighbors, $u^1,\ldots,u^k$ (with repetition) of u, we add the hyper-edge $\Big((u^1,b^1,y^1\circ\pi^u_{(u,u^1)}),\ldots,(u^k,b^k,y^k\circ\pi^u_{(u,u^k)})\Big)$ to $\mathcal{F}$. The constraint for the these hyper-edges is the same as that for e.

Completeness.

THEOREM B.3. *Suppose there is a labeling* λ *for* $\mathcal{U}$ *and a subset* U_0 *of* U_{left}*,* $\sum_{v\in U_0} p_v \geq 1-\eta$*, such that* λ *satisfies all edges incident on* U_0*. Then there is a subset of vertices in* $\mathcal{F}$ *which satisfy all the constraints in* $\mathcal{F}$ *and has weight at-most* $\mathsf{val}(\mathcal{J},x)+\eta$*.*

Proof. Consider the labeling λ. We now show how to pick a set F of vertices from $V(\mathcal{F})$ which satisfies all the hyper-edge constraints. For each $u \in U_0$, define J_u as $\{(u,b,y) \in V(\mathcal{F}) : y_{\lambda_u} = 1\}$. For each $u \in U_{\text{left}} - U_0$, define J'_u as the set $\{(u',b',y') \in V(\mathcal{F}) : u' = u\}$. Now define $F = \cup_{u\in U_0} J_u \bigcup \cup_{u\in U_{\text{left}}-U_0} J'_u$.

We now show that F satisfies all hyper-edge constraints. Fix a hyper-edge $e = \big((b^1,y^1),\ldots,(b^k,y^k)\big)$ in $\mathcal{D}$. Let $u \in U_{\text{right}}$ and $u^1,\ldots,u^k$ be k neighbors of u. Consider a corresponding edge $f = ((u^1,b^1,y^1\circ\pi^u_{(u,u^1)}), \ldots,(u^k,b^k,y^k\circ\pi^u_{(u,u^k)}))$ in $\mathcal{F}$. Lemma 3.2 shows that the set $C_i = \{(b,z) : z_i = 1\}$ satisfies the edge constraint for e for any i. Let us pick $i = \lambda_u$. It will be enough to prove that if (b^l,y^l) satisfies $y^l_i = 1$, then the vertex $w = (u^l,b^l,y^l\circ\pi^u_{(u,u^l)})$ is in F. But this is indeed the case because if $u^l \in U_0$, then $\lambda_u = \pi^u_{(u,u^l)}(\lambda_{u^l})$. Therefore, $y^l\circ\pi^u_{(u,u^l)}$ has coordinate λ_{u^l} equal to 1. Hence, $w \in J^l_u$. If $u^l \in U_{\text{left}} - U_0$, then we add $w \in J'_{u^l}$ trivially. Thus, we have shown that F satsifies the edge constraint for the hyper-edge f.

Let us now compute the weight of F. If $u \in U_0$, then Lemma 3.2 shows that the weight of J_u is at-most $p_u \cdot \mathsf{val}(\mathcal{J},x)$. If $u \notin U_0$, then the weight of J'_u is p_u. Thus, the weight of F is at-most

$$\mathsf{val}(\mathcal{J},x)\cdot\sum_{u\in U_0} p_u + \sum_{u\notin U_0} p_u \leq \mathsf{val}(\mathcal{J},x)+\eta.$$

Soundness.

Theorem B.4. *Suppose there is a subset of vertices F which satisfies all the constraints in $\mathcal{F}$ and $w_{\mathcal{F}}(F) < \mathsf{opt}(\mathcal{J}) - \delta$. Then there is a constant $\zeta(\delta)$ such that there is a labeling for $\mathcal{U}$ for which the set of satisfied edges has weight at-least $\zeta(\delta)$.*

Proof. Consider a set F satisfying the conditions of the theorem. Let $\mathbf{I}_F(\cdot)$ be the indicator function for F. For a vertex $u \in U_{\text{right}}$, let $N(u) \subseteq U_{\text{left}}$ denote the neighbors of u. Recall that every vertex of $\mathcal{F}$ can be written as (w, z), where $w \in U_{\text{left}}$ and $z \in V(\mathcal{D})$. Since the distribution $\{p_w\}_{w \in U_{\text{left}}}$ is same as first picking a vertex $u \in U_{\text{right}}$ with probability p_u and then picking a random neighbor of u (according to edge weights), we get

$$w_{\mathcal{F}}(F) = \mathbf{E}_{u \in U_{\text{right}}} \mathbf{E}_{w \in N(u)} \mathbf{E}_{z \in V(\mathcal{D})} \mathbf{I}_F((w, z \circ \pi^u_{(u,w)})),$$

where z is picked according to vertex weights in $\mathcal{D}$. For a vertex $u \in U_{\text{right}}$, let $G(u)$ denote the quantity

$$\mathop{\mathbf{E}}_{w \in N(u)} \mathop{\mathbf{E}}_{z \in V(\mathcal{D})} \mathop{\mathbf{I}}_{F}((w, z \circ \pi^u_{(u,w)})).$$

We can therefore state the condition of the Theorem as $\mathbf{E}_{u \in U_{\text{right}}} G(u) < \mathsf{opt}(\mathcal{J}) - \delta$. Call a vertex $u \in U_{\text{right}}$ *good* if $G(u) < \mathsf{opt}(\mathcal{J}) - \delta/2$. A simple averaging argument shows that the weight of good vertices is at-least $\delta/2$.

Fix a good vertex u. Let $\mathcal{D}^{(u)}$ be a *copy* of the instance $\mathcal{D}$. We construct a solution $S^{(u)}$ for $\mathcal{D}^{(u)}$ as follows : for each $(b, y) \in V(\mathcal{D}^{(u)})$, we pick a random neighbor u^i of u according to edge weights wt in the instance $\mathcal{U}$. If $(u^i, b, y \circ \pi^u_{(u,u^i)}) \in F$, we add (b, y) to $S^{(u)}$.

Claim B.5. *$S^{(u)}$ satisfies all the constraints in $\mathcal{D}^{(u)}$.*

Proof. Let $e = \big((b^1, y^1), \ldots, (b^k, y^k)\big)$ be a hyper-edge in $\mathcal{D}^{(u)}$. Suppose while constructing the set $S^{(u)}$, we decide to add (b^i, y^i) to this set based on whether $(u^i, b^i, y^i \circ \pi^u_{(u,u^i)}) \in F$. Now observe that the instance $\mathcal{F}$ has the hyper-edge $\Big((u^1, b^1, y^1 \circ \pi^u_{(u,u^1)}), \ldots, (u^k, b^k, y^k \circ \pi^u_{(u,u^k)})\Big)$. Since this hyper-edge is satisfied by F, the claim follows.

Note that $\mathbf{E}[S^{(u)}]$ is exactly $G(u)$, where the expectation is over the choice of random neighbors of u. For each vertex $w \in U_{\text{left}}$ and $b \in V$, define a 0-1 function $f_b^{F,w}$ on $\{0,1\}^r$ as follows –

$$f_b^{F,w}(y) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } (w, b, y) \notin F \\ 0 & \text{otherwise} \end{cases}$$

Note that $f_b^{F,w}$ is the indicator function for complement of F for the set of vertices $\{(w, b, y) : y \in \{0,1\}^r\}$. For the vertex u, we now define the function $f_b^{F,u}(y)$ which is the average of the corresponding functions for the neighbours of u.

$$f_b^{F,u}(y) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{w \in N(u)} f_b^{F,w}\left(y \circ \pi^u_{(u,w)}\right).$$

Observe that $f_b^{F,u}(y) = \mathbf{P}[(u, b, y) \notin S^{(u)}]$, where the probability is over the choice of $S(u)$. The following is identical to the soundness proof in the analysis of the dictatorship test gadget. (Stated here in the contra-positive form.)

Lemma B.1. *There exist values $b \in V, i \in [r]$ and constants d, τ depending on δ and k only such that $\mathrm{Inf}_i^{\{<d\}}(f_b^{F,u}) \geq \tau$.*

Using an application of Jensen's Lemma, it follows that for a good vertex $u \in U_{\text{left}}$ there is an $i \in [r]$ such that for at least $\tau/2$ fraction of its neighbors

$$\mathrm{Inf}^{\{<d\}}_{\pi^u_{(u,w)}(i)} f_b^{F,w} \geq \tau/2.$$

It follows from Lemma 3.1 that the number of such influential variables is at-most $O(2d/\tau)$. Hence, we can satisfy at-least $\delta/2 \cdot \tau/2 \cdot \tau/2d$ fraction of the Unique Games-instance $\mathcal{U}$. This completes the proof this theorem.

Thus, to appeal to the Strong-UGC, we need to pick $\eta \leq \delta$, where δ is as in the statement of Theorem 1.1 and $\zeta < \zeta(\delta)$ and conclude the proof of Theorem 1.1.

C Extension to q-ary Alphabet

In this section, we show how our results extended to the case when variables take values from a larger alphabet $[q] = \{0, \ldots, q-1\}$. We first need some definitions in the q-ary world.

C.1 Preliminaries. Given $x, y \in [q]^k$, we say that $y \succeq x$, if, $y_i \geq x_i$ for all i, $1 \leq i \leq k$. A set $A \subseteq [q]^k$ is said to be *upward monotone* if for every $x \in A$, and every y such that $y \succeq x$, it follows that $y \in A$. For sake of brevity, we assume that the alphabet size, q, is implicit in the definition below.

Definition C.1. (The class k-strict-CSP) *Let k be a positive integer. An instance of type k-strict-CSP is given by*

$$\mathcal{I} = (V, E, \{A_e\}_{\{e \in E\}}, \{w_v\}_{v \in V}) \text{ where :}$$

- *$V = \{v_1, v_2, \ldots, v_n\}$ denotes a set of variables/vertices taking values over $[q]$ along with non-negative weights such that $\sum_{v \in V} w_v = 1$.*
- *E denotes a collection of hyper-edges, each on at most k vertices. For each hyper-edge $e \in E$, there is a constraint A_e.*

The objective is to find an assignment $\Lambda : V \mapsto [q]$ for the vertices in V that minimizes $\sum_{v\in V} w_v \Lambda(v)$ such that for each $e = (v_1, v_2, \ldots, v_l)$, $(\Lambda(v_1), \ldots, \Lambda(v_l)) \in A_e$. A k-strict$^\uparrow$-CSP is one where every A_e is upward monotone while in a k-strict$^\downarrow$-CSP every A_e is downward monotone. We often refer to a k-strict$^\uparrow$-CSP as a covering problem and a k-strict$^\downarrow$-CSP as a packing problem. k-will be assumed to be constant throughout.

LPrelaxation We now give an LPrelaxation for a problem in k-strict-CSP. The following definition allows us to map values in $[q]$ to vectors whose coordinates lie between 0 and 1.

DEFINITION C.2. *Let Δ_q denote the set of vectors $\{ (z_0, \ldots, z_{q-1}) : z_i \geq 0$ for all $i \in [q]$ and $\sum_{i\in[q]} z_i = 1 \}$. There is a natural mapping $\Psi_q : \Delta_q \mapsto [q]$ defined as $\Psi_q((z_0, \ldots, z_{q-1})) = \sum_{i\in[q]} z_i \cdot i$. Let e_i, for $i \in [q]$, be the unit vector in $\mathbb{R}^q$ which has value 1 at coordinate i, and 0 elsewhere. It is easy to check that Δ_q is the convex hull of the vectors $\{e_i : i \in [q]\}$. It follows that a vector $x \in \Delta_q$ can also be thought of as a probability distribution over $[q]$.*

DEFINITION C.3. *Given an integer $i \in [q]$, define $\Phi_q(i)$ as the vector $e_i \in \mathbb{R}^q$. Given a sequence $\sigma \in [q]^k$, for some parameter k, define $\Phi_q(\sigma) = (\Phi_q(\sigma_1), \ldots, \Phi_q(\sigma_k))$. Note that $\Phi_q(\sigma)$ is a vector in $\mathbb{R}^{q\cdot k}$.*

The LP relaxation for an instance $\mathcal{I}$ of a problem $\Pi \in$ k-strict-CSP is described in Figure 3.

$$\mathsf{lp}(\mathcal{I}) \stackrel{\text{def}}{=} \text{minimize} \quad \sum_{v\in V} w_v \Psi_q(x_v) \tag{C.1}$$

$$\text{subject to} \quad \forall_{e=(v_1,v_2,\ldots,v_k)\in E} \quad (x_{v_1}, x_{v_2}, \ldots, x_{v_k}) \in \texttt{ConvexHull}(A_e) \tag{C.2}$$

$$\forall_{v\in V} \quad x_v \in \Delta_q \tag{C.3}$$

Figure 3: LPfor k-strict-CSP

Here, $\texttt{ConvexHull}(A_e)$ is the convex hull of the set $\{\Phi_q(\sigma) : \sigma \text{ is a satisfying assignment for } A_e\}$. It is easy to check that this indeed is a linear program. Given a solution x to $\mathsf{LP}(\mathcal{I})$, let $\mathsf{val}(\mathcal{I}, x)$ denote the objective function value for x. Let $\mathsf{opt}(\mathcal{I})$ denote the value of the optimal integral solution for $\mathcal{I}$.

C.2 Results The following are equivalents of Theorem 1.1, Corollary 1.1, Theorem 1.2 and Corollary 1.2 respectively. We skip the proofs and just highlight the important points in this section.

THEOREM C.4. (LP-INTEGRALITY GAP BASED INAPPROXIMABILITY) *Let Π be a k-strict-CSP for $k = O(1)$, and $\mathcal{J}$ be a constant-sized instance of Π. Let x be a feasible-connected solution for $\mathsf{LP}(\mathcal{J})$. Then for every $\delta > 0$, it is UNIQUE GAMES-hard to distinguish between the following instances $\mathcal{I}$ of Π*

- **YES.** $\mathsf{opt}(\mathcal{I}) \leq \mathsf{val}(\mathcal{J}, x) + \delta$
- **NO.** $\mathsf{opt}(\mathcal{I}) \geq \mathsf{opt}(\mathcal{J}) - \delta$.

Hence, if it is the case that x is also an optimal solution to $\mathsf{LP}(\mathcal{J})$, then, assuming the UGC, the LP captures the approximability of the problem Π. For k-strict$^\uparrow$-CSP and k-strict$^\downarrow$-CSP we can easily convert any optimal LP solution to a connected one with at-most a δ loss in the LP value, for arbitrarily small constant δ. Hence, we get the following important corollary which proves that the LP of Figure 3 captures precisely the approximability of all covering and packing problems with $k = O(1)$.

COROLLARY C.1. (OPTIMAL INAPPROXIMABILITY FOR COVERING A *Let Π be a k-strict$^\uparrow$-CSP or a k-strict$^\downarrow$-CSP for $k = O(1)$, and $\mathcal{J}$ be a constant-sized instance of Π. Then for every $\delta > 0$, it is UNIQUE GAMES-hard to distinguish between the following instances $\mathcal{I}$ of Π*

- **YES.** $\mathsf{opt}(\mathcal{I}) \leq \mathsf{lp}(\mathcal{J}) + \delta$
- **NO.** $\mathsf{opt}(\mathcal{I}) \geq \mathsf{opt}(\mathcal{J}) - \delta$.

Note that the form makes sense for both covering and packing problems if one notices that in the case of packing problems both the LP value and the optimal value are negative. We will, henceforth, keep the discussion just to covering problems. All results can be directly translated in the packing world and we omit the details.

Rounding for covering-packing problems. For a k-strict$^\uparrow$-CSP Π we give a rounding algorithm called ROUND_q (see Figure 4) for the LPof Figure 3. For an instance $\mathcal{I}$ of Π, a solution x to $\mathsf{LP}(\mathcal{I})$, and a parameter $\varepsilon > 0$, which should be ignored for this discussion, let $\mathsf{round}_q(\mathcal{I}, x, \varepsilon)$ denote the value of the integral solution that ROUND_q produces for $\mathcal{I}$ starting from the LP solution x. We show that ROUND_q (unconditionally) achieves an approximation ratio equal to the integrality gap, up to an arbitrarily small additive constant, of the LPrelaxation.

THEOREM C.5. (ROUNDING ACHIEVES INTEGRALITY GAP) *Let $\gamma^*(\Pi)$ be the worst-case approximation ratio (integrality gap) achieved by the LP relaxation for the problem Π, i.e., $\gamma^*(\Pi) \stackrel{\text{def}}{=} \sup_{\mathcal{J}}(\mathsf{opt}(\mathcal{J})/\mathsf{lp}(\mathcal{J}))$, where the supremum is taken over all instances $\mathcal{J}$ of Π. Then, for any given instance $\mathcal{I}$, optimal LPsolution $x^\star$ and $\varepsilon > 0$, $\mathsf{round}_q(\mathcal{I}, x^\star, \varepsilon) \leq \gamma^*(\Pi) \cdot (\mathsf{opt}(\mathcal{I}) + O_q(\varepsilon))$.*

For covering and packing problems, we show how to start with an instance $\mathcal{J}$ of Π and a solution x to $\mathsf{LP}(\mathcal{J})$, , and give a UNIQUE GAMES-based reduction for Π whose soundness and completeness are roughly $\mathsf{val}(\mathcal{J}, x)$ and $\mathsf{round}_q(\mathcal{J}, x, \varepsilon)$ respectively.

COROLLARY C.2. (LP-ROUNDING GAP BASED INAPPROXIMABILITY) *Let Π be a k-strict$^{\uparrow}$-CSP for $k = O(1)$, and $\mathcal{J}$ be a constant-sized instance of Π, and x a solution to $\mathsf{LP}(\mathcal{J})$, Then for every $\delta > 0$, it is* UNIQUE GAMES*-hard to distinguish instances $\mathcal{I}$ of Π with optimal less than $\mathsf{val}(\mathcal{J}, x) + O_q(\delta)$ from those with optimal more than $\mathsf{round}_q(\mathcal{J}, x, \delta) - \Omega_q(\delta)$.*

The reduction in this theorem is slightly different from that in the corollary. This theorem is useful compared to the in the case when it is easier to come up with a LP-rounding gap rather than an integrality gap.

The Rounding Algorithm. We now describe the rounding algorithm for a k-strict$^{\uparrow}$-CSP. The algorithm uses a perturbation parameter ε. We first argue that we can perturb a feasible solution to $\mathsf{LP}(\mathcal{I})$ such that the number of distinct (vector) values taken by the variables are small. This perturbation will not affect the objective value significantly. We shall assume without loss of generality that $1/\varepsilon$ is an integer.

DEFINITION C.6. *For a parameter $\varepsilon > 0$, define $\Delta_q^{\varepsilon,i}, 0 \le i < q$, as the set of points $z \in \Delta_q$ satisfying the following conditions – (1) $z_0, \ldots, z_{i-1}$ are multiples of ε, and (2) $z_{i+1} = \cdots = z_{q-1} = 0$. Observe that z_i must equal $1 - \sum_{j=0}^{i-1} z_j$. Let Δ_q^{ε} denote $\bigcup_i \Delta_q^{\varepsilon,i}$.*

It is easy to check that $|\Delta_q^{\varepsilon}|$ is at most $1/(\varepsilon+1)^q$. We now show how a vector $x \in \Delta_q$ can be perturbed to a vector in Δ_q^{ε}.

DEFINITION C.7. *Let $a \in [0, 1]$ be a real number. Define a^{ε} as the smallest multiple of ε greater than or equal to a. Consider $x \in \Delta_q$. Let i be the largest integer such that $x_0^{\varepsilon} + \cdots + x_{i-1}^{\varepsilon} \le 1$. Then, define x^{ε} to be the vector $(x_0^{\varepsilon}, \ldots, x_{i-1}^{\varepsilon}, 1 - \sum_{j=0}^{i-1} x_j^{\varepsilon}, 0, \ldots, 0) \in \Delta_q^{\varepsilon,i}$.*

The rounding algorithm is described in Figure 4. Let $\mathsf{round}_q(\mathcal{I}, x, \varepsilon)$ denote the objective value of the solution returned by $\mathsf{ROUND}_q(\mathcal{I}, x, \varepsilon)$.
Since m_q is $O(1/\varepsilon^q)$, the running time of ROUND_q is $O(\mathrm{poly}(n^k, 1/\varepsilon^q))$. We state the following fact without proof.

FACT C.1. *Let x be a feasible solution to $\mathsf{LP}(\mathcal{I})$. Then*

1. *x^{ε} is feasible for $\mathsf{LP}(\mathcal{I})$.*

2. *$\mathsf{val}(\mathcal{I}, x^{\varepsilon}) \le \mathsf{val}(\mathcal{I}, x) + \varepsilon \cdot q^2$.*

C.3 Dictatorship Test Gadget

In this section we outline the proof of Theorem C.4. For this discussion, we restrict ourselves to the dictatorship test gadget and the case that $w_v \ge 0$ and $\sum_v w_v = 1$. First, we need some preliminaries.

DEFINITION C.8. *Let x be a feasible solution to $\mathsf{LP}(\mathcal{J})$. For a hyper-edge e, let $x|_e$ be the vector obtained by restricting x to only those vertices which belong to e. P_e shall denote the probability distribution over vectors in Δ_q^k corresponding to elements of A_e that arises from the fact that $x|_e \in \mathtt{ConvexHull}(A_e)$. Given a vector $p \in \Delta_q$, let μ_p be the corresponding distribution over $[q]$ (as described in Definition C.2).*

Consider an instance $\mathcal{J} = (V, E, \{A_e\}_{e \in E}, \{w_v\}_{v \in V})$ of a k-strict-CSP problem Π, and a feasible-connected solution x to $\mathsf{LP}(\mathcal{J})$. As is key in basing hardness results on UGC, we will first construct, for an integer $r \ge 1$, a bigger instance (*dictatorship test gadget*) $\mathcal{D}_{\mathcal{J},x}^r$ of Π and then compose it in a standard way with a UNIQUE GAMES instance. For this discussion, we restrict ourselves to the dictatorship test gadget. The instance $\mathcal{D}_{\mathcal{J},x}^r$ will have the following components:

– **Vertex Set.** The vertex set of $\mathcal{D}_{\mathcal{J},x}^r$ is $V \times [q]^r$.

– **Vertex Weights.** The weight of a vertex $(v, (a_1, \ldots, a_r))$ will be $w_v \cdot \mu_{x_v}((a_1, \ldots, a_k))$.

– **Edges and Constraints.** Recall that for every hyper-edge $e = (v_1, \ldots, v_k)$ in $E(\mathcal{J})$, from the solution x, we can read off a probability distribution P_e on $[q]^k$. Moreover the constraint in the LP requires that this is distribution is supported on A_e, and the hypothesis requires that this support is connected. For every $e = (v_1, \ldots, v_k) \in E(\mathcal{J})$ and every $a^{(1)}, \ldots, a^{(k)} \in [q]^r$, there will be an hyper-edge in $\mathcal{D}_{\mathcal{J},x}^r$ between the vertices $((v_1, a^{(1)}), \ldots, (v_k, a^{(k)}))$ with the constraint A_e.

  We will also associate a *weight* with this edge which is $\prod_{i=1}^r P_e(a_i^{(1)}, \ldots, a_i^{(k)})$. We will not keep any hyper-edges with 0 weight. These weights will be useful for the analysis and will not show up in the actual instance produced by the reduction.

Influence in the q-ary world We will be interested in functions on $\Omega^r = [q]^r$ along with a product probability measure. As in the binary case, for $r = 1$, there are functions $(\chi_0 = 1, \chi_1, \chi_{q-1})$ that form an orthonormal basis for all functions $f : [q] \to [0, 1]$ which can be tensored to obtain a orthonormal basis $\{\chi_S\}_{\{S \in [q]^r\}}$. Thus, every function $f : [q]^r \to [0, 1]$ can be written in a multilinear representation:

$$f = \sum_{S \subseteq [r]} \hat{f}(S) \chi_S.$$

Input: An instance $\mathcal{I} = (V, E, \{A_e\}_{e\in E}, \{w_v\}_{v\in V})$ of a problem in k-$\mathsf{strict}^{\uparrow}$-$\mathsf{CSP}$, a feasible solution x to $\mathsf{LP}(\mathcal{I})$ and a parameter $\varepsilon > 0$. Let m_q denote $|\Delta_q^\varepsilon|$.

Output: A labeling $\Lambda : V \mapsto [q]$.

1. Construct the solution x^ε.
2. Let I denote the set Δ_q^ε arranged in some order.
3. For every $z \in [q]^{m_q}$, construct an integral solution Λ^z as follows : $\Lambda_u^z \stackrel{\text{def}}{=} z_j$ if $x_u^\varepsilon = I_j$.
4. Output the solution Λ^{z^*} which has the smallest objective value among all feasible solutions in $\{\Lambda^z | z \in [q]^{m_q}\}$.

Figure 4: Algorithm ROUND_q

The definition of influence and pseudo-randomness are exactly as in the binary case.

Definition C.9. (Low Degree Influence) *The d-degree influence of the ith coordinate of $f : [q]^r \to [0,1]$ is given by:*

$$\mathrm{Inf}_i^{\{<d\}}(f) \stackrel{\text{def}}{=} \sum_{|S|<d} \hat{f}^2(S).$$

Note that the definition of influence implicitly depends on the probability measure on $\Omega^r = [q]^r$. In our setting, the measure will be clear from the function we measure the influence of.

Definition C.10. (τ-pseudo-random function) *A function, $f : [q]^r \to [0,1]$, is said to be τ-pseudo-random if for $d = \lceil 1/\tau \rceil$ and every i, $\mathrm{Inf}_i^{\{<d\}}(f) \le \tau$.*

Invariance Principle The space $\Omega^k = [q]^k$ along with a probability measure P is called a correlated space. Two points $(x_1, \ldots, x_k), (y_1, \ldots, y_k) \in [q]^k$ are said to be connected by an edge if they differ in at-most one position. A subset $S \subseteq [q]^k$ is said to be connected if the subgraph induced by the vertices of S along with the edges is connected. For an instance $\mathcal{I}$ of a k-strict-CSP, given a solution x to $\mathsf{LP}(\mathcal{I})$, x-is said to be connected if for every edge $e = (v_1, \ldots, v_l)$, $(x_{v_1}, \ldots, x_{v_l})$ can be written as a convex combination of points in A_e such that the support of this convex combination is connected. As in the binary world, we have the following powerful theorem of Mossel.

Theorem C.11. (Invariance Principle, Mossel [Mos08]) *For every integer k, q, and numbers $0 < \delta, \alpha < 1/2$, there exists a $\Gamma = \Gamma(k, q, \delta, \alpha) > 0$ such that, for every $\varepsilon > 0$, there exists a $\tau > 0$, such that for all connected correlated space P on $[q]^k$ such that the minimum probability of any event is α, and τ-pseudo-random functions $f_1, \ldots, f_k$ such that $\mathbf{E}[f_i] \ge \delta$.*

$$\mathop{\mathbf{E}}_{(a^{(1)}, a^{(2)}, \ldots, a^{(k)}) \leftarrow P^{(r)}} [\Pi_i f_i(a^{(i)})] + \varepsilon \ge \Gamma(k, q, \delta, \alpha) > 0.$$

Note that the pseudo-randomness and non-triviality of measure of the functions are defined with respect to the corresponding marginal measures induced by $P^{(r)}$.

As before, we set $\varepsilon < \Gamma(k, q, \delta, \alpha)/2$, obtaining the following corollary.

Corollary C.3. *For every integer k, q, and numbers $0 < \delta, \alpha < 1/2$, there exists a $\tau > 0$ such that, given a connected correlated space P on $[q]^k$ such that the minimum probability of any event is α, and τ-pseudo-random functions $f_1, \ldots, f_k : [q]^r \to [0,1]$ such that $\mathbf{E}[f_i] \ge \delta$,*

$$\mathop{\mathbf{E}}_{(a^{(1)}, a^{(2)}, \ldots, a^{(k)}) \leftarrow P^{(r)}} [\Pi_i f_i(a^{(i)})] > 0.$$

Definition C.12. *An assignment $f : V \times [q]^r \mapsto [q]$ is said to be a dictator if there exists an $j \in [r]$ such that $f(v, z) = z_j$.*

Given $v \in V$, let f_v denote the restriction of f to $(v, [q]^r)$. Given $p \in \Delta_q$, let $z \leftarrow \mu_p^r$ denote a string in $[q]^r$ drawn from the product distribution μ_p^r. We can think of an assignment $f_v : [q]^r \mapsto [q]$ also as a a function from $[q]^r$ to Δ_q (where the value $i \in [q]$ gets associated with $e_i \in \Delta_q$). Further, let $f_v^{(1)}, f_v^{(2)}, \ldots, f_v^{(q)} : [q]^r \to [0,1]$ denote the q-components of q. Thus, $\sum_i f_v^{(i)}(x) = 1$ for every $x \in [q]^r$.

Definition C.13. *Given $\tau, d \ge 0$, an assignment $f : V \times [q]^r \mapsto \Delta_q$ is said to be (τ, d)*-pseudo-random *if for every $v \in V$, every $i \in [q]$ and every $j \in [r]$, $\mathrm{Inf}_j^{\le d}(f_v^{(i)}) \le \tau$.*

Now we state the completeness and soundness properties of the dictatorship function. The proof of the theorems is very similar to the proofs in the binary case and hence omitted. Let $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{D}^r_{\mathcal{J},x}$.

Lemma C.1. (Completeness) *Let $f : V \times [q]^r \mapsto [q]$ be a dictator. Then f satisfies all the constraints of $\mathcal{D}$ and $\sum_{v \in V(\mathcal{D})} w_{\mathcal{D}}(v) \cdot f(v) \leq \mathsf{val}(\mathcal{J}, x)$.*

Theorem C.14. (Soundness) *For every small enough $\delta > 0$, there exists a d, τ such that if $f : V \times [q]^r \mapsto [q]$ satisfies all the constraints of $\mathcal{D}$ and is (τ, d)-pseudo-random, then*

$$\sum_{v \in V(\mathcal{D})} w_{\mathcal{D}}(v) \cdot \Psi_q f((v, a)) \geq \mathsf{opt}_q(\mathcal{J}) - \Omega_q(\delta).$$