

Max-3-Lin over Non-Abelian Groups with Universal Factor Graphs

Amey Bhangale ✉

University of California, Riverside, CA, USA

Aleksa Stanković ✉ 

Department of Mathematics, KTH Royal Institute of Technology, Sweden

Abstract

Factor graph of an instance of a constraint satisfaction problem with n variables and m constraints is the bipartite graph between $[m]$ and $[n]$ describing which variable appears in which constraints. Thus, an instance of a CSP is completely defined by its factor graph and the list of predicates. We show inapproximability of Max-3-LIN over non-abelian groups (both in the perfect completeness case and in the imperfect completeness case), with the same inapproximability factor as in the general case, even when the factor graph is fixed.

Along the way, we also show that these optimal hardness results hold even when we restrict the linear equations in the Max-3-LIN instances to the form $x \cdot y \cdot z = g$, where x, y, z are the variables and g is a group element. We use representation theory and Fourier analysis over non-abelian groups to analyze the reductions.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Universal factor graphs, linear equations, non-abelian groups, hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.21

Related Version *Full Version:* <https://arxiv.org/abs/2111.09256>

Funding *Aleksa Stanković:* Research supported by the Approximability and Proof Complexity project funded by the Knut and Alice Wallenberg Foundation.

1 Introduction

Constraint Satisfaction Problems (CSPs), and especially k -LIN, are the most fundamental optimization problems. An instance of a CSP consists of a set of n variables and a set of m local constraints where each constraint involves a small number of variables. The goal is to decide if there exists an assignment to the variables that satisfies all the constraints. 3-LIN is a special type of CSP where each constraint is a *linear equation* in the variables involved in the constraint. More specifically, a 3-LIN instance over a (non-abelian or abelian) group G has the constraints of the form $a_1 \cdot x_1 \cdot a_2 \cdot x_2 \cdot a_3 \cdot x_3 = b$, where a_1, a_2, a_3, b are the group elements and x_1, x_2, x_3 are the variables. One can also sometimes allow inverses of the variables in the equations.

For most CSPs, the decision version is NP-complete. Therefore, from the algorithmic point of view, one can relax the goal to finding an assignment that satisfies as many constraints as possible. An α -approximation algorithm for a Max-CSP is an algorithm that always returns a solution that satisfies at least $\alpha \cdot \text{OPT}$ many constraints, where OPT is the maximum number of constraints that can be satisfied by an assignment.

The famous PCP theorem [2, 3, 12] shows that certain Max-CSPs are hard to approximate within a factor $c < 1$. A seminal result of Håstad [14] gives optimal inapproximability results for many CSPs including Max- k -SAT, Max- k -LIN over abelian groups, Set Splitting, etc. Once we understand the optimal *worst-case* complexity of a Max-CSP, it is interesting to



© Amey Bhangale and Aleksa Stanković;

licensed under Creative Commons License CC-BY 4.0

13th Innovations in Theoretical Computer Science Conference (ITCS 2022).

Editor: Mark Braverman; Article No. 21; pp. 21:1–21:21

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

understand how the complexity of the problem changes under certain restrictions on the instances. One such example of restrictions is the study of promise CSPs [5, 7] in which it is guaranteed that a richer solution exists (e.g., a given graph has a proper 3-coloring) and the goal is to find or even approximate a weaker solution (e.g., find a coloring with 10 colors that maximizes the number of non-monochromatic edges). Another example is the study of complexity of Max-CSP when the instance is generated using a certain random process [11, 16].

In this paper we study the complexity of Max-3-LIN when the underlying constraints *vs.* variables graph is fixed. A factor graph of an instance is a bipartite graph between $\{C_1, C_2, \dots, C_m\}$ and $\{x_1, x_2, \dots, x_n\}$ where we connect C_i to x_j if the constraint C_i depends on the variable x_j . An instance can be completely described by its factor graph and by specifying which predicate to use for each constraint. Feige and Jozeph [13] were interested in understanding the effect of the factor graph on the complexity of approximating CSPs. Namely, for a given CSP, is there a factor graph G_n for each input length n such that the CSP remains hard even after restricting its factor graph to G_n ? They answered this positively by showing that there exists a family of factor graphs for Max-3-SAT such that it is NP-hard to approximate within a factor of $s \frac{77}{80} + \varepsilon$, for every $\varepsilon > 0$. They defined such a family of graphs as the *universal factor graphs*.

► **Definition 1.** A family of (c, s) -universal factor graphs for a Max-CSP is a family of factor graphs, one for each input length, such that given a Max-CSP instance restricted to the factor graphs in the family, it is NP-hard to find a solution with value at least s even if it is guaranteed that there is a solution with value at least c .

In a follow-up work by Jozeph [15], it was shown that there are universal factor graphs for every NP-hard Boolean CSP and for every APX-hard Boolean Max-CSP. However, the inapproximability factors in these results are weaker than what was known in the standard setting.

The result of Feige and Jozeph [13] was vastly improved recently by Austrin, Brown-Cohen and Håstad [4]. They gave optimal inapproximability results for many well-known CSPs including Max-k-SAT, Max-TSA¹, “ $(2 + \varepsilon)$ -SAT” and any predicate supporting a pairwise independent subgroup. Namely, they show the existence of $(1, \frac{7}{8} + \varepsilon)$ -universal factor graphs for a Max-3-SAT, $(1 - \varepsilon, \frac{1}{2} + \varepsilon)$ -universal factor graphs for a Max-3-LIN over \mathbb{Z}_2 , etc., for every $\varepsilon > 0$. The optimal universal factor graph inapproximability of Max-3-LIN over any finite *abelian* groups was also shown in [4].

In this paper, we investigate the existence of universal factor graphs for Max-3-LIN over finite *non-abelian* groups. Engebretsen, Holmerin and Russell [10] showed optimal inapproximability of Max-3-LIN over non-abelian groups with *imperfect* completeness.² Recently, Bhangale and Khot [6] showed optimal inapproximability of Max-3-LIN over non-abelian groups with *perfect* completeness. Our main theorems extend both these results by showing the existence of universal factor graphs with the same hardness factor.

In the imperfect completeness case, we show that it is NP-hard to do better than the random assignment even if the factor graph is fixed, thereby extending the result of [10] to the universal factor graph setting.

► **Theorem 2.** For every $\varepsilon > 0$ and any finite non-abelian group G , there are $(1 - \varepsilon, \frac{1}{|G|} + \varepsilon)$ -universal factor graphs for Max-3-LIN over G .

¹ Each constraint is a Tri-Sum-And constraint, i.e., of the form $x_1 + x_2 + x_3 + x_4 \cdot x_5 = b \pmod{2}$.

² The instances in [10] involve inverses in the equations, e.g., they are of the form $a_1 \cdot x_1 \cdot a_2 \cdot x_2^{-1} \cdot a_3 \cdot x_3 = b$.

For a non-abelian group G , we denote by $[G, G]$ a commutator subgroup of G , i.e., the subgroup generated by the elements $\{g^{-1}h^{-1}gh \mid g, h \in G\}$. The factor $\frac{1}{|[G, G]|}$ comes naturally in the results on approximating Max-3-LIN over G in the perfect completeness situation. This is because of the fact that $G/[G, G]$ is an abelian group and this can be used in getting the $\frac{1}{|[G, G]|}$ -approximation algorithm for Max-3-LIN. For a concrete example, consider the group S_3 , the group of all permutations of a three-element set. The commutator subgroup of S_3 is $\{(), (1, 2, 3), (1, 3, 2)\}$ which is isomorphic to \mathbb{Z}_3 . In this case, $S_3/\mathbb{Z}_3 \cong \mathbb{Z}_2$ which is an abelian group. More generally, a given Max-3-LIN instance ϕ over G can be thought of as a Max-3-LIN over $G/[G, G]$ by replacing the group constant by its coset in $G/[G, G]$. If ϕ is satisfiable over G , then ϕ' is satisfiable over $G/[G, G]$. The satisfying assignment for ϕ' can be found in polynomial time as ϕ' is a collection of linear equations over an abelian group. To get the final assignment for ϕ , we can assign a variable x a random element from the coset assigned by the satisfying assignment of ϕ' . It is not very difficult to see that this random assignment satisfies each equation of ϕ with probability $\frac{1}{|[G, G]|}$.

In the perfect completeness situation too, we get the optimal universal factor graph hardness for Max-3-LIN, matching the inapproximability threshold of [6].

► **Theorem 3.** *For every $\varepsilon > 0$ and any finite non-abelian group G , there are $(1, \frac{1}{|[G, G]|} + \varepsilon)$ -universal factor graphs for Max-3-LIN over G .*

The actual theorem statements are stronger than what are stated. Along with the universal factor graph hardness, the instances have the following additional structure.

1. In the imperfect completeness case, our hardness result from Theorem 2 holds for constraints of the form $x_1 \cdot x_2 \cdot x_3 = g$ for some $g \in G$. In [10], the constraints involve inverses of the variables as well as group constants on the left hand side, for example, the constraints can be of the form $a_1 \cdot x_1 \cdot a_2 \cdot x_2^{-1} \cdot a_3 \cdot x_3 = b$.
2. Similar to the above, our hardness result from Theorem 3 holds even if we restrict the constraints in the Max-3-LIN instance to the form $x_1 \cdot x_2 \cdot x_3 = g$ for some $g \in G$. In comparison, in [6], the definition of a linear equation involves using constants on the left-hand side of the equations.

To sum it up, our results show that the exact “literal patterns” as well as the factor graphs are not the main reasons for the optimal NP-hardness of the aforementioned results [10, 6] on Max-3-LIN over finite non-abelian groups.

1.1 Techniques

In this section, we highlight the main differences between the previous works [10, 4, 6] and this work.

A typical way of getting optimal inapproximability result is to start with a gap $(1, 1 - \varepsilon)$ NP-hard instance of a Max-CSP and apply parallel repetition on it to create a 2-CSP (a.k.a. Label Cover, see Definition 29) with arbitrarily large gap of $(1, \delta)$, for any constant $\delta > 0$. Each vertex on one side of the Label Cover corresponds to a subset of constraints of the initial Max-CSP instance. In order to reduce a Label Cover to a given Max-CSP over smaller alphabet, one key component in the reduction is to use a long-code encoding of the labels (i.e., the assignments to the variables in the constraints associated with the vertex) in the Label Cover instance. A long code of a label $i \in [n]$ is given by the truth-table of a function $f : [q]^n \rightarrow [q]$ defined as $f(x) = x_i$. One of the main reasons for using long code is that its high redundancy allows one to implicitly check if the assignment satisfies the given set of constraints associated with a vertex using the operation called *folding*. Thus, the only thing to check is if the given encoding is indeed (close to) a long code encoding of a label and hence it is successful in getting many tight inapproximability results.

One issue with the foldings that appeared before the work of [4] is that the folding structure changes if we change the literal structure of the underlying constraints. Therefore, even if we start with a universal factor graph hard instance of a gap $(1, 1 - \varepsilon)$ Max-CSP, different literal patterns give different constraints *vs.* variables graphs for the final Max-CSP instances. Therefore, the reduction template is not enough to get the same factor graph.

To overcome this difficulty the *functional folding* was introduced in [4]. This folding is 'weaker' than the previously used foldings in terms of decoding to a label that satisfies the underlying predicate in the Label Cover instance. However, they show that this type of folding can be used to get the tight inapproximability results. The key lemma that was proved in [4] is that non-zero Fourier coefficients of such a folded function correspond to sets of assignments with an odd number of them satisfying the constraints.

We cannot directly use the functional folding defined in [4] for two main reasons. Firstly, the way functional folding was defined, the underlying group is always an abelian group. Secondly, the aforementioned main lemma from [4] talks about the non-zero Fourier coefficient of the folded function when viewed as a function over an abelian group. Since our soundness analyses use Fourier analysis over non-abelian groups (similar to [10, 6]), we cannot directly use their folding.

In this work, we define functional folding for functions $f : G^n \rightarrow G$, where G is a finite non-abelian group (See Definition 33). One of the main contributions of this work is to prove that non-zero Fourier coefficients (Fourier matrices to be precise) of such functionally folded functions also have properties that are sufficient to complete the soundness analysis.

For the proof of Theorem 2, it is enough to show that for functionally folded functions, any non-zero Fourier coefficient of the function has at least one assignment that satisfies all the constraints of the Label Cover vertex. We prove this in Lemma 37. This conclusion is analogous to the one in [4]. In order to make sure that we do not use inverses in the constraints as mentioned in the previous section, we follow the proof strategy of [6] in the imperfect completeness case. Compared to [6], in the imperfect completeness case, one needs to control extra terms related to dimension 1 representations of the group which necessitates a different approach for the soundness analysis. In particular, we use the independent noise to take care of all high-dimensional terms in the Fourier expansion of the test acceptance probability.

For proving Theorem 3, we need a stronger conclusion on the non-zero Fourier coefficients of functionally folded functions. This is because the decoding strategy in [6] is highly non-standard; The reduction can only decode from a very *specific subset* of the list corresponding to non-zero Fourier coefficients. Therefore, we need to show that for this type of non-zero Fourier coefficients, there exists an assignment from that subset satisfying the constraints of the Label Cover vertex. This is done in Lemma 38.

We also observe that for the soundness proof to work, one does not need to fold all the functions. We use this observation to conclude that the reduced instance of Max-3-LIN has constraints of the form $x_1 \cdot x_2 \cdot x_3 = g$ for some $g \in G$. Since the factor graph is fixed, these instances are completely specified by specifying only one group element per constraint.

1.2 Organization

The paper is organized as follows. We start with preliminaries in Section 2. In Section 2.1, we give an overview of representation theory and Fourier analysis over non-abelian groups. In Section 2.2, we formally define the problem that we study and universal factor graphs. In Section 2.3, we define functional folding. In Section 3 we give an brief overview of the result presented in this paper intended for the reader who might not be interested in all the details.

In the same section, we also discuss two key lemmas related to functional folding that will be used in the main soundness analysis. In Section 4, we give a reduction that shows the existence of $(1 - \varepsilon, \frac{1}{|G|} + \varepsilon)$ -universal factor graphs for Max-3-LIN over a non-abelian group G with imperfect completeness. Finally, in Section 5, we give a reduction that shows the existence of $(1, \frac{1}{|G|} + \varepsilon)$ -universal factor graphs for Max-3-LIN over a non-abelian group G with perfect completeness.

2 Preliminaries

2.1 Representation Theory and Fourier analysis on non-Abelian groups

In this section we give a brief description of concepts from representation theory used in this work. We will not prove any claims in this section; instead we refer the reader interested in a more detailed treatise to [19, 20].

Let us start by introducing a definition of representation.

► **Definition 4.** *A representation (V_ρ, ρ) of a group G is a group homomorphism $\rho: G \rightarrow \text{GL}(V_\rho)$, where V_ρ is a complex vector space.*

For the sake of brevity throughout this article we will use only symbol ρ to denote a representation, and we will assume that the vector space V_ρ can be deduced from the context. We work with finite groups, and hence assume that $V_\rho = \mathbb{C}^n$, where $n \in \mathbb{N}$, and that $\text{GL}(V_\rho)$ is a space of invertible matrices. Furthermore, we always work with unitary representations.

Study of representations can be reduced to the study of irreducible representations. In order to introduce them, we first bring in the following definition.

► **Definition 5.** *Let ρ be a representation of a group G . A vector subspace $W \subseteq V_\rho$ is G -invariant if and only if*

$$(\forall g \in G, \forall w \in W) \quad \rho(g)w \in W.$$

Observe that if W is G -invariant then the restriction $\rho|_W$ of ρ to W is a representation of G . We can now introduce irreducible representations.

► **Definition 6.** *A representation ρ of a group G is irreducible if $V_\rho \neq \emptyset$ and its only G -invariant subspaces are $\{0\}$ and V_ρ .*

We use $\text{Irrep}(G)$ to denote the set of all irreducible representations of G up to an isomorphism, where an isomorphism between representations is given by the following definition.

► **Definition 7.** *Two representations ρ and τ of a group G are isomorphic if there is an invertible linear operator $\varphi: V_\rho \rightarrow V_\tau$ such that*

$$\varphi \circ \rho(g) = \tau(g) \circ \varphi, \quad \forall g \in G.$$

We write $\rho \cong \tau$ to denote that ρ is isomorphic to τ .

Representation theory is used in this work because it is a natural language for expressing Fourier analysis of the space $L^2(G) = \{f : G \rightarrow \mathbb{C}\}$, which will be an ubiquitous tool in this work. We endow the space $L^2(G)$ with the scalar product defined as

$$\langle f_1, f_2 \rangle_{L^2(G)} = \frac{1}{|G|} \sum_{g \in G} f_1(g) \overline{f_2(g)},$$

21:6 Max-3-Lin over Non-Abelian Groups with Universal Factor Graphs

which induces a norm on $L^2(G)$ via

$$\|f\|_{L^2(G)} = \sqrt{\langle f, f \rangle_{L^2(G)}} = \sqrt{\frac{1}{|G|} \sum_{g \in G} |f(g)|^2},$$

We can now introduce the Fourier transform as follows.

► **Definition 8.** For $f \in L^2(G)$, the Fourier transform of f is an element \hat{f} of $\prod_{\rho \in \text{Irrep}(G)} \text{End}(V_\rho)$ given by

$$\hat{f}(\rho) = \mathbb{E}_{x \in G} [f(x)\rho(x)].$$

In the definition above we use $\text{End}(V_\rho)$ to denote a set of endomorphisms of the vector space V_ρ . In particular, once we fix bases of $\{V_\rho\}_{\rho \in \text{Irrep}(G)}$, we can identify \hat{f} with a matrix. Throughout this article we will consider the space of matrices of same dimension to be equipped by the scalar product defined as

$$\langle A, B \rangle_{\text{End}(V_\rho)} = \text{tr}(AB^*). \quad (1)$$

Note that if we consider A and B as operators on a vector space V the definition of $\langle A, B \rangle_{\text{End}(V)}$ does not change under unitary transformations of basis. Let us now state the Fourier inversion theorem which shows that the function is uniquely determined by its Fourier coefficients.

► **Lemma 9 (Fourier inversion).** For $f \in L^2(G)$ we have

$$f(x) = \sum_{\rho \in \text{Irrep}(G)} \dim(\rho) \langle \hat{f}(\rho), \rho(x) \rangle_{\text{End}(V_\rho)}.$$

Plancherel's identity can be written in this setting as follows.

► **Lemma 10. (Plancherel's identity)**

$$\langle f, g \rangle_{L^2(G)} = \sum_{\rho \in \text{Irrep}(G)} \dim(\rho) \langle \hat{f}(\rho), \hat{g}(\rho) \rangle_{\text{End}(V_\rho)}.$$

A straightforward corollary of Plancherel's identity is Parseval's identity, given in the following lemma.

► **Lemma 11. (Parseval's identity)** For $f: G \rightarrow \mathbb{C}$ we have

$$\|f(x)\|_{L^2(G)}^2 = \sum_{\rho \in \text{Irrep}(G)} \dim(\rho) \|\hat{f}(\rho)\|_{HS}^2,$$

where $\|\cdot\|_{HS}$ is a norm induced by the scalar product (1), i.e., it is the Hilbert-Schmidt norm defined on a set of linear operators on V_ρ by

$$\|A\|_{HS} = \sqrt{\langle A, A \rangle} = \sqrt{\text{tr}(AA^*)} = \sqrt{\sum_{ij} |A_{ij}|^2}.$$

The following lemma characterizes the values of scalar products between matrix entries of two representations. In particular, it shows that the matrix entries of two representations are orthogonal, and that $L^2(G)$ -norm of each entry of representation ρ equals to $1/\dim(\rho)$.

► **Lemma 12.** *If ρ and τ are two non-isomorphic irreducible representations of G then for any i, j, k, l we have*

$$\langle \rho_{ij}, \tau_{kl} \rangle_{L^2(G)} = 0.$$

Furthermore,

$$\langle \rho_{ij}, \rho_{kl} \rangle_{L^2(G)} = \frac{\delta_{ik} \delta_{jl}}{\dim(\rho)},$$

where δ_{ij} is the Kronecker delta function.

By taking $\tau \cong 1$ in the previous lemma we obtain the following corollary.

► **Lemma 13.** *Let $\rho \in \text{Irrep}(G) \setminus \{1\}$. Then*

$$\sum_{g \in G} \rho(g) = 0.$$

Let us now associate a character with each representation.

► **Definition 14.** *The character $\chi_\rho: G \rightarrow \mathbb{C}$ of a representation $\rho: G \rightarrow \text{GL}(V)$ is a function defined by*

$$\chi_\rho(g) = \text{tr}(\rho(g)).$$

Characters are orthogonal to each other as shown by the following lemma.

► **Lemma 15.** *For $\rho, \tau \in \text{Irrep}(G)$ we have*

$$\langle \chi_\rho(g), \chi_\tau(g) \rangle_{L^2(G)} = \begin{cases} 1, & \rho \cong \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Another nice identity that characters satisfy is given in the following lemma.

► **Lemma 16.**

$$\sum_{\rho \in \text{Irrep}(G)} \dim(\rho) \chi_\rho(g) = \begin{cases} |G|, & \text{if } g = 1_G, \\ 0, & \text{otherwise.} \end{cases}$$

Taking $g = 1_G$ in the previous lemma implies that $\sum_{\rho \in \text{Irrep}(G)} \dim(\rho)^2 = |G|$ and hence for every $\rho \in \text{Irrep}(G)$ we have $\dim(\rho) \leq \sqrt{|G|}$.

In this article we will also encounter convolution of functions in $L^2(G)$, which is defined as follows.

► **Definition 17.** *Given $f, g \in L^2(G)$, their convolution $f * g \in L^2(G)$ is defined as*

$$f * g(x) = \mathbb{E}_{y \in G} [f(y)g(y^{-1}x)].$$

Fourier analysis interacts nicely with the convolution, as shown by the following lemma.

► **Lemma 18.** *For $f, g \in L^2(G)$ we have*

$$\widehat{f * g}(\rho) = \hat{f}(\rho) \cdot \hat{g}(\rho).$$

Given a group G we can define the group G^n as the set of n -tuples of elements of G on which the group operation is performed coordinate-wise. It is of interest to study the structure of representations of G^n , particularly in relation to representations of G . In order to do so, let us first introduce direct sum and tensor product of representations.

► **Definition 19.** Let ρ and τ be two representations of G . We define their direct sum $\rho \oplus \tau$ to be a representation of G defined on vectors $(v, w) \in \rho_V \oplus \tau_V$ by

$$\rho \oplus \tau(v, w) = \rho(v) \oplus \tau(w).$$

Observe that if a representation ρ is not irreducible then there are G -invariant vector subspaces $V_1, V_2 \subseteq V_\rho$ such that V_ρ is a direct sum of V_1 and V_2 , i.e., $V_\rho = V_1 \oplus V_2$. Hence, by fixing a suitable basis of V_ρ we can write ρ as a block diagonal matrix with two blocks on the diagonal corresponding to $\rho|_{V_1}$ and $\rho|_{V_2}$, i.e., we have that $\rho \cong \rho|_{V_1} \oplus \rho|_{V_2}$. Since we assumed that $\dim(\rho) < \infty$ it follows by the principle of induction that we can represent each ρ as

$$\rho \cong \bigoplus_{s=1}^u \tau^s,$$

where $\tau^s \in \text{Irrep}(G)$. We now introduce the tensor product of representations.

► **Definition 20.** Let ρ and τ be representations of G . Tensor product $\rho \otimes \tau$ of ρ and τ is a representation of $V_\rho \otimes V_\tau$ defined by

$$(\rho \times \tau)(u \otimes v) = \rho(g)(u) \otimes \tau(g)(v),$$

and extended to all vectors of $V_\rho \otimes V_\tau$ by linearity.

Following lemma characterizes irreducible representations of G^n in terms of irreducible representations of G .

► **Lemma 21.** Let G be a group and let $n \in \mathbb{N}$. All irreducible representations of the group G^n are given by

$$\text{Irrep}(G^n) = \{ \otimes_{d=1}^n \rho_d \mid \rho_d \in \text{Irrep}(G) \}.$$

We will sometimes use $\rho = (\rho_1, \dots, \rho_n)$ to denote that $\rho = \otimes_{d \in [n]} \rho_d$. Furthermore, we will use $|\rho|$ to denote the number of $\rho_d \not\cong 1$.

2.2 Max-3-Lin and Universal Factor Graphs

We begin by introducing the Max-3-Lin problem over a group G .

► **Definition 22.** In the Max-3-Lin problem over a group G input is given by n variables x_1, \dots, x_n taking values in G and m constraints where i -th constraint is of the form

$$x_{i_1} \cdot x_{i_2} \cdot x_{i_3} = c_i \quad \text{for some } i_1, i_2, i_3 \in [n] \text{ and } c_i \in G.$$

Note that in this definition we do not allow for constants between the variables, i.e., we do not allow equations of the form $x_{i_1} \cdot g_i \cdot x_{i_2} \cdot g'_i \cdot x_{i_3} = c_i$, with $g_i, g'_i \in G$, which was not the case with the previous works [6, 10] on hardness of Max-3-Lin over non-abelian groups. Furthermore, this definition does not allow for inverses in equations, for example the constraint $x_{i_1} x_{i_2}^{-1} x_{i_3}^{-1} = c_i$ is not allowed, while in [10] these equations appeared since the analysis of the soundness required certain functions to be skew-symmetric (i.e., check Lemma 15 and Lemma 23 from [10]). Since in this work we consider hardness results, our proofs will also imply the hardness of instances defined in the sense of [6, 10].

Another strengthening of the previous results considered in this work comes from assuming that the instances have universal factor graphs. In order to formally explain this strengthening, we need to introduce the notion of a factor graph of a constraint satisfaction problem. We first recall the definition of a constraint satisfaction problem.

► **Definition 23.** A constraint satisfaction problem (CSP) over a language $\Sigma = [q]$, $q \in \mathbb{N}$, is a finite collection of predicates $K \subseteq \{P : [q]^k \rightarrow \{0, 1\} \mid k \in \mathbb{N}\}$.

We use $\text{ar}(P) = k$ to denote the arity of a predicate $P : [q]^k \rightarrow \{0, 1\}$. As an input to our problem we get an instance of a CSP K , which is defined as follows.

► **Definition 24.** An instance \mathcal{I} of a CSP K consists of a set $X = \{x_1, \dots, x_n\}$ of n variables taking values in Σ and m constraints C_1, \dots, C_m , where each constraint C_i is a pair (P_i, S_i) , with $P_i \in K$ being a predicate with arity $k_i := \text{ar}(P_i)$, and S_i being an ordered tuple containing k_i distinct variables which we call scope.

Let us denote by $\sigma : X \rightarrow \Sigma$ an assignment to the variables X of a CSP instance \mathcal{I} . We interpret $\sigma(S_i)$ as a coordinate-wise action of σ on S_i . Given σ , we define the value $\text{Val}_\sigma(\mathcal{I})$ of σ as

$$\text{Val}_\sigma(\mathcal{I}) = \sum_{r=1}^m P_r(\sigma(S_r)). \quad (2)$$

We work with Max-CSPs in which we are interested in maximizing $\text{Val}_\sigma(\mathcal{I})$. We use $\text{Opt}(\mathcal{I})$ to denote the optimal value of \mathcal{I} , i.e., $\text{Opt}(\mathcal{I}) = \max_{\sigma} (\text{Val}_\sigma(\mathcal{I}))$. Observe that Max-3-Lin over G can be seen as a Max-CSP with predicates $\bar{K} = \{P_g\}_{g \in G}$ where

$$P_g(x, y, z) = \begin{cases} 1, & \text{if } x \cdot y \cdot z = g, \\ 0, & \text{otherwise.} \end{cases}$$

We can now introduce factor graphs and the notion of hardness we are interested in this article.

► **Definition 25.** The factor graph \mathcal{F} of an instance \mathcal{I} consists of the scopes $S_r, r = 1, \dots, m$. A family $\{\mathcal{F}_n\}_{n \in \mathbb{N}}$ is explicit if it can be constructed in time that is polynomial in n .

► **Definition 26.** We say that the Max-CSP(K) is (c, s) -UFG-NP-hard if there is an explicit family of factor graphs $\{\mathcal{F}_n\}_n$ for K and a polynomial time reduction R from 3-Sat instances I_n on n variables to Max-CSP(K) instances $R(I_n)$ with the factor graph \mathcal{F}_n such that the following holds

- Completeness: If I_n is satisfiable then $\text{Opt}(R(I_n)) \geq c$.
- Soundness: If I_n is not satisfiable then $\text{Opt}(R(I_n)) \leq s$.

We will say that Max-CSP(K) has “universal factor graphs” to mean that Max-CSP(K) is (c, s) -UFG-NP-hard, in which case the values of c and s will be clear from the context.

► **Definition 27.** A polynomial time reduction R from CSP K to CSP K' is factor graph preserving if it satisfies the following property:

- Whenever two instances I_1, I_2 of K have the same factor graphs, the respective instances $R(I_1)$ and $R(I_2)$ output by the reduction R have the same factor graphs as well.

The definition of the (c, s) -UFG-NP-hardness given here matches the one from [4]. Note that another view is given by the definition from [13] which considered hardness from the perspective of circuit complexity, i.e., with the starting point being $\text{NP} \not\subseteq \text{P/Poly}$ one would

aim to show non-existence of polynomially sized circuits which distinguish soundness from completeness of instances with fixed factor graphs. This difference is only technical, and in particular it is not hard to see that the analogues of all the results given in this work also hold in the alternative setting considered in [13].

Majority of the strong hardness of approximation reductions such as [8, 14] use as an intermediary step in a reduction the parallel repetition [17, 18] of 3-Sat instances output by the PCP theorem [2, 3, 9]. Let us briefly describe the parallel repetition as a game played between two provers that can not communicate with each other and a verifier. We first fix the number of repetitions $r \in \mathbb{N}$. Then, in each round the verifier picks r random constraints $C_{i_1}, C_{i_2}, \dots, C_{i_r}, 1 \leq i_1, \dots, i_r \leq m$, where C_{i_j} are constraints of the starting 3-Sat instance with m constraints. Then, the verifier sends these r constraints to the prover P_1 . The verifier also picks one variable from each constraint sent to P_1 and sends these variables to the prover P_2 . The first prover responds with a satisfying assignment to the r constraints, while P_2 responds with values to r variables. The verifier accepts if the assignments of the provers P_1 and P_2 agree on the same variables.

Parallel repetition game is usually conceptualized by the Label Cover problem, whose definition can be given as follows.

► **Definition 28.** A Label Cover instance $\Lambda = (V, W, E, [L], [R], \Pi)$ is a tuple in which

- (V, W, E) is a bipartite graph with vertex sets V and W , and an edge set E .
- $[L]$ and $[R]$ are alphabets, where $L, R \in \mathbb{N}$.
- Π is a set which consists of projection constraints $\pi_e : [R] \rightarrow [L]$ for each edge $e \in E$.

The value of a Label Cover instance under an assignment $\sigma_L : V \rightarrow [L]$, $\sigma_R : W \rightarrow [R]$, is defined as the fraction of edges $e \in E$ that are satisfied, where an edge $e = (v, w)$ is satisfied if $\pi_e(\sigma_R(w)) = \sigma_L(v)$. We will write $\text{Val}_\sigma(\Lambda)$ for the value of the Label Cover instance Λ under the assignment $\sigma = (\sigma_L, \sigma_R)$.

In this definition one can see $[R]$ as the set of all possible satisfying assignments of the prover P_1 for a question $w \in W$. Hence, the set $[R]$ depends not only on the factor graph of the starting 3-Sat instance but also on the predicates applied to each triplet of variables. Due to this obstacle we can not use Label Cover as the black box input to our problem. In particular, we need to resort to bookkeeping the types of predicates received by the prover P_1 . Let us now describe the approach taken in this work.

The first difference compared to the “classical” approach to parallel repetition described above comes from the fact that as in [4] we apply parallel repetition to Max-TSA problem which is $(1, 1 - \varepsilon)$ -UFG-NP-hard [13] for some $\varepsilon > 0$. Max-TSA problem is a CSP with predicates $TSA_b : \{0, 1\}^5 \rightarrow \{0, 1\}$, $b \in \{0, 1\}$, given by

$$TSA_b(x_1, x_2, x_3, x_4, x_5) = \begin{cases} 1, & \text{if } x_1 + x_2 + x_3 + x_4 \cdot x_5 = b, \\ 0, & \text{otherwise.} \end{cases}$$

The reason for using Max-TSA problem instead of 3-Sat is technical in nature and it has to do with the fact in the case we use Max-TSA and we identify with $[R]$ all the possible assignments to the $5r$ variables received by P_1 , then checking whether an assignment $\ell \in [R]$ returned by the prover satisfies all the constraints is equivalent to checking whether r equations

$$p_j(\ell) = b_{i_j}, \quad j \in [r],$$

are true. This is achieved by setting each p_j to be the function that extracts values of variables appearing in j -th constraint received by P_1 , and if the values of the variables are

x_1, x_2, \dots, x_5 , then the function p_j returns the value $x_1 + x_2 + x_3 + x_4 \cdot x_5$. Observe that the parallel repetition applied to 3-Sat instances does not admit this description, in particular since for 3-Sat a single assignment to a triplet of variables can satisfy two different 3-Sat predicates. This difference turns out to be crucial for the functional folding which will be introduced in Section 2.3.

We can now conceptualize the game between the two provers described above with the following definition.

► **Definition 29.** A UFG Label Cover instance Λ_{UFG} is defined as $\Lambda_{UFG} = (\Lambda, U)$, where $\Lambda = (V, W, E, [L], [R], \Pi)$ is a Label Cover instance, while $U = (\mathcal{P}, \mathbf{b}, I)$ consists of

■ \mathcal{P} , which is a set of functions $p_1, \dots, p_r: [R] \rightarrow \{0, 1\}$, and

■ $\mathbf{b} \in \{0, 1\}^m$, $\mathbf{b} = (b_1, \dots, b_m)$,

■ I , which is a collection of functions $I_w: [r] \rightarrow [m]$, $w \in W$.

The value of a UFG Label Cover instance under an assignment $\sigma = (\sigma_L, \sigma_R)$ is the fraction of satisfied edges of Λ , where an edge $e = (w, v)$ is satisfied if it satisfies the projective constraint and if furthermore $p_i(\sigma_R(w)) = b_{I_w(i)}$ for $i = 1, \dots, r$.

In the definition above I_w returns the indices of the constraints received by the prover P_1 when given a query w .

Since the starting Max-TSA problem is (1, 0.51)-UFG-NP-hard [4], by the parallel repetition theorem [18] we have the following lemma.

► **Lemma 30.** For any $\gamma > 0$ there is a reduction from (1, 0.51)-Max-TSA to UFG Label Cover $\Lambda_{UFG}(\Lambda, U)$ such that it is NP-hard to distinguish the following cases

■ Completeness $\text{Opt}(\Lambda_{UFG}) = 1$,

■ Soundness $\text{Opt}(\Lambda_{UFG}) < \gamma$.

Furthermore, starting from two instances of Max-TSA with the same factor graphs, the reduction will produce the instances $\Lambda_{UFG}, \Lambda'_{UFG}$ which differ only in their respective values of vectors \mathbf{b}, \mathbf{b}' .

In our work for technical reasons we actually use smooth parallel repetition, which we describe in terms of two-prover game as follows. In this version apart from r we fix $t \in \mathbb{N}$, $1 < t < r$ as well, and instead of picking r variables from the constraints C_{i_1}, \dots, C_{i_r} sent to P_1 and sending them to P_2 , the verifier selects t constraints at random and sends one variable from each of these constraints to P_2 . The verifier also sends $r - t$ remaining constraints to P_2 . Note that in this version we do not need to force P_2 to satisfy these constraints since this is enforced by the agreement test performed by the verifier. We use smooth parallel repetition because we want to say that for any two answers given by P_1 it is highly likely that P_2 will have only one answer that will be accepted by the verifier. UFG Label Cover described in this paragraph will be referred to as (r, t) -smooth UFG Label Cover.

In order to express the smoothness property of Λ_{UFG} we will be interested in we introduce the following definition.

► **Definition 31.** Consider $\pi: [R] \rightarrow [L]$ and let $S \subseteq [R]$. We use $\mathcal{C}(S, \pi)$ to denote the indicator variable which is equal to 1 if and only if there are two distinct $i, j \in S$ such that $\pi(i) = \pi(j)$, or formally:

$$\mathcal{C}(S, \pi) = \begin{cases} 1, & \text{if } (\exists i, j \in S, i \neq j), \pi(i) = \pi(j), \\ 0, & \text{otherwise.} \end{cases}$$

Now, as shown by Claim 2.20 in [4], we can assume that the UFG Label Cover instance satisfies *smoothness* property defined as follows.

21:12 Max-3-Lin over Non-Abelian Groups with Universal Factor Graphs

► **Lemma 32.** *Let $0 \leq t \leq r, t \in \mathbb{N}$, consider (r, t) -smooth UFG Label Cover Λ_{UFG} and let $w \in W$ be any vertex of W . If we denote with E_w the set of all edges e incident to w , then for $S \subseteq [R]$ we have that*

$$\mathbb{E}_{e \in E_w} [\mathcal{C}(S, \pi_e)] \leq \frac{|S|^2 t}{r}.$$

The difference in the lemma above and Claim 2.20 in [4] is due to the fact that we use r and t here in the role played by $(r + t)$ and r in [4], respectively. Let us also observe that choosing sufficiently large t ensures that the parallel repetition of $(1, 0.51)$ -UFG-NP-hard Max-TSA still yields Λ_{UFG} instance with completeness 1 and soundness γ .

2.3 Functional Folding Meets Representation Theory

The next step in the “classical reduction” consists in encoding the answers of provers to a query w for P_1 (or v for P_2) by a long code which can be thought of as a function $f_w: G^R \rightarrow G$ for P_1 (or $f_v: G^L \rightarrow G$ for P_2), and using a suitable test along each edge of a Label Cover instance that depends on the hardness result we are trying to prove. Typically in order to avoid degenerate cases we require functions f_w and f_v to be folded, by requiring them to satisfy

$$f_w(c\mathbf{x}) = cf_w(\mathbf{x}), \quad f_v(c\mathbf{x}) = cf_v(\mathbf{x}),$$

where $c \in G$ and for $\mathbf{x} = (x_1, \dots, x_R)$ the action of c on \mathbf{x} is performed coordinate-wise, i.e., $c\mathbf{x} = (cx_1, \dots, cx_R)$. However, this folding does not depend on the values of b_i received by the prover P_1 , and hence the provers can come up with a strategy for only one \mathbf{b} which invalidates desired soundness property of the reduction. For that reason we use here functional folding which was the technical novelty introduced in [4].

► **Definition 33.** *Given a fixed collection of functions $p_1, p_2, \dots, p_r: [R] \rightarrow \{0, 1\}$ let us introduce an equivalence relation \sim on G^R by*

$$\mathbf{x} \sim \mathbf{y} \iff (\exists F: \{0, 1\}^r \rightarrow G) \text{ such that } (\forall d \in [R]) x_d = F(p_1(d), p_2(d), \dots, p_r(d))y_d.$$

Let $\mathbf{b} \in \{0, 1\}^r$, $\mathbf{b} = (b_1, \dots, b_r)$. We say that a function $f_{\mathbf{b}}: G^R \rightarrow G$ is functionally folded with respect to $(\{p_i\}_{i=1}^r, \mathbf{b})$ if for every $\mathbf{x} \sim \mathbf{y}$ such that

$$(\forall d \in [R]) x_d = F(p_1(d), p_2(d), \dots, p_r(d))y_d \quad \text{for some } F: \{0, 1\}^r \rightarrow G,$$

we have

$$f_{\mathbf{b}}(\mathbf{x}) = F(b_1, b_2, \dots, b_r)f_{\mathbf{b}}(\mathbf{y}). \tag{3}$$

For the sake of notational convenience we will usually omit the subscript \mathbf{b} and also not mention that the folding is with respect to $(\{p_i\}_{i=1}^r, \mathbf{b})$, especially when this can be easily inferred from the context.

Note that in order to construct a functionally folded function f we only need to define it on a fixed representative \mathbf{y} of each equivalence class $[\mathbf{y}]$. We can then extend the value of f to any $\mathbf{x} \sim \mathbf{y}$ by saying that

$$f(\mathbf{x}) = F(b_1, b_2, \dots, b_r)f(\mathbf{y}).$$

As an immediate consequence of the definition we have the following lemma.

► **Lemma 34.** *If $f: G^R \rightarrow G$ is functionally folded, then for any $c \in G$ and any $\mathbf{x} \in G^n$ we have*

$$f(c\mathbf{x}) = cf(\mathbf{x}).$$

Proof. By choosing $F \equiv c$ we have that $c\mathbf{x} \sim \mathbf{x}$, and since f is folded by (3) we have $f(c\mathbf{x}) = cf(\mathbf{x})$, irrespective of the values of b_i since F is constant. ◀

Following lemma allows us to eliminate trivial representations in the Fourier spectrum of a function $\rho \circ f$, where f is functionally folded.

► **Lemma 35.** *Let $f: G^n \rightarrow G$ satisfy $f(c\mathbf{x}) = cf(\mathbf{x})$ for every $c \in G$ and every $\mathbf{x} \in G^n$, and let $\rho \in \text{Irrep}(G) \setminus \{1\}$. Then for $g = [\rho \circ f]_{pq}$, where $1 \leq p, q \leq \dim(\rho)$, if $\alpha \cong 1$ we have $\hat{g}(\alpha) = 0$. The same statement holds if we replace the condition that f satisfies $f(c\mathbf{x}) = cf(\mathbf{x})$ by $f(\mathbf{x}c) = f(\mathbf{x})c$ for every $c \in G$.*

Proof.

$$\begin{aligned} \hat{g}(1) &= \mathbb{E}_{\mathbf{x} \in G^n} [g(\mathbf{x}) \cdot 1] = \mathbb{E}_{\mathbf{x} \in G^n} [\rho(f(\mathbf{x}))_{pq}] = \frac{1}{|G|} \mathbb{E}_{\mathbf{x} \in G^n} \left[\sum_{c \in G} \rho(f(c\mathbf{x}))_{pq} \right] \\ &= \frac{1}{|G|} \mathbb{E}_{\mathbf{x} \in G^n} \left[\sum_{c \in G} \sum_{1 \leq r \leq \dim(\rho)} \rho(c)_{pr} \rho(f(\mathbf{x}))_{rq} \right] = \frac{1}{|G|} \sum_{1 \leq r \leq \dim(\rho)} \mathbb{E}_{\mathbf{x} \in G^n} [\rho(f(\mathbf{x}))_{rq}] \sum_{c \in G} \rho(c)_{pr} = 0, \end{aligned}$$

where the last inequality holds since $\sum_{c \in G} \rho(c)_{pr} = 0$ by Lemma 13. The proof in case $f(\mathbf{x}c) = f(\mathbf{x})c$ is analogous. ◀

The lemma above will be useful when deriving the result for Max-3-Lin with almost perfect completeness. For Max-3-Lin over G with perfect completeness we will actually view all representations $\beta \in \text{Irrep}(G^L)$ with $\dim(\beta) = 1$ as trivial, and hence we will need the following lemma.

► **Lemma 36.** *Let $f: G^n \rightarrow G$ satisfy $f(c\mathbf{x}) = cf(\mathbf{x})$ for every $c \in G$ and every $\mathbf{x} \in G^n$, and let $\rho \in \text{Irrep}(G)$, $\dim(\rho) \geq 2$. Then for $g = [\rho \circ f]_{pq}$ where $1 \leq p, q \leq \dim(\rho)$, and $\alpha \in \text{Irrep}(G^n)$, $\dim(\alpha) = 1$, we have that $\hat{g}(\alpha) = 0$. The same statement holds if we replace the condition that f satisfies $f(c\mathbf{x}) = cf(\mathbf{x})$ by $f(\mathbf{x}c) = f(\mathbf{x})c$ for every $c \in G$.*

The statement and the proof of this lemma already appeared in [6] as Lemma 2.29, and hence we omit the details here.

Observe that the proof of Lemma 35 and Lemma 36 did not require any additional properties of functional folding, and therefore they can be applied for f that is “classically folded” as well. We prove strengthenings of Lemma 35 and Lemma 36 for functionally folded functions f in Section 3. These strengthenings are stated in Lemma 37 and Lemma 38 respectively.

3 Brief Overview

Throughout this work we consider a finite non-abelian group G and a CSP in which instances are defined by a set of variables x_1, \dots, x_n , taking values in G , and m constraints, in which every constraint is of form $x_i \cdot x_j \cdot x_k = c_t$, where $i, j, k \in [n]$, $t \in [m]$, $c_t \in G$. The triplet (i, j, k) is referred to as a scope of the constraint. The set of scopes of an instance is known as the factor graph. We are interested in approximability of instances in which the factor

graph is fixed. In particular, the question we ask is whether the problem remains equally hard even if we fix the factor graph? In this case we also say that the problem is Universal Factor Graph hard and use abbreviation UFG-hard.

In order to do so, we need to give a reduction from 3-Sat instances I_n on n variables to instances $R(I_n)$ of Max-3-Lin with a fixed factor graph for each $n \in \mathbb{N}$. Furthermore, since we are interested in inapproximability, we are also interested in showing the completeness/soundness of our reduction for some $0 < s < c \leq 1$, i.e., we would like to show

- *Completeness*: If the I_n is satisfiable then $\text{Opt}(R(I_n)) \geq c$.
- *Soundness*: If the starting I_n is not satisfiable then $\text{Opt}(R(I_n)) \leq s$.

We use Opt above to denote the maximum fraction of satisfiable constraints in a given instance under any assignment of values to the variables. Existence of a reduction with such properties implies UFG-hardness of approximating CSP within a ratio of s/c , and we usually say that such CSP is (s, c) -UFG-hard.

Standard reductions not concerned with UFG property usually consist in parallel repetition³ of the starting 3-Sat instance to obtain a Label Cover instance, on which one typically performs dictatorship testing. In our case for technical reasons instead of 3-Sat instance we start from Max-TSA instance with predicates TSA_0 and TSA_1 given by

$$TSA_b(x_1, x_2, x_3, x_4, x_5) = \begin{cases} 1, & \text{if } x_1 + x_2 + x_3 + x_4 \cdot x_5 = b, \\ 0, & \text{otherwise.} \end{cases}$$

We remark that this is the same approach as the one taken by the previous work [4], and that the reasoning behind this choice is explained in more detail in Section 2.2. It is already shown that Max-TSA is $(0.51, 1)$ -UFG-hard [4], and hence in order to give UFG-hardness result it is sufficient to give a reduction from Max-TSA to Max-3-Lin over G which preserves factor graphs and has appropriate soundness and completeness. In particular, we are interested in a reduction which, starting from instances of Max-TSA with a fixed factor graph (and possibly different predicates applies to the scopes), produces instances of Max-3-Lin over G with a fixed factor graph as well.

In particular, in order to explain the parallel repetition, let us first consider the standard approach which is not concerned with factor graphs. In this approach, we consider two provers, P_1 and P_2 , who cannot communicate with each other, and a verifier. We use $r \in \mathbb{N}$ to denote the constant number of repetitions. In each round of the game the verifier picks r random constraints $C_{i_1}, C_{i_2}, \dots, C_{i_r}, 1 \leq i_1, \dots, i_r \leq m$, of the starting Max-TSA instance with m constraints. Then, the verifier sends these r constraints to the prover P_1 . The verifier also picks one variable from each constraint sent to P_1 and sends these variables to the prover P_2 . The first prover responds with a satisfying assignment to the r constraints, while P_2 responds with values to r variables. The verifier accepts if the assignments of the provers P_1 and P_2 agree on the same variables.

One can check that the existing reductions [6, 10] would produce instances of Max-3-Lin with different factor graphs even if we fix the factor graph of the starting instance (i.e., only right hand sides of Max-TSA are varied). This is due to the fact that the alphabet available to the first prover consists of satisfying assignments, and hence depends not only on the factor graph but also on the right hand side of the starting Max-TSA instance.

In order to deal with this issue, in this work we generalize the approach taken in [4]. In particular, we make sure that the questions asked to prover P_1 do not depend on the right hand sides by relying on the following procedure. First, the verifier picks r random

³ We actually require smooth parallel repetition in our proofs, but we omit that detail in this section for the sake of clarity.

constraints with right hand sides b_1, \dots, b_r . Then, the verifier sends the scopes of these r constraints to the first verifier, and r variables, one from each of these scopes, to the second verifier. The answer of the first prover is long coded, i.e., if we identify the set of all possible assignments to the scopes received by P_1 with $[R]$, $R \in \mathbb{N}$, then the answer is given as the function $f: G^R \rightarrow G$, intended to be a dictator, but allowed to be any other function subject to the restriction called “folding” which we elaborate on soon. The answer of the second prover is long coded as well. In order to avoid trivial strategies we need to fold the answers. The standard folding [6, 10] does not force the answer of the first prover to depend on the right hand sides (or, in other words, the tables are not folded with respect to the values of b_1, \dots, b_r). For this reason this folding is not sufficiently strong for our result, and in particular the protocol with this folding has soundness 1.

Hence, instead of this classical approach, we use here “functional folding”, which is the main technical contribution of [4]. In particular, let us explain how the long code of the first prover P_1 is functionally folded. First, given an alphabet $[R]$, let us use $p_1, p_2, \dots, p_r: [R] \rightarrow \{0, 1\}$ to denote the functions which evaluate to the left hand side of the constraints received by the first prover. In particular, given $\mathbf{x} \in [R]$, the value of $p_i(\mathbf{x})$ for $i \in [r]$ is calculated as follows. First, we extract the values of variables x_1, x_2, x_3, x_4, x_5 that appear in the i -th scope received by the first prover. We then evaluate the left hand side of the TSA predicate, i.e., we evaluate $x_1 + x_2 + x_3 + x_4 \cdot x_5$, and return that as the value of $p_i(x)$.

We then split G^R into equivalence classes, and we query f by using only one fixed representative from each class. The equivalence relation \sim on G^R which is used to define equivalence classes is given by

$$\mathbf{x} \sim \mathbf{y} \iff (\exists F: \{0, 1\}^r \rightarrow G) \text{ such that } (\forall d \in [R]) x_d = F(p_1(d), p_2(d), \dots, p_r(d))y_d.$$

Then, when querying the function $f: G^R \rightarrow G$ at \mathbf{x} , we actually ask the prover for the value of f at the representative $\bar{\mathbf{x}} \sim \mathbf{x}$, and then calculate the value of $f(x)$ using the rule

$$f(\mathbf{x}) = F(b_1, \dots, b_r)f(\bar{\mathbf{x}}).$$

Since equivalence classes do not change with b_1, \dots, b_r , the choice of our representative $\bar{\mathbf{x}}$ does not change either, and this is an important property required for factor preserving reduction. Observe that however the value of $f(x)$ depends on b_1, \dots, b_r , and for that reason we also say that f is functionally folded with respect to $(\{p_i\}_{i=1}^r, \mathbf{b})$, where $\mathbf{b} = (b_1, \dots, b_r)$. Let us now describe the test performed by the verifier which queries the tables of provers and is used to create hard instances of Max-3-Lin over a group G with imperfect completeness. The test is performed on three tables:

- $f_{P_1}^F: G^R \rightarrow G$, the table of the first prover that is functionally folded,
- $f_{P_1}: G^R \rightarrow G$, the table of the first prover that is not folded,
- $f_{P_2}^F: G^L \rightarrow G$, the table of the second prover, where $L \in \mathbb{N}$ corresponds to the size of the alphabet $[L]$ available to P_2 , and f_{P_2} is “classically folded”, i.e., for every $c \in G$ we require

$$f_{P_2}^F(\mathbf{x} \cdot c) = f_{P_2}^F(\mathbf{x})c.$$

The test is then described as

- Sample \mathbf{y} uniformly at random from G^L .
- Sample \mathbf{x} uniformly at random from G^R .
- Set $\mathbf{z} \in G^R$ such that $z_i = x_i^{-1} \cdot \eta_i \cdot (y \circ \pi_e)_i^{-1}$, where $\eta_i = 1$ w.p. $1 - \varepsilon$, and $\eta_i \sim G$ w.p. ε .
- Test $f_{P_1}^F(\mathbf{x})f_{P_1}(\mathbf{z})f_{P_2}^F(\mathbf{y}) = 1_G$.

21:16 Max-3-Lin over Non-Abelian Groups with Universal Factor Graphs

In the test above, since $f_{P_1}^F$ is functionally folded, and we use the notation $f_{P_1}^F(\mathbf{x})$ to mean $F(b_1, \dots, b_r) f_{P_1}^F(\bar{\mathbf{x}})$, and similarly, $f_{P_2}^F(\mathbf{y})$ to mean $f_{P_2}^F(\bar{\mathbf{y}}) \cdot c$. Hence, we indeed get a Max-3-Lin instance where each constraint is of form $x_i x_j x_k = c_t$. Furthermore, since the representatives $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ do not change with b_1, \dots, b_r , the factor graph is constant, and hence it is sufficient to prove soundness and completeness of our reduction. In particular, we need to show the following:

- *Completeness*: If the starting Max-TSA instance is satisfiable, then the optimal value of Max-3-Lin output by the reduction is at least $1 - \varepsilon$.
- *Soundness*: If the optimal value of starting Max-TSA instance is at most 0.51 , then the optimal value⁴ of Max-3-Lin is at most $1/|G| + o(1)$.

Showing completeness is straightforward, and the main challenge lies in proving soundness. The proof of soundness of this reduction uses ideas involving Fourier analysis introduced in the seminal work of Håstad [14], as well as its generalizations [6, 10]. However, the proof departs from the previous works in two important ways. First, applying the test from [10] in our setting would give us the test

$$(f_{P_1}^F(\mathbf{x}^{a_1}))^{a_1} (f_{P_1}(\mathbf{z}^{a_2}))^{a_2} f_{P_2}^F(\mathbf{y}) = 1_G,$$

where $a_1, a_2 \in \{-1, 1\}$ are chosen uniformly at random. This would give us instances in which some variables are inverted in some equations. Since we aim here for a nicer form (and hence stronger hardness result) of Max-3-Lin, we opt for the test we initially described, and due to this choice our analysis parts ways with the previous work [10] and in particular requires arguably more careful treatment of different terms appearing in the expressions.

Another difference from the previous works comes from the fact that in our setting we are interested in showing UFG-hardness results. The main new challenge is to show that one can extract useful information from non-zero Fourier coefficients of the function $f_{P_1}^F$, in particular information which would be useful for decoding the strategies of the provers and proving the contrapositive of the soundness statement. In the classical setting, it was enough to say that constant Fourier character has weight 0 for the folded function. For UFG-hardness, in addition to this, we also need to show that non-zero Fourier coefficients “carry” information sufficient to extract assignment which satisfy all the constraints of the query received by the first prover. The exact formulation of this lemma is given below.

► **Lemma 37.** *Let $f: G^R \rightarrow G$ be functionally folded with respect to $(\{p_i\}_{i=1}^r, \mathbf{b})$, and let $\rho \in \text{Irrep}(G)$ be a representation such that $\rho \not\cong 1$. Given some $1 \leq p, q \leq \dim(\rho)$, let $g(\mathbf{x}) = \rho(f(\mathbf{x}))_{pq}$, and consider $\alpha \in \text{Irrep}(G^R)$, $\alpha = (\alpha_1, \dots, \alpha_R)$. If for every $d \in [R]$ such that $\alpha_d \not\cong 1$ we have $(p_1(d), \dots, p_r(d)) \neq \mathbf{b}$, then $\hat{g}(\alpha) = 0$.*

Proof. For $c \in G$ let us define a function $F_c: \{0, 1\}^r \rightarrow G$ by

$$\begin{cases} F_c(\mathbf{t}) = c, & \text{if } \mathbf{t} = \mathbf{b}, \\ F_c(\mathbf{t}) = 1_G, & \text{otherwise,} \end{cases}$$

and let us denote with \vec{F}_c the vector in G^R defined by $(\vec{F}_c)_d = F_c(p_1(d), \dots, p_r(d))$, $1 \leq d \leq R$. Then for $1 \leq i, j \leq \dim(\alpha)$ we can write

$$\hat{g}(\alpha)_{ij} = \mathbb{E}_{\mathbf{x} \in G^R} [g(\mathbf{x}) \alpha(\mathbf{x})_{ij}] = \frac{1}{|G|} \mathbb{E}_{\mathbf{x} \in G^R} \left[\sum_{c \in G} g(\vec{F}_c \mathbf{x}) \alpha(\vec{F}_c \mathbf{x})_{ij} \right].$$

⁴ The factor $o(1)$ depends on the number of rounds and we don’t explicitly express it here. The exact choice is explained in Section 4. We can think of $o(1)$ here as any small constant.

Let us now fix $\mathbf{x} \in G^R$ and study the term $\sum_{c \in G} g(\vec{F}_c \mathbf{x}) \alpha(\vec{F}_c \mathbf{x})_{ij}$. Denote with $\mathbf{y}^c := \vec{F}_c \mathbf{x}$, and observe that for each $c \in G$ we have $\mathbf{y}^c \sim \mathbf{x}$ since

$$y_d^c = F_c(p_1(d), \dots, p_r(d))x_d, \quad (\forall d \in [R]).$$

Therefore, since f is functionally folded we have that

$$g(\vec{F}_c \mathbf{x}) = \rho(f(\vec{F}_c \mathbf{x}))_{pq} = \rho(c(f(\mathbf{x})))_{pq}.$$

Let us now study the term $\alpha_{ij}(\vec{F}_c \mathbf{x})$. Using Lemma 21 let us represent α as the tensor product of irreducible representations $\alpha_d \in \text{Irrep}(G)$, $d = 1, \dots, R$, i.e., let us write $\alpha(\mathbf{x}) = \alpha_1(x_1) \otimes \alpha_2(x_2) \otimes \dots \otimes \alpha_R(x_R)$. Consider any $d \in [R]$ such that $\alpha_d \not\cong 1$. By the assumption of the lemma we have $(p_1(d), \dots, p_r(d)) \neq \mathbf{b}$. But in this case $F_c(p_1(d), \dots, p_r(d)) = 1_g$ and since

$$y_d^c = F_c(p_1(d), \dots, p_r(d))x_d,$$

we have that $x_d = y_d^c$. Since for the remaining d we have that α_d are constant (i.e., $\alpha_d \cong 1$), we have that $\alpha(\vec{F}_c \mathbf{x})_{ij} = \alpha(\mathbf{x})_{ij}$. Hence we can write

$$\hat{g}(\alpha)_{ij} = \frac{1}{|G|} \mathbb{E}_{\mathbf{x} \in G^R} \left[\sum_{c \in G} \rho(c(f(\mathbf{x})))_{pq} \alpha(\mathbf{x})_{ij} \right].$$

Finally, we have that

$$\sum_{c \in G} \rho(c(f(\mathbf{x})))_{pq} = \sum_{c \in G} (\rho(c) \rho(f(\mathbf{x})))_{pq} = \sum_{c \in G} \sum_{1 \leq r \leq \dim(\rho)} \rho(c)_{pr} \rho(f(\mathbf{x}))_{rq} = 0,$$

where we used the fact that $\sum_{c \in G} \rho(c)_{pr} = 0$ which holds since $\rho \not\cong 1$ allows us to use Lemma 12. \blacktriangleleft

We also prove hardness for Max-3-Lin with perfect completeness. The proof strategy follows closely the proof of Bhangale and Khot [6], and the main new contribution is in form of showing that in UFG setting non-zero Fourier coefficients of functionally folded functions “carry” information which is useful for decoding the strategies of the provers. In this case what we consider to be useful information is different from the imperfect completeness case. In particular, for decoding we are interested in representations which have dimension higher than or equal to 2. The statement and the proof of the main new contribution introduced in this paper which is used to show hardness of Max-3-Lin can be found below.

► **Lemma 38.** *Let $f: G^R \rightarrow G$ be functionally folded with respect to $(\{p_i\}_{i=1}^r, \mathbf{b})$, and let $\rho \in \text{Irrep}(G)$ be a representation such that $\dim(\rho) \geq 2$. Given some $1 \leq p, q \leq \dim(\rho)$, let $g(\mathbf{x}) = \rho(f(\mathbf{x}))_{pq}$, and consider $\alpha \in \text{Irrep}(G^R)$, $\alpha = (\alpha_1, \dots, \alpha_R)$. If for each $d \in [R]$ such that $\dim(\alpha_d) \geq 2$ we have that $(p_1(d), \dots, p_r(d)) \neq \mathbf{b}$, then $\hat{g}(\alpha) = 0$.*

Proof. For $c \in G$ we let a function $F_c: \{0, 1\}^r \rightarrow G$ and a vector \vec{F}_c be defined in the same way as in the proof of Lemma 37. For $1 \leq i, j \leq \dim(\alpha)$ we can write

$$\hat{g}(\alpha)_{ij} = \mathbb{E}_{\mathbf{x} \in G^R} [g(\mathbf{x}) \alpha(\mathbf{x})_{ij}] = \frac{1}{|G|} \mathbb{E}_{\mathbf{x} \in G^R} \left[\sum_{c \in G} g(\vec{F}_c \mathbf{x}) \alpha(\vec{F}_c \mathbf{x})_{ij} \right].$$

We now fix $\mathbf{x} \in G^R$ and study the term $\sum_{c \in G} g(\vec{F}_c \mathbf{x}) \alpha(\vec{F}_c \mathbf{x})_{ij}$. Denote with $\mathbf{y}^c := \vec{F}_c \mathbf{x}$, and observe that for each $c \in G$ we have $\mathbf{y}^c \sim \mathbf{x}$ since

$$y_d^c = F_c(p_1(d), \dots, p_r(d))x_d, \quad (\forall d \in [R]).$$

21:18 Max-3-Lin over Non-Abelian Groups with Universal Factor Graphs

Hence, since f is functionally folded we have that

$$g(\vec{F}_c \mathbf{x}) = \rho(f(\vec{F}_c \mathbf{x}))_{pq} = \rho(c(f(\mathbf{x})))_{pq}.$$

Let us now study the term $\alpha(\vec{F}_c \mathbf{x})_{ij}$. By Lemma 21 we can write $\alpha(\mathbf{x})$ as $\alpha(\mathbf{x}) = \alpha_1(x_1) \otimes \alpha_2(x_2) \otimes \dots \otimes \alpha_R(x_R)$. For each $d \in [R]$ such that $\dim(\alpha_d) \geq 2$ by the assumption of the lemma we have that $(\rho_1(d), \dots, \rho_r(d)) \neq \mathbf{b}$. Hence, for such d we have that $F_c(\rho_1(d), \dots, \rho_r(d)) = 1_G$ and

$$y_d^c = F_c(\rho_1(d), \dots, \rho_r(d))x_d = x_d.$$

Therefore, in the expression

$$\alpha(\vec{F}_c \mathbf{x}) = \alpha_1(\vec{F}_{c_1} x_1) \otimes \alpha_2(\vec{F}_{c_2} x_2) \otimes \dots \otimes \alpha_R(\vec{F}_{c_R} x_R),$$

each representation α_d of dimension greater than 1 satisfies $\alpha_d(\vec{F}_{c_d} x_d) = \alpha_d(x_d)$, and in particular for each such d the value of $\alpha_d(\vec{F}_{c_d} x_d)$ does not change with c . Hence, for a fixed \mathbf{x} if we denote with J the set of all the indices $d \in [R]$ such that $(\rho_1(d), \dots, \rho_r(d)) = \mathbf{b}$ we can write

$$\alpha(\vec{F}_c \mathbf{x})_{ij} = \prod_{d \in J} [\alpha_d(c x_d)]_{d_i d_j} \cdot C_{\alpha, \mathbf{x}},$$

where $C_{\alpha, \mathbf{x}} \in \mathbb{C}$ is a constant that depends on α, \mathbf{x} , but does not depend on c , and $1 \leq d_i, d_j \leq \dim(\alpha_d)$. Observe that for each α_d , where $d \in J$, is one-dimensional. Hence, we can write

$$\prod_{d \in J} [\alpha_d(c x_d)]_{d_i d_j} = \prod_{d \in J} \alpha_d(c x_d) = \prod_{d \in J} \alpha_d(c) \prod_{d \in J} \alpha_d(x_d).$$

Finally, since the product of one-dimensional representations is a one-dimensional representation, we can define with $\beta(c)$ the representation on G given by

$$\beta(c) = \prod_{d \in J} \alpha_d(c).$$

Then, we have that $\tilde{\beta}: G^R \rightarrow \mathbb{C}$ defined by $\tilde{\beta}(\mathbf{x}) = \overline{\beta(\mathbf{x})}$ is also a one-dimensional representation as shown in Claim 2.25 in [6]. But then

$$\begin{aligned} \sum_{c \in G} g(\vec{F}_c \mathbf{x}) \alpha(\vec{F}_c \mathbf{x})_{ij} &= \sum_{c \in G} \rho(c f(\mathbf{x}))_{pq} \beta(c) \prod_{d \in J} \alpha_d(x_d) \cdot C_{\alpha, \mathbf{x}} \\ &= \sum_{c \in G} \sum_{1 \leq r \leq \dim(\rho)} \rho(c)_{pr} \rho(f(\mathbf{x}))_{rq} \overline{\tilde{\beta}(c)} \prod_{d \in J} \alpha_d(x_d) \cdot C_{\alpha, \mathbf{x}} \\ &= \sum_{1 \leq r \leq \dim(\rho)} \left(\sum_{c \in G} \rho(c)_{pr} \overline{\tilde{\beta}(c)} \right) \rho(f(\mathbf{x}))_{rq} \prod_{d \in J} \alpha_d(x_d) \cdot C_{\alpha, \mathbf{x}} = 0, \end{aligned}$$

where the last inequality holds because $\sum_{c \in G} \rho(c)_{pr} \overline{\tilde{\beta}(c)} = |G| \langle \rho_{pr}, \tilde{\beta} \rangle_{L^2(G)} = 0$ by Lemma 12 and since $\tilde{\beta}$ and ρ are of different dimensions, and hence non-isomorphic. \blacktriangleleft

4 Hardness of Max-3-Lin with Almost Perfect Completeness with Universal Factor Graphs

In this section we describe a reduction from (r, t) -smooth UFG Label Cover to instances of Max-3-Lin over a non-abelian G with factor graph independent of $\mathbf{b} = (b_1, \dots, b_m)$. Let us introduce some notation first. For each $w \in W$ let $f_w^F: G^R \rightarrow G$ be functionally folded function with respect to $(\{\rho_i\}_{i=1}^r, (b_{I_w(1)}, \dots, b_{I_w(r)}))$, and for each $v \in V$ let $f_v^F: G^L \rightarrow G$ be “classically folded”, i.e., function f_v^F that satisfies

$$f_v^F(\mathbf{x}c) = f_v(\mathbf{x})c.$$

Finally, let $f_w: G^R \rightarrow G$ be a function without any restriction. When querying the function $f_w^F(\mathbf{x})$ in the procedure we describe below, we actually do not query $f_w^F(\mathbf{x})$ directly. Instead, we find the equivalence class of \mathbf{x} and the fixed representative $\bar{\mathbf{x}}$ of the equivalence class $[\mathbf{x}]$. Since $\mathbf{x} \sim \bar{\mathbf{x}}$ we have that $\mathbf{x}_d = F(\rho_1(d), \dots, \rho_r(d))\bar{\mathbf{x}}_d$. Because f_w^F is functionally folded we have

$$f_w^F(\mathbf{x}) = F(b_{I_w(1)}, \dots, b_{I_w(r)})f_w^F(\bar{\mathbf{x}}),$$

and instead of $f_w^F(\mathbf{x})$ we actually query $f_w^F(\bar{\mathbf{x}})$ from the prover P_1 and whenever we use $f_w^F(\mathbf{x})$ we actually mean $F(b_{I_w(1)}, \dots, b_{I_w(r)})f_w^F(\bar{\mathbf{x}})$. Note that $f_w^F(\bar{\mathbf{x}})$ does not depend on the values of $b_{I_w(1)}, \dots, b_{I_w(r)}$. Similarly, when querying $f_v^F(\mathbf{x} \cdot c)$ in the test below we use $f_v^F(\mathbf{x} \cdot c)$ to mean $f_v^F(\bar{\mathbf{x}})c$. With this in mind let us now we describe the Max-3-Lin instance we reduce to as a distribution of constraints generated by the following algorithm.

- Sample an edge $e = (v, w)$ from E .
- Sample \mathbf{y} uniformly at random from G^L .
- Sample \mathbf{x} uniformly at random from G^R .
- Set $\mathbf{z} \in G^R$ such that $z_i = x_i^{-1} \cdot \eta_i \cdot (y \circ \pi_e)_i^{-1}$, where $\eta_i = 1$ w.p. $1 - \varepsilon$, and $\eta_i \sim G$ w.p. ε .
- Test $f_w^F(\mathbf{x})f_w(\mathbf{z})f_v^F(\mathbf{y}) = 1_G$.

Since f_w^F and f_v^F are folded, the test $f_w^F(\mathbf{x})f_w(\mathbf{z})f_v^F(\mathbf{y}) = 1_G$ is actually

$$F(b_{I_w(1)}, \dots, b_{I_w(r)})f_w^F(\bar{\mathbf{x}})f_w(\mathbf{z})f_v^F(\bar{\mathbf{y}})c = 1_G,$$

which by rearranging gives us

$$f_w^F(\bar{\mathbf{x}})f_w(\mathbf{z})f_v^F(\bar{\mathbf{y}}) = F(b_{I_w(1)}, \dots, b_{I_w(r)})^{-1}c^{-1}.$$

Since $f_w^F(\bar{\mathbf{x}}), f_w(\mathbf{z}), f_v^F(\bar{\mathbf{y}})$ do not depend on the values of $b_{I_w(1)}, \dots, b_{I_w(r)}$ we get a Max-3-Lin instance whose factor graph does not depend on b_1, \dots, b_m . The following theorem proves the required hardness result. We refer the readers to the full version of the paper for the proof [1].

► **Theorem 39.** *Let $\varepsilon, \delta > 0$, let $\zeta = \frac{\log(2|G|^3/\delta)}{\varepsilon}$, and consider (r, t) -smooth UFG Label Cover Λ_{UFG} obtained by the parallel repetition of $(1, 0.51)$ -UFG-NP-hard Max-TSA, where t is chosen such that Λ_{UFG} has soundness at most $\frac{\delta^4}{32|G|^{12}}\zeta^{-1}$, and r is chosen such that*

$$\frac{t}{r} \leq \frac{\delta^2}{4|G|^6\zeta^2}.$$

If I is the instance of Max-3-Lin produced by the procedure described above with Λ_{UFG} as the starting point, then the following holds:

- Completeness: *If $\text{Opt}(\Lambda_{UFG}) = 1$ then $\text{Opt}(I) \geq 1 - \varepsilon$.*
- Soundness *If $\text{Opt}(\Lambda_{UFG}) \leq \frac{\delta^4}{32|G|^{12}}\zeta^{-1}$ then $\text{Opt}(I) \leq 1/|G| + \delta$.*

5 Hardness of Max-3-Lin with Perfect Completeness with Universal Factor Graphs

In this section we discuss the hardness result for Max-3-Lin with perfect completeness.

In order to apply the argument from [6] for high-dimensional terms we will show that we can suitably pick (r, t) such that (r, t) -smooth UFG Label Cover Λ_{UFG} satisfies the same *smoothness* property as the starting Label Cover instance in [6]. In particular, we have the following lemma.

► **Lemma 40.** *Consider (r, t) -smooth UFG Label Cover Λ_{UFG} constructed by the parallel repetition of $(1, 0.51)$ -UFG-NP-hard Max-TSA instance. There is a constant $d_0 \in (0, 1/3)$ such that given any fixed t for all sufficiently big r we have that*

$$\Pr_{e \in E_w} [|\pi_e(S)| < |S|^{d_0}] \leq \frac{1}{|S|^{d_0}}, \quad (\forall w \in W, S \subseteq [R]).$$

We provide the proof of this lemma in the full version of the paper [1]. We remark that one can prove also prove this lemma by applying Lemma 6.9 from [14] to our setting, in which case the statement holds for any $r \geq t$. However, since in this work we use parallel repetition of Max-TSA instead of Max-3-Sat we opt to reprove this lemma for the sake of completeness, and use less involved argument at the expense of having possibly much larger r which increases the size of starting Λ_{UFG} instance polynomially and hence does not affect the overall argument of this section.

As in the previous section we start from (r, t) -smooth UFG Label Cover, and along each edge we test functions $f_w^F, f_w: G^R \rightarrow G, f_v: G^L \rightarrow G$, where f_w^F and f_v^F are folded in the same way as previously. The reduction to Max-3-Lin instance is given as a distribution of constraints generated by the following algorithm.

- Sample an edge $e = (v, w)$ from E .
- Sample \mathbf{y} uniformly at random from G^L .
- Sample \mathbf{x} uniformly at random from G^R .
- Set $\mathbf{z} \in G^R$ such that $z_i = x_i^{-1} \cdot (y \circ \pi_e)_i^{-1}$.
- Test $f_w^F(\mathbf{x})f_w(\mathbf{z})f_v^F(\mathbf{y}) = 1_G$.

This is essentially the same test as in [6]; the only difference comes from the fact that we are folding two instead of three functions and that we are using functional folding on one of the functions. Let us now state the main theorem of this section. The proof can be found in the full version of the paper [1].

► **Theorem 41.** *Let $\delta > 0$, and C be a constant such that $C^{-d_0/2} \leq \frac{\delta^2}{12|G|^6}$. Consider (r, t) -smooth UFG Label Cover Λ_{UFG} obtained by the parallel repetition of $(1, 0.51)$ -UFG-NP-hard Max-TSA, where t is chosen such that Λ_{UFG} has soundness at most $\frac{\delta^4}{257|G|^{12+2C}}C^{-1}$, and r is chosen sufficiently big so that Lemma 40 holds and*

$$\frac{t}{r} \leq \frac{\delta^2}{8|G|^{6+C}C^2}.$$

If I is the instance of Max-3-Lin produced by the procedure described above with Λ_{UFG} as the starting point, then the following holds:

- **Completeness:** *If $\text{Opt}(\Lambda_{UFG}) = 1$ then $\text{Opt}(I) = 1$.*
- **Soundness** *If $\text{Opt}(\Lambda_{UFG}) \leq \frac{\delta^4}{257|G|^{12+2C}}C^{-1}$ then $\text{Opt}(I) \leq 1/|[G, G]| + \delta$.*

References

- 1 Aleksa Stankovic Amey Bhangale. Max-3-lin over non-abelian groups with universal factor graphs, 2021. [arXiv:2111.09256](#).
- 2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.
- 3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, January 1998. doi:10.1145/273865.273901.
- 4 Per Austrin, Jonah Brown-Cohen, and Johan Håstad. Optimal inapproximability with universal factor graphs. In *Proc. 51st Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 434–453, 2021. doi:10.1137/1.9781611976465.27.
- 5 Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\epsilon)$ -sat is NP-hard. *SIAM Journal on Computing*, 46(5):1554–1573, 2017. doi:10.1137/15M1006507.
- 6 Amey Bhangale and Subhash Khot. Optimal inapproximability of satisfiable k-lin over non-abelian groups. In *Proc. 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1615–1628, 2021. doi:10.1145/3406325.3451003.
- 7 Jakub Bulín, Andrei A. Krokhin, and Jakub Oprsal. Algebraic approach to promise constraint satisfaction. In *Proc. 51st ACM Symp. on Theory of Computing (STOC)*, pages 602–613, 2019. doi:10.1145/3313276.3316300.
- 8 Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3):27:1–27:32, 2016. doi:10.1145/2873054.
- 9 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.
- 10 Lars Engebretsen, Jonas Holmerin, and Alexander Russell. Inapproximability results for equations over finite groups. *Theor. Comput. Sci.*, 312(1):17–45, 2004. doi:10.1016/S0304-3975(03)00401-8.
- 11 Uriel Feige. Relations between average case complexity and approximation complexity. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 534–543, 2002. doi:10.1145/509907.509985.
- 12 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, March 1996. doi:10.1145/226643.226652.
- 13 Uriel Feige and Shlomo Jozeph. Universal factor graphs. In *Proc. 39th International Colloq. of Automata, Languages and Programming (ICALP), Part I*, volume 7391 of *LNCS*, pages 339–350. Springer, 2012. doi:10.1007/978-3-642-31594-7_29.
- 14 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.
- 15 Shlomo Jozeph. Universal factor graphs for every NP-hard boolean csp. In *Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques (APPROX/RANDOM)*, volume 28 of *LIPICs*, pages 274–283, 2014. doi:10.4230/LIPICs.APPROX-RANDOM.2014.274.
- 16 Pravesh K. Kothari, Ryuhei Mori, Ryan O’Donnell, and David Witmer. Sum of squares lower bounds for refuting any csp. In *Proc. 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 132–145, 2017. doi:10.1145/3055399.3055485.
- 17 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing*, 40(6):1871–1891, 2011. doi:10.1137/080734042.
- 18 Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998. doi:10.1137/S0097539795280895.
- 19 Jean-Pierre Serre. *Linear representations of finite groups*. Springer, 1977. doi:10.1007/978-1-4684-9458-7_1.
- 20 Audrey Terras. *Fourier Analysis on Finite Groups and Applications*. London Mathematical Society Student Texts. Cambridge University Press, 1999. doi:10.1017/CB09780511626265.