Baby PIH: Parameterized Inapproximability of Min CSP

Venkatesan Guruswami
Simons Institute for the Theory of Computing, Berkeley, CA, USA
Departments of EECS and Mathematics, University of California, Berkeley, CA, USA

Xuandi Ren
Department of EECS, University of California, Berkeley, CA, USA

Sai Sandeep
Department of EECS, University of California, Berkeley, CA, USA

Abstract

The Parameterized Inapproximability Hypothesis (PIH) is the analog of the PCP theorem in the world of parameterized complexity. It asserts that no FPT algorithm can distinguish a satisfiable 2CSP instance from one which is only $(1-\varepsilon)$-satisfiable (where the parameter is the number of variables) for some constant $0 < \varepsilon < 1$.

We consider a minimization version of CSPs (Min-CSP), where one may assign r values to each variable, and the goal is to ensure that every constraint is satisfied by some choice among the $r \times r$ pairs of values assigned to its variables (call such a CSP instance r-list-satisfiable). We prove the following strong parameterized inapproximability for Min CSP: For every $r \geq 1$, it is W[1]-hard to tell if a 2CSP instance is satisfiable or is not even r-list-satisfiable. We refer to this statement as "Baby PIH", following the recently proved Baby PCP Theorem (Barto and Kozik, 2021). Our proof adapts the combinatorial arguments underlying the Baby PCP theorem, overcoming some basic obstacles that arise in the parameterized setting. Furthermore, our reduction runs in time polynomially bounded in both the number of variables and the alphabet size, and thus implies the Baby PCP theorem as well.

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

Keywords and phrases Parameterized Inapproximability Hypothesis, Constraint Satisfaction Problems

Digital Object Identifier 10.4230/LIPIcs.CCC.2024.27

Funding *Venkatesan Guruswami*: Research supported in part by NSF grants CCF-2228287 and CCF-2211972 and a Simons Investigator award.
Xuandi Ren: Supported in part by NSF CCF-2228287.
Sai Sandeep: Supported in part by NSF CCF-2228287.

1 Introduction

Approximation algorithms and fixed parameter-tractabililty (FPT) are two ways to cope with NP-hard problems. Recently, there have been many works that unite the two by obtaining approximation algorithms for NP-Hard problems that run in FPT time. Examples include VERTEX COLORING[13, 40], k-PATH DELETION[27], VERTEX CYCLE PACKING[35], FLOW TIME SCHEDULING[43], MAX k-VERTEX COVER in d-uniform hypergraphs [42, 38], k-MEANS and k-MEDIAN [28, 5, 23, 1, 7, 12]. On the other hand, there are also various developments in FPT hardness of approximation, for example, for k-BICLIQUE [29], k-CLIQUE [6, 31, 32, 24, 34, 8], k-SETCOVER [6, 10, 25, 30, 26, 33] and so on. We refer to the survey by Feldmann, Karthik, Lee, and Manurangsi [19] for a more comprehensive list of both FPT approximation algorithms and FPT hardness of approximation results.

39th Computational Complexity Conference (CCC 2024).
Editor: Rahul Santhanam; Article No. 27; pp. 27:1–27:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

However, it is worth pointing out that the techniques used in proving FPT hardness of approximation results have been rather problem-specific, and there remain other basic problems for which even constant inapproximability is unknown. This situation is due to the lack of a PCP-like theorem in the FPT world. In classical polynomial time hardness of approximation, the PCP theorem is a fundamental result that leads to a plethora of other results via reductions. The analog of the PCP theorem in the FPT regime was explicitly proposed by Lokshtanov et al [36] and named *Parameterized Inapproximability Hypothesis (PIH).* The PIH states that there is an absolute constant $\varepsilon > 0$ such that for a 2CSP on k variables with alphabet size n, there is no algorithm running in time $f(k) \cdot n^{O(1)}$ that can distinguish a satisfiable instance from one where at most $(1 - \varepsilon)$ fraction of constraints can be simultaneously satisfied.

Analogous to the PCP theorem, PIH not only implies many FPT time inapproximability results including k-Clique, k-SetCover, k-ExactCover, and k-SetPacking, but also unifies the inapproximability landscape by serving as a versatile problem-agnostic starting point. Previously, PIH was known to be implied by Gap-ETH [16, 11], a strong assumption with an inherent gap in it. Establishing PIH under a gap-free hypothesis has been a significant open problem in parameterized complexity. Very recently, following the posting of this work, PIH was established under ETH [22] using the technique of parallelization. The authors first reduce 3SAT to a vectorized problem called Vector-Valued CSP, and then design parallel probabilistically checkable proofs to verify its satisfiability. However, the proof in [22] fails to get any inapproximability of 2CSP under $\mathsf{W}[1] \neq \mathsf{FPT}$, since it is not known how to reduce a $\mathsf{W}[1]$-complete problem, say k-Clique, to their Vector-Valued CSP. In fact, it was pointed out in [22] that Vector-Valued CSP is likely to be $\mathsf{M}[1]$-complete and thus not likely to be $\mathsf{W}[1]$-hard, where $\mathsf{M}[1]$ is an intermediate complexity class between FPT and $\mathsf{W}[1]$ [20, 9]. Proving PIH under $\mathsf{W}[1] \neq \mathsf{FPT}$, therefore, still remains an important open problem.

In this work, we prove a list version of the PIH, which we call *Baby PIH*, under the minimal hypothesis $\mathsf{W}[1] \neq \mathsf{FPT}$. In Baby PIH, we study r-list assignments to the underlying 2CSP instance. An r-list assignment has a list $L(u)$ of at most r values to each variable u, and a constraint between variables u, v is said to be satisfied by the r-list assignments if there is at least one pair of values $x \in L(u), y \in L(v)$ such that (x, y) satisfies the constraint between the variables u, v.

▶ **Definition 1** (Baby PIH). *For all constants $r, C \geq 1$ and every computable function f, there is no algorithm running in time $f(k) \cdot n^C$ that can distinguish between the following cases, on input a 2CSP instance on k variables with alphabet size n.*

- *(Completeness) There is an assignment satisfying all the constraints.*
- *(Soundness) No r-list assignment satisfies all the constraints.*

The Baby PIH can be compared with the "Baby PCP theorem," which was established via a purely combinatorial gap theorem in a remarkable recent work of Barto and Kozik [4] (who also coined the term Baby PCP). The Baby PCP theorem asserts the NP-completeness of distinguishing satisfiable instances of a 2CSP from those which lack a satisfying r-list assignment, for all constants $r > 1$. Baby PCP differs from Baby PIH in the sense that it concerns 2CSP on n variables with constant alphabet size. The concept of Baby PCP is itself not new and was studied in the early PCP days under the guise of a certain minimization version of Label Cover [2] (see also [17]).[1]

[1] Label Cover is stronger than Baby PCP as stated above, as it imposes that the 2CSP relations are functions (usually referred to as projection property in PCP parlance). However, Barto and Kozik's version also has this projection property.

The term “Baby” stems from the fact that after reducing the soundness parameter to an arbitrarily small positive constant via parallel repetition, one can deduce the Baby PCP theorem from the PCP theorem. This strategy also works in the PIH setting. Namely, we can use the canonical “clause-variable” construction to build a 2CSP instance with projection property, then apply parallel repetition [41] to amplify the soundness to below $\frac{1}{r^2}$. Suppose a 2CSP instance is r-list satisfiable, then by randomly picking a value from each variable’s list, we can conclude there is a single assignment that satisfies at least $\frac{1}{r^2}$ fraction of constraints. Therefore, being unable to approximate 2CSP within a $\frac{1}{r^2}$ factor implies that it is hard to distinguish whether a instance is satisfiable or not r-list satisfiable. In other words, PIH implies Baby PIH. Thus establishing the Baby PIH is a necessary step towards proving the PIH itself, and we also believe that it is a valuable intermediate step.

In this work, we prove Baby PIH under the minimal complexity assumption $\mathsf{W[1]} \neq \mathsf{FPT}$.

▶ **Theorem 2.** *Assuming* $\mathsf{W[1]} \neq \mathsf{FPT}$*, Baby PIH is true.*

Our proof of Theorem 2 is combinatorial and follows the framework of Barto and Kozik’s proof of the Baby PCP theorem. Specifically, given a 2CSP instance with variable set X and a list size r, we choose large enough integers a, b depending on r, and construct a bipartite direct product 2CSP instance, whose variable set is $\binom{X}{a} \cup \binom{X}{b}$. Given that the product instance is r-list satisfiable, we can repeatedly choose smaller integers $a' \leq a, b' \leq b$ and extract assignments for the instance with variable set $\binom{X}{a'} \cup \binom{X}{b'}$. The new assignments still list satisfy the smaller instance, but the size of each list on one side is decreased by 1, which helps us to do induction.

We highlight that although this proof strategy looks simple, there are basic obstacles to employ it in the Baby PIH setting (compared to the Baby PCP setting). In the PCP world, the alphabet size $|\Sigma|$ is at most some constant. Thus, to extract assignments for the smaller instance, it is affordable to pick a, b large enough depending on $|\Sigma|$. The running time of there reduction is therefore $|X|^{O_{r,|\Sigma|}(1)}$. However in the PIH case, $|\Sigma|$ can be as large as the input size, and the running time of the reduction can only be $f(|X|) \cdot |\Sigma|^{O_r(1)}$ for some computable function f. To overcome this barrier, we non-trivially use the underlying structure of r-list satisfying assignments. We further note that our reduction runs in time $|X|^{O_r(1)}$, which is polynomial also in $|X|$. Therefore, our methods give a *unified proof of both the Baby PCP theorem and Baby PIH.* [2]

As we mentioned earlier, PIH implies Baby PIH, and thus our result can be viewed as a first step towards proving the former. An intermediate problem between them is the following average version of Baby PIH: let an r-average list assignment be a labeling $L(u)$ for each variable u such that the average cardinality of $L(u)$ over all the variables u is at most r.

▶ **Conjecture 3** (Average Baby PIH). *For any constants $r > 1, C$ and any computable function f, no algorithm can given as input a 2CSP instance on k variables with size n, distinguish between the following two cases in $f(k) \cdot n^C$ time:*

- *(Completeness) There is an assignment satisfying all the constraints.*
- *(Soundness) No r-average list assignment satisfies all the constraints.*

[2] Barto and Kozik [4] derive Baby PCP with the stronger projection property. Using our techniques, we can get Baby PCP or Baby PIH with rectangular constraints, which is a slightly weaker property but still enough for many downstream reductions.

Note that the difference between the Baby and Average Baby versions is to use the ℓ_∞ vs. ℓ_1 norms of the number of values given to the variables. Once again, Average Baby PIH has a counterpart in the PCP world, namely the minimization version of Label Cover with ℓ_1 total cost, which fueled early inapproximability results for SetCover and basic problems that concern linear codes and lattices [2], as surveyed in [3].

Average Baby PIH is an intriguing open problem and could help in making further progress towards PIH. Furthermore, the Average Baby PIH is strong enough to imply some non-trivial inapproximability results. Notably, with an additional property called rectangular constraints (which we will define later), it implies constant inapproximability of k-ExactCover, which we previously only knew under PIH [39].

Towards a better understanding of Average Baby PIH, we give a counterexample showing that the direct product construction we use to prove Baby PIH as is cannot establish the average version. This suggests in order to get Average Baby PIH or full PIH, one may need other techniques or constructions. As a candidate, we pose a question that whether the W[1]-hardness of approximating k-Clique can help us bypass this counterexample. Please refer to Section 5 for details.

Organization. In Section 2, we introduce some preliminaries, including the problems considered in this paper and related complexity hypotheses. In Section 3, we prove Baby PIH via the direct product construction. We then discuss Average Baby PIH in Section 4, and conclude with some open problems in Section 5.

2 Preliminaries

We first start by formally defining 2-CSP.

▶ **Definition 4** (2CSP). *An instance of arity-2 constraint satisfaction problem (2CSP) is a tuple $\Pi = (X, \Sigma, \Phi)$, where:*

- *$X = \{x_1, \dots, x_k\}$ is the set of variables;*
- *$\Sigma = \{\Sigma_{x_1}, \dots, \Sigma_{x_k}\}$ is the set of their respective domains of values. We use $|\Sigma|$ to denote the maximum size of any domain, and call it the alphabet size of Π.*
- *$\Phi = \{\phi_1, \dots, \phi_m\}$ is the set of constraints. Each constraint ϕ_i is ordered tuple $\langle w_i, R_i\rangle$, where $w_i = (w_{i,1}, w_{i,2}) \in X^2$ is a pair of variables, and R_i is a 2-ary relation on $\Sigma_{w_{i,1}}$ and $\Sigma_{w_{i,2}}$.*

An assignment of the 2CSP instance is a function from all variables in X to their respective domains. A constraint ϕ_i is said to be satisfied by an assignment σ if $(\sigma(w_{i,1}), \sigma(w_{i,2})) \in R_i$. We say an assignment σ satisfies the 2CSP instance if all constraints are satisfied by σ.

For each constraint ϕ_i, we can without loss of generality assume $w_{i,1} \neq w_{i,2}$, since unary constraints can be removed by restricting the domain of that variable.

We define r-list satisfiability, which generalizes the above satisfiability.

▶ **Definition 5** (r-List Satisfiability). *Given a 2CSP instance $\Pi = (X, \Sigma, \Phi)$, a multi-assignment is a function mapping each variable to a subset of its domain. We define the size of a multi-assignment σ as $\max_{x\in X} |\sigma(x)|$.*

We say a multi-assignment σ r-list satisfies Π if σ is of size at most r, and for every constraint ϕ_i, there exists a pair of values $u \in \sigma(w_{i,1})$ and $v \in \sigma(w_{i,2})$, such that $(u, v) \in R_i$.

Normal satisfiability can be viewed as 1-list satisfiability. Note that as r increases, it becomes easier to r-list satisfy a 2CSP instance.

The relations in the 2CSPs that we construct using the direct product (see Definition 7) satisfy a useful structural property, namely, rectangular relations.

▶ **Definition 6** (Rectangular Relation). *A relation $R \subseteq A \times B$ is said to be rectangular if there is a set C and functions $\pi : A \to C$ and $\sigma : B \to C$ such that $(a, b) \in R$ if and only if $\pi(a) = \sigma(b)$. Equivalently, R is rectangular if for all $a, a' \in A$ and $b, b' \in B$ such that $(a, b) \in R, (a, b') \in R$, and $(a', b) \in R$, we have $(a', b') \in R$.*

Rectangular relations can be informally viewed as consistency checks, and they are often satisfied by 2CSPs in product constructions. Projection relation, a stronger version of rectangular relation, is ubiquitous in PCP-based hardness reductions.

We now formally define the direct product construction that we use in our proof.

▶ **Definition 7** (Direct Product Construction). *Given a 2CSP instance $\Pi = (X, \Sigma, \Phi)$, its t-wise direct product, denoted as $\Pi^{\odot t}$, is the following 2CSP instance (X', Σ', Φ'):*

- *$X' = \binom{X}{t}$, where each variable is a t-sized subset of variables in Π.*
- *The domain of each variable $S \in X'$ is the set of all partial satisfying assignments for S in Π, i.e., all function σ that maps each $x \in S$ to its domain in Π, such that all constraints in Π induced by S are satisfied.*
- *Φ' has of a consistency constraint for each pair of distinct variables in X'. For $S, T \in X'$, the assignments σ_S, σ_T satisfy the constraint if and only if they are consistent on the values for variables in $S \cap T$.*

Our results are based on the hypothesis $\mathsf{W[1]} \neq \mathsf{FPT}$, which is closely related to k-Clique, a fundamental problem in parameterized complexity theory.

▶ **Definition 8** (k-Clique). *An instance of (multicolored) k-Clique problem is an undirected graph $G = (V = V_1 \dot\cup \ldots \dot\cup V_k, E)$, where each V_i is an independent set. The goal is to decide whether we can find $v_1 \in V_1, \ldots, v_k \in V_k$ which form a clique. For $r > 1$, the r-gap version of k-Clique asks to distinguish between the following two cases:*

- *(Yes) There exists $v_1 \in V_1, \ldots, v_k \in V_k$ which form a clique.*
- *(No) The maximum clique in G has size at most k/r.*

The $\mathsf{W[1]} \neq \mathsf{FPT}$ hypothesis states that for any computable function f, no algorithm can solve a k-Clique instance with size n in $f(k) \cdot n^{O(1)}$ time. For a k-Clique instance $G = (V = V_1 \dot\cup \ldots \dot\cup V_k, E)$, we can build k variables $x_1, \ldots, x_k$, letting V_i be the domain of variable x_i and making the edge set between V_i and V_j the constraint relation for x_i and x_j. Thus, G corresponds to a 2CSP instance with k variables and alphabet size at most n. It is easy to see that there is a clique of size k in G if and only if the 2CSP instance is satisfiable. Thus, we can restate the $\mathsf{W[1]} \neq \mathsf{FPT}$ hypothesis as follows.

▶ **Hypothesis 9** ($\mathsf{W[1]} \neq \mathsf{FPT}$). *For any computable function f, no algorithm can decide whether a given 2CSP instance $\Pi = (X, \Sigma, \Phi)$ is satisfiable in $f(|X|) \cdot |\Sigma|^{O(1)}$ time.*

The approximation version (with respect to the fraction of constraints that can be satisfiable) of $\mathsf{W[1]} \neq \mathsf{FPT}$ is called Parameterized Inapproximability Hypothesis (PIH).

▶ **Hypothesis 10** (Parameterized Inapproximability Hypothesis (PIH)[36]). [3] *For any computable function f and some constant $\varepsilon > 0$, no algorithm can given as input a 2CSP instance $\Pi = (X, \Sigma, \Phi)$, distinguish between the following two cases in $f(|X|) \cdot |\Sigma|^{O(1)}$ time:*

- *(Completeness) Π is satisfiable.*
- *(Soundness) Any assignment of Π violates at least ε fraction of constraints.*

[3] Note that the original PIH in [36] states that constant approximating 2CSP parameterized by $|X|$ is W[1]-hard. Here we use a relaxed form.

We formally define k-SetCover, the parameterized version of the classical SetCover problem, and the exact version of it.

▶ **Definition 11** (k-SetCover, k-ExactCover). *An instance of k-SetCover problem is a tuple $\Pi = (\mathcal{S}, U)$, where $\mathcal{S}$ is a collection of subsets $\{S_1, \ldots, S_n\}$ over the universe U, and the goal is to decide whether there are k sets in $\mathcal{S}$, whose union is U. For $r > 1$, the r-gap version of k-SetCover asks to distinguish between the following two cases:*

- *(Yes) There are k sets whose union is U.*
- *(No) The union of any $r \cdot k$ sets is not U.*

Furthermore, if in the yes case, the k sets are non-intersecting, i.e., they form a partition of U, then we also denote this gap problem as k-ExactCover.

Finally, we define the (T, m)-set gadget, an important gadget used in reductions to k-SetCover and k-ExactCover.

▶ **Definition 12** ((T, m)-Set Gadget). *A (T, m)-set gadget consists of a universe $\mathcal{M}$ and some of its subsets $C_1, \ldots, C_m$ with the following property: Every collection of at most T sets out of $C_1, \overline{C_1}, C_2, \overline{C_2}, \ldots, C_m, \overline{C_m}$ that is a set cover for $\mathcal{M}$ must include both C_i and $\overline{C_i}$ for some i.*

It was proved in [37] that a (T, m)-set gadget can be constructed efficiently:

▶ **Lemma 13.** *There is an algorithm that given any T, m, runs in time poly$(m, 2^T)$ and outputs a (T, m)-set gadget with universe size poly$(m, 2^T)$.*

3 Baby PIH

In this section, we analyze the direct product construction to prove Baby PIH under $\mathsf{W}[1] \neq \mathsf{FPT}$.

▶ **Theorem 14** (Main). *For any integer $r > 1$, there is an integer $t > 0$ such that for any 2CSP instance $\Pi = (X, \Sigma, \Phi)$ and its t-wise direct product instance $\Pi^{\odot t} = (X', \Sigma', \Phi')$:*

- *(Completeness) If Π is satisfiable, then $\Pi^{\odot t}$ is satisfiable as well.*
- *(Soundness) If Π is not satisfiable, then $\Pi^{\odot t}$ is not r-list satisfiable.*

Since t is a constant depending solely on r, the number of variables in the new instance $|X'| = \binom{|X|}{t}$ depends only on $|X|$ rather than $|\Sigma|$, and the alphabet size of the new instance $|\Sigma'| \leq |\Sigma|^t$ is polynomial in $|\Sigma|$. Thus for any constant C and any computable function f, $f(|X'|) \cdot |\Sigma'|^C$ time is upper bounded by $g(|X|) \cdot |\Sigma|^{C \cdot t}$ time for some computable function g. Therefore, we have the following corollary from Theorem 14:

▶ **Corollary 15.** *Assuming $\mathsf{W}[1] \neq \mathsf{FPT}$, Baby PIH is true.*

Note that the completeness in Theorem 14 follows trivially by assigning the restriction of the satisfying assignment on X to each subset of variables. The main challenge is to show that when $\Pi^{\odot t}$ has a r-list satisfying assignment, Π is satisfiable. To prove this, we will first work on a bipartite version of the direct product of 2CSP.

▶ **Definition 16** (Bipartite Direct Product Construction). *Given a 2CSP instance $\Pi = (X, \Sigma, \Phi)$ and positive integers a, b, the (a, b)-bipartite direct product 2CSP instance $\Pi^{\odot(a,b)} = (X', \Sigma', \Phi')$ is constructed as follows.*

- *The variable set X' consists of all a-sized subsets of X on the left side, and all b-sized subsets of X on the right side. With a little abuse of notation, we have $X' = \binom{X}{a} \cup \binom{X}{b}$.*

- *The domain of each variable $S \in X'$ is the set of all partial satisfying assignments for S in Π.*
- *For every $S \in \binom{X}{a}$ and $T \in \binom{X}{b}$, we have a constraint in Φ' that checks whether σ_S and σ_T are consistent on the values for variables in $S \cap T$.*

If for a 2CSP instance Π, $\Pi^{\odot t}$ is r-list satisfiable for $t = \max(a, b)$, by taking restrictions of its assignments on the smaller sets, it is easy to see $\Pi^{\odot(a,b)}$ is r-list satisfiable as well. Thus, our goal is to show that if the bipartite instance $\Pi^{\odot(a,b)}$ is r-list satisfiable, then the original instance Π is satisfiable.

Our proof idea is borrowed from the Baby PCP theorem recently proved by Barto and Kozik [4]. However, their theorem crucially relies on the fact that the alphabet $|\Sigma|$ is as small as a constant, which helps them to extract satisfying assignments for the smaller instance. The running time of their reduction is, therefore, $|X|^{O_{|\Sigma|}(1)}$, which is not affordable here since $|\Sigma|$ is as large as the input size. We resolve this issue by making use of the structural properties of the assignments in the direct product construction arising from the fact that they satisfy r-list consistency. If we fix some set S on one side and consider its r assignments, each set on the other side that intersects S must have one of the r, which is a constant, assignments for their intersection part. We use this simple yet very useful observation when extracting the assignments in the inductive proof.

In the following, we first prove Lemma 17, which is crucial to extract list satisfying assignments for the smaller subsets. Then in Lemma 18, we analyze a special case of the bipartite direct product construction when each variable on the right (bigger) side has only one assignment, and the consistency requirement is a slightly weaker one. In Lemma 19, we finish the analysis of the bipartite direct product construction, from which we get Theorem 14 as a corollary.

▶ **Lemma 17.** *Let k, r, q, a, b, b' be integers satisfying $r, q > 0, a \geq k, b \geq r \cdot b' + a$. Consider the (a, b)-bipartite direct product 2CSP instance based on $\Pi = (X, \Sigma, \Phi)$. Let u be an r-sized multi-assignment for $\binom{X}{a}$ and v be a q-sized multi-assignment for $\binom{X}{b}$. Suppose for every $S \in \binom{X}{a}, T \in \binom{X}{b}$ with $T \supseteq S$, $v(T)_{|S} \cap u(S) \neq \emptyset$. Then for every $A \in \binom{X}{k}$, there is an assignment f_A for A such that for every $T' \in \binom{X}{b'}$, there is some $T \in \binom{X}{b}$ satisfying $T \supseteq T' \cup A$ and $v(T)_{|A} \ni f_A$.*

Proof. Suppose for the sake of contradiction that there is no such f_A for some set $A \in \binom{X}{k}$. In other words, for any assignment f on A, there exists a set $T'_f \in \binom{X}{b'}$, such that for every $T \in \binom{X}{b}$ satisfying $T \supseteq T'_f \cup A$, $v(T)_{|A} \not\ni f$.

Pick an arbitrary $S \in \binom{X}{a}$ with $S \supseteq A$. This can be done since $a \geq k$. Consider any set $T \in \binom{X}{b}$ which contains $\cup_{f \in u(S)_{|A}} (T'_f \cup S)$. Such a T exists since $b \geq r \cdot b' + a$. By the assumption about A, $v(T)_{|A}$ does not contain any value in $u(S)_{|A}$. This contradicts the consistency guarantee in the hypothesis of the lemma, namely that for any $T \in \binom{X}{b}$ with $T \supseteq S$, $v(T)_{|S}$ must contain some value in $u(S)$. ◀

See Figure 1 for an illustration of Lemma 17.

▶ **Lemma 18.** *For any integer $r > 0$, let $a = r$ and $b = (2r)^r$. Consider the (a, b)-bipartite direct product 2CSP instance based on $\Pi = (X, \Sigma, \Phi)$. Let u be an r-sized multi-assignment for $\binom{X}{a}$ and v be a 1-sized assignment for $\binom{X}{b}$. Suppose for every $S \in \binom{X}{a}$ and $T \in \binom{X}{b}$ with $S \subseteq T$, $v(T)_{|S} \in u(S)$. Then there is a global satisfying assignment σ to the 2CSP instance Π.*

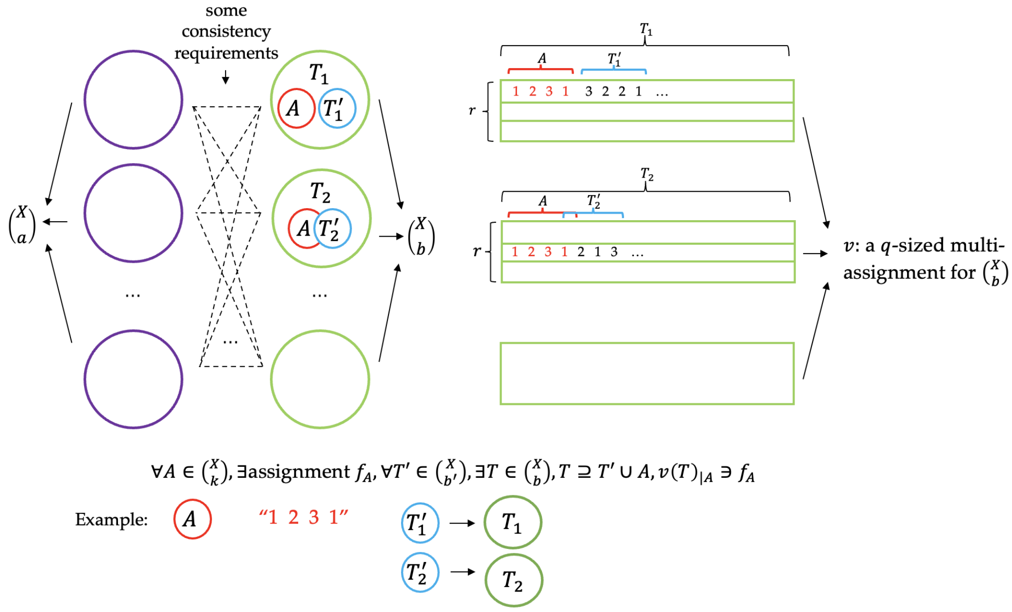
some consistency requirements
T_1
A
T_1'
T_2
A
T_2'
$\binom{X}{a}$
$\binom{X}{b}$
...
T_1
A
T_1'
r
1 2 3 1 3 2 2 1 ...
T_2
A
T_2'
r
1 2 3 1 2 1 3 ...
v: a q-sized multi-assignment for $\binom{X}{b}$
$\forall A \in \binom{X}{k}, \exists$assignment $f_A, \forall T' \in \binom{X}{b'}, \exists T \in \binom{X}{b}, T \supseteq T' \cup A, v(T)_{|A} \ni f_A$
Example:
A
"1 2 3 1"
T_1'
T_1
T_2'
T_2

Figure 1 An illustration of Lemma 17.

Proof. We apply induction on r. When $r = 1$, we have $a = 1, b = 2$, and both u, v are 1-sized assignments. We claim that u is the desired global satisfying assignment of Π. For each constraint on $(x, y) \in X^2$, $v(\{x, y\}) = (x \mapsto u(x), y \mapsto u(y))$ by our consistency guarantee. By the construction of bipartite direct product 2CSP instance, the domain for each $T \in \binom{X}{b}$ consists only of partial satisfying assignments. Thus the fact that $(x \mapsto u(x), y \mapsto u(y))$ lies in the domain of $T = \{x, y\}$ implies u satisfies the constraint on (x, y).

When $r > 1$, the idea is to extract consistent assignments for the (a', b')-bipartite direct product 2CSP instance and to decrease r, for some $a' \leq a, b' \leq b$. At a high level, if for some $x \in X$, every set in $\binom{X}{a}$ has at least two different values for it under u, we can keep only one of them and peel the other off to decrease r by 1; otherwise we can prove the unique assignments are already satisfying.

In the following, define $k = 1$, and $a' = a - 1, b' = (2(r-1))^{r-1}$, i.e., a', b' are parameters with respect to $r - 1$. It's easy to see $k \leq a$ and $r \cdot b' + a \leq r \cdot (2r)^{r-1} + r \leq (2r)^r = b$. According to Lemma 17, for every $A = \{x_i\} \in \binom{X}{k}$, there is an assignment f_A for A such that for every $T' \in \binom{X}{b'}$, there is some $T \in \binom{X}{b}$ satisfying $T \supseteq T' \cup A$ and $v(T)_{|A} \ni f_A$. Now $v(T)$ is of size-1, so we can simply write $v(T)_{|A} = f_A$.

Consider the following two cases:

Case 1. For some $A = \{x_i\}$, Lemma 17 holds for different assignments p, q. In other words, for every $T' \in \binom{X}{b'}$, there are $T_1, T_2 \in \binom{X}{b}$ satisfying $T_1, T_2 \supseteq T' \cup \{x_i\}$ and $v(T_1)_{|\{x_i\}} = p, v(T_2)_{|\{x_i\}} = q$.

Since for $r \geq 2$ we have $a = r \leq (2(r-1))^{r-1} = b'$, for each $S \in \binom{X}{a}$ containing x_i, we can pick an arbitrary $T' \supseteq S$ and consider the corresponding sets T_1, T_2 above. By the consistency assumption between (S, T_1) and between (S, T_2) in Lemma 18, we can infer $u(S)_{|\{x_i\}} \ni p, q$.

We construct new assignments u', v' for the (a', b')-bipartite direct product 2CSP instance of Π, such that they still meet the consistency requirements in Lemma 18, and the size of u' is at most $r-1$. For each $S' \in \binom{X}{a'}$, $u'(S')$ will be inherited from $u(S)$ for some $S \in \binom{X}{a}$ satisfying $S \supseteq S' \cup \{x_i\}$. Similarly for each $T' \in \binom{X}{b'}$, $v'(T')$ will be inherited from $v(T)$ for some $T \in \binom{X}{b}$ satisfying $T \supseteq T' \cup \{x_i\}$.

Suppose there is a arbitrary fixed order of all variables in X. We construct S from S' as follows, and output $u'(S') = \{\sigma \in u(S) | \sigma(x_i) = p\}_{|S'}$:

- If $x_i \notin S'$, let $S = S' \cup \{x_i\}$.
- If $x_i \in S'$, S is obtained by adding the lexicographical smallest element not in S' to S'.

Note that we only keep the assignments in $u(S)$ whose restriction on $\{x_i\}$ is p, and discard those whose restriction on $\{x_i\}$ is q. Thus u' is of size at most $r-1$ as desired.

For each $T' \in \binom{X}{b'}$, we first construct $T'' \in \binom{X}{b'}$ as follows:

- If $x_i \notin T'$, simply let $T'' = T'$.
- If $x_i \in T'$, T'' is obtained by adding the lexicographical smallest element not in T' to T', and delete x_i.

After that, we find a $T \in \binom{X}{b}$ satisfying $T \supseteq T'' \cup \{x_i\}$ and $v(T)_{|\{x_i\}} = p$, and output $v'(T') = v(T)_{|T'}$.

By our construction, $S' \subseteq T'$ implies $S \subseteq T$, so the new instance still meets the required consistency requirements, and the induction proceeds.

Case 2. Suppose for A being each $\{x_i\}$, Lemma 17 holds for a unique assignment z_i.

We claim that $(x_i \mapsto z_i)_{x_i \in X}$ is a global satisfying assignment. Indeed, consider an arbitrary constraint ψ between variables $(x_i, x_j) \in X^2$. Applying Lemma 17 to the choice $A = \{x_i, x_j\}$, we know there is an assignment f_A such that for every $T' \in \binom{X}{b'}$, there is some $T \in \binom{X}{b}$ satisfying $T \supseteq T' \cup A$ and $v(T)_{|A} = f_A$. Since $v(T)$ satisfies all constraints within T, this means that f_A must satisfy the constraint ψ. By the uniqueness assumption of this case, we must have $f_A(x_i) = z_i$ and $f_A(x_j) = z_j$, which means that the assignment (z_i, z_j) satisfies the constraint ψ between (x_i, x_j). ◀

We will now finish the analysis of the bipartite direct product construction.

▶ **Lemma 19.** *For integers $r, q > 0$, let $a = (2r)^{r+2q}, b = (2r)^{r+2q+2}$. Consider the (a, b)-bipartite direct product 2CSP instance based on $\Pi = (X, \Sigma, \Phi)$. Let u be an r-sized multi-assignment for $\binom{X}{a}$ and v be a q-sized multi-assignment for $\binom{X}{b}$. If u, v list-satisfy all bipartite constraints, then there is a global satisfying assignment σ to the 2CSP instance Π.*

Proof. We apply induction on q, the size of the right multi-assignment. When $q = 1$, we have $a = (2r)^{r+2} \geq r$ and $b = (2r)^{r+4} \geq (2r)^r$. We can extract consistent satisfying assignments for the $(r, (2r)^r)$-bipartite direct product 2CSP instance, and invoke Lemma 18 to prove there is a global satisfying assignment.

When $q > 1$, we categorize discussions based on whether the left multi-assignment u satisfies a certain property. Either we can still extract consistent assignments for the smaller (a', b')-bipartite direct product 2CSP instance while decreasing q and leaving r unchanged, or we can build multi-assignments that satisfy requirements in Lemma 18, and therefore directly invoke that lemma and stop the induction.

Define $k = (2r)^r$. The parameters a', b' with respect to $q-1$ would be $a' = (2r)^{r+2q-2}$ and $b' = (2r)^{r+2q}$. In our case-analysis we will use the following inequalities, which we first prove here.

- $b \geq r \cdot b' + a$. Indeed
$$\begin{aligned} r \cdot b' + a &= r \cdot (2r)^{r+2q} + (2r)^{r+2q} \\ &\leq 2 \cdot (2r)^{r+2q+1} \\ &\leq (2r)^{r+2q+2} = b. \end{aligned}$$
- $a \geq (r+1) \cdot (a' + k)$. Indeed
$$\begin{aligned} (r+1) \cdot (a' + k) &= (r+1) \cdot ((2r)^{r+2q-2} + (2r)^{r}) \\ &\leq (2r) \cdot (2 \cdot (2r)^{r+2q-2}) \\ &\leq (2r)^{r+2q} = a. \end{aligned}$$

Now we consider the following two cases based on u. The criterion is whether u satisfies a property which is reminiscent of the result of Lemma 17, except in Lemma 17 the property is for v while here we check it for u.

1. Suppose there exists $A \in \binom{X}{k}$ such that for every assignment f_A on A and every $S' \in \binom{X}{a'}$, there exists $S \in \binom{X}{a}$ satisfying $S \supseteq S' \cup A$ and $u(S)_{|A} \not\ni f_A$.
Given $a \geq k$ and $b \geq r \cdot b' + a$, we can apply Lemma 17 to know that there is an assignment f_A on A, such that for every $T' \in \binom{X}{b'}$, there is $T \in \binom{X}{b}$ satisfying $T \supseteq T' \cup A$ and $v(T)_{|A} \ni f_A$.
We therefore build multi-assignments u', v' for the smaller (a', b')-bipartite 2CSP instance as follows.
For every $S' \in \binom{X}{a'}$, we pick the set S guaranteed by the assumption of this case, and define $u'(S') = u(S)_{|S'}$. Note that u' still has size at most r.
For every $T' \in \binom{X}{b'}$, we pick the set T guaranteed by Lemma 17, and define $v'(T') = \{\sigma \in v(T) | \sigma_{|A} \neq f_A\}_{|T'}$. By the assumption, there exists $S \in \binom{X}{a}$ satisfying $S \supseteq A$ and $u(S)_{|A} \not\ni f_A$. Thus to be consistent with $u(S)_{|A}$, we can conclude for every $T \in \binom{X}{b}$ containing A, $v(T)_{|A}$ should also contain some value other than f_A. Therefore, our constructed v' is non-empty and has size at most $q - 1$.
It's also easy to see u', v' still satisfy list consistency, since in v' we only discard the assignments whose restriction on A equals to f_A, which are not consistent with any u'.
2. Suppose for every $A \in \binom{X}{k}$, there is an assignment f_A for A and a set $S' \in \binom{X}{a'}$, such that for every $S \in \binom{X}{a}$ satisfying $S \supseteq S' \cup A$, we have $u(S)_{|A} \ni f_A$.
In this case we can construct r-sized multi-assignment u' and 1-sized assignment v' for left and right part of the $(r, k = (2r)^r)$-bipartite direct product 2CSP of Π respectively, that meets the requirements of Lemma 18. Furthermore, u', v' are built purely based on u.
For every $B \in \binom{X}{r}$, we define
$$u'(B) = \bigcup_{A \in \binom{X}{k}, A \supseteq B} (f_A)_{|B}$$
For every $A \in \binom{X}{k}$, we define $v'(A) = f_A$.
We first claim the size of u' is at most r. Suppose it is not, there are $r+1$ sets $A_1, \ldots, A_{r+1} \in \binom{X}{k}$ with f_{A_i} all different. Let $S'_1, \ldots, S'_{r+1} \in \binom{X}{a'}$ be the corresponding sets guaranteed in the assumption of this case. Consider a $S \in \binom{X}{a}$ which contains $\cup_{i=1}^{r+1}(S'_i \cup A_i)$. Such S exists since $a \geq (r+1) \cdot (a' + k)$. Thus, $u(S)_{|A}$ contains $(r+1)$ different values, contradicting the fact that u is of size r.
It's easy to see u', v' meets the requirement of Lemma 18 by the definition of u'. Thus using Lemma 18, there is a satisfying assignment to the original instance Π. ◀

Our main result, Theorem 14, now follows immediately from Lemma 19.

Proof of Theorem 14. Given an integer r, we pick $t = (2r)^{r+2r+2}$, and prove that if a 2CSP instance $\Pi = (X, \Sigma, \Phi)$ is satisfiable, then so is $\Pi^{\odot t}$; if Π is not satisfiable, then $\Pi^{\odot t}$ is not r-list satisfiable.

The completeness case is easy: let σ be a satisfying assignment for Π, then we can assign each set $S \in \binom{X}{t}$ the function that maps any $x \in S$ to $\sigma(x)$.

For soundness case, suppose $\Pi^{\odot t}$ is r-list satisfiable by an assignment σ, we take $a = (2r)^{r+2q}, b = (2r)^{r+2q+2}$ and build multi-assignments u, v for the left and right part of the (a, b)-bipartite direct product 2CSP instance $\Pi^{\odot(a,b)}$. For each set $S' \in \binom{X}{a}$, we pick an arbitrary $S \in \binom{X}{t}$ with $S \supseteq S'$, and define $u(S') = \sigma(S)_{|S'}$. Similarly for each $T' \in \binom{X}{b}$, we pick an arbitrary $T \in \binom{X}{t}$ with $T \supseteq T'$, and define $v(T') = \sigma(T)_{|T'}$. It's easy to see u and v are r-list consistent. Thus by Lemma 19, Π is satisfiable. ◀

4 Average Baby PIH

Let us recall the average Baby PIH conjecture.

▶ **Hypothesis 20** (Average Baby PIH). *Given a 2CSP instance $\Pi = (X, \Sigma, \Phi)$, we say a multi-assignment σ r-average-list satisfies Π if $\sum_{x \in X} |\sigma(x)| \leq r \cdot |X|$, and for every constraint ϕ_i, there exists $u \in \sigma(w_{i,1})$ and $v \in \sigma(w_{i,2})$, such that $(u, v) \in R_i$.*

Average Baby PIH states that for any constant $r > 1$ and any computable function f, no algorithm can given as input a 2CSP instance $\Pi = (X, \Sigma, \Phi)$, distinguish between the following two cases in $f(|X|) \cdot |\Sigma|^{O(1)}$ time:

- *(Completeness) Π is satisfiable.*
- *(Soundness) Π is not r-average-list satisfiable.*

4.1 A Counter Example for Direct Product Construction

We use the following counter example to show that the Direct Product construction does not give us Average Baby PIH. Specifically, for any $t > 0$ and $\varepsilon > 0$, there exists an 2CSP instance which is not satisfiable but its t-wise direct product is $(1 + \varepsilon)$-average-list satisfiable.

▶ **Example 21.** The 2CSP instance $\Pi = (X, \Sigma, \Phi)$ is defined as follows.

- $X = \{x_1, \dots, x_n\}$.
- $\Sigma_{x_1} = \{2, \dots, n\}$, and for every $i \in \{2, \dots, n\}$, $\Sigma_{x_i} = \{1\}$.
- For every $i \in \{2, \dots, n\}$, there is a constraint $\phi_i = \langle w_i, R_i \rangle \in \Phi$, where
  - $w_i = (x_1, x_i)$.
  - $R_i = \{(j, 1) | j \neq i\}$.

Π is not satisfiable since no value for x_1 can satisfy all constraints. Specifically, for $i \in \{2, \dots, n\}$, $x_1 = i$ would violate constraint ϕ_i.

However, for every integer $t > 0$, $\Pi^{\odot t}$ can be list satisfied by the following multi-assignment σ. For every $S \in \binom{X}{t}$,

- if $x_1 \notin S$, define $\sigma(S)$ to be the single assignment that maps every $x_i \in S$ to 1;
- if $x_1 \in S$, then for every $2 \leq j \leq 2t$ with $x_j \notin S$, let $\sigma(S)$ contain an assignment that maps x_1 to j, and maps everything else to 1.

It's easy to see σ satisfies all constraints induced by S. It remains to show that σ is consistent on every different $S, T \in \binom{X}{t}$.

If $x_1 \notin S$ or $x_1 \notin T$, $\sigma(S)$ and $\sigma(T)$ are trivially consistent since every variable in $S \cap T$ is always assigned 1. Now suppose $x_1 \in S \cap T$, we have $|S \cup T| \leq 2t - 1$. By Pigeonhole Principle, there is a variable x_j with $j \leq 2t$, which is neither in S nor in T. Thus the assignment that maps x_1 to j and everything else to 1 appears in both $\sigma(S)$ and $\sigma(T)$, proving that σ list-satisfies $\Pi^{\odot t}$. The total size of the lists is

$$\begin{aligned}\frac{1}{\binom{|X|}{t}} \sum_{x \in X} |\sigma(x)| &\leq \frac{\binom{|X|-1}{t}}{\binom{|X|}{t}} \cdot 1 + \frac{\binom{|X|-1}{t-1}}{\binom{|X|}{t}} \cdot 2t \\ &= \left(1 - \frac{t}{|X|}\right) \cdot 1 + \frac{t}{|X|} \cdot 2t \\ &\leq 1 + \frac{2t^2}{|X|},\end{aligned}$$

which is smaller than $1 + \varepsilon$ when $|X|$ goes to infinity.

4.2 Towards the Inapproximability of k-ExactCover

In this subsection, we show that a slightly strengthened version of Average Baby PIH, namely, Average Baby PIH with rectangular relations, implies constant inapproximability of k-ExactCover problem. The latter, to our best knowledge, is currently not known under W[1] $\neq$ FPT. Formally, we have the following theorem:

▶ **Theorem 22.** *Suppose Average Baby PIH holds even when the 2CSP instance has rectangular relations (see Definition 6), then for any constant $r \geq 1$ and any function f, no algorithm can approximate k-ExactCover problem within factor r in $f(k) \cdot n^{O(1)}$ time. Specifically, no algorithm can given a k-SetCover instance $\Pi = (\mathcal{S}, U)$ with size n, distinguish between the following two cases in $f(k) \cdot n^{O(1)}$ time:*

- *There exists k sets in $\mathcal{S}$ which form a partition of U.*
- *The union of any $r \cdot k$ sets in $\mathcal{S}$ is not U.*

Proof. We reduce a 2CSP instance $\Pi = (X, \Sigma, \Phi)$ with rectangular relations to a k-SetCover instance $\Pi' = (\mathcal{S}, U)$ with $k = |X|$ in the following way. For every $(x, v) \in X \times \Sigma$, we build a set $S_{x,v}$. For each constraint $\phi_i = \langle (w_{i,1}, w_{i,2}), R_i \rangle$ in Φ, let Γ_i be the underlying set in the rectangular relation R_i and let $\pi_i, \sigma_i : \Sigma \to \Gamma_i$ be the underlying mappings. We build a $(r \cdot k, |\Gamma_i|)$-set gadget $(\mathcal{M}^{(i)}, C_1^{(i)}, \dots, C_m^{(i)})$. For every $(a, b) \in R_i$, we add the set $C_{\pi_i(a)}^{(i)}$ to $S_{w_{i,1},a}$, and add the set $\overline{C_{\sigma_i(b)}^{(i)}}$ to $S_{w_{i,2},b}$. Let the final universe U be the union of every $\mathcal{M}^{(i)}$.

In the completeness case, let $\sigma : X \to \Sigma$ be a satisfying assignment of Π. It is easy to see the k sets $\{S_{x,\sigma(x)}\}_{x \in X}$ cover each element of the universe exactly once.

In the soundness case, let $\mathcal{S}' \subseteq S$ be a collection of sets that covers U. Assuming $|\mathcal{S}'| \leq r \cdot k$, we claim the multi-assignment σ, which maps $x \in X$ to $\{v \in \Sigma \mid S_{x,v} \in \mathcal{S}'\}$, is a list satisfying assignment of Π. For every constraint $\phi_i = \langle (w_{i,1}, w_{i,2}), R_i \rangle$ with Γ_i being the image of the rectangular mapping, by the property of $(r \cdot k, |\Gamma_i|)$-set gadget and the fact that $|\mathcal{S}'| \leq r \cdot k$, $\mathcal{S}'$ must include $C_j^{(i)}$ and $\overline{C_j^{(i)}}$ for some $1 \leq j \leq |\Gamma_i|$, which implies that $\mathcal{S}'$ includes $S_{w_{i,1},a}$ and $S_{w_{i,2},b}$ for some $(a, b) \in R_i$, and thus ϕ_i is list satisfied.

From Lemma 13, a $(r \cdot k, |\Gamma_i|)$-set gadget can be constructed in time poly$(|\Gamma_i|, 2^{r \cdot k})$. The whole reduction runs in FPT time while preserving $k = |X|$. Thus, an $f(k) \cdot n^{O(1)}$ time algorithm for r-approximating k-ExactCover would give an $f(k) \cdot |\Sigma|^{O(1)}$ algorithm for the r-average-list satisfiability of 2CSP with rectangular relations, contradicting the strengthened version of Average Baby PIH. ◀

We remark that the direct product construction does have rectangular relations, although it does not directly give us Average Baby PIH, in view of Example 21.

5 Discussion and Open Problems

In this concluding section, we speculate on some possible avenues to attack Average Baby PIH or even PIH itself.

5.1 Average Baby PIH from Clique Hardness?

In [31], the author proved that even constant approximating k-Clique is W[1]-hard. In [24, 8], the inapproximability ratio was improved to $k^{o(1)}$. Using CSP words, their results can be described as the following theorem:

▶ **Theorem 23** ([24, 8]). *Assuming* W[1] $\neq$ FPT*, no algorithm can, given a 2CSP instance* $\Pi = (X, \Sigma, \Phi)$*, distinguish between the following two cases in* $f(|X|) \cdot |\Sigma|^{O(1)}$ *time, for any computable function* f*:*

- Π *is satisfiable.*
- *The constraints induced by any* $|X|^{1-o(1)}$ *variables are not simultaneously satisfiable.*

Thus, it is natural to ask whether we can get Average Baby PIH by applying direct product construction to a 2CSP instance with the above "clique-like" soundness? More formally, we have the following open question:

▶ **Open Question 1.** *Is it true that for any integer* $r > 1$*, there is an integer* $t > 0$*, such that for any 2CSP instance* $\Pi = (X, \Sigma, \Phi)$*, if* $\Pi^{\odot t}$ *is* r*-average-list satisfiable, then there are* $|X|^{1-o(1)}$ *variables in* Π *such all constraints amongst them are simultaneously satisfiable?*

Note that the counterexample for proving average baby PIH using the direct product construction (Example 21) does not apply here, since there are $|X| - 1$ variables in Π that are simultaneously satisfiable.

5.2 PIH via Direct Product Testing Theorems?

Our constructed instance in proving Baby PIH is reminiscent of the direct product testing, which has been studied in a recent line of work [18, 15, 14]. These results have shown that if the t-sized subsets of variables have good local consistency, then there is a global function which agrees with most of the subsets. Formally, we have the following theorem from [14].

▶ **Theorem 24.** *There is an absolute constant* $C > 1$*, such that for any* $\alpha, \beta \in (0, 1)$ *with* $\alpha + \beta \leq 1$*, there exists a constant* $Q(\alpha, \beta)$*, such that given any* k, t, m *with* $k \geq C \cdot t$ *and* $\alpha t \leq m \leq (1 - \beta)t$ *and any finite alphabet* Σ *, we have the following.*

Let $\mathcal{F} = \{f_S : S \to \Sigma \mid S \in \binom{[k]}{t}\}$ *be an ensemble of functions, one for every size-t subset of* $[k]$*. Let* $\mathcal{D}(m)$ *be the distribution as follows:*

- *Choose* $I \in \binom{[k]}{m}$ *uniformly at random.*
- *Choose* A, B *from the set* $\{X \mid X \in \binom{[k]}{t}, X \supseteq I\}$ *uniformly at random.*

Suppose the following holds:

$$\Pr_{A,B \sim \mathcal{D}(m)}[f_{A|A \cap B} = f_{B|A \cap B}] \geq 1 - \varepsilon,$$

then there exists a global function $g : [k] \to \Sigma$ *such that*

$$\Pr_{A \sim \binom{[k]}{t}}[f_A = g_{|A}] \geq 1 - Q(\alpha, \beta) \cdot \varepsilon.$$

By setting the alphabet of each f_S to be the set of all partial satisfying assignments for S and adding the consistency checks according to the above theorem, one may wonder whether we can "extract" a large clique from the $1 - Q(\alpha, \beta) \cdot \varepsilon$ fraction of subsets that are globally consistent. Formally, let $\mathcal{S} = \{A | A \in \binom{[k]}{t}, f_A = g_{|A}\}$ as in Theorem 24, can we prove there exists a subset $T \subseteq [k]$ of size $\geq k^{1-o(1)}$, such that the following holds?

- For every $(i, j) \in \binom{T}{2}$, there exists $A \in \mathcal{S}$ with $(i, j) \subseteq A$.

However, this might not be true if we only use the size bound on $\mathcal{S}$ and not the additional structures. Consider the following counter-example:

▶ **Example 25.** Mark each pair $(i, j) \in \binom{[k]}{2}$ as 1 independently with probability $1 - \gamma$ with γ to be determined. Take $\mathcal{S}'$ to be the collection of t-sized subsets of $[k]$, with all pairs in it marked as 1: $\mathcal{S}' := \{A \mid A \in \binom{[k]}{t}, \forall (i, j) \subseteq A, (i, j) \text{ is marked as } 1\}$.

The probability that a t-sized subset belongs to $\mathcal{S}'$ is $(1 - \gamma)^{\binom{t}{2}}$, which can be made arbitrarily close to 1 when we set $\gamma = \gamma(t)$ to be some small enough constant depending on the constant t.

However, it was known that (see e.g. [21]) the maximum clique size in this Erdős-Rényi graph is only $2 \log k / \log(1/(1 - \gamma))$, far smaller than $k^{1-o(1)}$.

This example suggests that, in order to potentially prove PIH from direct product testing theorems, one may need to analyze the structure of the collection $\mathcal{S}$. We leave this as an interesting future direction:

▶ **Open Question 2.** *Can we prove PIH under* $\mathsf{W}[1] \neq \mathsf{FPT}$ *using some appropriate form of direct product testing theorems?*

References

1 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM J. Comput.*, 49(4), 2020. doi:10.1137/18M1171321.

2 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

3 Sanjeev Arora and Carsten Lund. *Hardness of Approximations*, pages 399–446. In Dorit S. Hochbaum, editor, *Approximation algorithms for NP-hard problems.* PWS Publishing, 1996.

4 Libor Barto and Marcin Kozik. Combinatorial gap theorem and reductions between promise csps. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1204–1220. SIAM, 2022. doi:10.1137/1.9781611977073.50.

5 Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017. doi:10.1145/2981561.

6 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.74.

7 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002. doi:10.1006/jcss.2002.1882.

8 Yijia Chen, Yi Feng, Bundit Laekhanukit, and Yanlin Liu. Simple combinatorial construction of the $k^{o(1)}$-lower bound for approximating the parameterized k-clique. *CoRR*, abs/2304.07516, 2023. doi:10.48550/arXiv.2304.07516.

9 Yijia Chen and Martin Grohe. An isomorphism between subexponential and parameterized complexity theory. *SIAM Journal on Computing*, 37(4):1228–1258, 2007.

10 Yijia Chen and Bingkai Lin. The constant inapproximability of the parameterized dominating set problem. *SIAM J. Comput.*, 48(2):513–533, 2019. doi:10.1137/17M1127211.

11 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized approximation algorithms for directed steiner network problems. *CoRR*, abs/1707.06499, 2017. arXiv:1707.06499.

12 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 42:1–42:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.42.

13 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 637–646. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.14.

14 Irit Dinur, Yuval Filmus, and Prahladh Harsha. Analyzing boolean functions on the biased hypercube via higher-dimensional agreement tests: [extended abstract]. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2124–2133. SIAM, 2019. doi:10.1137/1.9781611975482.128.

15 Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 974–985. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.94.

16 Irit Dinur and Pasin Manurangsi. ETH-hardness of approximating 2-csps and directed steiner network. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 36:1–36:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.36.

17 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5):247–254, 2004. doi:10.1016/j.ipl.2003.11.007.

18 Irit Dinur and David Steurer. Direct product testing. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 188–196. IEEE Computer Society, 2014. doi:10.1109/CCC.2014.27.

19 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/a13060146.

20 Michael R Fellows. Blow-ups, win/win's, and crown rules: Some new directions in fpt. In *Graph-Theoretic Concepts in Computer Science: 29th International Workshop, WG 2003. Elspeet, The Netherlands, June 19-21, 2003. Revised Papers 29*, pages 1–12. Springer, 2003.

21 G. R. Grimmett and C. J. H. McDiarmid. On colouring random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 77(2):313–324, 1975. doi:10.1017/S0305004100051124.

22 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Parameterized inapproximability hypothesis under eth, 2023. arXiv:2311.16587.

23 Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004. doi:10.1016/j.comgeo.2004.03.003.

24 Karthik C. S. and Subhash Khot. Almost polynomial factor inapproximability for parameterized k-clique. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 6:1–6:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.6.

25 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019. doi:10.1145/3325116.

26 Karthik C. S. and Inbal Livni Navon. On hardness of approximation of parameterized set cover and label cover: Threshold graphs from error correcting codes. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 210–223. SIAM, 2021. doi:10.1137/1.9781611976496.24.

27 Euiwoong Lee. Partitioning a graph into small pieces with applications to path transversal. *Math. Program.*, 177(1-2):1–19, 2019. doi:10.1007/s10107-018-1255-7.

28 Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016. doi:10.1137/130938645.

29 Bingkai Lin. The parameterized complexity of the k-biclique problem. *J. ACM*, 65(5):34:1–34:23, 2018. doi:10.1145/3212622.

30 Bingkai Lin. A simple gap-producing reduction for the parameterized set cover problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 81:1–81:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.81.

31 Bingkai Lin. Constant approximating k-clique is w[1]-hard. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1749–1756. ACM, 2021. doi:10.1145/3406325.3451016.

32 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. On lower bounds of approximating parameterized k-clique. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 90:1–90:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.90.

33 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. Constant approximating parameterized k-setcover is w[2]-hard. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3305–3316. SIAM, 2023. doi:10.1137/1.9781611977554.ch126.

34 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. Improved hardness of approximating k-clique under ETH. *CoRR*, abs/2304.02943, 2023. doi:10.48550/arXiv.2304.02943.

35 Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Lossy kernelization. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 224–237. ACM, 2017. doi:10.1145/3055399.3055456.

36 Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2181–2200. SIAM, 2020. doi:10.1137/1.9781611975994.134.

37 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994. doi:10.1145/185675.306789.

38 Pasin Manurangsi. A note on max k-vertex cover: Faster fpt-as, smaller approximate kernel and improved approximation. In Jeremy T. Fineman and Michael Mitzenmacher, editors, *2nd Symposium on Simplicity in Algorithms, SOSA 2019, January 8-9, 2019, San Diego, CA, USA*, volume 69 of *OASIcs*, pages 15:1–15:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/OASIcs.SOSA.2019.15.

39 Pasin Manurangsi. Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 62–81. SIAM, 2020. doi:10.1137/1.9781611975994.5.

40 Dániel Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008. doi:10.1093/comjnl/bxm048.

41 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM J. Comput.*, 40(6):1871–1891, 2011. doi:10.1137/080734042.

42 Piotr Skowron and Piotr Faliszewski. Chamberlin-courant rule with approval ballots: Approximating the maxcover problem with bounded frequencies in FPT time. *J. Artif. Intell. Res.*, 60:687–716, 2017. doi:10.1613/jair.5628.

43 Andreas Wiese. Fixed-parameter approximation schemes for weighted flowtime. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 28:1–28:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.28.