PIH under ETH

Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, Kewen Wu

Presented by Matej Lieskovský

For a positive integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use log to denote the logarithm with base 2. Throughout the paper, we use $O(\cdot), \Theta(\cdot), \Omega(\cdot)$ to hide absolute constants that do not depend on any other parameter.

CSP. In this paper, we only focus on constraint satisfaction problems (CSPs) of arity two. Formally, a CSP instance G is a quadruple $(V, E, \Sigma, \{\Pi_e\}_{e \in E})$, where:

- V is for the set of variables.
- E is for the set of constraints. Each constraint $e = \{u_e, v_e\} \in E$ has arity 2 and is related to two distinct variables $u_e, v_e \in V$.

  The *constraint graph* is the undirected graph on vertices V and edges E. Note that we allow multiple constraints between a same pair of variables and thus the constraint graph may have parallel edges.
- Σ is for the alphabet of each variable in V. For convenience, we sometimes have different alphabets for different variables and we will view them as a subset of a grand alphabet Σ with some natural embedding.
- $\{\Pi_e\}_{e \in E}$ is the set of constraint validity functions. Given a constraint $e \in E$, the validity function $\Pi_e(\cdot, \cdot) \colon \Sigma \times \Sigma \to \{0, 1\}$ checks whether the constraint e between u_e and v_e is satisfied.

We use $|G| = (|V| + |E|) \cdot |\Sigma|$ to denote the *size* of a CSP instance G.

Assignment and Satisfiability Value. An *assignment* is a function $\sigma \colon V \to \Sigma$ that assigns each variable a value in the alphabet. The *satisfiability value* for an assignment σ, denoted by $\mathrm{val}(G, \sigma)$, is the fraction of constraints satisfied by σ, i.e., $\mathrm{val}(G, \sigma) = \frac{1}{|E|} \sum_{e \in E} \Pi_e(\sigma(u_e), \sigma(v_e))$. The satisfiability value for G, denoted by $\mathrm{val}(G)$, is the maximum satisfiability value among all assignments, i.e., $\mathrm{val}(G) = \max_{\sigma \colon V \to \Sigma} \mathrm{val}(G, \sigma)$. We say that an assignment σ is a *solution* to a CSP instance G if $\mathrm{val}(G, \sigma) = 1$, and G is *satisfiable* iff G has a solution.

When the context is clear, we omt σ in the description of a constraint, i.e., $\Pi_e(u_e, v_e)$ stands for $\Pi(\sigma(u_e), \sigma(v_e))$.

Parameterization and Fixed Parameter Tractability. For an instance G, the *parameterization* refers to attaching the parameter $k := |V|$ (the size of the variable set) to G and treating the input as a (G, k) pair. We think of k as a growing parameter that is much smaller than the instance size $n := |G|$. A promise problem $L_{\mathsf{yes}} \cup L_{\mathsf{no}}$ is *fixed parameter tractable (FPT)* if it has an algorithm which, for every instance G, decides whether $G \in L_{\mathsf{yes}}$ or $G \in L_{\mathsf{no}}$ in $f(k) \cdot n^{O(1)}$ time for some computable function f.

FPT Reduction. An *FPT reduction* from $L_{\sf yes} \cup L_{\sf no}$ to $L'_{\sf yes} \cup L'_{\sf no}$ is an algorithm $\mathcal{A}$ which, on every input $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ outputs another instance $G' = (V', E', \Sigma', \{\Pi'_e\}_{e \in E'})$ such that:

- COMPLETENESS. If $G \in L_{\sf yes}$, then $G' \in L'_{\sf yes}$.
- SOUNDNESS. If $G \in L_{\sf no}$, then $G' \in L'_{\sf no}$.
- FPT. There exist universal computable functions f and g such that $|V'| \leq g(|V|)$ and the runtime of $\mathcal{A}$ is bounded by $f(|V|) \cdot |G|^{O(1)}$.

ε-Gap k-CSP. We mainly focus on the gap version of the parameterized CSP problem. Formally, an ε-Gap k-CSP problem needs to decide whether a given CSP instance $(G, |V|)$ with $|V| = k$ satisfies $\mathrm{val}(G) = 1$ or $\mathrm{val}(G) < 1 - \varepsilon$. The exact version is equivalent to 0-GAP k-CSP.

Parameterized Inapproximability Hypothesis (PIH). *Parameterized Inapproximability Hypothesis (PIH)*, first formulated by Lokshtanov, Ramanujan, Saurabh, and Zehavi [1], is a central conjecture in the parameterized complexity theory, which, if true, serves as a parameterized counterpart of the celebrated PCP theorem. Below, we present a slight reformulation of PIH, asserting fixed parameter intractability (rather than $W[1]$-hardness specifically) of gap CSP.

Hypothesis 1 (PIH)**.** *For an absolute constant $0 < \varepsilon < 1$, no FPT algorithm can decide ε-Gap k-CSP.*

Exponential Time Hypothesis (ETH). *Exponential Time Hypothesis (ETH)*, first proposed by Impagliazzo and Paturi, is a famous strengthening of the $\mathsf{P} \neq \mathsf{NP}$ hypothesis and provides a foundation for fine-grained understandings in the modern complexity theory.

Definition 1 (3SAT)**.** A 3CNF formula φ on n Boolean variables is a conjunction of m clauses, where each clause is a disjunction of three literals and each literal is a variable or its negation. The goal of the 3SAT problem is to decide whether φ is satisfiable or not.

The original ETH is stated in the general 3SAT problem. In this paper, for convenience, we use the following variant due to the sparsification lemma and Tovy's reduction, which gives 3SAT additional structure.

Hypothesis 2 (ETH)**.** *No algorithm can decide* 3SAT *within runtime $2^{o(n)}$, where additionally each variable is contained in at most four clauses and each clause contains exactly three distinct variables.*[2]

[1] Prior to their work, this hypothesis was already informally stated by quite a few researchers as a natural formulation of the PCP theorem in parameterized complexity.

[2] We say a variable x is contained in a clause C if the literal x or $\neg x$ appears in C.