

# Open Problems: SPP Workshop on Advances in Theoretical Computer Science 2026

June 18, 2026

The following open problems were discussed in the workshop being organized on Advances in Theoretical Computer Science at the Department of CSE, IIT Kharagpur, to celebrate the significant contributions of Prof. Sudebkumar Prasant Pal on the occasion of his 65th birth year on June 10-11, 2026.

## 1 Learning Functions on the Cube (Prof. Arijit Ghosh)

**Can the Goldreich–Levin Query Complexity Be Improved?** Given query access to the truth table of a Boolean function  $f : \mathbb{F}_2^n \rightarrow \{-1, +1\}$ , a fundamental task in the analysis of Boolean functions is not only to estimate the largest Fourier coefficient

$$\text{FBias}(f) = \max_{\alpha \in \mathbb{F}_2^n} |\widehat{f}(\alpha)|,$$

but also to identify a character attaining (or approximately attaining) this maximum. While the value of  $\text{FBias}(f)$  can be estimated using only  $\tilde{O}(1/\tau^2)$  oracle queries, the best-known general procedure for recovering an index  $\alpha$  with

$$|\widehat{f}(\alpha)| \geq \text{FBias}(f) - \tau$$

relies on the Goldreich–Levin algorithm [GL89], whose query complexity is  $O(n/\tau^3)$ . This raises the following natural question.

**Open problem 1.** *Given oracle access to a Boolean function  $f : \mathbb{F}_2^n \rightarrow \{-1, 1\}$  and the promise that  $\text{FBias}(f) \geq \tau$ , can one find an  $\alpha \in \mathbb{F}_2^n$  satisfying*

$$|\widehat{f}(\alpha)| \geq \tau$$

*using only  $O(n/\tau^2)$  queries? More generally, can the additional  $1/\tau$  factor in the Goldreich–Levin complexity be removed while retaining polynomial dependence on  $n$ ?*

A positive answer would narrow the gap between estimating the magnitude of the largest Fourier coefficient and explicitly recovering a corresponding parity.

**Closing the gap for testing Fourier sparsity.** Given oracle access to a Boolean function  $f : \mathbb{F}_2^n \rightarrow \{-1, 1\}$ , the problem of testing whether  $f$  is  $s$ -Fourier sparse has received considerable attention in recent years. The best known upper bound is  $\tilde{O}(s^4)$  [GMR26], whereas the best known lower bound is  $\Omega(s)$ . The correct dependence on the sparsity parameter  $s$  remains unknown.

**Open problem 2.** *Determine the query complexity of testing whether a Boolean function  $f : \mathbb{F}_2^n \rightarrow \{-1, 1\}$  is  $s$ -Fourier sparse. In particular, is the optimal query complexity  $\Theta(s)$ ,  $\Theta(s^2)$ ,  $\Theta(s^4)$ , or something in between?*

Resolving this question would yield a tight characterization of the query complexity of testing Fourier sparsity. The current upper bound is based on implicit sensing and compressed sensing techniques, while the best known lower bound follows from communication complexity arguments, suggesting that substantial new ideas may be needed to close the gap.

**Dependence of distance parameter for sparsity testing.** A recent work [GMR26] established an  $\Omega(s)$  lower bound for testing Fourier sparsity. However, the lower bound is proved only for a fixed constant distance parameter  $\varepsilon$ .

**Open problem 3.** *Determine the query complexity of testing  $s$ -Fourier sparsity as a function of both  $s$  and  $\varepsilon$ . In particular, extend the current  $\Omega(s)$  lower bound [GMR26] to arbitrary  $\varepsilon > 0$ , and characterize the optimal dependence on the distance parameter.*

Understanding the role of  $\varepsilon$  would provide a more complete picture of the complexity of Fourier sparsity testing, and may require new lower-bound techniques beyond those currently known.

**Dimension-independent sparsity testing beyond the Boolean cube.** Dimension-independent testers are now known for Fourier sparsity of Boolean functions over  $\mathbb{F}_2^n$ . A natural question is whether similar phenomena occur in other function domains and representation systems. In particular, many applications involve sparse representations with respect to bases or dictionaries other than the Fourier basis on the Boolean cube.

**Open problem 4.** *Develop dimension-independent testers for sparsity with respect to other structured families of functions, such as wavelet bases, Fourier bases over  $\mathbb{Z}_N$ , or more general dictionaries. More broadly, identify properties of a basis or dictionary that permit query complexity depending primarily on the sparsity parameter rather than the ambient dimension.*

A successful resolution would substantially broaden the scope of sparsity testing beyond Boolean functions on  $\mathbb{F}_2^n$  and could lead to a unified theory of property testing for sparse representations. Such results may also reveal which aspects of existing Fourier sparsity testers are specific to the Boolean cube and which reflect more general principles of sparse recovery and compressed sensing.

## 2 Geometric Packing (Prof. Arindam Khan)

Minimum-volume container: Given a set of  $n$  convex polytopes, find the minimum-volume container that contains all polytopes (packing by translation only; rotations are not allowed). Can we even get some  $O(1)$ -approximation algorithm?

Dominating set of rectangles: In the geometric intersection graph of rectangles, find the minimum dominating set. Can we even get some  $\log \log n$ -approximation algorithm?

## 3 Problems on Planar Graphs, Disk Graphs, and Unit Interval Graphs (Prof. Bodhayan Roy)

1. A **bisection** of a graph  $G$  with an even number of vertices is a partition of  $V(G)$  into  $V_1$  and  $V_2$  such that  $|V_1| = |V_2|$ . The size of  $(V_1, V_2)$  is the number of edges with one endpoint in  $V_1$  and the other in  $V_2$ . In the **Minimum Bisection** problem, we try to find a bisection of minimum size for a given graph.

A **planar graph** is a graph which can be embedded in the plane such that no two edges intersect except at their endpoints.

**Can the Minimum Bisection problem be solved in polynomial time on planar graphs?**

2. Given a graph  $G(V, E)$ , a subset of vertices  $C \subseteq V$  such that every two distinct vertices of  $C$  are adjacent in  $G$ , is called a **clique** of  $G$ . The problem of finding a maximum sized clique in a graph is called the **Maximum Clique** problem.

A **disk graph** is an intersection graph of a set of disks in the Euclidean plane. Each vertex corresponds to a disk, and an edge exists between two vertices if and only if their corresponding disks intersect.

**Can the Maximum Clique problem be solved in polynomial time on disk graphs?**

3. Given a graph  $G(V, E)$  **Maximum Cut** problem asks for a partition of  $V$  into  $V_1$  and  $V_2$  such that the number of edges with with one endpoint in  $V_1$  and the other in  $V_2$  is maximized.

A **unit interval** graph is an intersection graph of unit length intervals on the real line. Each vertex corresponds to an interval, and an edge exists between two vertices if and only if their corresponding disks intersect.

**Can the Maximum Cut problem be solved in polynomial time on unit interval graphs?**

## 4 Many Sorting with Few Queries (Prof. Palash Dey)

Suppose we are given oracle access to  $n$  rankings (complete orders) over a set of  $m$  items: we can query the oracle with an integer  $i \in [n]$  and two items  $a$  and  $b$  and the oracle will let us know if  $a$  is ranked above  $b$  in the  $i$ -th ranking or  $b$  is ranked above  $a$ . We want to output these  $n$  rankings. Our goal is to minimize the total number of oracle queries. Because of the sorting lower bound, we know that  $\Theta(mn \log m)$  queries are both necessary and sufficient. The problem of preference elicitation studies tight bounds on the query complexity for this problem under various restrictions that real-world rankings often exhibit. One such restriction is the Euclidean domain. Here, the  $m$  items are points in  $\mathbb{R}^d$  for some integer  $d$  and, for every  $i \in [n]$ , there exists a point  $p_i$  in  $\mathbb{R}^d$  that explains the  $i$ -th ranking in the following sense. Item  $a$  is ranked above item  $b$  if and only if the point  $a$  is closer to  $p_i$  than the point  $b$ . If the items and the points lie in one-dimensional Euclidean space, i.e., in  $\mathbb{R}$ , then there is an easy algorithm (basically binary search) to know all these rankings in  $\Theta(n \log m)$  query complexity, and this is information theoretically tight. However, we do not know any interesting bound in  $\mathbb{R}^2$ . Can we get tight query complexity bounds in  $\mathbb{R}^2$  for this problem? Here are some references for the problem, although I am not sure if the techniques there will be of any use [Dey25, Con09].

### References

- [Con09] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries. *J. Artif. Intell. Res.*, 35:161–191, 2009.
- [Dey25] Palash Dey. Recognizing and eliciting weakly single crossing profiles on trees. *Theor. Comput. Sci.*, 1055:115487, 2025.
- [GL89] Oded Goldreich and Leonid A. Levin. A Hard-Core Predicate for All One-Way Functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (STOC)*, pages 25–32, 1989.
- [GMR26] Arijit Ghosh, Subhamoy Maitra, and Manmatha Roy. Testing Fourier Sparsity via Implicit Sensing. In *International Conference on Learning Representations (ICLR)*, 2026.