Recent Developments in Testing Bounded-Degree Graphs

Oded Goldreich

Weizmann Institute of Science

Although Dana's recent focus has not been on this model, much of the pioneering work was done by her.
This includes the introduction of the model, its initial study, and the celebrated tester for Bipartiteness, which relies on random walks.

These slides https://www.wisdom.weizmann.ac.il/~oded/T/bdg24.pptx

Outline

1. The bounded-degree graph testing model (G. and Ron, 1997).
2. Studies of (the fixed graph and two graph versions of) the **Graph Isomorphism** problem, including **determining the complexity of testing isomorphism to a fixed graph**, for almost all regular graphs.
3. Transporting hardness results about functions to hardness results about graphs by using **robustly self-ordered graphs**.
   E.g., a **separation between <u>tolerant</u> and standard testing** in the bounded-degree graph model (G. and Wigderson, 36th CCC, 2021).
4. The work of Adler, Kohler and Peng (32nd SODA and 36th CCC, 2021), which is pivoted at constructing **locally-characterized expander graphs.** (The construction makes inherent use of the iterative and local nature of the Zig-Zag construction.)
   This yields a **locally-characterizable graph property** that cannot be tested within a number of queries that does not depend on the size of the graph.

The Bounded-Degree Graph model

For a degree bound d, graphs are represented by incidence functions such that g(v,i) = w if w is the i^{th} neighbor of v.

On (explicit) input n and ε, a tester for property Π
is given **oracle access** to g, representing G, and should satisfy:

- If G is in Π, then the tester accepts w.p at least 2/3 (or 1 (“one-sided”)).
- If G is ε-far from Π, then the tester should reject w.p. at least 2/3, where G is **ε-far** from Π if for every G’$\in\Pi$ the symmetric difference between G and G’ is greater than εdn/2.

Focus: query complexity (as a function of n and ε).

Testing Graph Isomorphism: The two-input-graphs version

(The model has to be revised for two input oracles;
alternatively pack two oracles in one.)

THM: Testing requires query complexity $\tilde{\Omega}(n^{2/3})$, n = #vertices.

Proved by reduction to testing equality between (n/polylog n)-long sequences over [n], which in turn is closely related to testing equality between two distributions.

(The tested pairs of graphs consists of polylog-sized connected components.)

Open problem: What is the query complexity of this testing problem?

Testing Graph Isomorphism: The **fixed**-graph version [G. and Tauber]

(This is a massively parameterized property; for a **fixed** n-vertex graph H, the question is whether the input graph G is isomorphic to H.)

THM: Testing requires query complexity $\Omega(n^{1/2})$.
Furthermore, this holds for almost all d-regular n-vertex graphs H.

THM: For almost all d-regular n-vertex graphs H,
testing can be done in query complexity $\tilde{O}(n^{1/2})$.

Open problem: What about other graphs H?
E.g., other there graphs H for which the query complexity is higher?

Robustly self-ordered graphs and transporting results from functions to graphs [G. and Wigderson]

- A **self-ordered** graph = asymmetric. (Recall: Focus on bounded-degree.)
- G=(V,E) is **robustly** self-ordered if for every permutation π:V→V with **k** non-fixed-points, G and π(G) differ on $\mathbf{\Omega(k)}$ entries.
- G is **locally** self-ordered if given a vertex **v' in G'** isomorphic to G, we can find the location of **v'** in **G** using **polylog(|V|)** queries to **G'**.
- THM1 [GW]: There exists robustly and locally self-ordered graphs. Furthermore, they are strongly constructible.
- THM2 [GW]: If G is LSO and has polylog diameter, then given a vertex **v** in **G**, we can find its location in **G'** using **polylog(|V|)** queries to **G'**.

Robustly self-ordered graphs and **transporting results from functions to graphs** [G. and Wigderson]

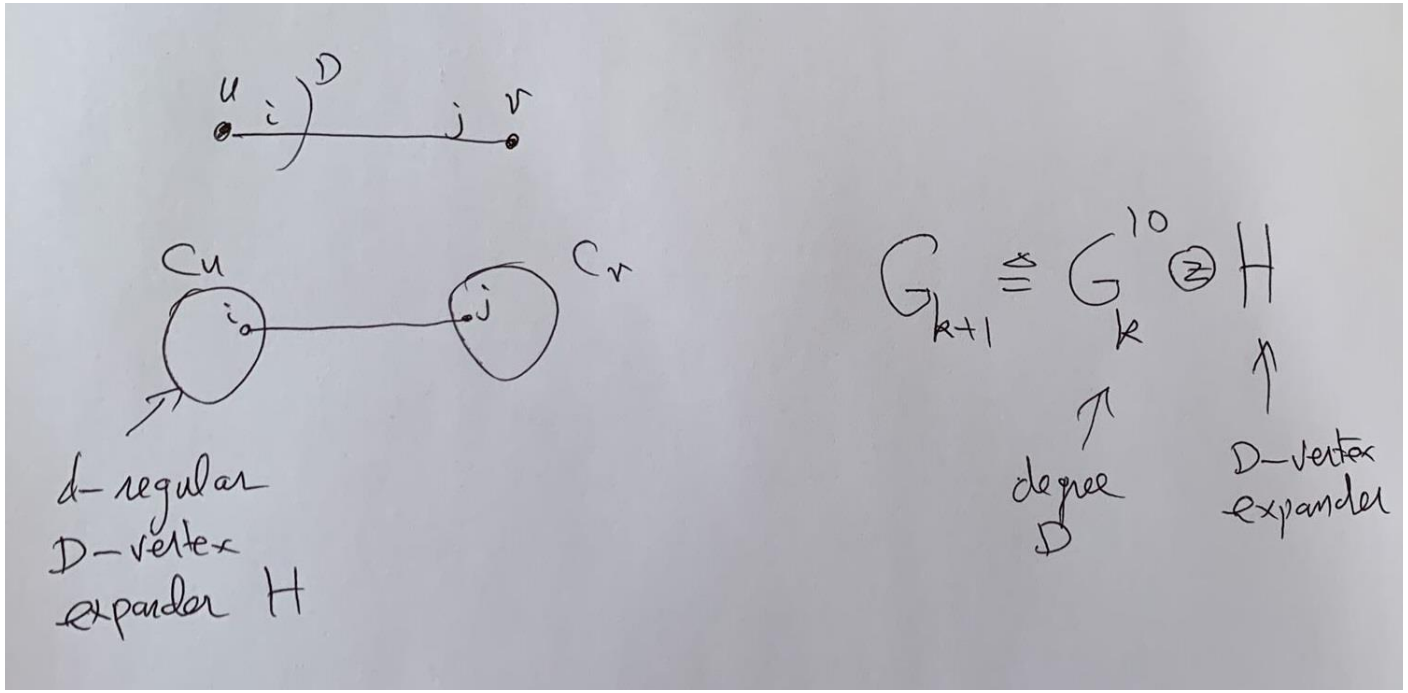
- THM3 [GW]: There exists a graph property that is easily testable but tolerantly testing it requires almost linear query complexity. (Recall: this is for the BDG model.)
- Proof idea: Start from an analogous result for properties of strings (obtained via PCPs by [FF]).
- Encode the bits of the (input) string by attaching tiny gadgets to a RSO and LSO graph (of Thm1). The RSO feature guarantees that the distance between strings is preserves (between the corres. graphs).
- Hardness of tolerant testing: Reduce (tolerantly) testing the string to (tolerantly) testing the graph, relying on the RSO feature.
- Easiness of standard testing: Reduce testing the graph to testing the corresponding string, using the LSO feature and Thm2.

Locally-characterized expander graphs [AKP]

- An n-vertex graph G is **locally-characterized** if it is uniquely determined (up to labeling) by a set of tuples of local conditions (i.e., all O(1)-neighborhoods should satisfy some prescribed condition)**.**
- E.g., a graph consisting of isolated triangles is locally characterized. In contrast, typical expanders are not; the O(1)-neighborhoods in typical expanders are regular trees.
- THM: See title. Furthermore, they are strongly constructible.
- Proof idea: Superimpose a full D-ary tree with the Zig-Zag construction (of [RVW]), where D is the ``cloud size''. **This construction is locally characterizable.**

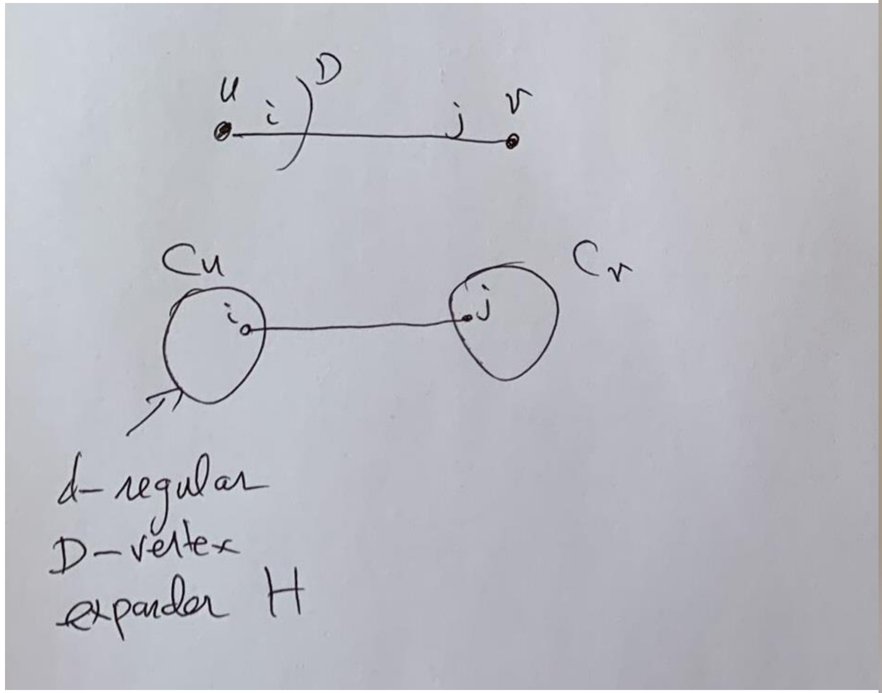
Details re locally-characterized expander graphs

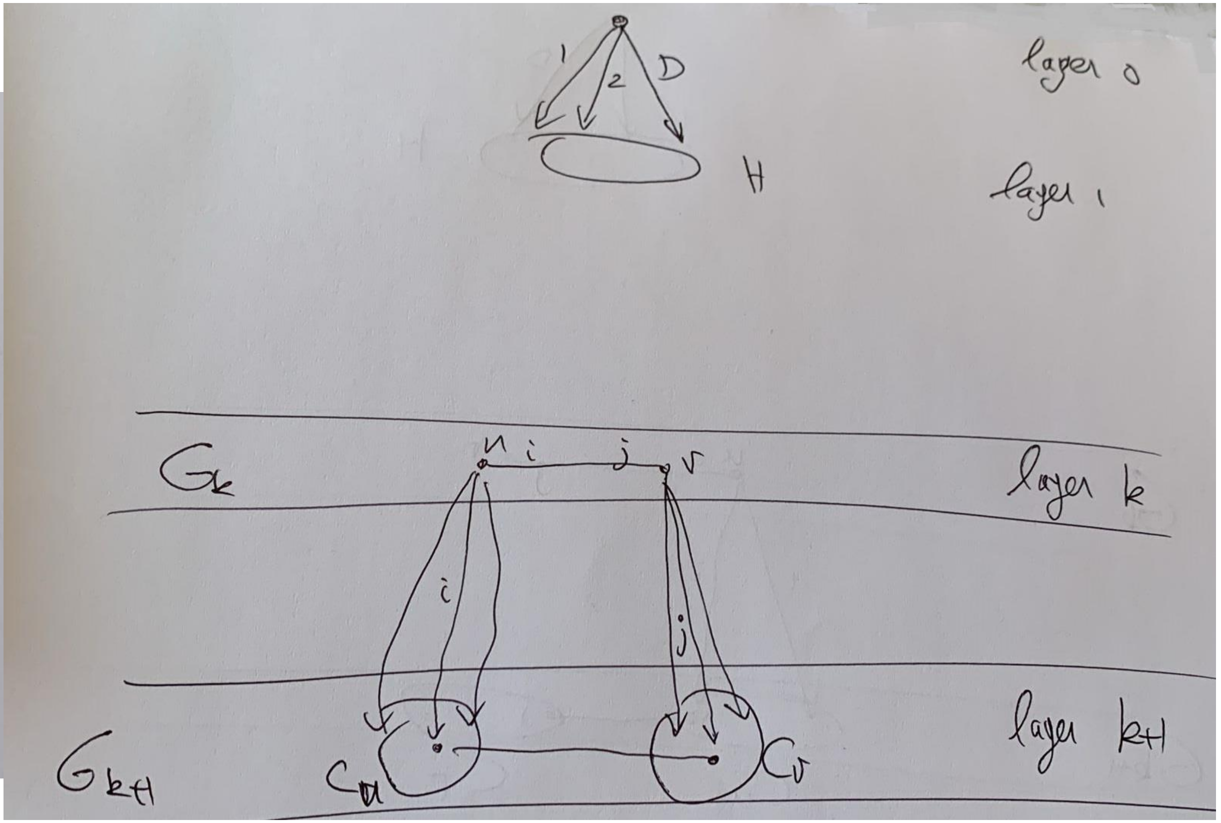
Proof idea: Superimpose a full D-ary tree with the Zig-Zag construction (of [RVW]), where D is the ``cloud size''.
This construction is locally characterizable.



Details re locally-characterized expander graphs

Proof idea: Superimpose a full D-ary tree with the Zig-Zag construction (of [RVW]), where D is the ``cloud size”.
This construction is locally characterizable.





Lower bound on the query complexity of some locally-characterized properties [AKP, FPS]

- A graph property is **locally-characterized** if it can be expressed a set of tuples of local conditions. This extends the notions of induced and non-induced subgraph freeness.
- E.g., the set of regular graphs is characterized by asserting that every two vertices have the same number of neighbors.
- THM: There exist locally characterized graph properties that are not **robustly** locally-characterized. Furthermore, they are not testable within **size-oblivious** query complexity.
  Specifically, their query complexity is at least logloglogloglog n.
- Proof idea: Use the locally characterized expander graphs.
  In contrast, by [FPS], every (infinite) property that is testable within size-oblivious query complexity contains a hyperfinite sub-property.
  A direct analysis gives a logloglogloglog bound.
  (Can get log in a relaxed model…)

References

- For the **two-graph** version of the **Graph Isomorphism**:
  Testing Isomorphism in the Bounded-Degree Graph Model, ECCC TR19-102, 2019.
- For the **fixed graph** version of the **Graph Isomorphism**:
  G. and Tauber, *On Testing Isomorphism to a Fixed Graph in the Bounded-Degree Graph Model*, ECCC TR23-146, 2023.
- For transporting results about functions to results about graphs by using **robustly self-ordered graphs:**
  G. and Wigderson, *Robustly Self-Ordered Graphs: Constructions and Applications to Property Testing*, in 36th CCC (2021) and TheoretiCS, Vol. 1, Art. 1, 2022.
- For constructing **locally-characterized expander graphs** and their use for lower bounds on the query complexity of testing a locally-characterizable graph property:
  * *On locally-characterized expander graphs (a survey)*, ECCC TR24-013, 2024.
  * *On the query complexity of testing local graph properties in the bounded-degree graph model*, ECCC TR24-047, 2024.

These slides https://www.wisdom.weizmann.ac.il/~oded/T/bdg24.pptx