On the Testability of Graph Partition Properties

Yonatan Nakar[1]
Tel Aviv University, Tel Aviv, Israel
yonatannakar@mail.tau.ac.il

Dana Ron[1]
Tel Aviv University, Tel Aviv, Israel
danaron@tau.ac.il

Abstract

In this work we study the testability of a family of graph partition properties that generalizes a family previously studied by Goldreich, Goldwasser, and Ron (*Journal of the ACM, 1998*). While the family studied by Goldreich, Goldwasser, and Ron includes a variety of natural properties, such as k-colorability and containing a large cut, it does not include other properties of interest, such as split graphs, and more generally (p, q)-colorable graphs. The generalization we consider allows us to impose constraints on the edge-densities within and between parts (relative to the sizes of the parts). We denote the family studied in this work by $\mathcal{GPP}$.

We first show that all properties in $\mathcal{GPP}$ have a testing algorithm whose query complexity is polynomial in $1/\epsilon$, where ϵ is the given proximity parameter (and there is no dependence on the size of the graph). As the testing algorithm has two-sided error, we next address the question of which properties in $\mathcal{GPP}$ can be tested with one-sided error and query complexity polynomial in $1/\epsilon$. We answer this question by establishing a characterization result. Namely, we define a subfamily $\mathcal{GPP}_{0,1}$ of $\mathcal{GPP}$ and show that a property $P \in \mathcal{GPP}$ is testable by a one-sided error algorithm that has query complexity $\mathrm{poly}(1/\epsilon)$ if and only if $P \in \mathcal{GPP}_{0,1}$.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Graph Partition Properties

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2018.53

Acknowledgements We thank the anonymous reviewers of this paper for their helpful comments.

1 Introduction

In graph property testing, the goal is to decide whether a graph satisfies a prespecified property P or is far from satisfying P. To this end, the testing algorithm is given query access to the adjacency matrix of the input graph so that the algorithm can check whether there is an edge between any given pair of vertices.[2] A graph G over n vertices is said to be ϵ-far from satisfying P if it is necessary to add or delete more than ϵn^2 edges in order to turn G into a graph satisfying P. A *tester* for a graph property P is a randomized algorithm, which given query access to the graph, distinguishes with high constant probability between the case where G satisfies P and the case where G is ϵ-far from satisfying P. The tester should make the distinction between the two cases by observing a very small portion of the input graph. In other words, the tester must have sublinear query complexity.

We focus on properties that can be tested with no dependence on n. In particular, the query complexity of the testers we consider depends only on the proximity parameter ϵ,

[1] This research was partially supported by the Israel Science Foundation grant No. 671/13.
[2] Here we refer to what is known as the Dense Graph Model or the Adjacency Matrix Model [12].

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018).
Editors: Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer; Article No. 53; pp. 53:1–53:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and the decisions of the testers do not depend on n as well. We call such graph properties *input-size oblivious testable.* Alon et al. [3] presented a complete characterization of input-size oblivious testable graph properties. Independently, Borgs et al.[9] obtained an analytic characterization of such properties through the theory of graph limits. However, while the query complexity of the tester emerging from the characterization of Alon et al. does not depend on the graph size, it could be super-polynomial in $\frac{1}{\epsilon}$. For example, the property of being triangle-free is input-size oblivious testable, but the query complexity of the best known tester for triangle-freeness is a tower function of $\frac{1}{\epsilon}$ [10]. Further, there exists a super-polynomial lower bound on the query complexity of testing triangle-freeness [1, 6]. Naturally, we strive to design testers with query complexity that is polynomial in $\frac{1}{\epsilon}$.

In this paper we consider a family of graph partition properties. This family of properties, which will be defined shortly, generalizes a family of graph partition properties that was introduced by Goldreich, Goldwasser, and Ron [12]. Examples of properties covered by their framework include bipartiteness, k-colorability, and the property of having a cut of at least βn^2 edges for some $\beta \in [0, 1]$. Their framework, while fairly general, lacks an ingredient that is necessary for specifying many natural graph partition properties such as split graphs (or more generally (p, q)-colorable graphs), probe complete graphs, and bisplit graphs.[3]

Given a graph $G = (V, E)$, a partition $(V_1, ..., V_k)$ of V, and a pair of parts V_i, V_j (possibly $i = j$) we denote by $e_G(V_i, V_j)$ the number of edges in G between the part V_i and part V_j (if $i = j$, then the notation refers to the number of edges within the part). Following the definitions in [12], the notation $e_G(V_i, V_j)$ counts each edge twice (both (u, v) and (v, u)) and when $i = j$ we also allow self-loops. That is, $e_G(V_i, V_j)$ counts the number of ones in the adjacency matrix representing the graph G. Also, we denote by $\overline{e}_G(V_i, V_j)$ the number of nonedges between the two parts.

Each property in the family of *Graph Partition Properties* considered in this work, is defined by an integer parameter k and $O(k^2)$ additional parameters in $[0, 1]$. Informally, a graph has the property if its vertices can be partitioned into k subsets such that the sizes of the subsets and the number of edges between pairs of subsets and within the subsets obey the constraints defined by the parameters of the property. Formally, a Graph Partition Property P is parameterized by an integer k denoting the number of parts and by the following parameters in the interval $[0, 1]$:

1. Bounds on each part's size: for each $1 \leq i \leq k$ we have ρ_i^L, ρ_i^U s.t. part V_i must satisfy $\rho_i^L n \leq |V_i| \leq \rho_i^U n$.
2. Absolute bounds on the number of edges within each part: for each $1 \leq i \leq k$ we have ρ_{ii}^L, ρ_{ii}^U s.t. part V_i must satisfy $\rho_{ii}^L n^2 \leq e_G(V_i, V_i) \leq \rho_{ii}^U n^2$.
3. Absolute bounds on the number of edges between each pair of parts: for each pair $1 \leq i, j \leq k$ we have ρ_{ij}^L, ρ_{ij}^U s.t. the pair of parts V_i, V_j must satisfy $\rho_{ij}^L n^2 \leq e_G(V_i, V_j) \leq \rho_{ij}^U n^2$.
4. Relative bounds on the number of edges within each part: for each $1 \leq i \leq k$ we have $\alpha_{ii}^L, \alpha_{ii}^U$ s.t. part V_i must satisfy $\alpha_{ii}^L |V_i|^2 \leq e_G(V_i, V_i) \leq \alpha_{ii}^U |V_i|^2$.
5. Relative bounds on the number of edges between each pair of parts: for each pair $1 \leq i, j \leq k$ we have $\alpha_{ij}^L, \alpha_{ij}^U$ s.t. the pair of parts V_i, V_j must satisfy $2\alpha_{ij}^L |V_i| \cdot |V_j| \leq e_G(V_i, V_j) \leq 2\alpha_{ij}^U |V_i| \cdot |V_j|$.

[3] A graph is a *split graph* if it can be partitioned into an independent set and a clique. A graph is (p, q)*-colorable* if it can be partitioned into p cliques and q independent sets. A graph is *probe-complete* if it can be partitioned into an independent set and a clique such that every vertex in the independent set is adjacent to every vertex in the clique. A graph is *bisplit* if it can be partitioned into an independent set and a bi-clique.

The original graph partition framework that was introduced in [12] includes only Items 1–3. The absence of relative edge bounds makes the original framework weaker than the general framework we consider in this paper. In particular, using the original framework, one cannot express the notion of parts being cliques or the notion of a pair of parts being fully connected to each other. More generally, our framework enhances the expressive power of the original framework by adding the notion of edge densities,[4] a notion that does not exist in the original framework. We denote the class of graph partition properties (as defined above) by $\mathcal{GPP}$. We denote the class of graph partition properties that have no relative bounds on the number of edges (the one introduced in [12]) by $\mathcal{GPP}_{NR}$ (NR stands for Non-Relative).

We say that a graph property is poly$\left(\frac{1}{\epsilon}\right)$-testable if it is input-size oblivious testable and the tester's query complexity is polynomial in $\frac{1}{\epsilon}$. All the properties in the class $\mathcal{GPP}_{NR}$ are poly$\left(\frac{1}{\epsilon}\right)$-testable [12]. In this work, we first show how to use the algorithm presented in [12] as a subroutine to devise a tester for all the partition properties covered by our generalized framework, thus obtaining the following theorem:

▶ **Theorem 1.** *Every property $P \in \mathcal{GPP}$ is poly$\left(\frac{1}{\epsilon}\right)$-testable.*

While the query complexity of the tester implied by 1 is a polynomial function of $\frac{1}{\epsilon}$ as desired, it has the disadvantage of having two-sided error (just like the algorithm described in [12]). A tester has one-sided error if, whenever a graph G satisfies P, the tester determines this with probability 1. Clearly, a one-sided error tester is preferable to a two-sided error tester because a one-sided error tester is capable of providing a witness demonstrating that the property is not satisfied by the input graph. Combining the two desired features of polynomial dependence on $\frac{1}{\epsilon}$ and having one-sided error leads to the definition of *easily testable graph properties* (as defined in e.g. [7, 4, 11]).

▶ **Definition 2.** *A graph property P is* easily-testable *if P is poly$\left(\frac{1}{\epsilon}\right)$-testable and the tester has one-sided error.*

An example of an easily testable graph partition property is the property of being k-colorable [12, 5]. In this paper we address the question of characterizing the easily testable graph partition properties. We show that every graph partition property belonging to a restricted subset of $\mathcal{GPP}$, which we denote by $\mathcal{GPP}_{0,1}$ (and formally define below), is easily testable, and every graph partition property $P \notin \mathcal{GPP}_{0,1}$ is not easily testable. That is, while Theorem 1 implies that every graph partition property P is poly$\left(\frac{1}{\epsilon}\right)$-testable, only those properties in $\mathcal{GPP}_{0,1}$ are poly$\left(\frac{1}{\epsilon}\right)$-testable with one-sided error. An analogous result was established for the class $\mathcal{GPP}_{NR}$ by Goldreich and Trevisan [13]. However, as the class $\mathcal{GPP}$ is more general, the class $\mathcal{GPP}_{0,1}$ contains properties that are not covered by the class $\mathcal{GPP}_{NR}$. We build on some techniques used in [13] to establish our characterization, but our characterization does not result from [13] and we rely on different ideas to arrive at it.

The class $\mathcal{GPP}_{0,1}$ is a subclass of $\mathcal{GPP}$ for which the following holds. For every property $P \in \mathcal{GPP}_{0,1}$, there are no absolute bounds on the number of edges between or within parts. If P has a constraint on the edge density between a pair of parts, or within a part, the constraint must be either that the edge density is exactly 0 or that the edge density is exactly 1. In addition, P does not constrain the sizes of the parts. Formally, P is parameterized by an integer k denoting the number of parts and by a function $d_P : [k] \times [k] \to \{0, 1, \perp\}$ denoting the relative edge density that P imposes on parts i and j:

[4] From now on, when using the term edge density, we refer to the fraction of edges between the parts (or within the part) relative to the number of vertex pairs between the parts (or within the part).

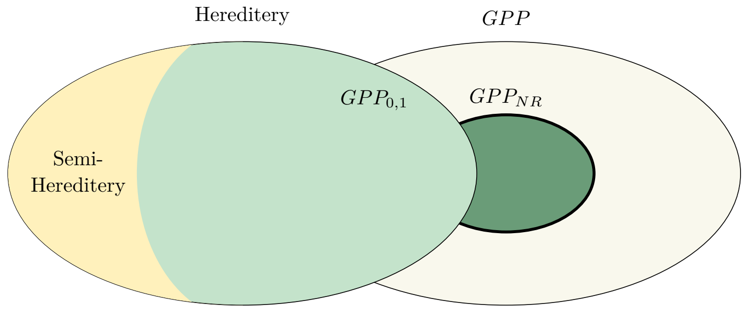

Figure 1 Inclusion relations among the graph partition properties.

$$d_P(i,j) = \begin{cases} 1 & \text{if every vertex in part } i \text{ should be connected to every vertex of part } j \\ 0 & \text{if there are no edges between part } i \text{ and part } j \\ \perp & \text{if any number of edges between part } i \text{ and } j \text{ is allowed} \end{cases}$$

Possibly, $i = j$ in which case it is the edge density within a single part. That is, if $d_P(i,i)$ is 0 or 1 then P forces part i to be an independent set or a clique respectively. It is clear from the definition that there are graph partition properties $P \in \mathcal{GPP}_{0,1}$ that are not part of the class $\mathcal{GPP}_{NR}$ (split graphs for instance).

The main result of our paper is a characterization of the easily testable graph partition properties.

▶ **Theorem 3.** *A graph property $P \in \mathcal{GPP}$ is easily testable if and only if $P \in \mathcal{GPP}_{0,1}$.*

Recall that a property is easily testable if it is testable by a one-sided error input-size oblivious tester whose query complexity is polynomial in $\frac{1}{\epsilon}$. If we remove the requirement that the dependence on $\frac{1}{\epsilon}$ is polynomial, then the property is said to be *strongly testable*. Alon and Shapira [8] define the notion of a property being *semi-hereditary* (which is a certain relaxation of being hereditary), and show that a graph property P is strongly testable if and only if P is semi-hereditary. Since the properties in $\mathcal{GPP}_{0,1}$ are clearly hereditary, and therefore semi-hereditary, the condition of Alon and Shapira implies that they are strongly testable. However, this is not enough to prove the "if" part of Theorem 3, because being strongly testable does not mean that the tester's query complexity is $\text{poly}\left(\frac{1}{\epsilon}\right)$. Therefore, to prove the "if" direction we give a $\text{poly}\left(\frac{1}{\epsilon}\right)$ one-sided error testing algorithm for the property. As for the "only if" direction, we could use [8] to get that if a property P in $\mathcal{GPP}$ is easily testable (and hence strongly testable), then it is semi-hereditary. We would then need to prove that if $P \in \mathcal{GPP}$ is semi-hereditary, then $P \in \mathcal{GPP}_{0,1}$. Establishing this claim would be essentially the same as our direct proof that if a property P in $\mathcal{GPP}$ is easily testable, then $P \in \mathcal{GPP}_{0,1}$, and would be based on the same proof ingredients.

We next give a brief summary of each of our results.

1.1 A Two-Sided Error Tester for properties in $\mathcal{GPP}$

In order to prove the existence of a (two-sided error) $\text{poly}\left(\frac{1}{\epsilon}\right)$-testing algorithm for $\mathcal{GPP}$ we show how to reduce the problem of testing properties in $\mathcal{GPP}$ to testing properties in $\mathcal{GPP}_{NR}$. Recall that the difference between properties in $\mathcal{GPP}$ and properties in $\mathcal{GPP}_{NR}$ is that the former include edge density constraints (that are relative to the sizes of the parts), while the latter include only absolute constraints on the sizes of the parts and the number of edges between/within them. We next give the high-level idea of the reduction.

Given a property $P \in \mathcal{GPP}$, we define a collection of properties in $\mathcal{GPP}_{NR}$, by *discretizing* P and replacing the edge-density constraints with absolute constraints on the number of edges. We then run the testing algorithm of [12], denoted $\mathcal{A}$, on G and each property in the constructed collection, with distance parameter $\frac{\epsilon}{2}$. If $\mathcal{A}$ accepts for at least one of these properties, then we accept, and otherwise we reject. The definition of the collection is such that if G satisfies P, then G satisfies at least one of the properties in the collection, so that our algorithm accepts with high constant probability. In order to show that if G is ϵ-far from P, then G is $\frac{\epsilon}{2}$-far from *every* property in the collection, we prove the contrapositive statement. That is, if for at least one of the properties P' in the collection, G is $\frac{\epsilon}{2}$-close to P', then G is $\frac{\epsilon}{2}$-close to P. While the first part of the analysis (regarding G that satisfies P) is fairly immediate, the second part (regarding G that is $\frac{\epsilon}{2}$-far from P) requires a more subtle analysis. Essentially, we need to show how to "fix" G (remove/add edges), so as to obtain a graph that satisfies P. This requires showing the existence of a partition $(V_1, \dots, V_k)$ that obeys all the constraints defined by P, while closeness to P' only ensures the existence of a partition $(V'_1, \dots, V'_k)$ that "almost" satisfies P'.

1.2 A One-Sided Error Tester for Properties in $\mathcal{GPP}_{0,1}$

The tester is very simple. It samples $\Theta\left(\frac{k\log(k)}{\epsilon^2}\right)$ vertices, checks whether or not the induced subgraph satisfies P and answers accordingly.[5] Since all the graph partition properties in $\mathcal{GPP}_{0,1}$ are hereditary, it clearly holds that if a graph G satisfies P, then every induced subgraph of G also does. Hence, if $G \in P$, the suggested tester accepts with a probability of 1.

The heart of the proof is in showing that if G is ϵ-far from satisfying P, where $P \in \mathcal{GPP}_{0,1}$, then with high constant probability, the subgraph induced by the sample does not satisfy P. In other words, we would like to show that with high constant probability over the choice of the sample S, every partition $(S_1, \dots, S_k)$ of the sample violates at least one of the constraints defined by the property P. That is, there is a pair (u, v), where $u \in S_i$ and $v \in S_j$ such that either $(u, v) \in E$ while $d_P(i, j) = 0$, or $(u, v) \notin E$ while $d_P(i, j) = 1$. Such a partition is said to be *invalid*. In order to prove this claim we extend the analysis of Alon and Krivelevich [5] for testing k-colorability. We next give a high-level description of the analysis.

Given the sample S, we construct a k-ary tree. Each node in the tree corresponds to a partial partition of the sample. That is, a partition of a subset of the sample. In particular, each internal node corresponds to a *valid* partition (where the root corresponds to a trivial partition of the empty set). If an internal node corresponds to a partition $(S'_1, \dots, S'_k)$ of a subset S' of the sample, then its children correspond to all partitions of $S' \cup \{u\}$ that extend the partition $(S'_1, \dots, S'_k)$ for some sample point $u \in S \setminus S'$. That is, partitions of the form $(S'_1, \dots, S'_{i-1}, S'_i \cup \{u\}, S'_{i+1}, \dots, S'_k)$. Observe that if we obtain a tree for which all leaves correspond to invalid partitions (i.e., that violates some constraint of P), then there is no valid partition of S.

Consider a node x in the tree, corresponding to a partition $(S'_1, \dots, S'_k)$ of $S' \subset S$. For each vertex $v \notin S'$, let $0 \le a_x(v) \le k$ be the number of parts in the partition to which u can be added so that the resulting partition is valid, and let a_x be the sum of $a_x(v)$ taken over all $v \notin S'$. Observe that for the root of the tree, r (which corresponds to $S' = \emptyset$), $a_r = k \cdot n$, and

[5] If the tested graph partition property is NP-hard to decide, then the running time is super-polynomial in the sample size, which is unavoidable assuming $P \neq NP$.

if y is a child of x, then $a_y \leq a_x$. If the partition corresponding to y is invalid, then $a_y = 0$. We show that with high constant probability over the choice of the sample, we can construct a tree for which the following holds. For every path in the tree, the value of a_x decreases in a relatively significant manner when comparing each node to its children. This allows us to show that we can obtain a tree in which all partitions corresponding to the leaves are invalid.

1.3 Easily Testable Graph Partition Properties Must be in $\mathcal{GPP}_{0,1}$

Our proof that if a property $P \in \mathcal{GPP}$ is easily testable, then P must be in $\mathcal{GPP}_{0,1}$ is the most technically involved part of this work. The proof consists of several steps, and we next give a high-level outline of these steps. We note that the proof uses the fact that easy testability implies strong testability. That is, we rely on the existence of a one-sided error tester for the property that is oblivious of the size of the graph, but we do not rely on the tester having complexity $\text{poly}\left(\frac{1}{\epsilon}\right)$.

Recall that properties in $\mathcal{GPP}_{0,1}$ are defined by the following types of constraints over graph partitions. First, for each part, either the edge density within the part is unconstrained, or it is constrained in an extreme manner. The latter means that no edges are allowed within the part, or that there must be all possible edges. We say in such a case that the part is *homogeneous*. Similarly, for each pair of parts, either there is no constraint on the edge density between the parts, or it is extreme (no edges, or all edges). Here too we say in the latter case that the pair is homogeneous. Finally, as opposed to $\mathcal{GPP}$, there are no constraints on the sizes of the parts. Observe that the trivial property, that is, the property that contains all graphs, belongs to $\mathcal{GPP}_{0,1}$ (since it can be defined by a single part with no edge-density constraints).

In what follows, for a property $P \in \mathcal{GPP}$ and a graph $G = (V, E)$ satisfying P, a partition $(V_1, \ldots, V_k)$ of V is said to be a *witness partition* with respect to P, if in G, $(V_1, \ldots, V_k)$ satisfies the constraints imposed by P. We say in such a case that the pair $(G, (V_1, \ldots, V_k))$ satisfies P. We first prove that if P is easily testable, then either it is trivial, or for every graph G satisfying P and witness partition $(V_1, \ldots, V_k)$, all parts are homogeneous. This is established by showing that if there exists a graph G in P with a witness partition that has some non-homogeneous part, then the premise that P is easily testable implies that P is trivial. The proof uses a type of "multiplying" operation on the graph G. Once we have only homogeneous parts, we can also establish the homogeneity of pairs (among those that are constrained in terms of edge-density). At this point it remains to show that there can be no size constraints on the parts.

To this end we prove a dichotomy claim. Let P' be the same property as P except that there are no size constraints. The claim is that either $P = P'$ or P' is in a certain sense far from P. We then show that the second case cannot hold if P is easily testable. In order to prove the dichotomy claim, we define a certain mathematical program, that, roughly speaking, is related to modifications of graph-partition pairs that satisfy P' to graph-partition pairs that satisfy P (by "fixing" the size constraints). In particular, the existence of a feasible solution corresponds to $P = P'$. On the other hand, if there is no feasible solution, then we show that P' is far from P. This proof involves a probabilistic construction of a graph that satisfies P but is sufficiently far from satisfying P'.

1.4 Related Work

Easily testable graph properties

Besides the class of graph partition properties, there are several results characterizing the set of easily testable graph properties among other classes. Alon [1] proved that the property of being H-free is easily testable if and only if H is bipartite. Alon and Shapira [7] proved that for any graph H besides P_2, P_3, P_4, C_4 and their complements, the property of being induced H-free is not easily testable. It was also shown in [7, 4] that induced H-freeness is easily-testable for P_2, P_3, P_4 and their complements, and the case of C_4 (and its complement) is the only one that remains open. In addition, the graph properties perfectness and comparability were shown to be not easily testable [4]. Gishboliner and Shapira [11] recently made significant progress by providing sufficient and necessary conditions for guaranteeing that a hereditary graph property is easily testable, implying all the positive and negative results mentioned above. It is worth noting, however, that their criteria do not apply to many properties in $\mathcal{GPP}_{0,1}$ (for example, (p, q)-colorability), that are shown to be easily testable in our work.

Testing properties in $\mathcal{GPP}_{NR}$ with one sided error

As mentioned previously, Goldreich and Trevisan [13] studied the one-sided error testability of $\mathcal{GPP}_{NR}$. They showed that every strongly testable property in $\mathcal{GPP}_{NR}$ belongs to a class of properties that generalizes k-colorability. Each property P in this class is defined by a set of pairs $A_P = \{(i, j)\,|0 \le i, j \le k\}$, where the property P is the set of k-colorable graphs with the additional constraint that if $(i, j) \in A_P$, then there are no edges between the vertices with color i and the vertices with color j. In addition, the property of being a clique and the property containing all graphs are both stongly testable graph properties in $\mathcal{GPP}_{NR}$.

We build on [13]'s technique of multiplying a graph-partition pair to derive the fact that all the easily testable properties in $\mathcal{GPP}$ only have homogeneous constraints on the edge density within and between parts. The idea of finding assignments to variables corresponding to moving vertices between parts also appears in [13], but they did not have to optimize over a mathematical program, and the assignments they defined could be used straightforwardly to establish the equivalence between a property and its relaxation. One of the main ideas used in [13] to derive the characterization was showing that strongly testable properties in $\mathcal{GPP}_{NR}$ are closed under removal of edges (except for the property of being a clique), and they rely on this fact heavily when deriving the implication regarding the assignments they define and when performing the multiplication. We could not use this idea as it does not hold in our case, because the existence of relative edge bounds in $\mathcal{GPP}$ enables easily testable properties to have lower bounds on the number of edges between or within parts. This is basically the main reason why the set of easily testable properties in $\mathcal{GPP}$ has a richer structure than in $\mathcal{GPP}_{NR}$. This is why we had to use notions that do not appear in their analysis such as weak and strong violations of assignments and rely on the probabilistic method to establish our result.

Organization

In Section 2 we expand on the result regarding the "only if" direction of Theorem 3, that all the easily testable properties in $\mathcal{GPP}$ are in $\mathcal{GPP}_{0,1}$. All the details regarding the other two results can be found in the accompanying full version of the paper, as well as the proofs of all claims stated in Section 2.

2 Easily Testable Graph Partition Properties Must be in $\mathcal{GPP}_{0,1}$

As stated in Theorem 3, a property $P \in \mathcal{GPP}$ is easily testable if and only if $P \in \mathcal{GPP}_{0,1}$. In this section we provide the proof structure and some of the proof details for the claim that if a graph partition property is easily testable, then $P \in \mathcal{GPP}_{0,1}$. We first define the concept of a t-multiplier, which is used several times throughout the proof. In what follows, given a graph $G = (V, E)$ and a set of vertices $U \subseteq V$ we denote by $G[U]$ the subgraph induced by U.

2.1 t-Multipliers

▶ **Definition 4.** Let $G = (V, E)$ be a graph over n vertices and let $(V_1, ..., V_k)$ be a partition of V. For an integer t we say that a graph-partition pair $(G', (V_1', ..., V_k'))$ is a ***t*-multiplier** of $(G, (V_1, ..., V_k))$, if the following holds (where $G' = (V', E')$).
Vertices: $|V'| = t \cdot n$.
Partition: For each $1 \le i \le k$, $|V_i'| = t \cdot |V_i|$.
Within Edges: Suppose $G[V_i]$ has $\alpha_{ii} |V_i|^2$ edges. Then $G'[V_i']$ has $\alpha_{ii} t^2 |V_i|^2$ edges.
Between Edges: Suppose G has $2\alpha_{ij} |V_i| \cdot |V_j|$ edges between V_i and V_j. Then G' has $2\alpha_{ij} \cdot t^2 |V_i| \cdot |V_j|$ edges between V_i' and V_j'.

Recall that given a graph $G = (V, E)$ and a partition $(V_1, ..., V_k)$ of V that satisfies the constraints imposed by property P in $\mathcal{GPP}$, we say that $(V_1, ..., V_k)$ is a *witness* partition to the fact that G satisfies P. In short, we say that the graph-partition pair $(G, (V_1, ..., V_k))$ *satisfies* P. The next claim trivially holds.

▶ **Claim 5.** *If $(G, (V_1, ..., V_k))$ is a graph-partition pair satisfying P and $(G', (V_1', ..., V_k'))$ is a t-multiplier of $(G, (V_1, ..., V_k))$, then the pair $(G', (V_1', ..., V_k'))$ also satisfies P.*

2.2 Easily Testable Graph Partition Properties are Homogeneous

Let P be a graph partition property. We note that even if there are no explicit bounds on a part's size or on the edge density within a part or between a pair of parts, such constraints may be implicitly induced by the combination of other constraints. This leads to the following definition.

▶ **Definition 6.** Given a pair of integers $(i, j) \in [k] \times [k]$, we say that P has **no constraints on the edge density between the parts** (i, j) if for every graph-partition pair $(G, (V_1, ..., V_k))$ satisfying P, any graph G', obtained from G by performing arbitrary vertex-pair modifications between V_i and V_j in G, satisfies P and $(V_1, ..., V_k)$ serves as a witness partition. Otherwise, we say that P **constrains the edge density between parts** (i, j). For the special case where $i = j$ we say that P **constrains the edge density within part** i.

We say that a graph is *homogeneous* if it is either an independent set or a clique. We can classify the properties in $\mathcal{GPP}$ into two sets, corresponding to the following complementary two cases.

Case (a): There exists a graph-partition pair $(G, (V_1, ..., V_k))$ satisfying P and an integer $1 \le i \le k$ such that $G[V_i]$ is non-homogeneous.

Case (b): For every graph-partition pair $(G, (V_1, ..., V_k))$ that satisfies P it holds that $G[V_i]$ is homogeneous for every $1 \le i \le k$.

We note that Case (b) can be shown to imply a stronger statement. If Case (b) holds, then not only every part is homogeneous, but rather the following holds: For each $1 \le i \le k$, either for every graph-partition pair $(G,(V_1,...,V_k))$ satisfying the property, $G[V_i]$ is an independent set, or for every graph-partition pair $(G,(V_1,...,V_k))$ satisfying the property, $G[V_i]$ is a clique. We next establish the following implication of Case (a).

▶ **Claim 7.** *Let P be an easily testable graph partition property for which Case (a) holds. Then for every proximity parameter ϵ, every graph is ϵ-close to satisfying P.*

Proof. Since Case (a) holds, there exists a graph-partition pair $(G,(V_1,...,V_k))$ satisfying P and an integer $1 \le i \le k$ such that $G[V_i]$ is non-homogeneous. Suppose by way of contradiction that there exists a one-sided error tester for P that is input-size oblivious. Denote the tester by $\mathcal{T}$. By [2, 13], we can assume without loss of generality that the algorithm $\mathcal{T}$ makes its decision based on an inspection of the subgraph induced by a random sample of s_ϵ vertices chosen independently and uniformly at random, where s_ϵ is a function of ϵ and is independent of n.

By Claim 5, for every $(G',(V'_1,...,V'_k))$ that is an s_ϵ-multiplier of $(G,(V_1,...,V_k))$, we have that G' satisfies P (with the witness partition $(V'_1,...,V'_k)$). Therefore, $\mathcal{T}$ must accept each such G' with probability 1. We next show that for every graph H over at most s_ϵ vertices, there exists at least one such graph G' for which $G'[V'_i]$ contains H as an induced subgraph. The claim will then follow since the tester must accept given any induced subgraph that it observes, implying that it accepts all graphs with probability 1.

Let $|V_i| = n_i$ and let $v_i^1,\dots,v_i^{s_\epsilon n_i}$ denote the vertices in V'_i. Observe that since $G[V_i]$ has at least one edge and at least one non-edge, for every $(G',(V'_1,...,V'_k))$ that is an s_ϵ-multiplier of $(G,(V_1,...,V_k))$, it holds that $G'[V'_i]$ has $m'_i \ge s_\epsilon^2$ edges and $(t\cdot n_i)^2 - m'_i \ge s_\epsilon^2$ non-edges. Let H be some fixed graph over $s \le s_\epsilon$ vertices with m_H edges (and $s^2 - m_H$ non-edges). Since $m_H \le s_\epsilon^2 \le m'_i$ and $s^2 - m_H \le s_\epsilon^2 \le (t\cdot n_i)^2 - m'_i$, the definition of an s_ϵ-multiplier allows to let the subgraph of $G'[V'_i]$ induced by the vertices $v_i^1,\dots,v_i^s$ be H.

That is, the tester $\mathcal{T}$ accepts every graph with probability 1, and hence, for every ϵ, every graph is ϵ-close to satisfying P. ◀

We emphasize that Claim 7 holds for every graph and not only for sufficiently large graphs. It follows that if P is an easily testable graph partition property that satisfies Case (a) then P is in fact the trivial graph partition property that contains all graphs. This property clearly belongs to $\mathcal{GPP}_{0,1}$. Hence, from now on we can assume that Case (b) holds. In other words, we consider properties $P \in \mathcal{GPP}$ for which every graph satisfying P can only be validly partitioned in such a way that every part of the partition is homogeneous. That is, if G is a graph satisfying P and $(V_1,...,V_k)$ is a witness partition, then for every pair $(i,j) \in [k]\times[k]$, the subgraph $G[V_i \cup V_j]$ is either a split graph, or a bipartite graph or a cobipartite graph. We can use this fact together with an application of an appropriate multiplier to establish the following claim.

▶ **Claim 8.** *Let P be an easily testable property in $\mathcal{GPP}$. If there exists a graph-partition pair $(G,(V_1,...,V_k))$ that satisfies P such that the edge density between a pair of parts is neither 0 nor 1, then P has no constraints on the edge density between the two parts.*

In this subsection we showed that if a graph partition property P is easily testable and P constrains the edge density within a particular part, then the part must be homogeneous. (To be precise, P either forces the part to be an independent set or it forces it to be a clique.) Similarly, if P constrains the edge density between a pair of parts then it either forces the edge density within the pair to be 0 or it forces it to be 1. We call such properties *homogeneous graph partition properties.* That is, a homogeneous graph partition property is defined similarly to a property in $\mathcal{GPP}_{0,1}$ except that unlike $\mathcal{GPP}_{0,1}$, a homogeneous graph partition property possibly has size constraints on the parts. In the next subsection we show that if such a property is easily testable, then it has no size constraints.

2.3 No Constraints on the Sizes of Parts

Suppose P is a homogeneous property in $\mathcal{GPP}$ that possibly has size constraints in its specification and is easily testable. We show that if this is the case, then there exists an equivalent formulation of P that has no size constraints. Namely, we show that the simple relaxation of P whose specification contains the same homogeneity constraints as those specified by P but excludes its size constraints, is equivalent to P (under the assumption that P is easily testable). From now on, given a graph partition property P, we denote the relaxation of P (obtained by deleting the size constraints) by P'.

In order to obtain the above we prove a *dichotomy of properties*. In particular, we show that every homogeneous graph partition property P falls into one of two disjoint categories.

▶ **Claim 9** (The Dichotomy of Properties). *Every homogeneous property P in the class $\mathcal{GPP}$ satisfies one of the following:*
1. $P = P'$.
2. *There exists $\epsilon > 0$ such that for every n_0 there exists a graph $G' \in P'$ of size $n > n_0$ where G' is ϵ-far from P.*

The implication of the *dichotomy of properties* is that a homogeneous graph partition property P with size constraints cannot be close to its relaxation P'. Either P is equivalent to P' or P is far from P' (for an appropriate distance measure). We claim that if the latter case holds then P is not easily-testable. Namely,

▶ **Claim 10.** *Suppose there exists $\epsilon > 0$ such that for every n_0 there exists a graph $G' \in P'$ of size $n > n_0$ where G' is ϵ-far from P. Then P is not easily testable.*

Combining Claim 10 with *the dichotomy of properties* (Claim 9) establishes the following: If a graph partition property P enforces size constraints, then P is not easily testable. It still remains to prove that *the dichotomy of properties* indeed holds. In order to do so, we first define the notions of a size vector and of a property's set of assignments. Then we show the existence of another dichotomy, *the trivial dichotomy*, which, as we state below, implies *the dichotomy of properties*.

▶ **Definition 11.** Given a homogeneous property P in $\mathcal{GPP}$ we define a set of variables $X = \{x_{ij} | (i,j) \in [k] \times [k]\}$. For a size vector $\vec{\rho}$, we say that an assignment $\varphi : X \to [0,1]$ is **sizewise valid** if: $\forall i' \in [k] : \sum_{i=1}^{k} \varphi(x_{i'i}) = \rho_{i'}$ and $\forall i \in [k] : \rho_i^L \le \sum_{i'=1}^{k} \varphi(x_{i'i}) \le \rho_i^U$.

We interpret an assignment φ as a transformation from a size vector $\vec{\rho}$ that violates the size constraints of P to a size vector that satisfies the size constraints. In particular, we interpret $\varphi(x_{ij})$ as the fraction of vertices (relative to n) that should be transferred from part i to part j (or stay in part i if $i = j$) in order to satisfy the size constraints imposed by P. Applying the transformation induced by a sizewise valid assignment φ to a graph G' that satisfies P' clearly results in a graph G that satisfies the size constraints imposed by P. However, the assignment being sizewise valid does not necessarily induce a transformation resulting in a graph that satisfies the edge density constraints. This leads to the notion of a *violation* defined next.

▶ **Definition 12.** Given a homogeneous property P in $\mathcal{GPP}$ and an assignment φ we say that a pair of variables $\{x_{i'i}, x_{j'j}\}$ constitutes a **violation** in the assignment φ with respect to P if $\varphi(x_{i'i}) \ne 0, \varphi(x_{j'j}) \ne 0, d_P(i,j) \ne \bot, d_P(i',j') \ne d_P(i,j)$. If $d_P(i',j') = \bot$, then we say that the violation is **weak**. Otherwise, we say the violation is **strong**.

The situation stated in the above definition is considered to be a violation of the edge density constraints because it can be interpreted as vertices being transferred to a pair of parts whose edge density differs from that of the pair of parts those vertices originated from. Given a violation $\{x_{i'i}, x_{j'j}\}$ we define the violation's *size* as $\min\{\varphi(x_{i'i}), \varphi(x_{j'j})\}$. We call a violation of size at least δ a δ*-violation*.

The following claim describes *the trivial dichotomy*.

▶ **Proposition 13** (The Trivial Dichotomy). *Let P be a homogeneous graph partition property. Exactly one of the following holds:*

1'. For every size vector $\vec{\rho}$ there exists a sizewise valid assignment φ which is free of violations with respect to P.

2'. There exists a size vector $\vec{\rho}$ for which every sizewise valid assignment φ contains a violation with respect to P.

2.3.1 The Trivial Dichotomy Implies The Dichotomy of Properties

The trivial dichotomy (Proposition 13) trivially holds as the second case is the complement of the first. In order to prove that *the trivial dichotomy* implies *the dichotomy of properties* (Claim 9) we have to prove that Case 1' in *the trivial dichotomy* implies Case 1 in *the dichotomy of properties* and that Case 2' in *the trivial dichotomy* implies Case 2 in *the dichotomy of properties*. We next give the high-level idea for the proof that Case 2' implies Case 2.

Let $\vec{\rho}$ be a size vector for which every assignment φ that is sizewise valid contains a violation. We define a set of decision variables: $Y = \{y_{i'i} | 1 \le i \le k, 1 \le i' \le k\}$.

▶ **Definition 14.** We say that a pair of decision variables $y_{i'i}, y_{j'j} \in Y$ are in potential conflict if $d_P(i,j) \neq \perp$ and $d_P(i',j') \neq d_P(i,j)$.

That is, $y_{i'i}$ is in potential conflict with $y_{j'j}$ if the pair $\{x_{i'i}, x_{j'j}\}$ constitutes a violation with respect to P in an assignment φ where $\varphi(x_{i'i}) > 0$ and $\varphi(x_{j'j}) > 0$.

We define a mathematical program on the set of decision variables Y. The objective of the program is to minimize the function $\max_{(y_{i'i}, y_{j'j}) \in Y} \{\min\{y_{i'i}, y_{j'j}\}\}$.

The feasible region is defined by the following system of linear constraints:

$$\forall i' \in [k] : \sum_{i=1}^{k} y_{i'i} = \rho_{i'}$$

$$\forall i \in [k] : \rho_i^L \le \sum_{i'=1}^{k} y_{i'i} \le \rho_i^U$$

$$\forall (i', i) \in [k] : 0 \le y_{i'i} \le 1$$

There is a one-to-one correspondence between feasible solutions to the program and sizewise valid assignments. Moreover, the correspondence between assignments and feasible solutions has the property that an assignment is free of violations if and only if the objective function attains a value of 0 for the corresponding feasible solution. It is shown in the full version that there exists a feasible solution (and a corresponding assignment φ) that minimizes $\max_{(y_{i'i}, y_{j'j}) \in Y} \{\min\{y_{i'i}, y_{j'j}\}\}$. Denote by δ the value of the objective function under that solution. Since we have assumed that every sizewise valid assignment φ contains a violation, there is no feasible solution attaining an objective function value of 0. In other words, $\delta > 0$. Therefore, every sizewise valid assignment contains a violation of size at least δ.

Let $\epsilon = \frac{1}{16}\delta^2$. We show that for every n_0 there exists a graph of size $n > n_0$ that is ϵ-far from P. Here we show this for the case where there are only weak δ-violations. We construct a random graph $G' = (V', E')$ of size n and prove that with positive probability G' is ϵ-far from every n-size graph G satisfying P. Let $V' = \{1, ..., n\}$ be the set of vertices of the graph G'. We arbitrarily partition $V[G']$ into k disjoint sets $U_1, ..., U_k$ in such a way that the number of vertices in U_i is $\rho_i n$. Let E' be defined as follows: For each pair of vertices $u \in U_i, v \in U_j$: If $d_P(i,j) = 0$ we do not connect u and v. If $d_P(i,j) = 1$ we do connect u and v. If $d_P(i,j) = \perp$ we connect u and v with probability $\frac{1}{2}$.

For each $1 \le i \le k$ where $d_P(i,i) = \perp$ we define the event $\mathcal{E}_i$ as follows: *For every set $A \subseteq U_i$ of size $|A| \ge \delta n$ it holds that $e_{G'}(A,A) \ge \epsilon n^2$ and $\overline{e}_{G'}(A,A) \ge \epsilon n^2$.*

For each pair $(i,j) \in [k] \times [k]$ where $d_P(i,j) = \perp$ we define the event $\mathcal{E}_{ij}$ as follows: *For every pair of disjoint sets $(A,B) \subseteq U_i \times U_j$ s.t. $A \subseteq U_i, B \subseteq U_j$ where $|A| \ge \delta n$ and $|B| \ge \delta n$ it holds that $e_{G'}(A,B) \ge \epsilon n^2$ and $\overline{e}_{G'}(A,B) \ge \epsilon n^2$..*

Note that the event $\mathcal{E}_i$ is not the same as event $\mathcal{E}_{ii}$. We denote by $\mathcal{E}$ the conjunction of all the events defined above. A detailed probabilistic analysis establishes the following claim.

▶ **Claim 15.** $\Pr[\mathcal{E}] > 0$.

Claim 15 implies the existence of a graph G^* for which $\mathcal{E}$ holds and the pair $(G^*, (V_1, ..., V_k))$ satisfies P'. We next show that G^* is ϵ-far from every n-vertex graph G satisfying P. Let $(G, (V_1, ..., V_k))$ be any graph-partition pair satisfying P where G is a graph over n vertices. We define the assignment φ as $\varphi(x_{i'i}) = \frac{|U_{i'} \cap V_i|}{n}$. Since the assignment φ can be shown to be sizewise valid, it contains a δ-violation $\{x_{i'i}, x_{j'j}\}$. Suppose the violation is weak. That is, $d_P(i',j') = \perp$ and $d_P(i,j) \ne \perp$. The fact that $\varphi(x_{i'i}) \ge \delta$ and $\varphi(x_{j'j}) \ge \delta$ implies that $|U_{i'} \cap V_i| \ge \delta n$ and $|U_{j'} \cap V_j| \ge \delta n$. The event $\mathcal{E}_{i'j'}$ holds and so do $\mathcal{E}_{i'}$ and $\mathcal{E}_{j'}$. Hence, $e_{G^*}(V_i, V_j) \ge \epsilon n^2$ and $\overline{e}_{G^*}(V_i, V_j) \ge \epsilon n^2$. However, since $d_P(i,j) \ne \perp$, the number of vertex-pair modifications required in order to obtain G from G^* is at least ϵn^2. In other words, the graph G^* is ϵ-far from every graph G that satisfies P.

It follows that easily-testable homogeneous graph partition properties cannot have size constraints, and if they do, then these size constraints are in fact redundant. In conclusion, if a graph partition property P is easily-testable then it is both homogeneous and it has no size constraints. That is, $P \in \mathcal{GPP}_{0,1}$.

References

1. Noga Alon. Testing subgraphs in large graphs. *Random Structures & Algorithms*, 21(3-4):359–370, 2002.
2. Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.
3. Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39(1):143–167, 2009.
4. Noga Alon and Jacob Fox. Easily testable graph properties. *Combinatorics, Probability and Computing*, 24(4):646–657, 2015.
5. Noga Alon and Michael Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Mathematics*, 15(2):211–227, 2002.
6. Noga Alon and Asaf Shapira. Testing subgraphs in directed graphs. *Journal of Computer and System Sciences*, 69(3):354–382, 2004.
7. Noga Alon and Asaf Shapira. A characterization of easily testable induced subgraphs. *Combinatorics, Probability and Computing*, 15(6):791–805, 2006.

8 Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008.

9 Christian Borgs, Jennifer Chayes, László Lovász, Vera T Sós, Balázs Szegedy, and Katalin Vesztergombi. Graph limits and parameter testing. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 261–270. ACM, 2006.

10 David Conlon and Jacob Fox. Graph removal lemmas. *Surveys in combinatorics*, 1(2):3, 2013.

11 Lior Gishboliner and Asaf Shapira. Removal lemmas with polynomial bounds. *arXiv preprint arXiv:1611.10315*, 2016.

12 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

13 Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Structures & Algorithms*, 23(1):23–57, 2003.