Abstract Combinatorial Programs and Efficient Property Testers [*]

Artur Czumaj
Department of Computer Science
New Jersey Institute of Technology
University Heights
Newark, NJ 07102-1982, USA
E-mail: `czumaj@cis.njit.edu`

Christian Sohler
Heinz Nixdorf Institute and
Department of Mathematics & Computer Science
University of Paderborn
D-33095 Paderborn, Germany
E-mail: `csohler@uni-paderborn.de`

Abstract

Property testing is a relaxation of classical decision problems which aims at distinguishing between functions having a predetermined property and functions being far *from any function having the property. In this paper we present a novel framework for analyzing property testing algorithms with one-sided error. Our framework is based on a connection of property testing and a new class of problems which we call* abstract combinatorial programs. *We show that if the problem of testing a property can be reduced to an* abstract combinatorial program *of small dimension, then the property has an efficient tester. We apply our framework to a variety of classical combinatorial problems. Among others, we present efficient property testing algorithms for* geometric clustering *problems, for the* reversal distance *problem, for* graph and hypergraph coloring *problems. We also prove that, informally, any* hereditary graph property *can be efficiently tested if and only if it can be reduced to an abstract combinatorial program of small size.*

Our framework allows us to analyze all our testers in a unified way and the obtained complexity bounds either match or improve the previously known bounds. We believe that our framework will help to better understand the structure of efficiently testable properties.

1. Introduction

In this paper, we consider *Property Testing* problems, that is, problems of determining whether a given function has a predetermined property or is "far" from any function having the property. A notion of property testing was first explicitly formulated by Rubinfeld and Sudan [25], who were motivated mainly by its connection to program checking. This notion arises naturally in the context of program verification [7, 25], learning theory, and, in a more theoretical setting, in probabilistically checkable proofs [6]. In [17], the study of property testing for *combinatorial objects* was initiated. In this and other more recent papers (see, the excellent surveys in [13, 16, 24] and the references therein), various algorithms have been proposed for testing graph and hypergraph properties, for testing geometric properties, for testing properties of metrics and matrices, for testing properties of regular languages and branching problems, for testing monotonicity, properties of Boolean functions, etc.

A property testing algorithm (*property tester*) is a randomized algorithm that distinguishes (with low error probability) between the functions that have a predetermined property and functions that are "far" from any function having the property. A good property tester is one that looks only at a small fraction of the function values. There are two error models for property testing algorithms. In this paper we consider only the *one-sided error* model, in which the tester must *accept* every function that has the property and must *reject* with probability at least $\frac{2}{3}$ every function that is "far" from having the property. To specify the notion of being "far" from having a property, one has to define a distance measure between functions. For a given parameter ϵ, a *function is ϵ-far from having a property if it has distance bigger than ϵ from any function having the property.*

Since property testing is a relaxation of the traditional decision problem, it is often possible to design algorithms that are much faster than their "classical" counterparts. In particular, there exist many property testing algorithms whose complexity is sublinear, or even independent of the input size (see, e.g., [13, 16, 24]). This, in turn, resulted in development of *sublinear*-time approximation algorithms (in the "traditional" sense) for many classical combinatorial problems, including bisection, metric Max-cut, clustering problems, estimating the cost of the minimum spanning tree, etc. (see, e.g., [4, 9, 10, 12, 14, 17, 19, 20]).

[*] Research supported in part by NSF grant CCR-0105701, by DFG grant Me872/8-1, and by the Future and Emerging Technologies program of the EU under contract number IST-1999-14186 (ALCOM-FT).

Although many efficient testing algorithms are known, most of them have been analyzed using ad-hoc techniques designed specially for the problem at hand. There is still insufficient methodology and a very few tools that could help in the analysis of efficient testers for new problems. Goldreich *et. al.* [17] (see also Theorem 4.3 in [13]) presented a fairly general framework (for the two-sided-error model; see [18] for a characterization in a one-sided-error model) of studying testing of certain graph partitioning problems. They were able to apply this framework to some graph problems, including graph coloring, clique, cut, and bisection. Another general approach, which uses the Szemerédi regularity lemma, has been proposed recently for studying graph problems and problems on matrices [3, 14, 21]. Even if this method is very powerful (and in particular, it allowed to prove that all first order graph properties without a quantifier alternation of type $\forall\exists$ have property testers whose complexity is independent of the size of the input graph), there are still some limitations of this approach. Furthermore, even though the bounds obtained by using the regularity lemma lead to the complexity bounds that are often independent from the input size, their dependence on the approximation parameter ϵ is often enormous (see, the "tower" bounds in [3] and superpolynomial lower bounds in [1]).

1.1. A Framework for Property Testing Problems

The main contribution of this paper is introduction of a novel framework to analyze property testing algorithms. We focus on functions *properties* that are *closed under taking restrictions*. That is, we consider properties of the following form: If a function $f : \mathcal{D} \to \mathcal{R}$ has a given property, then so has the function $f_{|X}$ (function f restricted to X), for any $X \subseteq \mathcal{D}$. This class of properties captures a type of monotonicity which is essential for almost all (if not all non-trivial) *one-sided error* property testers and it includes many natural properties. For properties closed under taking restrictions, we consider property testing algorithms that choose a sample of the domain $\mathcal{S} \subseteq \mathcal{D}$ uniformly at random and then verify if $f_{|\mathcal{S}}$ has the property:

```
SAMPLING PROPERTY TESTER
sample a set $\mathcal{S}$ of $s$ objects from $\mathcal{D}$ uniformly at random
if $f_{|\mathcal{S}}$ has the property then accept else reject
```

Property testing algorithms of this kind are simple to implement. The main difficulty with their use is the estimation of the sample size: what is the right sample size s (which is the *query complexity* of the tester) so that the algorithm is a correct property tester? It is easy to see that for properties closed under taking restrictions, if f has the required property then the algorithm always accepts f. Thus, the challenging part of the analysis is to estimate the value of s such that if f is far from having the property, then f will be rejected with probability at least $\frac{2}{3}$. Our framework is designed to help in the analysis needed in this step.

In order to define our framework, we first introduce briefly a notion of an *abstract combinatorial program (ACP)*. An ACP consists of a *ground set*, which is typically a set of basic objects underlying the property testing problem, a set of *bases*, where each basis is a configuration of a subset of the ground set, and a *violation function* that verifies the input constraints and specifies whether an element violates a given basis or not. We call a basis *feasible* if it is not violated by any object from the ground set. We investigate a generic problem of *testing feasibility of an ACP*, that is, for a given ACP, we want to distinguish the case when the ACP has a feasible basis from the case when any basis is violated by at least an ϵ fraction of objects from the ground set. We present a sampling theorem (Theorem 1) that gives a bound for the size of the random sample taken in the sampling property tester when testing feasibility of ACPs. We show that if a certain monotonicity property is satisfied by an ACP, then the sample size s depends only on the maximum size of the basis in the ACP. Thus, the sample size is independent of the size of the ground set.

The main idea behind introducing abstract combinatorial programs is that for many properties ACPs capture the structure essential for testing the property. Therefore, in our framework we reduce property testing of a given property Q to the problem of testing feasibility of related ACPs. We show that property Q (closed under taking restrictions) can be tested efficiently if there *exists* a *reduction* to abstract combinatorial programs that satisfies two properties:

- the reduction is *distance preserving*, that is, any function that is far away from Q is mapped to an ACP in which each basis is violated by many objects from the ground set, and
- the reduction is *feasibility preserving*, that is, if the function restricted to a sample set S has property Q, then the subset of the ground set corresponding to S has a feasible basis in the ACP.

To demonstrate the applicability of our framework, we show its generality by applying it to a variety of problems. We illustrate our framework on problems from geometric optimization, computational biology, and on graphs and hypergraphs. An important feature of the framework is that it provides a powerful tool that allows to concentrate on the combinatorial structure of the problem at hand rather than on probabilistic arguments about sampling. For example, our analysis of clustering has a similar flavor as the previous analysis of this problem in [2]. What distinguish our analysis, however, is that we do not have to deal with the probabilistic analysis of the sampling required by the tester that actually hides the combinatorial structure of the problem.

Problem	Source	Query complexity
k-diameter	[2]	$\Omega((1/\beta)^{(d-1)/4})$; $\Omega(\sqrt{n})$ for $\beta = 0$
clustering	[2]	$\widetilde{\mathcal{O}}(k^2\, d\, \epsilon^{-1}\, (2/\beta)^{2\, d})$; only in L_2 metric
in $\mathbb{R}^d$	*this paper*	$\widetilde{\mathcal{O}}(k\, \epsilon^{-1}\, (1 + 2/\beta)^{d})$; any L_p metric
	this paper	$\widetilde{\mathcal{O}}(k\, d\, \epsilon^{-1}\, (2/\beta)^{d-1})$; any L_p metric
sorting by reversals	*this paper*	$\mathcal{O}(k/\epsilon)$
k-coloring	[11]	$(\widetilde{\mathcal{O}}(k^2\, \ell^2/\epsilon^2))^{\ell}$
of ℓ-uniform	[5]	$(\widetilde{\mathcal{O}}(k^{\ell-1}/\epsilon^2))^{\ell}$
hypergraphs	*this paper*	$(\widetilde{\mathcal{O}}(k\, \ell/\epsilon^2))^{\ell}$

Table 1. Summary of selected specific results.

Instead, we can deal with pure combinatorial arguments and hence, simplify the proof to obtain a stronger bound for the complexity of the tester. Our tester for hypergraphs coloring is also of similar flavor (compare to [4, 5, 11], where more complicated arguments are used and weaker bounds are obtained).

In this paper, we first introduce abstract combinatorial programs and show that they are testable if certain requirements are satisfied. Then, we present a simple version of our main theorem and we illustrate it on the examples of the diameter clustering problem and the sorting by reversals problem. Then, we present our full framework and discuss its applicability on the example of hypergraph coloring problem, in which the new framework leads to compact and elegant proofs. In the last section, we show that for any testable hereditary graph property there exists a reduction to ACPs that proves that the property is testable.

We can apply our framework to some other combinatorial problems. However, due to space limitations we only consider the problems listed in Table 1.

2. Property Testing and Abstract Combinatorial Programs

Throughout the paper, we denote by $\mathcal{D}$ a finite set called *domain* and by $\mathcal{R}$ a set called *range*. By $\mathcal{F}$ we denote the set of functions from $\mathcal{D}$ to $\mathcal{R}$ and by $\mathcal{F}^*$ be the set of "restrictions" of functions in $\mathcal{F}$, that is, $\mathcal{F}^* = \{f : X \rightarrow \mathcal{R} : X \subseteq \mathcal{D}\}$. A set $\mathcal{Q} \subseteq \mathcal{F}^*$ is called a *property* of $\mathcal{F}$ (or a property defined on the elements of $\mathcal{F}^*$). A property $\mathcal{Q}$ of $\mathcal{F}$ is called *closed under taking restrictions* if $f_{|\, S} \in \mathcal{Q}$ holds for every $f \in \mathcal{Q}$ and any $S \subseteq \mathcal{D}$.

We assume there is given a (problem dependent) *distance measure* $\varsigma : \mathcal{F} \times \mathcal{F} \rightarrow [0, 1]$ that measures the distance between any two functions in $\mathcal{F}$ (it is *not* required for ς to be a metric). Typically, our distance measure will be the *relative distance between the functions* (see, e.g., [6, Definition 4.1]), that is, for any two functions $f, g \in \mathcal{F}$, we define $\varsigma(f, g) = \mathbf{Pr}_{x \in \mathcal{D}}[f(x) \neq g(x)]$, where the probability is taken according to the uniform probability distribution over $\mathcal{D}$. Given a real number ϵ, $0 \leq \epsilon < 1$, we say a function $f \in \mathcal{F}$ is *ϵ-far from (having a property) $\mathcal{Q}$* if $\varsigma(f, g) > \epsilon$ for any function $g \in \mathcal{Q} \cap \mathcal{F}$. An *$\epsilon$-tester for property $\mathcal{Q}$* is an algorithm that (i) accepts any function $f \in \mathcal{Q}$ and (ii) rejects with probability greater than or equal to $\frac{2}{3}$ any function that is ϵ-far from $\mathcal{Q}$.

We assume the access to any function $f \in \mathcal{F}$ is given by an *oracle* that can access values of f. Then the number of the queries to the values of the input function $f \in \mathcal{F}$ is the *query complexity* of the property tester.

2.1. Abstract Combinatorial Programs

In this section we describe the notion of *abstract combinatorial programs*. An *abstract combinatorial program* (*ACP*) is defined by an abstract set of objects, which we call a *ground set*, a set of *bases*, which consists of some "basic" configurations of subsets of the ground set, and a set of constraints described by a *violation function*.

The *ground set* depends on the problem under consideration (and in all our applications is independent of the input instance). For example, this may be a vertex set of a graph or a set of halfspaces describing a linear program.

A set of *bases* consists of some "basic" configurations of subsets of the ground set. And so, for example, if the ground set is a vertex set of a graph, then a basis may be defined as a subset of vertices, or as a subset of vertices X together with an associated k-vertex-coloring of X. If the ground set is a set of halfspaces in $\mathbb{R}^d$ defining a linear program, then we could take as the set of bases the intersection of any d halfspaces (which is known to define a point in $\mathbb{R}^d$ in a non-degenerated case). Because of technical reasons, we shall always assume that every basis is defined as a pair (X, ℓ), where X is a subset of the ground set and ℓ is an index describing a configuration of X (for example, in the graph-coloring example above, it is a coloring of vertices in X). Unlike the ground set, the set of bases usually depends on the input instance.

A *violation function* is used to determine which bases are *feasible*. Typically, the violation function depends on the input instance. To define a violation function, for example, in the linear programming case, we can say that a given halfspace $\mathcal{H}$ violates a given basis if and only if the basis determines a point p which is not contained in $\mathcal{H}$. For the graph-coloring example above one can define the violation function such that a vertex v violates a basis (colored vertex set X) if and only if in the input graph the k-coloring of X cannot be extended to a proper k-coloring of $X \cup \{v\}$.

Formally, we define an abstract combinatorial program in the following way.

Definition 2.1 *Let $\mathcal{C}$ be a finite set (called a* ground set*). An* abstract combinatorial program *(ACP) over $\mathcal{C}$ is a pair $(\mathcal{B}, \varpi)$, where*

- $\mathcal{B} \subseteq \{(K, \ell) : K \subseteq \mathcal{C}, \ell \in \mathbb{N}\}$ *is a set of* bases*, and*
- $\varpi : \mathcal{B} \times \mathcal{C} \rightarrow \{true, false\}$ *is a function defining whether a basis* $b \in \mathcal{B}$ *is* violated *by an element* $c \in \mathcal{C}$*.*

A basis b *is* feasible *if it is not violated by any* $c \in \mathcal{C}$*, that is, if* $\varpi(b, c) = false$ *for every* $c \in \mathcal{C}$*.*

An abstract combinatorial program is feasible *if it has a feasible basis.*

We study abstract combinatorial programs in the context of deciding whether a given ACP is *feasible* or not. In our framework we shall use also the following definitions.

Definition 2.2 (ACP Dimension) *An abstract combinatorial program* $\mathbb{A} = (\mathcal{B}, \varpi)$ *over* $\mathcal{C}$ *has* dimension (δ, ϱ) *if for all* $b = (K, \ell) \in \mathcal{B}$ *it holds that* $|K| \leq \delta$ *and* $\ell \leq \varrho$*.*

Definition 2.3 (Self-feasible bases) *Let* $(\mathcal{B}, \varpi)$ *be an abstract combinatorial program. We say a basis* $b = (K, \ell) \in \mathcal{B}$ *is* covered *by a subset* $C^* \subseteq \mathcal{C}$ *if* $K \subseteq C^*$*. We say that a basis* b *is* feasible *for a subset* $C^* \subseteq \mathcal{C}$*, if no* $c \in C^*$ *violates* b*. We say a subset* $C^* \subseteq \mathcal{C}$ contains a self-feasible basis *if there is a basis* b *that is covered by* C^* *and that is feasible for* C^**.*

Definition 2.4 ((Semi-)monotone ACPs) *A feasible abstract combinatorial program* $(\mathcal{B}, \varpi)$ *over* $\mathcal{C}$ *is* monotone *if any subset* $S \subseteq \mathcal{C}$ *contains a self-feasible basis. Let* s *be any integer. A feasible abstract combinatorial program* $(\mathcal{B}, \varpi)$ *over* $\mathcal{C}$ *is* s-semi-monotone *if any subset* $S \subseteq \mathcal{C}$ *with* $|S| \geq s$ *contains a self-feasible basis.*

2.2. Testing Abstract Combinatorial Programs

In this section we consider the problem of testing ACPs. An abstract combinatorial program is ϵ*-far from feasible* if any basis is violated by more than $\epsilon \cdot |\mathcal{C}|$ objects from the ground set $\mathcal{C}$. An ϵ*-tester* for ACPs is an algorithm that (i) accepts every feasible ACP and (ii) rejects with probability at least $\frac{2}{3}$ any ACP that is ϵ-far from feasible. The following key theorem characterizes testable ACPs.

Theorem 1 (Testing ACPs) *Let* $\mathcal{C}$ *be a finite ground set and let* $\mathbb{ACP}_{(\delta,\varrho)}(\mathcal{C})$ *be the set of abstract combinatorial programs over* $\mathcal{C}$ *of dimension* (δ, ϱ)*. Let* $s = \Theta(\epsilon^{-1} \cdot (\delta \cdot \ln(\delta/\epsilon) + \ln \varrho))$*. Then, the algorithm that takes as its input an ACP* $\mathbb{A} \in \mathbb{ACP}_{(\delta,\varrho)}(\mathcal{C})$*, samples a set* $\mathcal{S}$ *of* s *objects from* $\mathcal{C}$ *uniformly at random, and accepts* $\mathbb{A}$ *if* $\mathcal{S}$ *contains a self-feasible basis (and rejects otherwise), satisfies the following properties:*

1. *If* $\mathbb{A}$ *is* ϵ*-far from feasible, then* $\mathbb{A}$ *is rejected with probability at least* $\frac{2}{3}$*.*
2. *If* $\mathbb{A}$ *is feasible and it is either monotone or is* s*-semi-monotone, then* $\mathbb{A}$ *is accepted.* □

3. Simple Reductions to ACPs

Our main motivation to introduce abstract combinatorial programs was to study their relation to property testing algorithms. In this section, and later in Section 4, we show how the framework described in Section 2.1 can be applied to obtain various efficient property testers, where in many cases the structure of the problems on the first glace does not seem to fit into the framework of abstract combinatorial programs. We present a rather general reduction-based technique that can be used to prove the correctness of various property testing algorithms by reductions to abstract combinatorial programs.

Our approach of using the framework of abstract combinatorial programs to study property testers of functions $f \in \mathcal{F}$ is to reduce testing of f to testing certain ACP. In the simplest case this reduction is done in a rather easy way, because there is a one to one correspondence between the domain of f and the ground set of the ACP. A more complicated reduction requires some manipulations with the ground set, bases, and the violation function. Therefore, for simplicity of presentation, we first describe the simpler model and only later, in Section 4, discuss its extensions to the full framework.

The following theorem describes a simple version of our framework.

Theorem 2 *Let* $\mathcal{F}$ *be a set of functions from a finite set* $\mathcal{D}$ *to a set* $\mathcal{R}$ *and let* $\mathcal{Q}$ *be a property of* $\mathcal{F}$ *that is closed under taking restrictions. Let* $0 < \epsilon < 1$*. Let* $\mathbb{ACP}_{(\delta,\varrho)}(\mathcal{D})$ *be the set of abstract combinatorial programs over* $\mathcal{D}$ *of dimension* (δ, ϱ)*. Let* $s = \Theta(\epsilon^{-1} \cdot (\delta \cdot \ln(\delta/\epsilon) + \ln \varrho))$*. If for every* $f \in \mathcal{F}$ *there exists an abstract combinatorial program* $\mathbb{A}_f \in \mathbb{ACP}_{(\delta,\varrho)}(\mathcal{D})$ *such that:*

(Distance Preserving) *if* f *is* ϵ*-far from* $\mathcal{Q}$ *then* $\mathbb{A}_f$ *is* ϵ*-far from feasible and*

(Feasibility Preserving) *for every* $S \subseteq \mathcal{D}$*, if* S *contains no self-feasible basis then* $f_{|S} \notin \mathcal{Q}$*,*

then the following algorithm is an ϵ*-tester for* $\mathcal{Q}$ *with the query complexity of* $\Theta(\epsilon^{-1} \cdot (\delta \cdot \ln(\delta/\epsilon) + \ln \varrho))$:

```
        TESTER($f$)
    Sample a set $\mathcal{S}$ of $s$ elements in $\mathcal{D}$ uniformly at random
    if $f_{|\mathcal{S}} \in \mathcal{Q}$ then accept $f$ else reject $f$
```

Furthermore, the same algorithm is an ϵ*-tester for* $\mathcal{Q}$ *if the Feasibility Preserving property is replaced by the following* s**-semi Feasibility Preserving** *property:*

for every $S \subseteq \mathcal{D}$ *with* $|S| \geq s$*, if* S *contains no self-feasible basis then* $f_{|S} \notin \mathcal{Q}$*.*

Proof : We first observe that the query complexity of TESTER(f) follows directly from the fact that TESTER(f) queries for exactly s values of f.

In order to show that TESTER(f) is an ϵ-tester for $\mathcal{Q}$, we have to prove that any function having property $\mathcal{Q}$ is accepted by the tester and any function that is ϵ-far from having property $\mathcal{Q}$ is rejected with probability at least $\frac{2}{3}$. Since $\mathcal{Q}$ is closed under taking restrictions, if $f \in \mathcal{Q}$ then for any $X \subseteq \mathcal{D}$ (and in particular, for $X = \mathcal{S}$) $f_{|X} \in \mathcal{Q}$. This immediately implies that every $f \in \mathcal{Q}$ is accepted by TESTER(f). Therefore, it remains to prove that if f is ϵ-far from $\mathcal{Q}$, then the algorithm rejects the input with probability greater than or equal to $\frac{2}{3}$. We prove this by relating ACP-TESTER($\mathbb{A}_f$) with TESTER(f) and by applying Theorem 1.

By the Distance Preserving property, if f is ϵ-far from $\mathcal{Q}$ then $\mathbb{A}_f$ is ϵ-far from feasible. Furthermore, by Theorem 1, if $\mathbb{A}_f$ is ϵ-far from feasible then ACP-TESTER($\mathbb{A}_f$) rejects $\mathbb{A}_f$ with probability at least $\frac{2}{3}$. $\mathbb{A}_f$ is rejected by ACP-TESTER($\mathbb{A}_f$) only if the chosen sample set $\mathcal{S}$ contains no self-feasible basis. But now, accordingly, either the Feasibility Preserving or the s-semi Feasibility Preserving property implies that if $\mathcal{S}$ contains no self-feasible basis then $f_{|S} \notin \mathcal{Q}$. Therefore, we can conclude that if f is ϵ-far from $\mathcal{Q}$ then $f_{|S} \notin \mathcal{Q}$ with probability at least $\frac{2}{3}$, and hence, f is rejected by TESTER(f) with probability at least $\frac{2}{3}$. This implies that TESTER(f) is a proper ϵ-tester for $\mathcal{Q}$. □

Let us mention briefly that the ACP formulation is usually not equivalent to the problem under consideration: It is possible that the ACP has a self-feasible basis for a subset S of its ground set but $f_{|S}$ does not have property Q. For example, this is the case for the ACP formulations of the diameter clustering problem and the hypergraph coloring problem presented later in this paper.

3.1. Testing Diameter Clustering

In this section, we demonstrate how to apply our framework of testing ACPs to test the classical problem of *diameter clustering* in $\mathbb{R}^d$. For a given point set X in $\mathbb{R}^d$, the *diameter* of X is the maximum distance between any two points in X. The (decision version of the) *diameter clustering* problem (see, e.g., [2] and [15, Problem MS9]) is to decide if an input point set P in $\mathbb{R}^d$ can be partitioned into k sets (called *clusters*) such that the diameter of each cluster is bounded from above by a given real number 1. In this paper, we mainly focus on the problems under the L_2 metric (Euclidean), but we show also that our arguments can be carried over to an arbitrary L_p metric, $p \geq 1$.

We consider a *bicriteria relaxation* of the diameter k-clustering problem introduced by Alon *et. al.* [2]. We use the following notion (notice that Alon *et. al.* [2] proved that without using the bicriteria relaxation, that is, when $\beta = 0$, there is no ϵ-tester having the query complexity of $o(\sqrt{n})$ even in the most basic case of $k = 1$):

Definition 3.1 [2] *Let P be a point set in $\mathbb{R}^d$ and k be a positive integer. We say P is (ϵ, β)*-far from being k-clusterable *if for any partition of P into sets $C_0, C_1, \dots, C_k$ satisfying $dist(x, y) \leq 1 + \beta$ for all $1 \leq i \leq k$ and $x, y \in C_i$, it holds that $C_0 > \epsilon \cdot |P|$.*

With this definition, our goal is to design an efficient property tester that for given k, ϵ and $\beta > 0$ (i) always accepts any point set that is k-clusterable and (ii) rejects with probability at least $\frac{2}{3}$ any input that is (ϵ, β)-far from being k-clusterable.

For any non-empty set X of points in $\mathbb{R}^d$ with $dist(x, y) \leq 1$ for every $x, y \in X$, the *kernel* $kern(X)$ *of* X is defined as the intersection of unit balls with centers at the points in X.

Let P be a point set in $\mathbb{R}^d$, k a positive integer, and β a positive real. Let $X_1, \dots, X_k$ be any disjoint subsets of P. We say a point $p \in P$ is *β-covered* by $\{X_1, \dots, X_k\}$ if for some i and some $q \in X_i$ we have $p \in kern(X_i)$ and $dist(p, q) \leq \beta$.

To use the framework from Theorem 2 we define domain $\mathcal{D}$ to be the set $\{1, \dots, n\}$, range $\mathcal{R}$ to be the set $\mathbb{R}^d$, set of functions $\mathcal{F}$ to map the points to their locations in $\mathbb{R}^d$ (i.e., from $\mathcal{D}$ to $\mathcal{R}$), and property $\mathcal{Q}$ to correspond to all functions in $\mathcal{F}^*$ that represent point sets that are clusterable into at most k-clusters such that any pair of points in each cluster is at distance at most 1. Now, in order to use our framework from Theorem 2 we have to describe for any input set P of n points in $\mathbb{R}^d$ an ACP $\mathbb{A}_P$ over $\mathcal{D}$ that satisfies the preconditions of the theorem.

The bases in $\mathbb{A}_P$ are formed by k sets of points (ground set elements), one set for each cluster. (In the remainder of this section we assume that a basis is given as a partition of a set of points into k sets rather than a set of points with an encoding of such a partition.) The idea of introducing the sets associated with the clusters is to represent each cluster by a small set of points X for which the kernel will approximate the kernel in the real clustering. We want to define the bases such that if the input point set P is k-clusterable, then there is a basis $\{X_1, \dots, X_k\}$ such that each point $p \in P$ is β-covered by $\{X_1, \dots, X_k\}$. On the other hand, we define the bases such that if any k-clustering of P has diameter greater than $1 + \beta$, then for any $\{X_1, \dots, X_k\}$ there is a point $p \in P$ that is not β-covered by $\{X_1, \dots, X_k\}$. These two properties will then be used to distinguish between point sets that are k-clusterable and those for which any k-clustering has diameter greater than $1 + \beta$.

Bases for diameter clustering: We recursively define the set of bases as follows:

- $\{\emptyset, \dots, \emptyset\}$ is a basis (where $\{\emptyset, \dots, \emptyset\}$ is the set consisting of k empty sets)
- if $b = \{X_1, \dots, X_k\}$ is a basis then $\{X_1, \dots, X_{i-1}, X_i \cup \{p\}, X_{i+1}, \dots, X_k\}$ is also a basis if $p \in P$ is a point that is not β-covered by b and (i) either $X_i = \emptyset$ or (ii) $p \in kern(X_i)$.

A simple volume argument gives us the following result:

Lemma 3.1 *The ACP defining diameter clustering has dimension* $(k \cdot (1 + (2/\beta))^d, k^{k\cdot(1+(2/\beta))^d})$. □

Violation function for diameter clustering: *A basis* b *is* violated *by a point* p *if* p *is not* β*-covered by* b.

Now, we show that the Distance Preserving and the s-semi Feasibility Preserving properties of Theorem 2 are satisfied with $s = \Theta(k \cdot \epsilon^{-1} \cdot (1 + (2/\beta))^d \cdot \ln(k\,\epsilon^{-1}\,(1 + (2/\beta))^d))$.

Distance Preserving Property: The proof is by contradiction. Let us assume P is (ϵ, β)-far from being k-clusterable and suppose there is a basis $b = (X_1, \dots, X_k)$ that is violated by less than $\epsilon\, n$ points. We delete all points in P that violate b and let P^* be the remaining point set. Since all the points in P^* are β-covered by b, for each point $p \in P^*$ there is an X_i with $p \in kern(X_i)$ and for which there exists $q_p \in X_i$ with $dist(p, q_p) \le \beta$. We assign each such a point p to the cluster corresponding to X_i. Observe that all points in the cluster are contained in $kern(X_i)$. Furthermore, for any point $r \in kern(X_i)$ the distance between p and r is not larger than the distance from p to q_p plus the distance from q_p to r. Hence, we can conclude that the distance between two points in the cluster (both of which must be contained in $kern(X_i)$) is at most $1 + \beta$. This implies that P^* can be partitioned into k clusters of diameter at most $1 + \beta$ each, which is a contradiction.

s**-semi Feasibility Preserving Property:** Let S be a set of points with $|S| \ge s$ that contains no self-feasible basis. Then, every basis b that is covered by S is violated by certain $p \in S$. If p violates b, then p is either outside the kernel of every cluster in b or p is in some kernel but the distance to each other point defining the corresponding cluster is bigger than β. In the latter case, we can obtain a new basis b' covered by S by adding p to b. Since b' is also violated by some point in S and the size of each basis is bounded, we can conclude inductively that any basis b is violated by some point $q \in S$ that is outside the kernel of every cluster. But by our discussion about the bases, this implies that S is not k-clusterable. This yields the s-semi Feasibility Preserving property.

Now, by our discussion above, we can apply Theorem 2 to obtain a property tester for the diameter clustering problem under the L_2 metric having the query complexity of $\widetilde{\mathcal{O}}(k \cdot \epsilon^{-1} \cdot (1 + (2/\beta))^d)$. Actually, one can slightly modify our arguments to obtain even a stronger result that holds for arbitrary L_p metrics.

Theorem 3 *There is a property tester for the diameter clustering problem under the* L_p *metric,* $p \ge 1$*, that for any* β*,* $0 < \beta \le 1/d$*, always accepts a feasible input, with probability at least* $\frac{2}{3}$ *rejects any input which is* (ϵ, β)*-far from being* k*-clusterable, and has the query complexity of* $\widetilde{\mathcal{O}}(k \cdot d \cdot \epsilon^{-1} \cdot (2/\beta)^{d-1})$. □

3.2. Testing Reversal Distance

The study of genome comparisons and rearrangements is one of the major topics in modern molecular biology. Mathematical analysis of genome rearrangements was initiated by Sankoff, who introduced the *sorting by reversals problem* (see, e.g., [22, Chapter 10]). In sorting by reversals one asks to compute the *reversal distance* of a given permutation, which is the minimum number of *reversals* needed to be performed to transform the permutation into the identity permutation. Because of its applications in computational biology, sorting by reversals has been widely studied in the last years (see, e.g., [22, 23]).

In this paper, we introduce the notion of property testing in the context of sorting by reversals. We design a property testing algorithm that verifies if a given permutation has reversal distance at most k or is ϵ-far from having reversal distance at most k. We apply our framework to show that it has the query complexity of $\widetilde{\mathcal{O}}(k/\epsilon)$.

Let $\mathbb{S}_n$ denote the set of all permutations of $\{1, \dots, n\}$. A reversal $\varrho\langle i, j\rangle$ of an interval $[i, j]$, $1 \le i \le j \le n$, is the permutation that for each permutation $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$, $\varrho\langle i, j\rangle$ has the effect of reversing the order of $(\pi_i, \pi_{i+1}, \dots, \pi_j)$ and transforming π into $\pi \cdot \varrho\langle i, j\rangle = (\pi_1, \dots, \pi_{i-1}, \pi_j, \pi_{j-1}, \dots, \pi_i, \pi_{j+1}, \dots, \pi_n)$ (see, e.g., [22, Chapter 10]). Given a pair of permutations $\pi, \sigma \in \mathbb{S}_n$, the *reversal distance* between π and σ is the minimum number of reversals needed to transform π into σ (that is, the minimum number k such that there exists a sequence of reversals $\varrho_1, \varrho_2, \dots, \varrho_k$ with $\pi \cdot \varrho_1 \cdot \varrho_2 \cdots \varrho_k = \sigma$). The reversal distance between π and the identity permutation $id = (1, 2, \dots, n)$ is called the *reversal distance* of π. The *sorting by reversals* problem is for a given permutation $\pi \in \mathbb{S}_n$ to find the reversal distance of π.

To apply our framework in the context of sorting by reversals, we have to consider also restrictions of permutations. We say $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$ is a *restriction* of a permutation $\pi' = (\pi'_1, \dots, \pi'_n) \in \mathbb{S}_n$ if for each i, $1 \le i \le n$, either $\pi_i = \pi'_i$ or $\pi_i =$ *undefined*. Now, to apply our framework, we define domain $\mathcal{D}$ and range $\mathcal{R}$ to be both equal to $\{1, \dots, n\}$, and we define $\mathcal{F} = \mathbb{S}_n$ and $\mathcal{F}^*$ to be the set of restrictions of permutations in $\mathbb{S}_n$. We extend the reversal distance to functions in $\mathcal{F}^*$ in the following natural way: A restriction of a permutation $\pi \in \mathcal{F}^*$ has *reversal distance* less than or equal to k if there exist k reversals $\varrho_1, \dots, \varrho_k$ such that if $\pi \cdot \varrho_1 \cdots \varrho_k = \sigma = (\sigma_1, \dots, \sigma_n)$, then for any $1 \le i \le n$, either $\sigma_i = i$ or $\sigma_i =$ *undefined*.

We define the k*-reversal distance property* $\mathcal{Q}$ to be the set of all permutations $\pi \in \mathcal{F}^*$ that have reversal distance smaller than or equal to k. One can easily verify that $\mathcal{Q}$ is closed under taking restrictions.

In order to design a property testing algorithm we use the relative distance in our context. We say a permutation $\pi \in \mathbb{S}_n$ is ϵ-far from having reversal distance smaller than or equal to k if for any sequence of k reversals $\varrho_1, \varrho_2, \dots, \varrho_k$, permutation $\pi \cdot \varrho_1 \cdot \varrho_2 \cdots \varrho_k$ disagrees with the identity permutation on more than $\epsilon \cdot n$ places.

The *ground set* $\mathcal{C}$ in ACPs used in our framework is identical with the domain $\{1, \dots, n\}$ and, for simplicity of notation, we identify each $i \in \mathcal{C}$ with π_i.

Let us notice that we can encode an interval $[i, j]$ by π_i and π_j (using the fact that $\pi^{-1}(\pi_i) = i$ and $\pi^{-1}(\pi_j) = j$). If we apply a reversal ϱ to π then π_i and π_j *induce* the interval $[(\pi \cdot \varrho)^{-1}(\pi_i), (\pi \cdot \varrho^{-1})(\pi_j)]$. We denote the interval induced by two elements π_i and π_j by $[\pi_i, \pi_j]$.

We say a reversal $\varrho\langle r, s\rangle$ *splits* an interval $[\pi_i, \pi_j]$ if either $i < r \leq j$ or if $i \leq s < j$. We generalize this notion to k-reversals: A sequence of k reversals $\varrho_1, \dots, \varrho_k$ *splits* an interval $[\pi_i, \pi_j]$ if there exists ℓ, $0 \leq \ell < k$, such that $\varrho_{\ell+1}$ splits $[(\pi \cdot \varrho_1 \cdots \varrho_\ell)^{-1}(\pi_i), ((\pi \cdot \varrho_1 \cdots \varrho_\ell)^{-1}(\pi_j)]$. If $\varrho_1, \dots, \varrho_k$ does not split $[\pi_i, \pi_j]$ then we say $\varrho_1, \dots, \varrho_k$ is *safe* for $[\pi_i, \pi_j]$. Notice that if $\varrho_1, \dots, \varrho_k$ is safe for $[\pi_i, \pi_j]$, then each of the reversals $\varrho_1, \dots, \varrho_k$ either entirely contains $[\pi_i, \pi_j]$ or it does not contain any $\pi_s \in [\pi_i, \pi_j]$. Therefore, in this case, after applying $\varrho_1, \dots, \varrho_k$ the positions of $\pi_{i+1}, \dots, \pi_{j-1}$ are determined by the position of π_i and π_j.

Bases for the k-reversal property: Our goal is to define a basis as a set of $2k + 1$ intervals induced by pairs of the ground set elements of the basis. For each such a set we then consider only reversals that are safe for these intervals.

Let $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$. A set $\mathcal{I}$ of $2\,k + 1$ intervals is a valid basis for the reversal distance problem if there is a sequence $\varrho_1, \dots, \varrho_k$ of k reversals such that

- $(\pi \cdot \varrho_1 \cdots \varrho_k)^{-1}(\pi_i) = \pi_i$ and $(\pi \cdot \varrho_1 \cdots \varrho_k)^{-1}(\pi_j) = \pi_j$ for each interval $[\pi_i, \pi_j] \in \mathcal{I}$, and
- no interval $[\pi_i, \pi_j] \in \mathcal{I}$ is split by $\varrho_1, \dots, \varrho_k$.

If the set of intervals is a basis b, then we associate with it any such a k-reversal $\varrho_b = \varrho_1 \cdots \varrho_k$ (ties broken arbitrarily). It is easy to verify that the ACPs constructed this way have dimension $(4k + 2, (4k + 2)^{4k+2})$.

Violation function for the k-reversal property: Let b be a basis and let $\varrho_b = \varrho_1, \dots \varrho_k$ be the k-reversal associated with b. We say b *is violated by* $\pi_i \in \mathcal{C}$ *if* $(\pi \cdot \varrho_b)^{-1}(\pi_i) \neq \pi_i$.

Distance and Feasibility Preserving Property: With the above definition the Distance Preserving property is trivially satisfied. The difficult part is to prove the Feasibility Preserving property. Let $S \subseteq \mathcal{C}$ be a set of ground set elements and let $\varrho = \varrho_1 \cdots \varrho_k$ be a k-reversal with $(\pi \cdot \varrho)^{-1}(\pi_i) = \pi_i$ for each $\pi_i \in S$. We show that in this case S has a self-feasible basis. Let us consider a maximal set of maximal intervals not split by ϱ. We observe that this set has cardinality at most $2k + 1$ since a single reversal can cause splits at no more than 2 places. We conclude that these intervals form a basis b. It remains to prove that this basis is not violated (the k-reversal associated with the basis does not have to be identical with ϱ). By our construction of the intervals (i.e., by the maximality of the intervals) each $\pi_i \in S$ is contained in a safe interval. Therefore, its position after applying the reversal is uniquely determined by the positions of the endpoints of the interval. Let $S_I \subseteq S$ denote the set of endpoints of intervals of the basis b. Since b is a basis there is a k-reversal ϱ_b with $(\pi \cdot \varrho_b)^{-1}(\pi_i) = \pi_i = (\pi \cdot \varrho)^{-1}(\pi_i)$ for each $\pi_i \in S_I$. Since the endpoints are mapped to the identical positions when ϱ_b and ϱ are applied to π, we can conclude that each other point in S is also mapped to the identical position. Hence, no $\pi_i \in S$ violates b and the Feasibility Preserving property is satisfied. We conclude:

Theorem 4 *There exists an ϵ-tester for the k-reversal distance property with query complexity $\widetilde{\mathcal{O}}(k/\epsilon)$.* □

4. Full Framework of Testing Algorithms via Testing ACPs

In Section 3, we described a framework for testing problems via testing abstract combinatorial programs. The framework presented in that section has a few unnecessary assumptions that we want to address now.

The first restriction of the framework described in Section 3 is that the ground set $\mathcal{C}$ in ACPs is required to be identical with the domain $\mathcal{D}$ of the functions. In order to avoid this restriction, we introduce the notion of *interpretation*. An *interpretation* of $\mathcal{C}$ in $\mathcal{D}$ is a function I that maps each subset of the ground set $\mathcal{C}$ to a subset of the domain $\mathcal{D}$ of the functions we consider. For example, when we consider graph properties we identify the ground set for the ACPs with the set of vertices of the graph and the interpretation gives us for each set of vertices the submatrix corresponding to the induced subgraph. Since interpretations affect the query complexity of the tester we need another notion: We say that an interpretation I of $\mathcal{C}$ in $\mathcal{D}$ is *g-bounded* if for every $X \subseteq \mathcal{C}$ it holds $|I(X)| \leq g(|X|)$ where g is a function $g : \mathbb{N} \to \mathbb{N}$.

We adapt the definition of the property being closed under taking restrictions to interpretations in the following way: A property $\mathcal{Q}$ is *closed under taking restrictions* of I, if $\forall f \in \mathcal{Q}, \forall S \subseteq \mathcal{D}$ it holds that $f_{|I(S)} \in \mathcal{Q}$.

The main idea behind introducing these notions is to allow a more general analysis of algorithm TESTER(f) from

Section 3 via analyzing ACPs. As in the proof of Theorem 2, we want to test an input function $f \in \mathcal{F}$ via testing a related ACP $\mathbb{A}_f$. Since $\mathbb{A}_f$ is allowed to be an ACP over an arbitrary ground set $\mathcal{C}$, we use the interpretation I of $\mathcal{C}$ in $\mathcal{D}$ to link the domains of f and $\mathbb{A}_f$ in the reduction. The notion of g-bounded functions is used to describe the size of the random sample in the tester. That is, if the interpretation I is g-bounded and if in our analysis we require $\mathbb{A}_f$ to sample a set S of s elements in $\mathcal{C}$, then we shall require to sample set $I(S)$ from the domain $\mathcal{D}$ of f, where $|I(S)| \le g(s)$.

In Theorem 2 we used the Distance Preserving property that requires that if a function f is ϵ-far from property Q then the ACP is ϵ-far from feasible. In general, however, one can parameterize this property and require the **(ϵ, λ)-Distance Preserving** property: *if f is ϵ-far from property Q then the ACP is λ-far from feasible.*

Summarizing, in the framework defined above, it is easy to see that Theorem 2 can be generalized to the following theorem, which describes the main property of our framework in its full generality.

Theorem 5 *Let $\mathcal{F}$ be the set of functions from a finite set $\mathcal{D}$ to a set $\mathcal{R}$, and let $\mathcal{Q}$ be a property of $\mathcal{F}$. Let $0 < \epsilon < 1$. Let $\mathcal{C}$ be a finite ground set and let $\mathbb{ACP}_{(\delta,\varrho)}(\mathcal{C})$ be the set of abstract combinatorial programs of dimension (δ, ϱ) over $\mathcal{C}$. Let $I : 2^{\mathcal{C}} \to 2^{\mathcal{D}}$ be a g-bounded interpretation of $\mathcal{C}$ in $\mathcal{D}$ such that $\mathcal{Q}$ is closed under taking restrictions of I. Let $0 < \lambda \le 1$ and let $s = \Theta(\lambda^{-1} \cdot (\delta \cdot \ln(\delta/\lambda) + \ln \varrho))$.*

If for every $f \in \mathcal{F}$ there exists an abstract combinatorial program $\mathbb{A}_f \in \mathbb{ACP}_{(\delta,\varrho)}(\mathcal{C})$ such that:

((ϵ, λ)-Distance Preserving) *if f is ϵ-far from $\mathcal{Q}$ then any basis in $\mathbb{A}_f$ is λ-far from feasible and*

(Feasibility Preserving) *for every $S \subseteq \mathcal{C}$, if S contains no self-feasible basis then $f_{|I(S)} \notin \mathcal{Q}$,*

then algorithm TESTER(f) *is an ϵ-tester for $\mathcal{Q}$ with the query complexity of $g(s) = g(\Theta(\lambda^{-1} \cdot (\delta \cdot \ln(\delta/\lambda) + \ln \varrho)))$.*

*Furthermore, the same algorithm is an ϵ-tester for $\mathcal{Q}$ if the Feasibility Preserving property is replaced by the following s-***semi Feasibility Preserving** *property: for every $S \subseteq \mathcal{C}$ with $|S| \ge s$, if S contains no self-feasible basis then $f_{|I(S)} \notin \mathcal{Q}$.* □

5. Testing Hypergraph Coloring

In this section we demonstrate our framework from Theorem 5 to design a very efficient *property tester for testing hypergraph coloring*. A *hypergraph* is a pair $\mathcal{H} = (V, E)$ with a finite vertex set V and the edge set $E \subseteq 2^V$. A hypergraph $\mathcal{H}$ is *ℓ-uniform* if $|e| = \ell$ for all $e \in E$. A *k-coloring* of a hypergraph $\mathcal{H}$ is an assignment $\chi : V \to \{1, \dots, k\}$. A k-coloring is *proper* if no edge in E is *monochromatic*, that is, if for every edge $e \in E$ there are $v, u \in e$ with $\chi(v) \neq \chi(u)$. A hypergraph having a proper k-coloring is called *k-colorable*. The k-coloring problem for hypergraphs is to decide whether a given hypergraph is k-colorable. We assume that a ℓ-uniform hypergraph with n vertices is represented by its ℓ-dimensional adjacency matrix. We say a hypergraph is *ϵ-far from having a proper k-coloring* if one has to change more than $\epsilon\, n^\ell$ entries in the adjacency matrix to obtain a hypergraph with a proper k-coloring.

To apply our framework to hypergraph coloring, we identify the ground set $\mathcal{C}$ with the set of vertices V of the input hypergraph $\mathcal{H} = (V, E)$. Since $\mathcal{H}$ is represented by its adjacency matrix, we define the interpretation I to map each set of vertices to the submatrix induced by these vertices. That is, for any $W \subseteq V$, we have $I(W) = W \times \cdots \times W$. Clearly, the interpretation is N^ℓ-bounded.

Let $\langle S, \chi \rangle$ be a pair with $S \subseteq V$ and χ a proper k-coloring of vertices in S. We say a vertex v is *i-colorable* with respect to $\langle S, \chi \rangle$ if for every $e \in E$ with $v \in e$, either (i) there exists a vertex $u \in (S \cap e)$ with $\chi(u) \neq i$ or (ii) there exists a vertex $w \in e \setminus (S \cup \{v\})$.

In order to define bases we define a potential function for partial colorings. The potential function is a measure for the weighted number of "constraints" on the colors of the uncolored vertices in the hypergraph. For any integers i, j, let us define

$$\Lambda_{i,j}\langle S, \chi \rangle := \Big\{ X \subseteq V \ : \ |X| = \ell - j \ \ \& \qquad (1)$$
$$\exists e \in E \ (X \subseteq e \ \ \& \ \ \forall_{u \in e \setminus X} \ \chi(u) = i) \Big\} \ .$$

Then, the *potential* of $\langle S, \chi \rangle$ is defined as

$$\Phi_{\mathcal{H}}(\langle S, \chi \rangle) \ := \ \sum_{i=1}^{k} \sum_{j=1}^{\ell-1} n^{j-1} \cdot |\Lambda_{i,j}\langle S, \chi \rangle| \ .$$

Next, we introduce the notion of *conflict* and *heavy* vertices. A vertex $v \in V \setminus S$ is a *conflict vertex* with respect to $\langle S, \chi \rangle$ if for every i, $1 \le i \le k$, v is *not* i-colorable. A vertex $v \in V \setminus S$ is *heavy* with respect to $\langle S, \chi \rangle$ if (i) there is an i, $1 \le i \le k$, such that v is i-colorable and (ii) for every i, $1 \le i \le k$, if v is i-colorable and χ' is the extension of χ to $S \cup \{v\}$ by coloring v with color i then $\Delta\Phi_{\mathcal{H}}(v, i, \langle S, \chi \rangle) > \frac{\epsilon n^{\ell-1}}{3}$, where $\Delta\Phi_{\mathcal{H}}(v, i, \langle S, \chi \rangle) \ := \ \Phi_{\mathcal{H}}(\langle S \cup \{v\}, \chi' \rangle) - \Phi_{\mathcal{H}}(\langle S, \chi \rangle)$.

The bases for the ACPs correspond to colorings of subsets of vertices.

Bases for k-coloring:

- $\{\emptyset, 0\}$ is a basis (where 0 is the encoding of the coloring of the empty set of vertices) and
- if $b = (K, \chi)$ is a basis, v is a *heavy* vertex for b and χ^* is an encoding of the previous coloring χ of K extended by a proper coloring of v, then $(K \cup \{v\}, \chi^*)$ is a basis.

Violation function for k-coloring: *A basis $b = (K, \chi)$ is violated by a vertex $v \in V$ if either v is a heavy vertex for $\langle K, \chi \rangle$ or v is a conflict vertex for $\langle K, \chi \rangle$.*

It is easy to prove that the ACPs defined above have dimension $(3\,k\,\ell/\epsilon, k^{3\,k\,\ell/\epsilon})$ and that the corresponding reduction is feasibility preserving. The difficult part is to prove the distance preserving property:

Lemma 5.1 ($(\epsilon, \epsilon/3)$-Distance Preserving property) *Let $\mathcal{H} = (V, E)$ be a hypergraph that is ϵ-far from being k-colorable and let $S \subseteq V$ be any set of properly k-colored vertices with a proper coloring χ. Then, either V has more than $\epsilon\,n/3$ conflict vertices with respect to $\langle S, \chi \rangle$ or V has more than $\epsilon\,n/3$ heavy vertices for $\langle S, \chi \rangle$.*

Proof : The proof is by contradiction. Let us assume there are at most $\epsilon\,n/3$ heavy vertices and at most $\epsilon\,n/3$ conflict vertices with respect to $\langle S, \chi \rangle$. Then, we show that it is possible to extend coloring χ of S to a coloring χ^* of V that has at most $\epsilon\,n^\ell$ monochromatic edges in $\mathcal{H}$. This will yield contradiction.

We define χ^* as follows:

$$\chi^*(v) = \begin{cases} \chi(v) & \text{for any } v \in S \\ 1 & \text{if } v \in V \setminus S \text{ and } v \text{ is either a conflict vertex} \\ & \text{or a heavy vertex with respect to } \langle S, \chi \rangle \\ i & \text{if } v \in V \setminus S \text{ is } i\text{-colorable with respect to } \langle S, \chi \rangle \\ & \text{and } i \text{ minimizes (over all possible choices of} \\ & \text{proper coloring } i\text{) the increase in potential,} \\ & \text{i.e., } \Delta\Phi_{\mathcal{H}}(v, i, \langle S, \chi \rangle) \le \Delta\Phi_{\mathcal{H}}(v, j, \langle S, \chi \rangle) \\ & \text{for any proper coloring } j \text{ of } v \end{cases}$$

Now, we give an upper bound on the number of monochromatic edges in coloring χ^* of $\mathcal{H}$. Let us first consider heavy and conflict vertices. By our assumption, the number of such vertices is upper bounded by $\frac{2}{3}\,\epsilon\,n$. Therefore, the number of edges incident to these vertices is upper bounded by $\frac{2}{3}\,\epsilon\,n^\ell$. Hence, it is sufficient to show that there are at most $\frac{1}{3}\,\epsilon\,n^\ell$ monochromatic edges in $\mathcal{H}$ that are not incident to heavy or conflict vertices.

Let us fix a vertex v that is neither heavy nor conflict. We show that there are at most $\frac{1}{3}\,\epsilon\,n^{\ell-1}$ monochromatic edges incident to v in $\mathcal{H}$, which by our arguments above will complete the proof. Vertex v is colored in χ^* with color i such that (i) v is i-colorable with respect to $\langle S, \chi \rangle$ and (ii) the potential function satisfies

$$\Delta\Phi_{\mathcal{H}}(v, i, \langle S, \chi \rangle) \le \frac{1}{3}\,\epsilon\,n^{\ell-1} \quad . \tag{2}$$

Let χ' be the extension of coloring χ to $S \cup \{v\}$ by coloring v with color i. Notice that in order for an edge e incident to v to be monochromatic in coloring χ^*, for every vertex $u \in (e \cap S)$ it must hold $\chi(u) = i$. This motivates us to define the following set

$$E^v_{i,j}\langle S, \chi \rangle := \{e \in E : v \in e,\ |e \setminus S| = \ell - j,\ \forall_{u \in e \cap S} \chi(u) = i\}.$$

Thus, an edge e incident to v may be monochromatic in coloring χ^* only if $e \in \bigcup_{j=0}^{\ell-2} E^v_{i,j}\langle S, \chi \rangle$ (notice that χ^* ensures that $E^v_{i,\ell-1}\langle S, \chi \rangle = \emptyset$). We show that $\left|\bigcup_{j=0}^{\ell-2} E^v_{i,j}\langle S, \chi \rangle\right| \le \frac{1}{3}\,\epsilon\,n^{\ell-1}$, which implies that there are at most $\frac{1}{3}\,\epsilon\,n^{\ell-1}$ monochromatic edges incident to v in $\mathcal{H}$, and hence, yields the proof of the lemma.

Let us consider the subsets of vertices that belong to $\Lambda_{r,j}\langle S, \chi \rangle$ or to $\Lambda_{r,j}\langle S \cup \{v\}, \chi' \rangle$, for certain r and j. From (1), it is easy to see that if $X \in \Lambda_{r,j}\langle S, \chi \rangle$, then $X \in \Lambda_{r,j}\langle S \cup \{v\}, \chi' \rangle$ too. On the other hand, if $X \notin \Lambda_{r,j}\langle S, \chi \rangle$, then $X \in \Lambda_{r,j}\langle S \cup \{v\}, \chi' \rangle$ if and only if (i) $|X| = \ell - j$, (ii) $r = i$, (iii) $v \notin X$, and (iv) there exists $e \in E$ with $X \cup \{v\} \subseteq e$ such that every vertex $u \in e \setminus (X \cup \{v\})$ has $\chi(u) = i$. Therefore, if we define

$$\Upsilon^{v,i}_j(\langle S, \chi \rangle) := \Big\{ X \subseteq V : |X| = \ell - j \ \ \&\ \ v \notin X \ \ \& \\ \exists e \in E : \big(X \cup \{v\} \subseteq e \ \ \&\ \ \forall_{u \in e \setminus (X \cup \{v\})}\ \chi(u) = i\big) \Big\} \ ,$$

then
$$\Delta\Phi_{\mathcal{H}}(v, i, \langle S, \chi \rangle) = \sum_{j=1}^{\ell-1} n^{j-1} \cdot \left|\Upsilon^{v,i}_j(\langle S, \chi \rangle)\right| \quad . \tag{3}$$

Next, let us observe that if $e \in E^v_{i,j}\langle S, \chi \rangle$, then $X = e \setminus (S \cup \{v\})$ must belong to $\Upsilon^{v,i}_{j+1}(\langle S, \chi \rangle)$. Furthermore, for a set $X \in \Upsilon^{v,i}_{j+1}(\langle S, \chi \rangle)$, there can be at most $\binom{|S|}{j} \le n^j$ edges e such that $X = e \setminus (S \cup \{v\})$. Therefore, $\left|E^v_{i,j}\langle S, \chi \rangle\right| \le n^j \cdot \left|\Upsilon^{v,i}_{j+1}(\langle S, \chi \rangle)\right|$. Hence, we can combine this inequality with inequality (2) and with equation (3), to conclude that

$$\left|\bigcup_{j=0}^{\ell-2} E^v_{i,j}\langle S, \chi \rangle\right| = \sum_{j=0}^{\ell-2} \left|E^v_{i,j}\langle S, \chi \rangle\right| \le \sum_{j=0}^{\ell-2} n^j \left|\Upsilon^{v,i}_{j+1}(\langle S, \chi \rangle)\right| \\ = \Delta\Phi_{\mathcal{H}}(v, i, \langle S, \chi \rangle) \le \frac{1}{3}\,\epsilon\,n^{\ell-1} \ .$$

Therefore, by our arguments above, we have proven that if a vertex v is neither heavy nor conflict, not more than $\frac{1}{3}\,\epsilon\,n^{\ell-1}$ edges incident to v may be monochromatic in coloring χ^*. Since there are at most n such vertices, we get an upper bound of $\frac{1}{3}\,\epsilon\,n^\ell$ for the number of monochromatic edges in coloring χ^* that are not incident to heavy or conflict vertices. This implies that the total number of monochromatic edges in coloring χ^* of $\mathcal{H}$ is upper bounded by $\epsilon\,n^\ell$. This in turn, implies that the hypergraph $\mathcal{H}$ is *not* ϵ-far from being k-colorable. This yields contradiction. □

The above results and our framework from Theorem 5 imply the following result.

Theorem 6 *There is an ϵ-tester for the hypergraph k-colorability with the query complexity $\widetilde{\mathcal{O}}((k\,\ell/\epsilon^2)^\ell)$.* □

6. Hereditary Graph Properties and ACPs

In this section we consider *hereditary graph properties*. A *graph property* Π is any family of graphs that is preserved under graph isomorphism (that is, if G satisfies property Π and G' is a graph isomorphic to G then G' has property Π too). A graph property Π is *hereditary* if it is closed under taking induced subgraphs, that is, if for every graph G having property Π every induced subgraph of G has property Π too (see, e.g., [8]). We call a graph property Π *strongly-testable* [1] if for every $\epsilon > 0$ there exists a (one-sided error) ϵ-tester for Π whose query complexity is bounded only by a function of ϵ, which is independent of the size of the input graph. We consider the standard adjacency matrix model (see the previous section for the more general definition for hypergraphs).

In the previous section we gave a reduction from hypergraph coloring to ACPs that satisfies the requirements of our framework which proves that hypergraph coloring can be tested efficiently. We observe that the constructed ACPs are not equivalent to the hypergraph coloring problem in the following sense: There might be a subset S of vertices such that the subgraph induced by S does not have a proper coloring but the corresponding ACP has a self-feasible basis. Nevertheless, the bases of the ACPs have a nice interpretation on the corresponding hypergraph: each basis corresponds to a coloring of a certain subset of vertices. Two natural questions arise of whether it is possible to apply our framework to other graph properties and, if this is possible, whether there is a nice interpretation of bases for these properties. We answer the first question by showing that we can apply our framework to any testable hereditary graph property. The second question remains open.

We show that a *hereditary graph property* can be tested efficiently in the adjacency matrix model if and only if there is a reduction to ACPs. Although it is known [3, 18] that a testable hereditary graph property Π can be tested by a canonical tester (a tester that samples a set of vertices and accepts if and only if the induced subgraph has property Π) the straightforward reductions to ACPs either violate the distance preserving or the feasibility preserving property.

Theorem 7 *Let Π be a hereditary graph property. Let $0 < \epsilon < 1$. Let $\mathcal{G}$ be the set of all graphs on the vertex set $V = \{1, \dots, n\}$. For any $\delta, \varrho \in \mathbb{N}$, let $\mathbb{ACP}_{(\delta,\varrho)}(V)$ be the set of abstract combinatorial programs of dimension (δ, ϱ) over V. Then, Π is strongly-testable* **if and only if** *there are $\delta = \delta(\epsilon)$, $\varrho = \varrho(\epsilon)$, and $\lambda = \lambda(\epsilon)$, such that for every $G \in \mathcal{G}$ there exists an abstract combinatorial program $\mathbb{A}_G \in \mathbb{ACP}_{(\delta,\varrho)}(V)$ satisfying the following two properties:*

((ϵ, λ)-Distance Preserving) *if G is ϵ-far from Π then any basis in $\mathbb{A}_G$ is λ-far from feasible, and*

(Feasibility Preserving) *for any $S \subseteq V$, if the subgraph G_S satisfies property Π then there is a self-feasible basis for S in $\mathbb{A}_G$.* □

References

[1] N. Alon. Testing subgraphs in large graphs. *42th FOCS*, pp. 434–441, 2001.

[2] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. *41st FOCS*, pp. 240–250, 2000.

[3] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.

[4] N. Alon, W. Fernandez de la Vega, R. Kannan, and M. Karpinski. Random sampling and approximation of MAX-CSP problems. *34th STOC*, pp. 232–239, 2002.

[5] N. Alon and A. Shapira. Testing satisfiability. *13th SODA*, pp. 645–654, 2002.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *JACM*, 45(3):501–555, 1998.

[7] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *JCSS*, 47(3):549–595, 1993.

[8] B. Bollobás. Hereditary properties of graphs: Asymptotic enumeration, global structure, and colouring. *International Congress of Mathematicians*, vol. 3, pp. 333–342, 1998.

[9] B. Chazelle, R. Rubinfeld, and L. Trevisan. Approximating the minimum spanning tree weight in sublinear time. *28th ICALP*, pp. 190–200, 2001.

[10] A. Czumaj, F. Ergün, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler. Sublinear-time approximation of Euclidean minimum spanning tree. *14th SODA*, 2003.

[11] A. Czumaj and C. Sohler. Testing hypergraph coloring. *28th ICALP*, pp. 493–505, 2001.

[12] W. Fernandez de la Vega and C. Kenyon. A randomized approximation scheme for metric Max-Cut. *39th FOCS*, pp. 468–471, 1998.

[13] E. Fischer. The art of uniformed decisions. A primer to property testing. *Bull. EATCS*, 75:97–126, 2001.

[14] A. Frieze and R. Kannan. The regularity lemma and approximation schemes for dense problems. *37th FOCS*, pp. 12–20, 1996.

[15] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, New York, NY, 1979.

[16] O. Goldreich. Combinatorial property testing (a survey). *DIMACS Workshop on Randomization Methods in Algorithm Design*, pp. 45–59, 1997.

[17] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *JACM*, 45(4):653–750, 1998.

[18] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *42nd FOCS*, pp. 460–469, 2001.

[19] P. Indyk. Sublinear time algorithms for metric space problems. *30th STOC*, pp. 428–434, 1998.

[20] P. Indyk. A sublinear time approximation scheme for clustering in metric spaces. *39th FOCS*, pp. 154–159, 1998.

[21] Y. Kohayakawa, B. Nagle, and V. Rödl. Efficient testing of hypergraphs. *29th ICALP*, pp. 1017–1028, 2002.

[22] P. A. Pevzner. *Computational Molecular Biology*. MIT Press, 2000.

[23] P. A. Pevzner and M. S. Waterman. Open combinatorial problems in computational molecular biology. *3rd ISTCS*, pp. 158–173, 1995.

[24] D. Ron. Property testing. In *Handobook of Randomized Algorithms*. Kluwer Academic Publishers, 2001.

[25] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.