Smoothed Analysis:
Motivation and Discrete Models*

Daniel A. Spielman[1] and Shang-Hua Teng[2]

[1] Department of Mathematics, Massachusetts Institute of Technology
[2] Department of Computer Science, Boston University

Abstract. In smoothed analysis, one measures the complexity of algorithms assuming that their inputs are subject to small amounts of random noise. In an earlier work (Spielman and Teng, 2001), we introduced this analysis to explain the good practical behavior of the simplex algorithm. In this paper, we provide further motivation for the smoothed analysis of algorithms, and develop models of noise suitable for analyzing the behavior of discrete algorithms. We then consider the smoothed complexities of testing some simple graph properties in these models.

1 Introduction

We believe that the goals of research in the design and analysis of algorithms must be to develop theories of algorithms that explain how algorithms behave and that enable the construction of better and more useful algorithms. A fundamental step in the development of a theory that meets these goals is to understand why algorithms that work well in practice actually do work well. From a mathematical standpoint, the term "in practice" presents difficulty, as it is rarely well-defined. However, it is a difficulty we must overcome; a successful theory of algorithms must exploit models of the inputs encountered in practice. We propose using smoothed analysis to model a characteristic of inputs common in many problem domains: inputs are formed in processes subject to chance, randomness, and arbitrary decisions. Moreover, we believe that analyses that exploit this characteristic can provide significant insight into the behavior of algorithms. As such analyses will be difficult, and will therefore be instinctively avoided by many researchers, we first argue the necessity of resting analyses on models of inputs to algorithms.

Researchers typically avoid the need to model the inputs to algorithms by performing worst-case analyses. By providing an analysis that does not depend upon the inputs, worst-case analysis provides an incredibly strong guarantee, and it is probably one of the greatest achievement of the theoretical computer science community. However, worst-case analysis provides only one statistic about an algorithm's behavior. In many situations, and especially those in which algorithms are used, it is more important to understand the typical behavior of

* The first author was supported in part by NSF grant CCR-0112487, and the second author was supported in part by NSF grant 99-72532

an algorithm. Moreover, the typical behavior of an algorithm is often quite different from its worst-case behavior. If the mention of the ill-defined "typical" causes a mathematical mind run to the comfort of the cleanly defined worst-case analysis, it is understandable. It is not even clear that one should try to use mathematics to understand a notion such as "typical behavior", and it is clear that experiments must also play a role. However, the results of experiments are best understood in the context of an abstract theory. Experiments can confirm or contradict a theory; but, mathematically describable theories provide the most desirable encapsulations of knowledge about algorithms. It remains to be seen whether these theories will be mathematically rigorous, reason by analogy with mathematically rigorous statements, or combine theorems with heuristic mathematical arguments as is common in the field of Physical Applied Mathematics.

In smoothed analysis, we exploit the low-order random events influencing the formation of the inputs to algorithms. These influences have many sources including measurement error, constraints imposed by economics or management, and the chain of chance leading to the consideration of any particular situation. Consider, for example, the design of a bridge that may be input to an algorithm. Design constraints are imposed by the surface under the bridge, and the locations of the roadways available to connect the bridge at either edge. A governmental committee will provide a probability distribution over architects, and a given architect will choose different designs in different periods of her career. These designs will then be altered as local politicians push contracts to favored constituents, etc.

By examining different levels of the design process, one can obtain complexity measures varying from average case to worst case. If one just views the entire process as providing one distribution on bridges, then one obtains an average-case complexity measure. If one merely considers the finished bridge, and maximizes over the possible bridges, then one obtains a worst-case complexity measure. By considering the probability distribution after certain choices have been made, and taking a maximum over those choices, one obtains a model between the average-case and worst-case complexity measures.

Of course, we cannot hope to define a mathematical model that precisely captures any of these influences or that captures the levels of refinement of the actual process. But, we can try to define models that capture their spirit and then reason by analogy. Our first attempt [ST01] was to model these influences by subjecting inputs to perturbations. In this model we defined the smoothed complexity of an algorithm to be the maximum over its inputs of its expected running time over random perturbations of those inputs. This running time should be measured in terms of the input length and the magnitude of the perturbations. By varying the magnitudes of the perturbations, we smoothly generate complexity measures between the worst-case and average-case.

However, a model in which inputs are perturbed at random may be unnatural for some problems, and it might be necessary to place some constraints upon the perturbations by insisting that they respect some divisions of the input space. For example, it might be necessary that the bridge be able to support a 20-ton truck (or SUV), and we should not allow perturbations of the bridge that violate this constraint to enter our probability space. In general, perturbations should probably be restricted to preserve the most significant aspects of an input for a given situation. For example, a natural perturbation of a graph is obtained by adding edges between unconnected vertices and removing edges with some probability. However, a graph subject to such perturbations is highly unlikely to have a large clique, and so it may be meaningless to measure the performance of algorithms for clique under this model. We propose to avoid this problem by studying *property-preserving perturbations*, which we define by restricting a natural perturbation model to preserve certain properties of the input. For example, one could imagine perturbing a graph subject to preserving the size of its largest clique.

We remark that a notion such as property-preserving perturbations is necessary even in average-case analysis. For example, if one desires an average-case analysis of algorithms for max-clique, one should state the running times of the algorithms as functions of the size of the max-clique. Otherwise, the probability mass is concentrated on the graphs without large cliques, and for which the problem is much less interesting. One should not be distracted by the fact that it may be computationally difficult to sample from the resulting conditional distributions under which we must measure the complexity of our algorithms.

Of course, one should not just preserve only the property being calculated by the algorithm: it is natural to require that the perturbations preserve all the most relevant properties of the input. For example, when studying algorithms for minimum bisection, one might consider genus- and bisection-size-preserving graph perturbations. We note that the complexity measure of an algorithm under perturbations that preserve more properties properties is strictly closer to worst-case complexity that a measure under perturbations that preserve a subset of the properties.

1.1 A Mathematical Introduction

In our analysis of the simplex method [ST01], we exploited the most natural model of perturbation for real-number inputs—that of Gaussian random perturbations. This model has also been applied in the smoothed analysis of the Perceptron Algorithm by Blum and Dunagan [BD02], of Interior Point Methods by Spielman and Teng [ST03] and Dunagan, Spielman and Teng [DST02]. For a survey of some of these works, we refer the reader to [ST02]. It has been suggested by many that these analyses could be made to have a tighter analogy with practice if the perturbations preserved more properties of their input. For example, it would be reasonable to restrict perturbations to preserve feasibility, infeasibility, or even the condition number of the programs. It is also natural to restrict the perturbations so that zero entries remain zero.

In this paper, we will mainly concern ourselves with discrete problems, in which the natural models of perturbations are not nearly as clear. For graphs, the most natural model of perturbation is probably that obtained by XORing the adjacency matrix with the adjacency matrix of a random sparse graph. This model is captured by the following definition:

Definition 1. *Let $\bar{G}$ be a graph and $\sigma > 0$. We define the σ-perturbation of $\bar{G}$ to be the graph obtained by converting every edge of $\bar{G}$ into a non-edge with probability σ and every non-edge into an edge with probability σ. We denote this distribution on graphs by $\mathcal{P}(\bar{G}, \sigma)$.*

Unfortunately, there are many purposes for which such perturbations can radically change an input, rendering the model meaningless. For example, it would be pointless to study algorithms testing whether a graph is bipartite or has a ρn-clique under this model because it is highly unlikely that the σ-perturbation of any graph will have either of these properties.

Property preserving perturbations provide a modification of this model in which this study becomes meaningful. Given a property P, and a notion of perturbation, we define a P-preserving perturbation of an object $\bar{X}$ to be a perturbation X of $\bar{X}$ sampled subject to the condition $P(\bar{X}) = P(X)$. For example, if $\bar{G}$ is a graph and G is a P-preserving σ-perturbation of $\bar{G}$, then G has density

$$\frac{\mathbf{Pr}_{G \leftarrow \mathcal{P}(\bar{G},\sigma)}\left[G \text{ and } (P(\bar{G}) = P(G))\right]}{\mathbf{Pr}_{G \leftarrow \mathcal{P}(\bar{G},\sigma)}\left[P(\bar{G}) = P(G)\right]}.$$

We can then say that an algorithm A has smoothed error probability δ under P-preserving σ-perturbations if

$$\max_{\bar{G}} \mathbf{Pr}_{G \leftarrow \mathcal{P}(\bar{G},\sigma)}\left[A(G) \text{ is incorrect} | P(G) = P(\bar{G})\right] \leq \delta.$$

Property preserving perturbations are a special case of function preserving perturbations in which the function is binary valued.

Definition 2. *Let f be a function defined on the space of graphs, let $\bar{G}$ be a graph and $\sigma > 0$. We define the f-preserving σ-perturbation of $\bar{G}$ to be the random graph G with density:*

$$\frac{\mathbf{Pr}_{G \leftarrow \mathcal{P}(\bar{G},\sigma)}\left[G \textit{ and } (f(\bar{G}) = f(G))\right]}{\mathbf{Pr}_{G \leftarrow \mathcal{P}(\bar{G},\sigma)}\left[f(\bar{G}) = f(G)\right]}.$$

This function could represent many qualities of a graph. In addition to properties, f could measure numerical quantities such as diameter or conductance. In such cases, it might be more reasonable to merely require the perturbed graph to approximately preserve f.

In the remainder of this paper, we will derive some elementary results on the complexity of graph properties under perturbations that preserve these properties. In particular, we will measure the smoothed error probability of sub-linear time algorithms for these problems. In this sense, we consider a problem closely related to that studied in the field of property testing. In property testing, one measures the worst-case complexity of Monte Carlo algorithms solving a promise problem of the form: determine whether or not an input has a property given that the input either has the property or is far from those inputs that have the property. For many property testing problems, we find that under perturbations that preserve the same property, the input typically satisfies such a guarantee. Conversely, if one cannot construct a notion of property-preserving perturbations under which inputs typically satisfy such a guarantee, then we feel one should probably not assume such a guarantee is satisfied in practice.

In the following sections, we obtain some simple results on the complexity of testing if graphs have small cliques, bisections, or are bipartite under property-preserving perturbations. We hope stronger results will be obtained by considering perturbations that preserve even more properties of their inputs.

1.2 Comparison with the Semi-Random model

Another approach to interpolating between worst-case and average-case complexity appears in a line of work initiated by Blum and Spencer [BS95]. Blum and Spencer considered the problem of k-coloring k-colorable graphs generated by choosing a random k-colorable graph and allowing an adversary to add edges between color classes. Feige and Kilian [FK98a] extended their results and considered analogous models for finding large cliques and optimal bisections. For the clique problem, a large clique is planted in a random graph, and an adversary is allowed to remove edges outside the clique. Their model for bisection modifies Boppana's model of a random graph with a planted bisection [Bop87] by allowing an adversary to add edges not crossing the bisection and remove edges crossing the bisection. It is easy to show that these models are stronger than the analogous models in which an adversary constructs a graph with a large clique or small bisection and these graphs are then perturbed in a way that preserves the embedded clique or bisection. In Section 3, we show that the graphs produced by ρ-Clique preserving σ-perturbations are close to the graphs produced by this later model, and that we can use the algorithm for Feige and Kilian to produce a fast testing algorithm for these properties.

In contrast, the planted bisection model considered by Feige and Kilian seems to produce rather different graphs than the ρ-Bisection preserving σ-perturbations, and we cannot find a way to use their algorithm to test for small bisections in this model, let alone speed up a tester. The difference is that a ρ-Bisection preserving σ-perturbation may produce a graph with many small bisections of almost exactly the same size, while the model considered by Feige and Kilian produces graphs in which the smallest bisection is significantly smaller than all competitors.

Other work in similar models includes the analysis by Feige and Krauthgamer [FK98b] for bandwidth minimization algorithms and Coja-Oghlan [CO02] for finding sparse induced subgraphs.

1.3 Property Testing

Rubinfeld and Sudan [RS96] defined property testing to be a relaxation of the standard decision problem: rather than designing an algorithm to distinguish between inputs that have and do not have a property, one designs an algorithm to distinguish between those that have and those that are far from having a property. Under this relaxation, many properties can be tested by sub-linear time algorithms that examine random portions of their input. In this paper, we will examine the testers designed by Goldreich, Goldwasser and Ron [GGR98].

Goldreich, Goldwasser and Ron [GGR98] introduced the testing of graph properties. Their results included the development of testers that distinguished between graphs that are bipartite, have size ρn cliques, and size ρn bisections from those graphs that have distance ϵ to those with these properties, where distance is measured by the Hamming distance of adjacency matrices.

Formally speaking, an algorithm A is said to be a property tester for the property P if

1. for all x with property P, $\mathbf{Pr}\left[A(x,\epsilon)=1\right] \geq 2/3$; and
2. for all x of distance at least ϵ from every instance that has property P, $\mathbf{Pr}\left[A(x,\epsilon)=1\right] \leq 1/3$,

under some appropriate measure of distance on inputs (although some testers have one-sided error). A typical property testing algorithm will use a randomized process to choose a small number of facets of x to examine, and then make its decision. For example, a property tester for a graph property may query whether or not certain edges exist in the graph. The quality of a property testing algorithm is measured by its query complexity (the number of queries to the input) and its time complexity.

Since the seminal works of Rubinfeld and Sudan [RS96] and Goldreich, Goldwasser, and Ron [GGR98], property testing has become a very active area of research in which many different types of properties have been examined [GR97,GR98,KR00,Alo01,ADPR00,AKFS99,BR00,GGLR98,Ron01,EKK$^+$98] [PR99,DGL$^+$99,CSZ00,BM98,BM99,CS02,GT01]. In this work, we will restrict our attention to graph properties and geometric properties of point sets.

Following Goldreich, Goldwasser, and Ron [GGR98], we measure the distance between graphs by the Hamming distance between their adjacency matrices. That is, the distance between two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ on n vertices is defined as the fraction of edges on which G_1 and G_2 differ: $|E_1 \cup E_2 - E_1 \cap E_2| / \binom{n}{2}$. The properties considered in [GGR98] include Bipartite, the property of being bipartite; ρ-Clique, the property of having a clique of size at least ρn; and ρ-Bisection, the property of having a bisection crossed by fewer than ρn^2 edges. For these properties, they prove:

Theorem 1 (Goldreich-Goldwasser-Ron). *The properties ρ-Clique and ρ-Bisection have property testing algorithms with query complexity polynomial in $1/\epsilon$ and time complexity $2^{\tilde{O}(1/\epsilon^3)}$, and the property Bipartite has a property testing algorithm with query and time complexities polynomial in $1/\epsilon$.*

We remark that Goldreich and Trevisan [GT01] have shown that every graph property that can be tested by making a number of queries that is independent of the size of the graph, can also be tested by uniformly selecting a subset of vertices and accepting if and only if the induced subgraph has some fixed graph property (which is not necessarily the same as the one being tested).

We now state a lemma that relates the smoothed error probability of a testing algorithm with the probability that the property-preserving perturbation of an input is far from one having the property.

Lemma 1. *Let P be a property and A a testing algorithm for P with query complexity $q(1/\epsilon)$ and time complexity $T(1/\epsilon)$ such that*

$$\mathbf{Pr}\left[A(X) \neq P(X)\right] < 1/3,$$

for all inputs X that either have property P or have distance at least ϵ from those having property P. Then, if $\mathcal{P}(\bar{X}, \sigma)$ is a family of distributions such that for all $\bar{X}$ lacking property P,

$$\mathbf{Pr}_{X \leftarrow \mathcal{P}(\bar{X},\sigma)}\left[X \textit{ is } \epsilon\textit{-close to } P | P(X) = P(\bar{X})\right] \leq \lambda(\epsilon, \sigma, n),$$

then for all inputs $\bar{X}$,

$$\mathbf{Pr}_{X \leftarrow \mathcal{P}(\bar{X},\sigma)}\left[A(X) \neq P(X) | P(X) = P(\bar{X})\right] < 1/3 + \lambda(\epsilon, \sigma, n).$$

2 Smoothed Error Bound for Graph Property Testers

In this section, we prove that the ρ-Clique, ρ-Bisection and Bipartite property testers of [GGR98] may be viewed as sub-linear-time decision algorithms with low smoothed error probability under the corresponding property-preserving perturbations.

Lemma 2. *Let $\bar{G}$ be a graph on n vertices, let $\rho < 1/8$, and let $\sigma < 1/2$. If G is the ρ-Bisection preserving σ-perturbation of $\bar{G}$, then*

1. *if $\bar{G}$ has a ρ-Bisection, then G has a ρ-Bisection with probability 1, and*
2. *if $\bar{G}$ does not have a ρ-Bisection, then for any $\epsilon < \sigma(1/4 - 2\rho)$*

$$\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[\begin{array}{r} G \textit{ is } \epsilon\textit{-close to a graph with a } \rho\textit{-Bisection} \\ | \; G \textit{ does not have a } \rho\textit{-Bisection} \end{array}\right] < 2^{-\Omega(n^2)}.$$

Proof. The first part follows from the definition of a ρ-Bisection preserving perturbation.

To prove the second part, we first observe that G is ϵ-close to a graph with a ρ-Bisection if and only if G has a $(\rho+\epsilon)$-Bisection. We express the probability of this event in the property-preserving model as

$$\begin{aligned} &\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[G \text{ has a } (\rho+\epsilon)\text{-Bisection} \middle| G \text{ does not have a } \rho\text{-Bisection}\right] \\ &\qquad \leq \frac{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[G \text{ has a } (\rho+\epsilon)\text{-Bisection}\right]}{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[G \text{ does not have a } \rho\text{-Bisection}\right]}. \end{aligned} \tag{1}$$

We now proceed to bound these probabilities. If we flip every edge and non-edge of G with probability σ, then for every partition of the vertices of $\bar{G}$ into two equal-sized sets the expected number of edges crossing this partition in G is at least

$$(1-\sigma)\rho n^2 + \sigma(1/4-\rho)n^2.$$

Applying a Chernoff bound (see for example [MR97, Theorem 4.2]), we find the probability that there are fewer than $(\rho+\epsilon)n^2$ edges crossing this partition is at most

$$e^{-n^2 \frac{(\sigma(1/4-2\rho)-\epsilon)^2)}{\rho+\sigma(1/4-2\rho)}} = 2^{-\Omega(n^2)}.$$

As there are fewer than 2^n partitions, we may plug this inequality into (1) to conclude the proof.

The proofs of the following two lemmas for Bipartite and Clique are similar.

Lemma 3. *Let $\bar{G}$ be a graph of n vertices. If $\epsilon > 0$ and $\epsilon/\rho^2 < \sigma < 1/2$, and if G is the ρ-Clique preserving σ-perturbation of $\bar{G}$, then*

1. *if $\bar{G}$ is has a ρ-Clique, then G has a ρ-Clique with probability 1, and*
2. *if $\bar{G}$ does not have a ρ-Clique, then for any $\epsilon < \sigma(1/4-2\rho)$*

$$\mathbf{Pr}\left[\begin{array}{r} G \text{ is } \epsilon\text{-close to a graph with a } \rho\text{-Clique} \\ \mid G \text{ does not have a } \rho\text{-Clique} \end{array}\right] < 2^{-\Omega(n^2)}.$$

Lemma 4. *Let $\bar{G}$ be a graph of n vertices and let $0 < \epsilon < \sigma/4 < 1/8$. If G is the bipartite-preserving σ-perturbation of $\bar{G}$, then*

1. *if $\bar{G}$ is bipartite, then G is bipartite with probability 1, and*
2. *if $\bar{G}$ is not bipartite, then*

$$\mathbf{Pr}\left[G \text{ is } \epsilon\text{-close to bipartite} | G \text{ is not bipartite}\right] < 2^{-\Omega(n^2)}.$$

Remark 1. Bipartite and Clique differ from Bisection in this model as their natural testers have simple proofs of correctness in the smoothed model. In contrast, we are unaware of a means of proving the correctness of the Bisection tester that does not go through the machinery of [GGR98]. This seems to be related to the fact that we can find exponentially faster testers for Clique in this model.

Using Lemma 1 to combine Theorem 1 with Lemmas 2, 3 and 4, we obtain:

Theorem 2. *Let P be one of Bipartite, ρ-Clique, or ρ-Bisection. There exists an algorithm A that takes as input a graph G, examines poly$(1/\sigma)$ edges of G and runs in time $\tilde{O}(1/\epsilon^3)$ when P is Bipartite, and in $2^{\tilde{O}(1/\epsilon^2)}$ time when P is ρ-Clique or ρ-Bisection such that for every $\bar{G}$, if G is the P-property preserving σ-perturbation of $\bar{G}$, then*

$$\mathbf{Pr}\left[A(G) \neq P(G)\right] < 1/3 + o(1).$$

In the next section, we improve the time complexity of ρ-Clique testing under ρ-Clique preserving σ-perturbations.

3 A Fast Clique Tester

In this section we will consider a tester for ρ-Clique that samples a random set of k vertices and accepts if these vertices contain a $\rho k/2$ clique. In Lemma 5 we prove that this tester rarely accepts a graph without a ρ-Clique under ρ-Clique preserving σ-perturbations. The other lemmas of the section are devoted to adapting the machinery of Feige and Killian [FK98a] to quickly finding the $\rho k/2$ clique when it is present in the graph.

Theorem 3 (Fast Clique Tester). *Let ρ and $\sigma < 1/2$ be constants. There exists an algorithm A that takes as input a graph G, examines the induced subgraph of G on a randomly chosen set of $\frac{8}{\rho\sigma} \log\left(\frac{4}{\rho\sigma}\right)$ vertices of G and runs in time polynomial in $\frac{1}{\rho\sigma}$ such that for every graph $\bar{G}$, if G is the ρ-Clique preserving σ-perturbation of $\bar{G}$, then*

$$\mathbf{Pr}\left[A(G) \neq \rho\text{-}Clique(G)\right] < 1/4 + o(1).$$

In contrast, Goldreich, Goldwasser and Ron [GGR98] prove that the existence of a tester with such worst-case complexity would imply $NP \subseteq BPP$.

Proof. The algorithm A runs the algorithm of Lemma 8 below and accepts if it finds a clique of size at least $\rho k/2$. If $\bar{G}$ does not contain a ρ-Clique, then by Lemma 5 below the probability this algorithm will accept is at most $e^{-8}+o(1) \leq 1/4 + o(1)$.

On the other hand, if $\bar{G}$ does contain a ρ-Clique, We can apply Lemma 8 to show that

$$\begin{aligned}
\mathbf{Pr}_{\mathcal{Q}(\bar{G},\sigma)}\left[A(G) \text{ rejects}\right] &= \sum_S w(S)\mathbf{Pr}_{\mathcal{Q}(\bar{G},S,\sigma)}\left[A(G) \text{ rejects}\right] \\
&\leq \sum_S w(S)(1/4 + o(1)) \leq 1/4 + o(1).
\end{aligned}$$

The theorem then follows from Lemma 9 below which implies

$$\left|\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[A(G) \text{ accepts}|\rho\text{-Clique}(G)\right] - \mathbf{Pr}_{\mathcal{Q}(\bar{G},\sigma)}\left[A(G) \text{ accepts}\right]\right| < o(1).$$

The next lemma states that the tester is unlikely to accept if G does not contain a ρ-Clique.

Lemma 5. *Let $\bar{G}$ be a graph without a ρ-Clique and let G be the ρ-Clique preserving σ-perturbation of G. Let U be a randomly chosen subset of k vertices of G for $k \geq \frac{8}{\rho\sigma} \log\left(\frac{4}{\rho\sigma}\right)$. Then,*

$$\mathbf{Pr}\left[\textit{the vertices of } U \textit{ contain a } \rho k/2 \textit{ clique in } G\right] < e^{-8} + o(1).$$

Proof. We begin by observing that

$$\mathbf{Pr}_{U,G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[\begin{array}{r}\text{the vertices of } U \text{ contain a } \rho k/2 \text{ clique in } G\\ \mid G \text{ does not contain a } \rho n \text{ clique}\end{array}\right]$$
$$\leq \frac{\mathbf{Pr}_{U,G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[\text{the vertices of } U \text{ contain a } \rho k/2 \text{ clique in } G\right]}{1-\mathbf{Pr}_{G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[G \text{ contains a } \rho n \text{ clique}\right]}$$
$$\leq \mathbf{Pr}_{U,G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[\text{the vertices of } U \text{ contain a } \rho k/2 \text{ clique in } G\right] + o(1),$$

by Lemma 6.

To bound the last probability, we note that the probability that any particular set of $\rho k/2$ nodes in G is a clique is at most $(1-\sigma)^{\binom{\rho k/2}{2}}$ and that U contains $\binom{k}{\rho k/2}$ sets of $\rho k/2$ nodes, so

$$\begin{aligned}\mathbf{Pr}_{U,G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[\begin{array}{l}\text{the vertices of } U \text{ contain}\\ \text{a } \rho k/2 \text{ clique in } G\end{array}\right] &\leq \binom{k}{\rho k/2}(1-\sigma)^{\binom{\rho k/2}{2}}\\ &\leq \left(\frac{2e}{\rho}\right)^{\rho k/2} e^{-\sigma\binom{\rho k/2}{2}}\\ &\leq e^{\frac{\rho k}{2}\left(\ln\left(\frac{2e}{\rho}\right)-\sigma\left(\frac{\rho k-2}{4}\right)\right)}\\ &\leq e^{-\rho k} \leq e^{-8}.\end{aligned}$$

as $k \geq \frac{8}{\rho\sigma}\log\left(\frac{4}{\rho\sigma}\right)$ and $\sigma < 1$.

Lemma 6. *Let $\bar{G}$ be a graph without a ρn-Clique and let G be the σ-perturbation of G. Then,*

$$\mathbf{Pr}_{G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[G \textit{ contains a } \rho\textit{-Clique}\right] = 2^{-\Omega(n^2)}.$$

Proof. There are fewer than 2^n sets of ρn nodes, and the probability that any particular such set is a clique in G is at most $(1-\sigma)^{\binom{\rho n}{2}}$.

Lemma 7. *Let $\bar{G}$ be a graph that has a ρ-Clique. Then,*

$$\mathbf{Pr}_{G\leftarrow\mathcal{P}(\bar{G},\sigma)}\left[G \textit{ has at least two } \rho\textit{-Cliques} | G \textit{ has one } \rho\textit{-Clique}\right] \leq 2^{-\Omega(n)}.$$

Proof. By inclusion-exclusion,

$$\begin{aligned}&\mathbf{Pr}\left[G \text{ has one } \rho\text{-Clique}\right]\\ &\qquad \geq \sum_{|S_1|=\rho n}\mathbf{Pr}\left[K_{S_1}\subseteq G\right] - \sum_{|S_1|=|S_2|=\rho n}\mathbf{Pr}\left[K_{S_1}\subseteq G \text{ and } K_{S_2}\subseteq G\right],\end{aligned}$$

and

$$\mathbf{Pr}\left[G \text{ has at least two } \rho\text{-Cliques}\right] \geq \sum_{|S_1|=|S_2|=\rho n}\mathbf{Pr}\left[K_{S_1}\subseteq G \text{ and } K_{S_2}\subseteq G\right].$$

Therefore,

$$\begin{aligned}
&\mathbf{Pr}\left[G \text{ has at least two } \rho\text{-Cliques} | G \text{ has one } \rho\text{-Clique}\right] \\
&\leq \frac{\sum_{|S_1|=|S_2|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and } K_{S_2} \subseteq G\right]}{\sum_{|S_1|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G\right] - \sum_{|S_1|=|S_2|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and } K_{S_2} \subseteq G\right]} \\
&\leq \max_{|S_1|=\rho n} \frac{\sum_{|S_2|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and } K_{S_2} \subseteq G\right]}{\mathbf{Pr}\left[K_{S_1} \subseteq G\right] - \sum_{|S_2|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and } K_{S_2} \subseteq G\right]}
\end{aligned}$$

We now prove the lemma by demonstrating that for all $|S_1| = \rho n$,

$$\begin{aligned}
&\frac{\sum_{|S_2|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and } K_{S_2} \subseteq G\right]}{\mathbf{Pr}\left[K_{S_1} \subseteq G\right]} \\
&= \sum_{k=1}^{\rho n} \sum_{|U|=|V|=k} \frac{\mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and} K_{S_1 \setminus U \cup V} \subseteq G\right]}{\mathbf{Pr}\left[K_{S_1} \subseteq G\right]} \\
&\leq \sum_{k=1}^{\rho n} \binom{\rho n}{k} \binom{n - \rho n}{k} (1-\sigma)^{k(\rho n - k) + \binom{k}{2}} \\
&= 2^{-\Omega(n)},
\end{aligned}$$

where the last inequality follows from the fact that $k(\rho n - k) + \binom{k}{2}$ is an increasing function in k, and for $k \leq \rho n/2$, the terms in the sum decrease as k increases. In addition, when $k = \rho n/2$, $(1-\sigma)^{k(\rho n - k) + \binom{k}{2}} = 2^{-\Omega(n^2)}$. Therefore, the first term in the sum dominates, and hence the sum is no more than $2^{-\Omega(n)}$.

Feige and Kilian [FK98a] design a polynomial-time algorithm for finding cliques in random graphs with planted cliques which may be modified in a limited fashion by an adversary. A corollary of their work is that if one takes a graph with a large clique and then perturbs the edges not involved in the clique, then with high probability their algorithm will find the large clique. To facilitate the rigorous statement of this corollary and the application of their result to the smoothed model, we introduce the following notation:

Definition 3. *For a graph $\bar{G}$, a subset of its vertices S and σ between 0 and $1/2$, we define $\mathcal{Q}(\bar{G}, S, \sigma)$ to be the distribution on graphs obtained by sampling from $\mathcal{P}(\bar{G}, \sigma)$ and adding edges to create a clique among the nodes in S.*

For a graph $\bar{G}$ and a σ between 0 and $1/2$, we define $\mathcal{Q}(\bar{G}, \sigma)$ to be the distribution obtained by choosing a set S of vertices of size ρn with probability $w(S)$ and then sampling from $\mathcal{Q}(\bar{G}, S, \sigma)$ where

$$w(S) = \frac{\mu(S)}{\sum_{T:|T|=|S|} \mu(T)},$$

and

$$\mu(S) = \prod_{i,j} \sigma^{[(i,j) \notin \bar{G}]} (1-\sigma)^{[(i,j) \in \bar{G}]}.$$

Theorem 4 (Feige-Kilian). *For any positive constant ρ, there is a randomized polynomial time algorithm that with probability $1 - o(1)$ will find a clique of size ρn in a graph G drawn from the distribution $\mathcal{Q}(\bar{G}, S, \sigma)$ where S is a subset of the vertices of $\bar{G}$ of size ρn and $\sigma \geq 2 \ln n / \rho n$.*

From this theorem, we derive

Lemma 8. *Let $\rho > 0$ and let G be drawn from the distribution $\mathcal{Q}(\bar{G}, S, \sigma)$ where S is a subset of the vertices of $\bar{G}$ of size ρn and $1/2 \geq \sigma \geq 2 \ln n / \rho n$. Let U be a random subset of k vertices of G where $k = \min\left(k_0, \frac{8}{\rho\sigma} \log\left(\frac{4}{\rho\sigma}\right)\right)$, where k_0 is some absolute constant. Then, with probability $3/4 - o(1)$ the algorithm of Theorem 4 finds a clique of size at least $\rho k/2$ in the graph induced by G on U.*

Proof. We first note that the probability that U contains fewer than $\rho k/2$ vertices of S is at most

$$e^{-\rho k/8} + o(1) \leq e^{-3} + o(1)$$

as $\log\left(\frac{4}{\rho\sigma}\right) \geq 3$ and $\rho, \sigma < 1$.

Given that there are at least $\rho k/2$ points of S in U, the probability that the algorithm of Theorem 4 fails is at most $1/8$, provided that $\sigma > 2 \log k / (\rho k/2)$, which follows from our setting of $k \geq \frac{8}{\rho\sigma} \log\left(\frac{4}{\rho\sigma}\right)$, and that k is larger than some absolute constant, k_0. Thus, the failure probability is at most $e^{-3} + 1/8 + o(1) \leq 1/4 + o(1)$.

To transfer the result of Lemma 8 to graphs produced by ρ-Clique preserving σ-perturbations of graphs with ρ-Cliques, we show:

Lemma 9. *Let $\bar{G}$ be a graph with a ρ-Clique and $\sigma < 1/2$. Then,*

$$\sum_{G} \left| \mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)} \left[G | G \textit{ has a } \rho\textit{-Clique} \right] - \mathbf{Pr}_{\mathcal{Q}(\bar{G},\sigma)} \left[G \right] \right| < 2^{-\Omega(n)}.$$

Proof. For any graph G, we apply inclusion-exclusion to compute

$$\begin{aligned}
\frac{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}[G]}{\sum_{S:|S|=\rho n} \mu(S)} &\leq \mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}[G | G \text{ contains a } \rho n\text{-Clique}] \\
&\leq \frac{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}[G]}{\sum_{S:|S|=\rho n} \mu(S) - \sum_{|S_1|=|S_2|=\rho n} \mathbf{Pr}\left[K_{S_1} \subseteq G \text{ and} K_{S_2} \subseteq G\right]} \\
&\leq \frac{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}[G]}{\sum_{S:|S|=\rho n} \mu(S)} \left(1 + 2^{-\Omega(n)}\right),
\end{aligned}$$

by Lemma 7.

On the other hand,

$$\begin{aligned}
\mathbf{Pr}_{\mathcal{Q}(\bar{G},\sigma)}\left[G\right] &= \sum_{S:K_S\subseteq G,|S|=\rho n} \frac{\mu(S)}{\sum_{|T|=\rho n}\mu(T)}\mathbf{Pr}\left[G|K_S\subseteq G\right] \\
&= \sum_{S:K_S\subseteq G,|S|=\rho n} \frac{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[G\right]}{\sum_{|T|=\rho n}\mu(T)} \\
&= (\#\ \rho\text{-Cliques in } G)\,\frac{\mathbf{Pr}_{\mathcal{P}(\bar{G},\sigma)}\left[G\right]}{\sum_{|T|=\rho n}\mu(T)}.
\end{aligned}$$

We now conclude the proof by observing that if G has no ρn cliques then both probabilities are zero, if G has one ρn clique then the probabilities differ by at most a multiplicative factor of $(1+2^{-\Omega(n)})$, and, by Lemma 7, the probability under $\mathcal{P}(\bar{G},\sigma)$ that there are two ρn cliques is at most $2^{-\Omega(n)}$.

4 Discussion

4.1 Condition Numbers and Instance-Based Complexity

To obtain a finer analysis of algorithms for a problem than that provided by worst-case complexity, one should find a way of distinguishing hard problem instances from easy ones. A natural approach is to find a quantity that may be associated with a problem instance and which is indicative of the difficulty of solving that instance. For example, it is common in Numerical Analysis and Operations Research to bound the running time of an algorithm in terms of a condition number of its input. The condition number is typically defined to be the the reciprocal of the distance of the input to one on which the problem is ill-posed, or the sensitivity of the solution of a problem to slight perturbations of the input.

Thus, one can view the effort to measure the complexity of testing whether or not an input has a property in terms of its distance from having the property if it does not as being very similar. In fact, the perturbation distance used by Czumaj and Sohler [CS01] is precisely the the reciprocal of the condition number of the problem. Moreover, the natural definition of the condition number for a discrete function—the reciprocal of the minimum distance of an input to one on which the function has a different value—is precisely the measure of complexity used in the study of property testing: the larger the condition number the harder the testing.

In fact, in many smoothed analyses [BD02,DST02,ST03], an essential step has been the smoothed analysis of a condition number.

References

[ADPR00] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. In *41st Annual Symposium on Foundations of Computer Science*, pages 240–250. IEEE, 2000.

[AKFS99] N. Alon, M. Krivelevich, E. Fischer, and M. Szegedy. Efficient testing of large graphs. In *40th Annual Symposium on Foundations of Computer Science,*, pages 656–666. IEEE, 1999.

[Alo01] N. Alon. Testing subgraphs in large graphs. In IEEE, editor, *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science: proceedings*, pages 434–439, 2001.

[BD02] Avrim Blum and John Dunagan. Smoothed analysis of the perceptron algorithm for linear programming. In *Proceedings of the 13th Annual ACM-SIAM Symposium On Discrete Mathematics (SODA-02)*, pages 905–914. ACM Press, 2002.

[BM98] P. Bose and P. Morin. Testing the quality of manufactured disks and cylinders. In *ISAAC: 9th International Symposium on Algorithms and Computation*, volume 1533 of *Lecture Notes in Computer Science*, pages 129–138, 1998.

[BM99] Prosenjit Bose and Pat Morin. Testing the quality of manufactured balls. In *Workshop on Algorithms and Data Structures*, pages 145–156, 1999.

[Bop87] Ravi Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 28th Symposium on Foundation of Computer Science*, pages 280–285, 1987.

[BR00] Michael A. Bender and Dana Ron. Testing acyclicity of directed graphs in sublinear time. In *Automata, Languages and Programming*, pages 809–820, 2000.

[BS95] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *J. Algorithms*, 19(2):204–234, 1995.

[CO02] Amin Coja-Oghlan. Finding sparse induced subgraphs of semirandom graphs. In *Randomization and Approximation Techniques in Computer Science, 2002*, volume 2483 of *Lecture Notes in Computer Science*, pages 139–148. Springer, 2002.

[CS01] A. Czumaj and C. Sohler. Property testing with geometric queries. In *Proceedings of the 9th Annual European Symposium on Algorithms*, volume 2161 of *Lecture Notes in Computer Science*, pages 266–277. Springer-Verlag, 2001.

[CS02] A. Czumaj and C. Sohler. Abstract combinatorial programs and efficient property testers. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 83–92, 2002.

[CSZ00] Artur Czumaj, Christian Sohler, and Martin Ziegler. Property testing in computational geometry. In *European Symposium on Algorithms*, pages 155–166, 2000.

[DGL+99] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonocity. In *Proceedings of RANDOM*, pages 97–108, 1999.

[DST02] John Dunagan, Daniel A. Spielman, and Shang-Hua Teng. Smoothed analysis of interior point methods: Condition numbers. available at `http://arxiv.org/abs/cs.DS/0302011`, 2002.

[EKK+98] Funda Ergun, Sampath Kannan, S. Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. In *ACM Symposium on Theory of Computing*, pages 259–268, 1998.

[FK98a] U. Feige and J. Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 674–683. IEEE, 1998.

[FK98b] Uri Feige and Robert Krauthgamer. Improved performance guarantees for bandwidth minimization heuristics. Unpublished manuscript, 1998.

[GGLR98] Oded Goldreich, Shafi Goldwasser, Eric Lehman, and Dana Ron. Testing monotonicity. In *IEEE Symposium on Foundations of Computer Science*, pages 426–435, 1998.

[GGR98] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, July 1998.

[GR97] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 406–415, 1997.

[GR98] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. In ACM, editor, *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing*, pages 289–298. ACM Press, 1998.

[GT01] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 460–469, 2001.

[KR00] Michael Kearns and Dana Ron. Testing problems with sublearning sample complexity. *J. of Comput. Syst. Sci.*, 61(3):428–456, December 2000.

[MR97] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1997.

[PR99] Michal Parnas and Dana Ron. Testing the diameter of graphs. In *Random Structures and Algorithms*, volume 1671 of *Lecture Notes in Computer Science*, pages 85–96, 1999.

[Ron01] D. Ron. Property testing. In *Handbook on Randomized Computing (Vol. II)*. Kluwer Academic Publishers, 2001.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, April 1996.

[ST01] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC '01)*, pages 296–305, 2001.

[ST02] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms. In *Proceedings of the International Congress of Mathematicians*, volume 1, 2002. to appear.

[ST03] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of termination of linear programming algorithms. *Mathematical Programming B*, 2003. to appear.