Theoretical Computer Science 576 (2015) 18–29

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the average-case complexity of parameterized clique

Nikolaos Fountoulakis [a], Tobias Friedrich [b,*], Danny Hermelin [c]

[a] University of Birmingham, Edgbaston, United Kingdom
[b] Hasso Plattner Institute, Potsdam, Germany
[c] Ben-Gurion University, Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 14 February 2013
Received in revised form 23 October 2014
Accepted 23 January 2015
Available online 19 February 2015
Communicated by V.Th. Paschos

Keywords:
Parameterized complexity
Computational complexity
Average-case
Clique

ABSTRACT

The k-Clique problem is a fundamental combinatorial problem that plays a prominent role in classical as well as in parameterized complexity theory. It is among the most well-known NP-complete and W[1]-complete problems. Moreover, its average-case complexity analysis has created a long thread of research already since the 1970s. Here, we continue this line of research by studying the dependence of the average-case complexity of the k-Clique problem on the parameter k. To this end, we define two natural parameterized analogs of efficient average-case algorithms. We then show that k-Clique admits both analogues for Erdős–Rényi random graphs of *arbitrary* density. We also show that k-Clique is unlikely to admit either of these analogs for some specific computable input distribution.

1. Introduction

The k-Clique problem is one of the most fundamental combinatorial problems in graph theory and computer science. This problem asks to determine whether a given graph contains a clique of size k, *i.e.* a complete subgraph on k vertices. The k-Clique problem forms the groundwork for many worst-case hardness frameworks: It is one of Karp's famous initial lists of NP-complete problems [11], and its optimization variant is a classical example of a problem that is NP-hard to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [20]. In parameterized complexity theory [4], the k-Clique problem is textbook example complete for the class W[1], the parameterized analogue of NP, playing a prominent role in W[1]-hardness results very much akin to the role 3-SAT plays in the classical complexity.

In this paper we are interested in the parameterized complexity of the k-Clique problem on "average" inputs. For our purposes, an average k-Clique instance can be naturally and conveniently modeled using the thoroughly-studied Erdős–Rényi distributions on graphs. The class of these distributions is typically denoted by $\mathcal{G}(n, p)$, with $n \in \mathbb{N}$ and $p \in [0, 1]$, where on a graph with n vertices each pair of vertices are adjacent independently with probability p. Such random graphs have approximate density p, and it is well-known (see *e.g.* [1,10]) that the typical properties of these random graphs are essentially the typical properties of a random graph that is uniformly selected among all graphs on n vertices and $p\binom{n}{2}$ edges.

The question of finding cliques in $\mathcal{G}(n, p)$ random graphs has been raised by Karp [12] already in 1976. Karp observed that in $\mathcal{G}(n, 1/2)$ (note that this is in fact the uniform distribution over all graphs on n vertices) the maximum size of a clique is about $2 \log n$ with high probability, but the greedy algorithm only finds with high probability a clique that is approximately half this size. Karp asked whether in fact there is any polynomial-time algorithm that finds a clique of size $(1 + \varepsilon) \log n$, for some $\varepsilon > 0$. This question remains open until today.

* Corresponding author.

http://dx.doi.org/10.1016/j.tcs.2015.01.042
0304-3975/

Finding cliques in $\mathcal{G}(n, p)$ random has also been considered when the clique sought after have small size, which is the main theme of our paper. For a fixed integer $k \geq 3$, the random graph $\mathcal{G}(n, p)$ undergoes a phase transition regarding the (almost sure) existence of cliques of size k (cf. [1] or [10]) as the edge probability p grows. More specifically, it is known that when $p \ll n^{-2/(k-1)}$, then $\mathcal{G}(n, p)$ does not contain any cliques of size k, with high probability, but when $p \gg n^{-2/(k-1)}$, then in fact there are many k-cliques with high probability. However, inside the "critical window", that is when $p = \Theta(n^{-2/(k-1)})$, the maximum size of a clique could be either $k-1$ or k each one occurring with probability that is bounded away from 0 as n grows to infinity. More precisely, the number of cliques of size k follows asymptotically a Poisson distribution with parameter that depends on k. In this range, the greedy algorithm finds a clique of size $\lfloor \frac{k}{2} \rfloor$ or $\lceil \frac{k}{2} \rceil$, with high probability. Rossman [17, Remark 35] remarks that repeating the greedy algorithm $n^{k/4+O(1)}$ times, we can enumerate all the maximal cliques with high probability. This gives a randomized algorithm with runtime $n^{k/4+O(1)}$ for solving k-Clique with high probability.

Since the above algorithm is the fastest algorithm known, it seems that a typical instance of $\mathcal{G}(n, p)$ with $p = \Theta(n^{-2/(k-1)})$ is in fact a hard instance for k-Clique. This is also suggested by the lower bounds on the size of monotone circuits for k-Clique derived recently by Rossman [17] (see also [16]) for p in this range. Thus any substantial improvement to the $n^{k/4+O(1)}$ algorithm above would be a major breakthrough result; not to mention an FPT algorithm running in $f(k) \cdot n^{O(1)}$ time, which is perhaps far too much of an improvement than we can expect.[1] To avoid this obstacle, we consider distributions $\mathcal{G}(n, p)$ where p does not depend on k (but may depend on n). Apart from the obvious advantage that this gives a real chance at obtaining positive results, we also believe that this a very natural model of practical settings. Indeed, in many cases the distribution of the graphs we are interested in is fixed, while the size of the cliques we are looking for may vary.

We consider two types of algorithms running in FPT time on average. The first is an avgFPT-algorithm, which is an algorithm with expected $f(k) \cdot n^{O(1)}$ run-time. Thus, an avgFPT-algorithm is required to run in FPT-time on average according to the given input distribution. This means that the algorithm is allowed to be slow on some instances, so long as that its efficient on average. The notion of avgFPT-time is a natural parameterized analogue of an avgP-time algorithm (see *e.g.* [7]), and is perhaps the most natural definition of the notion "FPT on average".

We present a very simple avgFPT algorithm for k-Clique for essentially all distributions $p := p(n)$. By essentially, we mean all *natural* distributions that have typical properties, such as certain limit properties (this is made precise in Definition 5). The first result of this paper is thus the following theorem.

Theorem 1. *Let $p := p(n)$ denote a natural distribution function. There is an* avgFPT-*algorithm for k-*Clique *on graphs $G \in \mathcal{G}(n, p)$.*

The second type of average-case FPT algorithms we consider are algorithms that run in typical FPT (typFPT) time. By this we mean a running time of $f(k) \cdot n^{O(1)}$ with high probability, where high probability means that the algorithm is allowed to be slower only with probability smaller than any polynomial in n. Thus, one may view the difference between a typFPT-time algorithm and an avgFPT-time algorithm is that an avgFPT-time algorithm is allowed to be slightly slow on relatively many instances, while a typFPT-time algorithm is allowed to be extremely slow on relatively few instances. In stochastic terms, this is precisely the difference between bounding the expected value of a random variable and showing that it is bounded with high probability. Again, the analogous notion in classical complexity is typical P-time [7].

We show that the same algorithm used in Theorem 1 is actually a typFPT algorithm for k-Clique for any natural $p := p(n)$. However, the proof of this result is more involved than the former and requires a rather sophisticated tail bound argument.

Theorem 2. *Let $p := p(n)$ denote a natural distribution function. There is a* typFPT-*algorithm for k-*Clique *on graphs $G \in \mathcal{G}(n, p)$.*

It is worth mentioning that in both theorems above, our algorithms are completely deterministic and *always* correctly decide whether their input graph contains a clique of size k. This makes the proofs more challenging, since the algorithms cannot only assume that a k-clique is unlikely to exist in the input, but they must also certify this somehow. Furthermore, our algorithms can easily be modified to determining whether a $\mathcal{G}(n, p)$ random graph has an independent set of size k. Moritz Müller's PhD thesis [15] provides the first attempt at setting up a framework of parameterized average case complexity. In particular, he defined a notion very much similar to our avgFPT-algorithm, except that in his case the algorithm is allowed to have one-sides errors with constant probability. The notion of typFPT has not appeared elsewhere to the best of our knowledge. The distinction between these two types of average-case tractability notions is standard in the classical world, and in Section 2 we briefly argue why this distinction makes even more sense in the parameterized world. Müller also defined an average-case analogue of W[1], and showed that there is some (artificial) problem which is complete for it. We discuss this result in the last part of the paper, and show that the k-Clique problem is hard for this average-case analogue of W[1] on a specific distribution.

[1] Note that $f(k) \cdot n^{O(1)} \ll n^k$ for any function f, when k is fixed and n tends to infinity.