Electronic Colloquium on Computational Complexity, Revision 3 of Report No. 104 (2020)

On Counting t-Cliques Mod 2

Oded Goldreich*

July 16, 2020

Abstract

For a constant $t \in \mathbb{N}$, we consider the problem of counting the number of t-cliques *mod 2* in a given graph. We show that this problem is not easier than determining whether a given graph contains a t-clique, and present a simple worst-case to average-case reduction for it. The reduction runs in linear time when graphs are presented by their adjacency matrices, and average-case is with respect to the uniform distribution over graphs with a given number of vertices.

Correction (July 16th, 2020): It turns out that the foregoing results were previously obtained by Boix-Adsera, Brennan, and Bresler (FOCS'19), using a slightly more complex worst-case to average-case reduction.

1 Informal description

For a constant integer $t \geq 3$, finding t-cliques in graphs and determining their mere existence are archetypical computational problems within the frameworks of parameterized complexity and fine grained complexity (see, e.g., [FG06] and [W15], resp.). The complexity of counting the number of t-cliques has also been studied (see, e.g., [GR18, BBB19]). In this work, we consider a variant of the latter problem; specifically, the problem of counting the number of t-cliques mod 2.

Determining the number of t-cliques *mod 2* in a given graph is potentially easier than determining the number of t-cliques in the same graph. On the other hand, as shown in Theorem 1, determining the said number mod 2 is not easier (in the worst-case sense) than determining whether or not a graph contains a t-clique. Hence, the worst-case complexity of *counting t-cliques mod 2* lies between the worst-case complexity of *counting t-cliques* and the worst-case complexity of *determining the existence of t-cliques*. Consequently, as far as worst-case complexity is concerned, using the "counting mod 2 problem" as proxy for the "existence problem" is at least as justified as using the "counting problem" as such a proxy.

Our main result (presented in Theorem 2) is an efficient worst-case to average-case reduction for *counting t-cliques mod 2*. The reduction in efficient in the sense that it runs in linear time when graphs are presented by their adjacency matrices. Average-case is with respect to the uniform distribution over graphs with a given number of vertices, and it yields the correct answer (with high probability) whenever the average-case solver is correct on at least a $1 - 2^{-t^2}$ fraction of the instances. In other words, the average-case solver has error rate at most 2^{-t^2}. The question of whether the same result holds with respect to significantly higher error rates, and ultimately with error rate 0.49, is left open.

*Department of Computer Science, Weizmann Institute of Science, Rehovot, ISRAEL. E-mail: oded.goldreich@weizmann.ac.il

ISSN 1433-8092

Relation and comparison to prior work. Efficient worst-case to average-case reductions were presented before for the related problem of *counting t-cliques* (over the integers). Specifically, Goldreich and Rothblum provided such a reduction with respect to a relatively simple distribution over graphs with a given number of vertices, alas not the uniform distribution [GR18]. On the other hand, their reduction works even when the average-case solver has error rate that approaches 1; specifically, its error rate on n-vertex graphs may be as large as $1 - \frac{1}{\mathrm{poly}(\log n)} = 1 - o(1)$. In contrast, Boix-Adsera, Brennan, and Bresler provided an efficient worst-case to average-case reduction with respect to the uniform distribution, but their reduction can only tolerate a vanishing error rate [BBB19, Thm. II.8]; specifically, its error rate on n-vertex graphs is required to be $1/\mathrm{poly}(\log n) = o(1)$.

Hence, our worst-case to average-case reduction, which is for a related (but different) problem, matches the better aspects of the aforementioned results (see Table 1): It refers to the uniform distribution (as [BBB19, Thm. II.8]), and tolerates a constant error rate (which is better than [BBB19, Thm. II.8] but worse than [GR18]).

As stated in the abstract, it turns out that a similar result was proved before by Boix-Adsera, Brennan, and Bresler [BBB19, Thm. II.9], using a conceptually similar but slightly more complicated reduction (which is due to their obtaining this result by modifying the approach they used to obtain their other results).

problem	distribution	error rate	where
counting	relatively simple	$1 - 1/\mathrm{poly}(\log n) = 1 - o(1)$	[GR18]
counting	uniform	$1/\mathrm{poly}(\log n) = o(1)$	[BBB19, Thm. II.8]
counting mod 2	uniform	$\exp(-\widetilde{O}(t^2)) = \Omega(1)$	[BBB19, Thm. II.9]
counting mod 2	uniform	$\exp(-t^2) = \Omega(1)$	Theorem 2

Table 1: *Comparison of different worst-case to average-case reductions for variants of the t-*`CLIQUE` *problem, for the constant t, where n denotes the number of vertices. The first column indicates the version being treated, the second indicates the distribution for which average-case is considered, and the third indicates the error rate allowed for the average-case solver.*

Techniques. In contrast to [GR18, BBB19], which relate the t-clique counting problem to the evaluation of lower degree polynomials over large and medium sized fields, we related the counting *mod 2* problem to low degree polynomials over GF(2). This relation allows us to present reductions that are much simpler than those presented in [GR18, BBB19].

As noted above, we leave open the problem of improving the error rate that can be tolerated by a worst-case to average-case reduction (for counting t-cliques mod 2). We note that tolerating an error rate that approaches 0.5 presupposes that approximately half of the n-vertex graphs have an odd number of t-cliques (unless finding t-cliques can be done in $\widetilde{O}(n^2)$-time). This is indeed the case, as can be seen from a general result of Kolaitis and Kopparty [KK13, Thm. 3.2].

2 Formal statements and proofs

For a fixed integer $t \geq 3$ and a graph G, we denote by $\mathtt{CC}^{(t)}(G)$ the number of t-cliques in G, and let $\mathtt{CC}_2^{(t)}(G) \stackrel{\rm def}{=} (\mathtt{CC}^{(t)}(G) \bmod 2)$ denote the parity of this number. We often represent n-vertex graphs by their adjacency matrices; hence, $\mathtt{CC}_2^{(t)}(A) = \mathtt{CC}_2^{(t)}(G)$, where A is the adjacency matrix of G, and it follows that

$$\mathtt{CC}_2^{(t)}(A) = \sum_{i_1 < \cdots < i_t \in [n]} \prod_{j<k \in [t]} A_{i_j,i_k} \bmod 2, \tag{1}$$

where $A_{u,v}$ is the $(u,v)^{\rm th}$ entry of A (indicating whether or not $\{u,v\}$ is an edge in G).

Theorem 1 (deciding the existence of t-cliques reduces to computing $\mathtt{CC}_2^{(t)}$): *For every integer $t \geq 3$, there is a randomized reduction of determining whether a given n-vertex graph contains a t-clique to computing $\mathtt{CC}_2^{(t)}$ on n-vertex graphs such that the reduction runs in time $O(n^2)$, makes* $\exp(t^2)$ *queries, and has error probability at most* $1/3$.

(Added in revision: The proof of Theorem 1 is similar to the proof of [WWWY, Lem. 2.1].)[1] (Added in later revision: Theorem 1 is identical to [BBB19, Lem. A.1].)

Proof: Consider a randomized reduction that, on input $G = ([n], E)$, flips each edge to a non-edge with probability 0.5, leaves non-edges intact, and returns the value of $\mathtt{CC}_2^{(t)}$ on the resulting graph; that is, the reduction generates a random subgraph of G, denoted G', and returns $\mathtt{CC}_2^{(t)}(G')$.

To analyze the output of this procedure (on input G), consider a (symmetric) n-by-n matrix X such that $x_{i,j}$ is a variable if $\{i,j\} \in E$ and $x_{i,j} = 0$ otherwise. We view $\mathtt{CC}_2^{(t)}(X)$, which is defined as in Eq. (1), as a multivariate polynomial over GF(2), and observe that it has degree at most $\binom{t}{2}$. The key observation is that $\mathtt{CC}_2^{(t)}(X)$ *is a non-zero polynomial if and only if the graph G contains a t-clique* (i.e.,$\mathtt{CC}^{(t)}(G) > 0$). Hence, the foregoing reduction can be viewed as returning the value of $\mathtt{CC}_2^{(t)}(X)$ on a random (symmetric) assignment to the variables in X. It follows that the reduction always returns 0 if $\mathtt{CC}^{(t)}(G) = 0$, and returns 1 with probability at least $2^{-\binom{t}{2}}$ otherwise (i.e., when $\mathtt{CC}^{(t)}(G) > 0$). The latter assertion is due to the Schwartz–Zippel for small fields (i.e., for GF(2)).[2] Applying the foregoing reduction for $\exp(t^2)$ times, the claim follows. ■

Theorem 2 (worst-case to average-case reduction for $\mathtt{CC}_2^{(t)}$): *For every integer $t \geq 3$, there is a randomized reduction of computing $\mathtt{CC}_2^{(t)}$ on the worst-case n-vertex graph to correctly computing $\mathtt{CC}_2^{(t)}$ on at least a $1 - \exp(-t^2)$ fraction of the n-vertex graphs such that the reduction runs in time $O(n^2)$, makes $\exp(t^2)$ queries, and has error probability at most* $1/3$.

Proof: Setting $d = \binom{t}{2}$, consider the following random self-reduction of $\mathtt{CC}_2^{(t)}$. On input a symmetric and non-reflective n-by-n matrix, A:

1. Select uniformly d random (symmetric and non-reflective) n-by-n matrices, denoted $R^{(1)}, ..., R^{(d)}$, and let $R^{(0)} = A$.

[1] A result of similar nature appears in [AFW20, Thm. 2].

[2] See [G17, Exer. 5.1]. (Alternatively, see [WWWY, Lem. 2.2].)

2. Making adequate queries to $\texttt{CC}_2^{(t)}$, return $\sum_{I\subseteq\{0,1,...,d\}:I\neq\{0\}} \texttt{CC}_2^{(t)}(R^{(I)}) \bmod 2$, where $R^{(I)} \stackrel{\rm def}{=} \sum_{i\in I} R^{(i)} \bmod 2$ and $\texttt{CC}_2^{(t)}(R^{(\emptyset)}) = 0$.

Hence, the foregoing reduction performs $2^{d+1}-2$ queries, and each of these queries (i.e., each $R^{(I)}$ for $I \notin \{\emptyset, \{0\}\}$) is uniformly distributed over the set of all symmetric and non-reflective n-by-n matrices.

We claim that, for any fixed $R^{(0)}, R^{(1)}, ..., R^{(d)}$, it holds that $\sum_{I\subseteq\{0,1,...,d\}:I\neq\{0\}} \texttt{CC}_2^{(t)}(R^{(I)})$ equals $\texttt{CC}_2^{(t)}(R^{(0)}) \bmod 2$. This claim is proved by considering the multivariate polynomial $P(x_0, x_1, ..., x_d)$ over GF(2) that is defined to equal $\texttt{CC}_2^{(t)}(\sum_{i=0}^{d} x_i R^{(i)})$. Specifically, we use the following facts:

- $P(b_0, b_1, ..., b_d) = \texttt{CC}_2^{(t)}(R^{(\{i:b_i=1\})})$; in particular, $P(0,0,...,0) = 0$ and $P(1,0,...,0) = \texttt{CC}_2^{(t)}(R^{(0)})$.
- P has degree $\binom{t}{2} = d$, since $P(x_0, x_1, ..., x_d) = \texttt{CC}_2^{(t)}(L(x_0, x_1, ..., x_d))$ such that $L(x_0, ..., x_d)$ is a matrix of linear functions (i.e., the $(u,v)^{\rm th}$ entry of $L(x_0, ..., x_d)$ equals $\sum_{i=0}^{d} R_{u,v}^{(i)} x_i$).

  (Indeed, using Eq. (1), it follows that $P = \texttt{CC}_2^{(t)}(L)$ has degree $\binom{t}{2}$.)
- for any $(d+1)$-variate polynomial of degree at most d over GF(2) it holds that the sum of its evaluation over all 2^{d+1} points is 0.

  This general fact can be seen by considering an arbitrary monomial $M(x_0, x_1, ..., x_d) = \prod_{i\in I} x_i$, where $I \subset \{0, 1, .., d\}$. Indeed,

$$
\begin{aligned}
\sum_{(b_0,b_1,...,b_d)\in \mathrm{GF}(2)^{d+1}} M(b_0, b_1, ..., b_d) &= \sum_{(b_0,b_1,...,b_d)\in \mathrm{GF}(2)^{d+1}} \prod_{i\in I} b_i \\
&= 2^{d+1-|I|} \cdot \prod_{i\in I} \sum_{b_i\in \mathrm{GF}(2)} b_i
\end{aligned}
$$

  which equals 0 (mod 2), since $|I| \le d$.

Combining the foregoing facts, it follows that $\sum_{I\subseteq\{0,1,...,d\}:I\neq\{0\}} \texttt{CC}_2^{(t)}(R^{(I)})$ equals $\texttt{CC}_2^{(t)}(R_0)$ (mod 2).

Thus, given oracle access to a program Π such that $\Pr_R[\Pi(R) = \texttt{CC}_2^{(t)}(R)] \ge 1-\epsilon$, when making queries to Π rather than to $\texttt{CC}_2^{(t)}$, the foregoing reduction returns the correct value with probability at least $1 - (2^{d+1}-2)\cdot\epsilon$ (i.e., whenever all queries are answered correctly). Using $\epsilon = 2^{-t^2}$, we obtain a worst-case to average-case reduction that fails with probability less than $2^{d+1-t^2} = 2^{-(t^2+t-2)/2} < 1/3$ when given access to a procedure that is correct on at least a $1-2^{-t^2}$ fraction of the instances.[3] ∎

Remark 3 (the distribution of $\texttt{CC}_2^{(t)}(R)$ for random R): *The proof of Theorem 2 implies that* $2^{-t^2} < \Pr_R[\texttt{CC}_2^{(t)}(R)=1] < 1-2^{-t^2}$. *To see this, using notation as in the proof, suppose towards the contradiction that* $\Pr_R[\texttt{CC}_2^{(t)}(R)=b] \ge 1-2^{-t^2}$ *for some* b. *Then, for every* R_0, *it holds that*

$$
\Pr_{R_1,...,R_d}\left[\sum_{I\subseteq\{0,1,...,d\}:I\neq\{0\}} \texttt{CC}_2^{(t)}(R^{(I)}) \equiv 0 \pmod 2\right]
$$

[3] Indeed, we can slightly improve the bound by using any constant $\epsilon < 2^{-d-2} = 2^{-(t^2-t+4)/2}$.

$$\begin{aligned} &\geq \Pr_{R_1,...,R_d}\left[(\forall I \subseteq \{0,1,...,d\} \setminus \{\{0\},\emptyset\})\, \mathtt{CC}_2^{(t)}(R^{(I)})\!=\!b\right] \\ &\geq 1-(2^{d+1}-2)\cdot 2^{-t^2} \;>\; 0 \end{aligned}$$

where the last inequality uses $2^{d+1-t^2} = 2^{-(t^2+t-2)/2} < 1$. *But this is impossible when* $\mathtt{CC}_2^{(t)}(R_0) = 1$ (e.g., if $\mathtt{CC}^{(t)}(R_0) = 1$).

While Remark 3 only asserts that $\mathrm{E}_R[\mathtt{CC}_2^{(t)}(R)]$ is bounded away from both 0 and 1, it is known to be approximately 1/2. The latter fact follows as a special case of a general result of Kolaitis and Kopparty [KK13, Thm. 3.2].[4]

Open Problem 4 (stronger worst-case to average-case reduction for $\mathtt{CC}_2^{(t)}$): *For every integer* $t \geq 3$ *and* $\gamma > 0.5$, *is there a randomized reduction of computing* $\mathtt{CC}_2^{(t)}$ *on the worst-case* n*-vertex graph to correctly computing* $\mathtt{CC}_2^{(t)}$ *on at least a* γ *fraction of the* n*-vertex graphs such that the reduction runs in time* $\widetilde{O}(n^2)$, *and has error probability at most* $1/3$.

This strengthens Theorem 2 by requiring the reduction to tolerate error rate that is arbitrary close to 0.5 rather than error rate $\exp(-t^2)$. The fact that $\mathrm{E}_R[\mathtt{CC}_2^{(t)}(R)] \approx 0.5$ may be viewed as a sanity check for Problem 4, since $|\mathrm{E}_R[\mathtt{CC}_2^{(t)}(R)] - 0.5| > \delta$ would have implied that $\mathtt{CC}_2^{(t)}$ can be computed correctly with probability $0.5 + \delta$ in constant time.

3 Conclusion

Like [BBB19, Thm. II.9], Theorem 2 asserts an efficient worst-case to average-case reduction for *counting* t*-cliques mod 2*, where average-case is with respect to the uniform distribution over graphs with the given number of vertices. Specifically, for any integer $t \geq 3$, computing $\mathtt{CC}_2^{(t)}$ on the worst-case n-vertex graph is reducible (in $O(n^2)$-time) to computing $\mathtt{CC}_2^{(t)}$ correctly on a $1 - \exp(-t^2)$ fraction of all n-vertex graphs.

We believe that Theorem 2, which has a very simple proof, is as interesting as an analogous result that refers to counting t-cliques (i.e., computing $\mathtt{CC}^{(t)}$), because (as shown in Theorem 1 and [BBB19, Lem. A.1]), computing $\mathtt{CC}_2^{(t)}$ is not easier than determining whether a given graph contains a t-clique. The point is that the decisional problem (i.e., t-$\mathtt{CLIQUE}$) is the one that has received most attention in prior work, and results regarding either $\mathtt{CC}^{(t)}$ or $\mathtt{CC}_2^{(t)}$ are mostly proxies for it (i.e., for results regarding t-$\mathtt{CLIQUE}$). In particular, combining Theorems 1 and 2, it follows that deciding t-$\mathtt{CLIQUE}$ on the worst-case n-vertex graph is reducible (in $O(n^2)$-time) to computing $\mathtt{CC}_2^{(t)}$ correctly on a $1 - \exp(-t^2)$ fraction of all n-vertex graphs. (Recall that a similar result was established in [BBB19], by combining [BBB19, Lem. A.1] and [BBB19, Thm. II.9].)

We note that [GR18] and [BBB19, Thm. II.8], which refer to the counting problem, fall short of establishing results analogous to [BBB19, Thm. II.9] and Theorem 2: The results of [GR18] are not for the uniform distribution (but rather for a relatively simple but different distribution), where the result of [BBB19, Thm. II.8] holds for a notion of average-case that allows only a vanishing

[4] The comment was added in revision. The original version included proofs of the cases of $t \in \{3, 4\}$, since (at the time) I was unaware of the results of Kolaitis and Kopparty [KK13].

error rate (i.e., the "average-case algorithm" is required to be correct on at least a $1 - \frac{1}{\mathrm{poly}(\log n)}$ fraction of the n-vertex graphs).

As stated in Problem 4, we leave open the problem of obtaining a result analogous to Theorem 2 for "average-case algorithms" that are correct on a γ fraction of the instances, for every $\gamma > 1/2$.

Acknowledgements

I am grateful to Dana Ron and to Guy Rothblum for useful discussions.

The revisions have benefitted from comments of several readers (demonstrating the advatage of posting on ECCC): I am grateful to Swastik Kopparty for calling my attention to the results in [KK13], to Ryan Williams for calling my attention to [WWWY, Lem. 2.1], to Oren Weimann for calling my attention to the results in [AFW20], and to Guy Bresler for calling my attention to [BBB19, Thm. II.9].

References

[AFW20] Amir Abboud, Shon Feller, and Oren Weimann. On the Fine-Grained Complexity of Parity Problems. In *47th ICALP*, pages 5:1–5:19, 2020.

[BBB19] Enric Boix-Adsera, Matthew Brennan, and Guy Bresler. The Average-Case Complexity of Counting Cliques in Erdos-Renyi Hypergraphs. In *60th FOCS*, pages 1256–1280, 2019.

[FG06] Jorg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series, Springer, 2006.

[G17] Oded Goldreich. *Introduction to Property Testing.* Cambridge University Press, 2017.

[GR18] Oded Goldreich and Guy Rothblum. Counting t-Cliques: Worst-Case to Average-Case Reductions and Direct Interactive Proof Systems. In *59th FOCS*, 2018.

[KK13] Phokion Kolaitis and Swastik Kopparty. Random graphs and the parity quantifier. *Journal of the ACM*, Vol. 60 (5), pages 1–34, 2013.

[W15] Virginia Vassilevska Williams. Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis. In *10th Int. Sym. on Parameterized and Exact Computation*, pages 17–29, 2015.

[WWWY] Virginia Vassilevska Williams, Joshua Wang, Ryan Williams, and Huacheng Yu. Finding Four-Node Subgraphs in Triangle Time. In *26th SODA*, pages 1671–1680, 2015.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il