A note on the approximability of the tenacity of graphs

Vahid Heidari*1 and Dara Moazzami†2

1University of Tehran, Department of Algorithms and Computation.
2University of Tehran, College of Engineering, Faculty of Engineering Science.

ABSTRACT

In this paper we show that, if $NP \neq ZPP$, for any $\epsilon > 0$, the tenacity of graph with $n$ vertices is not approximable in polynomial time within a factor of $\frac{1}{2} \left(\frac{n-1}{2}\right)^{1-\epsilon}$.

Article history:
Received 19, September 2019
Received in revised form 12, July 2020
Accepted 30 September 2020
Available online 30, December 2020
Research paper

Keyword: $NP$-complete problem, tenacity, tenacious, $NP$-hard.

AMS subject Classification: 05C78.

1 Introduction

The concept of tenacity of a graph $G$ was introduced in [8,9], as a useful measure of the "vulnerability" of $G$. In [9] Cozzens et al. calculated tenacity of the first and second case of the Harary Graphs but they didn’t show the complete proof of the third case. In [28], Moazzami showed a new and complete proof for case three of the Harary Graphs. In [22], Moazzami compared integrity, connectivity, binding number, toughness, and tenacity for several classes of graphs. The results suggest that tenacity is a most suitable measure of stability or vulnerability in that for many graphs it is best able to distinguish between graphs that intuitively should have different levels of vulnerability. In [3, 8-11, 18-50],
the authors studied more about this invariant. We consider only graphs without loops or multiple edges. We use $V(G)$, $\beta(G)$, and $\omega(G)$ to denote the vertex set, independence number and number of components in a graph $G$, respectively. We consider only finite undirected graphs without loops and multiple edges. Let $G$ be a graph. We denote by $V(G)$, $E(G)$ and $|V(G)|$ the set of vertices, the set of edges and the order of $G$, respectively.

The tenacity of a graph $G$, $T(G)$, is defined by $T(G) = \min\{\frac{|S| + \tau(G - S)}{\omega(G - S)}\}$, where the minimum is taken over all vertex cutsets $S$ of $G$. We define $\tau(G - S)$ to be the number of the vertices in the largest component of the graph $G - S$, and $\omega(G - S)$ be the number of components of $G - S$. A connected graph $G$ is called $T$-tenacious if $|S| + \tau(G - S) \geq T\omega(G - S)$ holds for any subset $S$ of vertices of $G$ with $\omega(G - S) > 1$. If $G$ is not complete, then there is a largest $T$ such that $G$ is $T$-tenacious; this $T$ is the tenacity of $G$. On the other hand, a complete graph contains no vertex cutset and so it is $T$-tenacious for every $T$. Accordingly, we define $T(K_p) = \infty$ for every $p$ ($p \geq 1$). A set $S \subseteq V(G)$ is said to be a $T$-set of $G$ if $T(G) = \frac{|S| + \tau(G - S)}{\omega(G - S)}$.

The Mix-tenacity $T_m(G)$ of a graph $G$ is defined as

$$T_m(G) = \min_{A \subset E(G)} \left\{ \frac{|A| + m(G - A)}{\omega(G - A)} \right\}$$

where $m(G - A)$ denotes the order (the number of vertices) of a largest component of $G - A$ and $\omega(G - A)$ is the number of components of $G - A$. Cozzens et al. in [8], called this parameter Edge-tenacity, but Moazzami changed the name of this parameter to Mix-tenacity in [26]. It seems Mix-tenacity is a better name for this parameter. $T(G)$ and $T_m(G)$ turn out to have interesting properties.

After the pioneering work of Cozzens, Moazzami, and Stueckle in [8,9], several groups of researchers have investigated tenacity, and its related problems. In [43] and [44] Piazza et al. used the $T_m(G)$ as Edge-tenacity. But this parameter is a combination of cutset $A \subset E(G)$ and the number of vertices of a largest component, $\tau(G - A)$. It may be observed that in the definition of $T_m(G)$, the number of edges removed is added to the number of vertices in a largest component of the remaining graph. Also this parameter didn’t seem very satisfactory for Edge-tenacity. Thus Moazzami and Salehian introduced a new measure of vulnerability, the Edge-tenacity, $T_e(G)$, in [26]. The Edge-tenacity $T_e(G)$ of a graph $G$ is defined as

$$T_e = \min_{A \subset E(G)} \left\{ \frac{|A| + p(G - A)}{\omega(G - A)} \right\}$$

where $p(G - A)$ denotes the order (the number of edges) of a largest component of $G - A$ and $\omega(G - A)$ is the number of components of $G - A$. This new measure of vulnerability involves edges only and thus is called the Edge-tenacity. Since 1992 there were several interesting questions. But the question ”How difficult is it to recognize $T$-tenacious graphs? ” has remained an interesting open problem for some time. The question was first raised by Moazzami in [21]. Our purpose in [29] was to show that for any fixed positive rational
number $T$, it is $NP$-hard to recognize $T$-tenacious graphs. To prove this we showed that it is $NP$-hard to recognize $T$-tenacious graphs by reducing a well-known $NP$-complete variant of INDEPENDENT SET.

2 Preliminaries

In this paper we consider only finite, non-complete, undirected and connected graphs without loops or multiple edges. Consider a graph $G = (V, E)$ with $n$ vertices, then $\beta(G)$ denotes the maximum size of an independent set of $G$. For each subset of vertices $S$ of $G$, $\omega(G - S)$ denotes number of connected components of $G - S$ which is obtained by removing $S$ from $G$. We define $\tau(G - S)$ to be the number of vertices in the largest component of the graph $G - S$. The tenacity, $T(G)$, of the graph $G$ is defined as $T(G) = \min \left\{ \frac{|A| + \tau(G - A)}{\omega(G - A)} : A \subseteq V, \omega(G - A) \geq 2 \right\}$.

We consider the following minimization problem:

Min Tenacity

Input: A graph $G = (V, E)$.

Output: A set of vertices $S$ of $G$ with the property $\omega(G - S) \geq 2$ such that the ratio $\frac{|A| + \tau(G - A)}{\omega(G - A)}$ is minimized.

An algorithm is a $f(n)$-approximation algorithm for maximization (respectively minimization) problem if for any instance $x$ of the problem of size $n$, it returns a solution $y$ of value $m(x, y)$ such that $m(x, y) = \text{opt}(x) / f(n)$ (respectively $m(x, y) = f(n) \times \text{opt}(x)$). An algorithm is a constant factor approximation if $f(n)$ is a constant. An optimization problem is $f(n)$-approximable if there exists a polynomial time $f(n)$-approximation algorithm for it.

3 Result

Johan Hastad [17] studied the possible performance of polynomial time approximation algorithm for Max Clique. He demanded that the algorithm, on input a graph $G$ with $n$ vertices, outputs a number that is always at most the size of the largest clique in $G$. We say that we have an $f(n)$ approximation algorithm if this number is always at least the largest clique divided by $f(n)$. The best polynomial time approximation algorithm for Max Clique achieves an approximation ratio of $O\left(\frac{n}{\log n}\right)$, [7], and thus it is of the form $n^{1-o(1)}$.

The main question has been whether this is the correct form of the best approximation function. Hastad in [17] proved that this is the case. He used the connection, discovered by Feige et al. in their seminal paper [12], between multiprover interactive proofs and non-approximability results for Max Clique. There has been a sequence of papers [2, 1, 5, 13, 6, 4] giving stronger and stronger non-approximability results based on very plausible intractability assumption. The previously strongest non-approximability result
by Bellare, Goldreich, and Sudan [4] shows that unless NP=co R it is, for any \( \epsilon > 0 \),
infeasible to approximate Max Clique within a factor \( O(n^{1/2-\epsilon}) \). This is done by, for
any \( \delta > 0 \), constructing a probabilistically checkable proof for NP, with a polynomial
time verifier that uses logarithmic randomness, perfect completeness (the verifier always
accepts when the input is in the language) and amortized free-bit complexity \( 2 + \delta \).
This amortized free-bit complexity, which was introduced in [13] and named in [6], is central
to the Hastad’s paper and let us give an informal definition.

In general a verifier is trying to verify a proof of some NP-statement, i.e. that \( x \in L \) for
some input \( x \) and \( L \in NP \). For the sake of concreteness we assume that \( L \) is satisfiability.

In probabilistically checkable proofs we should think of the proof as written down and
the verifier as trying to check the proof as cheaply as possible. The verifier looks at the
number of bits. Sometimes the verifier has no idea what the value of the bit should be
while some other times the verifier is in a checking mode and knows what to expect and
when the value is not the expected value the verifier reject the input. The number of
questions of the first type is, informally speaking, the number of free bits. The amortized
number of free bits is \( \frac{f}{\log_2 g} \) where \( f \) is the number of free bits and \( g \) is the gap
which in our case, since we are assuming perfect completeness, is \( \frac{1}{\delta} \) where \( \delta \) is the probability
that the verifier accepts when \( x \notin L \). The important connection is now that if there is a
proof-system for NP with a polynomial time verifier which uses logarithmic randomness
and has perfect completeness with \( k \) amortized free bits, then for any \( \epsilon > 0 \) we get an
inapproximability result of \( n^{\frac{1}{2}+\epsilon} \), [51]. Bellare, Goldreich and Sudan [4] proved that
in fact we essentially have an equivalence, in that if it is NP-hard to approximate Max
Clique within a factor \( n^{\frac{1}{2}+\epsilon} \) then there is also a proof-system with essentially \( k \) amortized
bits.

Johan Hastad in his paper, for any \( \delta > 0 \), found a proof-system which uses \( \delta \) amortized
bits. This shows that unless NP = ZPP then for any \( \epsilon > 0 \) Max Clique cannot be
approximated within a factor \( n^{1-\epsilon} \) in polynomial time.

Hastad’s paper was the final version of the results announced in [15] and [16].

In this section we show that if \( NP \neq ZPP \), then for any \( \epsilon > 0 \), the tenacity of a graph
with \( n \) vertices is not approximable in polynomial time within a factor of \( \frac{1}{2}(\frac{n-1}{2})^{1-\epsilon} \).

**Theorem 1.** [17], If \( NP \neq ZPP \), for any \( \epsilon > 0 \), the maximum independent set of a
directed graph is not approximable within \( n^{1-\epsilon} \) approximation factor.

**Lemma 1.** [8], If \( G \) is not complete, then \( T(G) \leq \frac{n-\beta(G)+1}{\beta(G)} \).

**Theorem 2.** If \( NP \neq ZPP \), for any \( \epsilon > 0 \), the \textsc{Min Tenacious} problem is not
\( \frac{1}{2}(\frac{n-1}{2})^{1-\epsilon} \) approximable in polynomial time where \( n \) is the number of vertices of the graph.

**Proof:** We construct a reduction between \textsc{Max Independent Set} and \textsc{Min Tenacious}. Given a graph \( G \) instance of \textsc{Max Independent Set} on \( n \) vertices,
we construct a graph \( H \) from \( G \) by adding a clique \( C \) of size \( n+1 \) vertices, and making
each vertex of \( C \) adjacent to each vertex in \( G \). By lemma 1, we have:
\[
T(H) \leq \frac{2n + 2 - \beta(H)}{\beta(H)} \leq \frac{2n + 2}{\beta(H) - 1} \leq \frac{2n + 2}{\beta(H)}
\]
\[
\beta(H) \leq \frac{2n + 2}{T(H)}
\]

And because \( \beta(H) = \beta(G) \), then:
\[
\beta(G) \leq \frac{2n + 2}{T(H)}
\]

Now suppose that \textbf{MIN TENACIOUS} is \( \frac{1}{2} \left( \frac{n-1}{2} \right)^{1-\epsilon} \) approximable. Thus there is an algorithm that applied to \( H \) finds a set \( A \) of vertices such that:
\[
val = \frac{|A| + \tau(G - A)}{\omega(G - A)} \leq \frac{1}{2} \left( \frac{2n + 1 - 1}{2} \right)^{1-\epsilon} T(H) \leq \frac{1}{2} n^{1-\epsilon} T(H)
\]

We consider as solution for \( G \) an independent set that contains a vertex from each connected component of \( \omega(G - A) \). Thus the size of this independent set is at least \( n + 1 \), because \( A \) contains at least vertices of \( C \), then we have:
\[
val' = \omega(G - A) \geq \frac{\omega(G - A) \times (n + 1)}{|A| + \tau(G - A)} \geq \frac{2n + 2}{n^{1-\epsilon}}
\]

And by using the previous inequalities we obtain:
\[
val' \geq \frac{1}{n^{1-\epsilon}} \beta(G)
\]

The \( \beta(G) \) is the optimal solution for \textbf{MAX INDEPENDENT SET} problem, and because this is a maximization problem, and by using theorem 1, we can’t find an approximation algorithm with approximation factor better than \( n^{1-\epsilon} \), and the theorem follows.

\textbf{Acknowledgement}
This work supported by Tehran University. Our special thanks go to the University of Tehran, College of Engineering and Department of Engineering Science for providing all the necessary facilities available to us for successfully conducting this research.

\textbf{References}


