

On constant time approximation of parameters of bounded degree graphs

Noga Alon *

March 10, 2010

Abstract

How well can the maximum size of an independent set, or the minimum size of a dominating set of a graph in which all degrees are at most d be approximated by a randomized constant time algorithm? Motivated by results and questions of Nguyen and Onak, and of Parnas, Ron and Trevisan, we show that the best approximation ratio that can be achieved for the first question (independence number) is between $\Omega(d/\log d)$ and $O(d \log \log d/\log d)$, whereas the answer to the second (domination number) is $(1 + o(1)) \ln d$.

1 Introduction

The question of identifying the properties of bounded degree graphs in the model of [7] that can be tested efficiently, is that of recognizing the properties that are local in nature. These are properties for which the local structure of the graph supplies meaningful information about the global property. A related problem deals with efficient approximation algorithms for graph parameters, like the independence number, or the domination number of a given bounded degree graph. The question here is to decide how well we can approximate these quantities by observing the local structure of the graph. In this short paper we discuss several problems of this type, continuing the work in several earlier papers including [13] and [12] on related questions.

1.1 Notation and definitions

Following [13] and [12] we call a real number \bar{y} an (α, β) -approximation for a number y if $y \leq \bar{y} \leq \alpha y + \beta$. A randomized algorithm A is an (α, β) -approximation algorithm for a graph parameter $P(G)$ if given an input graph G the algorithm computes a value y which is an (α, β) -approximation for $P(G)$ with probability at least $2/3$ for any proper input graph G . Let $G(n, d)$ denote the family of all graphs on n vertices with maximum degree at most d , represented by their adjacency lists. In this note

*Schools of Mathematics and Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. Email: nogaa@tau.ac.il. Research supported in part by an ERC Advanced grant and by a USA-Israeli BSF grant.

we are interested in randomized $(\alpha, \epsilon n)$ -approximation algorithms for some graph parameters, that work on input graphs $G \in G(n, d)$ in constant expected time, that is, in time bounded by a function $f = f(d, \epsilon)$ of d and ϵ only, which is independent of n . In order to make the discussion cleaner, we will not be interested in the precise function f as long as it is independent of n . Our objective is to try to determine or estimate, for a given graph parameter P , the best possible α so that for any positive $\epsilon > 0$ there is a constant time, randomized $(\alpha, \epsilon n)$ -approximation algorithm for $P(G)$ for inputs $G \in G(n, d)$. In some cases the same α actually works for $\epsilon = 0$ as well.

We note that it is possible to replace the success probability $2/3$ by any larger number smaller than 1 (by running the algorithm several times, taking the median of all values computed) and it is also possible to replace the expected running time by worst case running time by stopping the algorithm if its running time exceeds its expectation by a large constant factor.

1.2 Examples

- Simple sampling of vertices, checking their degrees, shows that for every $\epsilon > 0$ there is a constant time randomized $(1, \epsilon n)$ approximation algorithm for estimating the sum of degrees of a graph $G \in G(n, d)$. The analysis is essentially trivial, and the example is listed here mainly to practice the definitions. A similar approach provides a constant time randomized $(1, \epsilon n)$ approximation algorithm for the number of triangles (or copies of any fixed connected graph) in a given $G \in G(n, d)$.
- As shown in [13] and [11] for every $\epsilon > 0$ there is a constant time randomized $(2, \epsilon n)$ approximation algorithm for the vertex cover of a graph $G \in G(n, d)$. Trevisan (c.f. [13]) observed that there is no such $(2 - \delta, \epsilon n)$ -approximation algorithm, for any fixed $\epsilon < \delta$.
- In [12] it is proved that for every $\epsilon > 0$ there is a constant time randomized $(H(d + 1), \epsilon n)$ approximation algorithm for the domination number of a graph $G \in G(n, d)$, where $H(d + 1) = 1 + \frac{1}{2} + \dots + \frac{1}{d+1} = \ln d + \Theta(1)$ and the domination number of $G = (V, E)$ is the minimum cardinality of a set of vertices $U \subset V$ so that each vertex $v \in V - U$ has a neighbor in U .
- Another result proved in [12] is that for every $\epsilon > 0$ there is a constant time randomized $(1, \epsilon n)$ approximation algorithm for the maximum size of a matching in a graph $G \in G(n, d)$.

1.3 A useful tool

The basic tool applied in the algorithms of [13] and [12] is the result that for every $\epsilon > 0$ there is a constant time randomized algorithm that computes, for a given $G \in G(n, d)$, a $(1, \epsilon n)$ -approximation of the size of some maximal (with respect to containment) independent set in G . The same proof provides such an approximation even in the weighted case, which we'll need here, where the weight of each vertex is, say, an integer between 1 and d .

1.4 The new results

Our first result deals with approximation of domination numbers, showing that the $(H(d+1), \epsilon n)$ approximation proved in [12] is essentially tight.

Theorem 1.1 *The smallest α so that for every $\epsilon > 0$ there is a constant time randomized $(\alpha, \epsilon n)$ approximation algorithm for the domination number of a graph $G \in G(n, d)$ is $(1 + o(1)) \ln d$, where the $o(1)$ -term tends to zero as d tends to infinity.*

The second result is about approximation of independence numbers.

Theorem 1.2 *There are two positive constants c_1, c_2 so that the following holds. The smallest α so that for every $\epsilon > 0$ there is a constant time randomized $(\alpha, \epsilon n)$ approximation algorithm for the independence number of a graph $G \in G(n, d)$ is at least $c_1 \frac{d}{\log d}$, and at most $c_2 \frac{d \log \log d}{\log d}$.*

2 Proofs

In this section we present the proofs of the two theorems stated above. Recall that the girth of a graph G is the length of a shortest cycle in it. We start with the proof of Theorem 1.1.

2.1 Dominating set

Lemma 2.1 *For every fixed d and large n divisible by $2d+2$, there are two d -regular graphs $G = (V, E)$ and $G' = (V', E')$ in $G(n, d)$ so that the following holds.*

- (i) *The girth of G is at least, say, $0.5 \log n / \log d$ and its domination number is precisely $\frac{n}{d+1}$.*
- (ii) *The girth of G' is at least $0.5 \log n / \log d$ and its domination number is $(1 + o(1)) \frac{n \ln d}{d}$.*

Proof:

(i) Start with an arbitrary set of $n/(d+1)$ pairwise vertex disjoint stars, each containing d edges. Let U denote the set of centers of these stars, and let W denote the set of all their end vertices. Note that $|W| = \frac{nd}{d+1}$ is even, and thus there is a $d-1$ -regular graph H on the set W . Let G be the d -regular graph on the n vertices $U \cup W$ consisting of all edges of the initial stars as well as all edges of H . The domination number of G is clearly $n/(d+1)$, as U is a dominating set in it. The girth, however, can be small, and our objective is to show that it can be increased without changing the domination number. This will be done by modifying the graph G , without touching the edges of the stars.

The method is similar to that of Erdős and Sachs in [6]. Call a cycle short if its length is at most $0.5 \log n / \log d$. As long as there is a short cycle, take arbitrarily a shortest cycle C , and an arbitrary non-star edge in it uv . Since the graph is d -regular, there is another non-star edge xy in it whose distance from uv is at least $\log n / \log d$. Omit the two edges uv and xy from G and replace them by the two new edges xu and yv . It is easy to check that this replacement does not create any new cycles of length at most that of C , and destroys the cycle C itself. Continuing in this manner until there are no short cycles left we obtain the desired graph G .

(ii) Let G' be a random d -regular graph on n vertices. It is known that with high probability the domination number of G' is $(1 + o(1))\frac{n \ln d}{d}$ (see, e.g., [5]). It is also not difficult to check that the expected number of cycles of length at most $0.5 \log n / \log d$ in G' is at most \sqrt{n} , and hence, by Markov's Inequality, with probability at least 0.5 there are at most $2\sqrt{n}$ such cycles. Fix a graph with domination number $(1 + o(1))\frac{n \ln d}{d}$ and at most $2\sqrt{n}$ short cycles, and modify it according to the process described in the proof of part (i), destroying all short cycles. Since each modification step touches at most 2 edges, and cannot create too many short cycles, the domination number changes during this process by at most $o(n)$, providing the desired result. ■

Proof of Theorem 1.1: The existence of the required approximation algorithm is proved in [12], as described in Section 1. It remains to show that no better approximation is possible.

Let G and G' be as in Lemma 2.1 and consider two distributions on graphs in $G(n, d)$: the first is a permuted copy of G , and the second is a permuted copy of G' . By the girth and regularity conditions the statistics of subgraphs of less than $0.5 \log n / \log d$ vertices and edges in both distributions is identical. Hence no constant time algorithm can distinguish between them. This completes the proof. Note that the two distributions here are not only computationally indistinguishable for randomized constant time algorithms, but are completely identical. It is possible to use computational indistinguishability here and show that in fact in order to obtain an $(\alpha, \epsilon n)$ -approximation for the domination of $G \in G(n, d)$, where $\alpha < (1 - \delta) \ln d$ and $\epsilon < \delta \ln d$, one needs to inspect at least $\Omega(\sqrt{n})$ vertices and edges, but as we care here only about algorithms whose running time is independent of n , we do not include the detailed analysis of this stronger claim. ■

2.2 Independence number

Proof of Theorem 1.2: The proof of the lower bound is simple: there is a d -regular bipartite graph G on n vertices with girth $\Omega(\log n)$ (and independence number $n/2$), and it is well known that a random d -regular graph on n vertices has, with high probability, independence number at most $O(\frac{n \log d}{d})$ and only a small number of cycles of length shorter than $0.5 \log n / \log d$. We can thus modify the graph as in the previous subsection and get a d -regular graph G' on n vertices with girth at least $\Omega(\log n / \log d)$ and independence number at most $O(\frac{n \log d}{d})$. As in the previous proof, no constant time algorithm will be able to distinguish between a permuted copy of G and a permuted copy of G' , providing the lower bound.

The proof of the upper bound requires a bit more work. It is, in fact, non trivial to get any constant time $(\alpha, \epsilon n)$ - approximation algorithm to the independence number, with $\alpha = o(d)$. We first describe a sequential deterministic algorithm and then observe that it can be converted into a randomized, constant time procedure.

Let $G \in G(n, d)$ be a given input graph. As long as there is a nonempty set X of vertices of G with a common neighbor in the graph, satisfying $|X| \leq \log^3 d$, so that the induced subgraph on X contains no independent set of size at least $|X| / \log d$, omit it. (When there are many choices for such a set X , pick one arbitrarily). Suppose that when there is no such set X left, there are t vertices in the remaining graph. By the result in [2] (see the remark following the proof of Theorem 1.1 in [2]), the

induced graph left on the t remaining vertices contains an independent set of size at least $\Omega(\frac{t \log d}{d \log \log d})$. It is also obvious that G has an independent set of size at least $\frac{n}{d+1}$. Thus, the independence number of G is at least the average between these two, that is, at least $\Omega(\frac{n}{d+1} + \frac{t \log d}{d \log \log d})$. On the other hand we know that there is no independent set of size bigger than $\frac{n-t}{\log d} + t$, providing the required approximation if the value of t is known with sufficient accuracy.

It thus remains to show how to approximate t in randomized constant time. Define an auxiliary weighted graph F whose set of vertices is the set of all nonempty subsets X of at most $\log^3 d$ vertices of G with a common neighbor, so that the induced subgraph of G on X contains no independent set of size at least $|X|/\log d$. Two such vertices X and X' of F are adjacent iff the two sets X and X' have a nonempty intersection. The weight of each vertex X is the cardinality $|X|$ of the subset corresponding to it. Note that one can easily check the adjacency relations in the graph F by observing the original graph G locally. We can therefore find, in randomized constant time, a good approximation to the weight of a maximal (with respect to containment) independent set of vertices of F , which will enable us to approximate the value of t defined in the sequential procedure described above. This completes the proof. ■

3 Concluding remarks and open problems

We have investigated the best possible approximation ratios that can be obtained for two graph parameters by randomized, constant time algorithms on bounded degree graphs represented by their adjacency lists. The problem for domination number is quite well understood, whereas in the case of independence number there is still a $\Theta(\log \log d)$ gap between the upper and lower bounds. It will be interesting to close this gap. We suspect that the $\log \log d$ term can be omitted, but this will require an additional argument. It is worth noting that the best known polynomial time algorithm for approximating the independence number of graphs $G \in G(n, d)$ provides an approximation ratio of $\Theta(d \log \log d / \log d)$, matching the approximation ratio obtained here. This has been found by Vishwanathan (first recorded in [8], see also [9]), and by Halperin [10], and is based on the method of [3] that applies semidefinite programming. It seems unlikely that these algorithms can be converted into constant time randomized ones. Austrin, Khot and Safra [4] have recently proved a hardness result of $\Omega(d/\log^2 d)$ for the problem, under the unique games conjecture.

A purely combinatorial problem, that seems related to the question above (although we do not know any direct relation) is the conjecture raised in [1] that for any fixed graph H , any graph $G \in G(n, d)$ that contains no copy of H has an independent set of size at least $c_H \frac{n \log d}{d}$. Here, too, the best known result, due to Shearer [14], is off by a factor of $\log \log d$, and it is only known that the independence number of any such graph is at least $c_H \frac{n \log d}{d \log \log d}$.

Acknowledgment I would like to thank Krzysztof Onak for helpful discussions, comments and suggestions.

References

- [1] M. Ajtai, P. Erdős, J. Komlós and E. Szemerédi, On Turan’s theorem for sparse graphs, *Combinatorica* 1 (1981), 313-317.
- [2] N. Alon, Independence numbers of locally sparse graphs and a Ramsey type problem, *Random Structures and Algorithms* 9 (1996), 271–278.
- [3] N. Alon and N. Kahale, Approximating the independence number via the θ -function, *Math. Programming* 80 (1998), 253-264.
- [4] P. Austrin, S. Khot and M. Safra, Inapproximability of Vertex Cover and Independent Set in Bounded Degree Graphs, *IEEE Conference on Computational Complexity* 2009, 74–80.
- [5] N. Alon and N. Wormald, High degree graphs contain large-star factors, to appear.
- [6] P. Erdős and H. Sachs, Reguläre Graphen gegebener Tailenweite mit minimaler Knotenzahl. (German), *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe* 12 (1963), 251–257.
- [7] O. Goldreich and D. Ron, Property testing in bounded degree graphs , *Proc. 29th STOC* (1997), 406–415.
- [8] M. M. Halldórsson, Approximations of Independent Sets in Graphs, *Proc. APPROX* 1998, 1-13.
- [9] M. M. Halldórsson, Approximations of Weighted Independent Set and Hereditary Subset Problems, *Journal of Graphs Algorithms and Applications* 4 (2000), 1-16.
- [10] E. Halperin, Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs, *Proc. Eleventh ACM-SIAM Symp. on Discrete Algorithms* (2000), 329–337.
- [11] S. Marko and D. Ron, Distance approximation in bounded-degree and general sparse graphs, *Proc. APPROX-RANDOM, Lecture Notes in Comput. Sci.*, 4110, Springer, Berlin, 2006, 475–486.
- [12] H. N. Nguyen and K. Onak , Constant-Time Approximation Algorithms via Local Improvements *Proc. 49th Annual Symposium on Foundations of Computer Science (FOCS 2008)*, 327–336.
- [13] M. Parnas and D. Ron, Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms, *Theoret. Comput. Sci.* 381 (2007), no. 1-3, 183–196.
- [14] J. B. Shearer, On the independence number of sparse graphs, *Random Structures and Algorithms* 7 (1995), 269-271.