Fast distributed approximations in planar graphs

A. Czygrinow[*]
Department of Mathematics and Statistics
Arizona State University
Tempe, AZ,85287-1804, USA
Email: andrzej@math.la.asu.edu

M. Hańćkowiak[†]
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Poznań, Poland
Email: mhanckow@main.amu.edu.pl

W. Wawrzyniak[‡]
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Poznań, Poland
Email: wwawrzy@amu.edu.pl

February 3, 2009

Abstract

We give deterministic distributed algorithms that given $\delta > 0$ find in a planar graph G, $(1 \pm \delta)$-approximations of a maximum independent set, a maximum matching, and a minimum dominating set. The algorithms run in $O(\log^* |G|)$ rounds. In addition, we prove that no faster deterministic approximation is possible and show that if randomization is allowed it is possible to beat the lower bound for deterministic algorithms.

1 Introduction

In recent years, there has been a growing interest in designing distributed approximation algorithms for special families of networks. In particular, efficient (in the model described below) distributed algorithms for some problems in constant-degree graphs, unit-disc graphs, or planar networks have been recently proposed. In contrast, for general networks most of the problems that admit easy solutions in special classes of graphs seem either unapproachable or are provably intractable (see for example [KMW04]). In this paper, we give deterministic distributed

---

approximation algorithms for the maximum independent set, the maximum matching, and the minimum dominating set problems in planar graphs. The algorithms run $O(\log^* |G|)$ rounds and find a $(1 \pm \delta)$-approximation in a planar graph G. In addition, we prove lower bounds for the time complexity of deterministic approximation algorithms and show that if randomization is allowed then it is possible to beat the lower bound for deterministic procedures and give faster solutions.

1.1 Model of Computations and Notation

We will work in a synchronous, message-passing model of computations (model $LOCAL$ in [Pe00]). In this model a graph is used to represent an underlying network. Vertices of the graph correspond to computational units, and edges represent communication links. The network is synchronized and in one round a vertex can send, receive messages from its neighbors, and can perform some local computations. Neither the amount of local computations nor the size of messages sent is restricted in any way. Consequently, in this model in a graph of diameter t, any graph-theoretic function can be computed in $O(t)$ rounds. In addition, we assume that vertices have unique identifiers from $\{1, \ldots, |G|\}$ where $|G|$ is the order of the underlying graph G. In some applications graphs will have additional weights. The interpretation of the weights depends on specific applications and they do not impact the communication in any way. We will mostly follow [D05] in graph-theoretic terminology. In particular, we will use $|G|$ to denote the order of graph G and $||G||$ to denote the size of G.

1.2 Related Work

Theory of distributed approximation algorithms has attracted some attention recently. For a nice overview of important results in this area the reader is referred to the survey by Elkin [E04]. Although there are very few deterministic distributed approximation algorithms that run in $o(|G|)$ rounds in a general graph G, in the case when the underlying network has additional properties, algorithms that give a non-trivial approximation are much easier to design. The most eminent example is the classical algorithm of Cole and Vishkin from [CV86]. The algorithm finds in $O(\log^* |G|)$ rounds a maximal independent set in a constant-degree graph G and provides therefore a constant approximation for the maximum independent set problem in this type of a network. Results of Linial ([L92]) and Naor ([N91]) give matching $\Omega(\log^* |G|)$ lower bounds for the running time of deterministic and randomized distributed algorithm that find a maximal independent set in a cycle and show that the $\log^* |G|$ running time cannot be beaten if one expects exact, non-approximate, solutions. Similarly, in the case of unit-disk graphs, it is possible (see [KMNW05a], [KMNW05b], [CH06b], or [SW08]) to give fast approximation algorithms for many graph-theoretic problems that seem intractable in general networks.

In planar graphs, approximations that run in the poly-logarithmic number of rounds and give the approximation error of $(1 \pm O(1/\log^k |G|)$ are known for all problems discussed in this paper ([CHS06], [CH06a]). However if one is willing to accept a larger approximation error (for example $(1 \pm \delta)$) but in a much faster fashion (say in $O(\log^* |G|)$ or $O(1)$ rounds) then the methods from [CHS06] or [CH06a] do not give any indication if such algorithms are possible to obtain. In addition, it has not been clear if approximation problems are significantly easier than the original Maximal Independent Set problem and if it is possible to beat the $\log^* |G|$ bound and give $O(1)$-running time algorithms that find approximate solutions. In this direction,

very recently and independently of the work in this paper, Lenzen, Oswald, and Wattenhofer ([LOW08]) gave a 71-approximation for the minimum dominating set problem that runs in $O(1)$ rounds and Lenzen, Wattenhofer proved in [LW08] a lower bound which is identical with our result from Section 4.1. The arguments from [LW08] and Section 4.1 are completely different.

1.3 Results

We will prove a few results on distributed approximations in planar graphs. The main result is a collection of deterministic distributed algorithms which approximate a maximum independent set, a maximum matching, and a minimum dominating set in planar graphs. The algorithms run in $O(\log^* |G|)$ rounds. In addition, we prove that no deterministic algorithm can run faster and give an approximation error achieved by our algorithms. At the same time, we note that an easy randomized procedure beats the lower bound for deterministic algorithms and with high probability finds a $(1 \pm \delta)$-approximation and runs in $O(1)$ rounds.

More formally, we show that there is a deterministic distributed algorithm which for a given $\delta > 0$ finds in a planar graph G an independent set I of size which is at least $(1 - \delta)\alpha(G)$ where $\alpha(G)$ is the independence number of G. The algorithm runs in $O(\log^* |G|)$ rounds (Theorem 3.2). Algorithms with a similar performance can be obtained for the maximum matching problem (Theorem 3.3), and for the minimum dominating set problem (Theorem 3.4). In addition, in the case of the maximum independent set problem, the weighted version of the problem can be approximated in $O(\log^* |G|)$ rounds (Theorem 3.2). These results are complemented by some lower bounds. We prove that for any $c > 0$, no deterministic distributed algorithm can find a c-approximation of the maximum independent set in a planar graph G in $o(\log^* |G|)$ rounds, nor there is a deterministic distributed algorithm that finds a c-approximation of a maximum matching in a planar graph G in $o(\log^* |G|)$ rounds (Section 4.1). In the case of the dominating set problem the situation is different and it is possible to find a $O(1)$-approximation in $O(1)$ rounds (see [LOW08] or Section 3.3) but for any $\delta > 0$ there is no deterministic algorithm which in $o(\log^* |G|)$ rounds finds a $(5 - \delta)$-approximation of a minimum dominating set in a planar graph G (Section 4.1). Finally, we note that an easy randomized procedure can find with high probability a $(1 - \delta)$-approximation of the maximum independent set in a planar graph in $O(1)$ rounds (Section 4.2) and so in the case randomness is allowed, approximation is significantly easier than solving the Maximum Independent Set problem.

1.4 Organization

In the next section, we describe a partitioning algorithm which is used in Section 3 to obtain deterministic approximations for the maximum independent set, the maximum matching, and the minimum dominating set problems. In Section 4 we give lower bounds and discuss randomized algorithms.

2 Clustering Algorithm

In this section, we give a deterministic algorithm which in $O(\log^* |G|)$ rounds finds a partition of a weighted planar graph with the property that the weight of the edges between different partition classes is significantly smaller than the total weight of the graph. The procedure is

invoked in the next section to give approximation algorithms. We will consider weighted graphs with weights defined on vertices as well as weights defined on edges. For a graph $G = (V, E)$ we will use $\omega : V \to R^+$ to denote a vertex-weight function and $\bar{\omega} : E \to R^+$ to denote a edge-weight function. In addition, we will sometimes slightly abuse the notation and if F is a subgraph of $G = (V, E)$ with $\bar{\omega} : E \to R^+$ (or $\omega : V \to R^+$) then $\bar{\omega}(F)$ (or $\omega(F)$) will be used to denote $\bar{\omega}(E(F))$ (or $\omega(V(F))$). If P is a path then the length of P will be the number of edges in P (i.e $|P| - 1$).

In the course of computations we will be contracting connected subgraphs of a planar graph and recomputing the weights. Specifically, if $G = (V, E)$ is a planar graph, $\bar{\omega} : E \to R^+$, and $U_1, U_2, \ldots, U_l$ are pair-wise disjoint subsets of V such that $G[U_i]$ is connected then we define $\tilde{G}$ to be the weighted graph in which every U_i is contracted to a vertex and for $u, u' \in V(\tilde{G})$ with $u \neq u'$ we set

$$\bar{\omega}_{\tilde{G}}(u, u') = \sum_{x \in U, y \in U'} \bar{\omega}(x, y) \tag{1}$$

where $U = U_i$ if u is obtained by contracting U_i and $U = \{u\}$ otherwise (the same for U'). We proceed with a few auxiliary definitions and facts.

Definition 2.1 *A directed graph $\overrightarrow{F}$ such that the maximum out-degree in F is one is called a pseudo-forest.*

If $\overrightarrow{F}$ is a directed graph then we use F to denote the graph obtained from $\overrightarrow{F}$ by ignoring the orientation of edges.

Fact 2.2 *Let $G = (V, E)$ be a planar graph with edge-weight function $\bar{\omega} : E \to R^+$. There is a distributed procedure which in two steps finds a pseudo-forest $\overrightarrow{F}$ such that F is a subgraph of G and $\bar{\omega}(F) \geq \bar{\omega}(G)/6$.*

Proof. First we show that there exists a pseudo-forest $\overrightarrow{F}$ such that F is a subgraph of G and $\bar{\omega}(F) \geq \bar{\omega}(G)/3$. Indeed, since G is planar its edge set is the union of at most 3 forests by the theorem of Nash-Williams. Now, select the heaviest forest and root the trees to obtain the desired graph F with $\bar{\omega}(F) \geq \bar{\omega}(G)/3$. Next we prove that a desired pseudo-forest can be obtained by a distributed procedure. Consider the following simple algorithm. Every vertex v selects the heaviest edge $\{u, v\}$ incident to v and puts the orientation from v to u. If an edge has been assigned the orientation in both directions then one of them is selected arbitrarily. Clearly the obtained graph is a pseudo-forest. Since G is the union of three forests F_1, F_2, F_3 and $2\bar{\omega}(F) \geq \max\{\bar{\omega}(F_i)\}$, we have $\bar{\omega}(F) \geq \bar{\omega}(G)/6$. $\square$ In addition we have the following simple fact.

Fact 2.3 *If $\overrightarrow{F}$ is a connected pseudo-forest such that the diameter of F is d then $\overrightarrow{F}$ contains a directed path of length at least $d/2$.*

We develop some more notation which will be used in the next procedure. Let $\overrightarrow{F}$ be a pseudo-forest vertices of which are properly colored with colors from some set S. For a vertex v and for a set of colors $C \subset S$, let $in(v, C)$ be the set of arcs (u, v) (from u to v) such that the color of u is in C and let $out(v, C)$ be defined analogously. If C is empty then $in(v, C)$ and $out(v, C)$ are empty and their weights are equal to zero.

HeavyStars

1. Find a pseudo-forset $\overrightarrow{F}$ in G using the procedure from Fact 2.2.

2. Use Cole-Vishkin algorithm [CV86] to properly color the vertices of $\overrightarrow{F}$ using colors from $\{1, 2, 3\}$.

3. For every non-isolated vertex v in parallel:

   (a) If color of v is 1 then v marks all edges from $in(v, \{2, 3\})$ if $\bar{\omega}(in(v, \{2, 3\})) > \bar{\omega}(out(v, \{2, 3\}))$ and v marks the edge from $out(v, \{2, 3\})$ otherwise.

   (b) If color of v is 2 then v marks all edges from $in(v, \{3\})$ if $\bar{\omega}(in(v, \{3\})) > \bar{\omega}(out(v, \{3\}))$ and v marks the edge from $out(v, \{3\})$ otherwise.

4. Let Q_i's denote connected components of the graph induced by marked edges. In parallel, find in each Q_i vertex-disjoint stars with weight of at least $\bar{\omega}(Q_i)/2$ and return them. (This is easily accomplished by rooting Q_i's and considering odd and even levels.)

Lemma 2.4
$$diam(Q_i) < 10.$$

Proof. Suppose $diam(Q_i) \geq 10$. Then from Fact 2.3, there is a directed path of length at least 5. If there is an internal vertex v in the path of color 1 then either the edge coming to v or coming out of v (but not both) is marked by 3(a). Otherwise, since the length is at least 5, there must be an internal vertex of color 2 with both of its neighbors of color 3 and by 3(b) only one of these edges can be marked. $\square$

Lemma 2.5 HEAVYSTARS *returns stars of weight at least $\bar{\omega}(G)/24$ and runs in $O(\log^* |G|)$ rounds.*

Proof. From Fact 2.2, we have $\bar{\omega}(F) \geq \bar{\omega}(G)/6$. Every edge of F has either one endpoint in color 1 and the second from $\{2, 3\}$ or one endpoint in color 2 another in color 3. Consequently the edges considered in steps 3(a) and 3(b) form a partition of $E(F)$ and so the weight of the union of Q_i's is at least half of the the weight of F. Finally, the stars will have at least half of the weight of Q_i's and so the weight of them is at least $\bar{\omega}(G)/24$. The first, third, and the fourth step require $O(1)$ rounds and the coloring from step two can be found in $O(\log^* |G|)$ rounds. $\square$

We will now describe the clustering algorithm. The procedure takes $0 < \epsilon < 1$ as an input.

CLUSTERING

1. $H := G$

2. Iterate $\lceil (\log (\frac{1}{\epsilon}) / \log (\frac{24}{23})) \rceil$ times:

   (a) Call HEAVYSTARS to find vertex-disjoint stars in H.

   (b) Modify H by contracting each star to a vertex and computing the weights as in (1).

3. Let W denote the set of vertices contracted to w. Return $\{W | w \in V(H)\}$.

5

Lemma 2.6 *Given $0 < \epsilon < 1$, CLUSTERING finds a partition $(V_1, \ldots, V_l)$ of $V(G)$ such that if $\tilde{G}$ is obtained by contracting each of V_i's and recomputing the weights as in (1) then*

$$\bar{\omega}(\tilde{G}) \leq \epsilon \bar{\omega}(G).$$

The algorithm runs in $O(\log^* |G|)$ rounds.

Proof. From Lemma 2.5 in each iteration the algorithm contracts the stars of weight which is at least $1/24$ of the total weight of the graph. Consequently after l iterations the weight of graph H is at most $\left(\frac{23}{24}\right)^l \bar{\omega}(G) \leq \epsilon \bar{\omega}(G)$ if $l = \lceil \left(\log \left(\frac{1}{\epsilon}\right) / \log \left(\frac{24}{23}\right)\right) \rceil$. The running time is $O(\log^* |G|)$ from Lemma 2.5. $\square$

3 Approximating Algorithms

In this section, we will use the clustering procedure from the previous section to give deterministic distributed approximations.

3.1 Maximum-Weight Independent Set

We will start with the maximum-weight independent set problem. Let $G = (V, E)$ be a planar graph with $\omega : V \to R^+$. For an edge $\{u, v\} \in E$, define

$$\bar{\omega}(\{u, v\}) = \min\{\omega(u), \omega(v)\}. \tag{2}$$

We have the following fact.

Fact 3.1

$$\bar{\omega}(G) \leq 3\omega(G).$$

Proof. From Nash-Williams theorem G has an orientation such the out-degree is at most three. For an oriented edge (u, v) (from u to v) we have the weight of the edge to be at most $\omega(u)$. Since the out-degree is at most three, a vertex can be the starting point of at most three edges. $\square$

To approximate a maximum-weight independent set we will invoke a modified procedure from [CH06c]. This algorithm proceeds as follows. First, consider the edge-weighted graph with weights from (2) and invoke CLUSTERING to find a partition $(V_1, \ldots, V_l)$ of $V(G)$. Then find optimal independent sets I_i in each of the $G[V_i]$. (Note that the diameter of $G[V_i]$ is $O(1)$.) Finally correct the solution obtained in the previous step by deleting u from I_i if $\{u, v\}$ is the edge with $v \in I_j$ and $\omega(u) < \omega(v)$ (In the case the weights are equal use the identifiers to break the symmetry). Using a similar argument to the one from [CH06c] we obtain:

Theorem 3.2 *There is a deterministic distributed algorithm which for given $0 < \delta < 1$ finds in a planar weighted graph $G = (V, E)$ with $\omega : V \to R^+$ an independent set I of weight which is at least $(1 - \delta)OPT$ where OPT denotes the weight of a max-weight independent set. The algorithm runs in $O(\log^* |G|)$ rounds.*

Proof Outline: Since G is planar, $OPT \geq \omega(G)/4$ as $\chi(G) \leq 4$. Let I be the union of I_i's before the correction step. Then $\omega(I) \geq OPT$ as the procedure finds the optimal solution in each of $G[V_i]$'s and the restriction of any independent set to V_i is an independent set. In the correction step, for every edge between two different clusters we delete one of its endpoints, weight of which is at most the weight of the edge by (2). Consequently, the total weight of deleted vertices is at most the weight of edges between different clusters. By Fact 2.6, the weight of deleted vertices is at most

$$\epsilon \bar{\omega}(G) \leq 3\epsilon \omega(G) \leq 12\epsilon OPT = \delta OPT$$

provided $\epsilon = \delta/12$. $\square$

Unlike the maximum independent set problem, in the case of matchings and dominating sets we can only give algorithms for the un-weighted versions of these problems. The reason is very simple, since G is planar, we know that the optimal solution to the maximum-weight independent set problem is of size proportional to $\omega(G)$. This however is not the case for the weight of an optimal solution to the max-wight matching or the min-weight dominating set problem. In fact, even when the weights are all equal to one, an optimal solution to the above two problems can be substantially smaller than the order of the graph. However, in the case a graph G is un-weighted, one can use simple preprocessing to reduce G to a graph where optimal solution is of size which is proportional to its order.

3.2   Maximum Matching

In the case of the maximum matching problem we can adopt the ideas from [CHS06] to show:

Theorem 3.3 *There is a deterministic distributed algorithm which for given $0 < \delta < 1$ finds in a planar graph $G = (V, E)$ in $O(\log^* |G|)$ rounds a matching M such that*

$$|M| \geq (1 - \delta)\beta(G)$$

where $\beta(G)$ is the size of a maximum matching in G.

Proof Outline.   The algorithm is similar to the one from [CHS06]. First, from [CHS06] (Lemma 6), we know that if induced stars of size at least two and induced double stars of size at least three are eliminated from G so that only one edge from a star and two paths of length two from each double star are left (which can be done in $O(1)$ rounds) then the matching in the new graph G' will be of size $\Omega(|G'|)$. Since at most one edge from a single star and at most two edges from a double star can be in a matching in G, we have $\beta(G) = \beta(G')$. Second, in G' we invoke the procedure CLUSTERING and find a partition of G' (the edge weights are initially equal to one). Finally, in each subgraph induced by partition classes we find an optimal solution, which can be done in $O(1)$ rounds, and return the union. The fact that the error of approximation is as claimed can be verified in the same way as in the argument for Theorem 3.2. $\square$

3.3   Minimum Dominating Set

In the case of the dominating set, we must do some more preprocessing. Specifically, the algorithm first finds a constant approximation of the minimum dominating set in $O(1)$ rounds

and then proceeds to improve this approximation and finds a dominating set of size $(1 - \delta)\gamma(G)$ where $\gamma(G)$ is the size of a minimum dominating set in G. To complete the second step $O(\log^* |G|)$ rounds are needed. As noted in the introduction, recently in [LOW08], a $O(1)$-rounds 71-approximation of the minimum dominating set for planar graphs is given. Although the constant approximation is not the focus of this paper we briefly describe an alternative way for finding a $O(1)$-approximation in Section **??**. Using a similar argument as in the case of matchings we have:

Theorem 3.4 *There is a deterministic distributed algorithm which for given $0 < \delta < 1$ finds in a planar graph $G = (V, E)$ in $O(\log^* |G|)$ rounds a dominating set D such that*

$$|D| \leq (1 + \delta)\gamma(G)$$

where $\gamma(G)$ is the size of a minimum dominating set in G.

Proof. The algorithm is similar to the one from [CH06a]. After a constant approximation is obtained using the algorithm from [LOW08], we proceed in the following fashion. Let $D = \{w_1, \ldots, w_k\}$ denote the dominating set obtained from the preprocessing phase. Then

$$|D| \leq c\gamma(G) \tag{3}$$

for some constant c. A partition $(W_1, \ldots, W_k)$ of $V(G)$ is obtained by every vertex of G joining the group of exactly one of the vertices from D that dominates it. Then $k = |D|$ and each V_i induces a graph of diameter at most two in G. We contract V_i's to obtain a planar graph H and set the weights of the edges to be equal to one. Note that $||H|| < 3k$. Set $\epsilon = \delta/(6 \cdot c)$ and use CLUSTERING to find a partition $(V_1, \ldots, V_l)$ of $V(H)$ which by Lemma 2.6 is such that the weight of the edges between different clusters is at most $\epsilon||H||$. Un-contract V_i's and W_i's to obtain a partition $(U_1, \ldots, U_l)$ of $V(G)$. Finally in each of $G[U_i]$'s (which has a constant diameter) find an optimal dominating set D_i and return the union of these dominating sets. The running time of the algorithm is $O(\log^* |G|)$. To prove the approximation error, let D^* be a minimum dominating set in G and let $\bar{D}$ be obtained from D^* by adding all vertices w_i from D with the property that a vertex in $W_i \subset U_j$ has a neighbor in $V \setminus U_j$. We have

$$|\bar{D}| \leq |D^*| + 2\epsilon||H|| < \gamma(G) + 6\epsilon|D| \leq \gamma(G)(1 + \delta)$$

from (3). In addition

$$|\bigcup D_i| \leq |\bar{D}|$$

as $\bar{D} \cap U_i$ is a dominating set in $G[U_i]$ and D_i is an optimal solution in $G[U_i]$. Therefore, $|\bigcup D_i| \leq (1 + \delta)\gamma(G)$. $\square$

4 Lower bounds and randomization

In this section, we will establish lower bounds for deterministic approximation algorithms and discuss randomized procedures.

4.1 Lower bounds

Our lower bounds will be based on the general Ramsey's theorem (see for example [We01]). It is known (see [Pa99]) that Ramsey's theorem can be used to argue that no deterministic distributed algorithm can properly color a cycle C with $O(1)$ colors and run in $o(\log^* |C|)$ rounds. We will first use this method to establish similar results for approximation algorithms. Let $R(2, m; l)$ denote the least number of vertices n such that in any 2-coloring of the edges of the complete l-uniform hypergraph on n vertices there is a monochromatic complete sub-hypergraph on m vertices. The general Ramsey's theorem shows that $R(2, m; l)$ is finite and a proof of the theorem provides a tower-type upper bound (height of tower is proportional to l) for $R(2, m; l)$.

Lemma 4.1 *For any positive integer T there is no deterministic distributed algorithm that finds in a cycle on n vertices an independent set of size $\Theta(n/\log^{(2T)} n)$ in T rounds. There is no deterministic distributed algorithm that finds an independent set of size $\Theta(n/\log^* n)$ in a cycle on n vertices in $o(\log^* n)$ rounds.*

Proof. For notational convenience we will assume that if $S = \{i_1, \ldots, i_l\}$ is a subset of $[n] := \{1, \ldots, n\}$ with l elements then the elements are indexed so that $i_k < i_l$ when $k < l$. Let C be a cycle with $V(C) = [n]$. For a distributed algorithm A that finds in T rounds an independent set in C, let $F_A : \binom{[n]}{2T+1} \to \{0, 1\}$ be defined by $F_A(\{i_1, \ldots, i_{2T+1}\}) = 1$ if and only if i_{T+1} is selected by A to the independent set provided $i_1, i_2, \ldots i_{2T+1}$ is a path in C. Then F_A is a 2-coloring of the edges of the complete $(2T + 1)$-uniform hypergraph H with $V(H) = [n]$. Let m be such that $n \geq R(2, m; 2T + 1)$. Then, from the Ramsey's Theorem, H contains a monochromatic complete hypergraph K on m vertices. Observe that if $m > 2T + 1$ then the color of the edges in K cannot be one. Indeed, if $\{i_1, \ldots, i_{2T+2}\}$ is a subset of $V(K)$ then by definition of F_A, i_{T+1} and i_{T+2} are selected by A to the independent set if the path $i_1, \ldots, i_{2T+2}$ is a subgraph of C. Consequently every edge in K has color zero and if $K = \{v_1, \ldots, v_m\}$ then none of $v_{T+1}, \ldots, v_{m-T}$ is in the independent set returned by A provided $P := v_1, \ldots, v_m$ is the subgraph of C. Therefore, out of m vertices in K, $m - 2T$ are not in the independent set returned by A. If $n - m \geq R(2, m; 2T + 1)$ then we can repeat the above argument to the hypergraph induced by $[n] \setminus V(K)$. Let p denote the number of times we will repeat the above reasoning. Then the size of the independent set returned by the algorithm A when the ordering of vertices in C is determined by the repeated application of the Ramsey's theorem is at most $2Tp + R(2, m; 2T + 1)$ which is at most $2nT/m + R(2, m; 2T + 1)$ as $pm \leq n$. It is known (see [N95]) that for some constant c,

$$R(2, m; 2T + 1) \leq 2^{2^{\cdots^{2^{cm}}}}$$

where the number of 2's in the tower is $2T$. Thus for any constant T if $m = \Theta(\log^{(2T)} n)$ then the size of the independent set is at most $O(n/\log^{(2T)} n)$. In addition, very similar computations give that for the size of an independent set to be $\Omega(n/\log^* n)$, T must be $\Omega(\log^* n)$. Indeed, let $\epsilon > 0$ and for n large enough let m and T be two integers with $(\log^* n)^2/(2c) \leq m \leq (\log^* n)^2/c$ and $\epsilon \log^* n/(8 \cdot c) \leq 2T + 1 < \epsilon \cdot \log^* n/(4 \cdot c)$. Then

$$2nT/m < \epsilon n/(2 \log^* n).$$

At the same time, for n large enough,

$$\log^{(2T+1)} R(2, m; 2T+1) \leq 2 \log \log^* n$$

and $2 \log \log^* n < 0.5 \log^{(2T+1)} n < \log^{(2T+1)} (\epsilon n/(2 \log^* n))$. Consequently, the size of the independent set returned by the algorithm is smaller than $\epsilon n/\log^* n$. $\square$

A very similar lower bound can be obtained for matchings. In the case of the dominating set the situation is different and it is possible to find a $O(1)$-approximation in zero rounds. On the other hand, one can show that no deterministic α-approximation with $\alpha < 2$ can run in $o(\log^* |G|)$ rounds. We will prove something slightly stronger for planar graphs.

Fact 4.2 *There is no deterministic distributed algorithm which for every $\delta > 0$ finds in $o(\log^* |G|)$ rounds a dominating set of size which is at most $(5 - \delta)\gamma(G)$ in a planar graph G.*

Proof. Let $\delta > 0$ be fixed and suppose that there is a deterministic distributed algorithm that finds a dominating set in any planar graph G of size at most $(5 - \delta)\gamma(G)$. From a cycle C on n vertices (with $10|n$), we create a graph G in the following way. Let $G = (V, E)$, where $V = V(C)$ and $E = E(C) \cup \{v, u \in V, d_C(v, u) = 2\}$. Note that this virtual graph can be obtained from C by a distributed algorithm, G is 4-regular, $\gamma(G) = n/5$, and G is planar. From a dominating set D in G returned by the algorithm we can obtain an independent set I in C as follows. Consider $C[D]$ and add to I all isolated vertices. For every vertex $v \in D$ with $deg_C(v) = 1$, if the neighbor of v, w, has $deg_C(w) = 1$ then add the vertex with a smaller identifier to I and if $deg_C(w) = 2$ then add v to I. Let D_i be the subset of D with vertices of degree i in $C[D]$. Note that if $v \in D_2$, that is v has degree two in $C[D]$, then every vertex from $V \setminus D$ which is dominated by v in G is also dominated by a vertex from D_1. Therefore, we have $|I| \geq |D_0| + |D_1|/2$ and $4|D_0| + 3|D_1| \geq n - |D|$. Since $|D| \leq (5 - \delta)\gamma(C) = (1 - \delta/5)n$, $4|D_0| + 3|D_1| \geq \delta n/5$ and so $|I| \geq |D_0| + |D_1|/2 \geq \delta n/30$. By Lemma 4.1 this is not possible. $\square$

4.2 Randomized algorithms

In this section we will briefly discuss randomized algorithms. We focus on the maximum independent set problem as algorithms for other problems can be obtained using similar consideration.

Lemma 4.3 *Let G be a graph with maximum degree at most Δ. Then there is distributed randomized algorithm which in two rounds finds an independent set I in G such that with probability larger than $1 - \exp\{-|G|/(2^{\Delta+4}(\Delta^2+1))\}$ the size of I is at least $|G|/(2^{\Delta+2}(\Delta^2+1))$.*

Proof. The algorithm is trivial and we will only outline it. It proceeds in two rounds and in the first round every vertex marks itself with probability $1/2$, choices made independently. In the second round a marked vertex un-marks itself if at least one of its neighbors is marked. Let S be a maximal set of vertices at distance at least three in G. Note that $|S| \geq |G|/(\Delta^2+1)$ and the events that two vertices from S are marked after the second round are independent. The expected number of vertices selected from S is at least $|G|/(2^{\Delta+1}(\Delta^2+1))$ and by Chernoff's bound the probability that it is smaller than $|G|/(2^{\Delta+2}(\Delta^2+1))$ is less than $\exp\{-|G|/(2^{\Delta+4}(\Delta^2+1))\}$. $\square$

Lemma 4.4 *Let K be a positive integer and let $G = (V, E)$ be a weighted planar graph with $\bar{\omega} : E \to \{1, \dots K\}$, the maximum degree Δ, and with no isolated vertices. Then there exist pair-wise disjoint subsets $V_1, \dots, V_l$ of $V(G)$ such that*

- $diam(G[V_i]) \leq 2(M-1)$;

- $\Delta(\tilde{G}) \leq \Delta^M$, $\max\{\bar{\omega}(e)|e \in E(\tilde{G})\} \leq K\Delta^M$;

- $\bar{\omega}(\tilde{G}) \leq 9\bar{\omega}(G)/10$

where $M := \lceil \log_{(1-1/(5 \cdot 82 \cdot 2^{10}))} \left(\frac{1}{20K\Delta} \right) \rceil$ and $\tilde{G}$ is obtained from G by contracting V_i's and re-computing the weights as in (1).

Proof. Let $\epsilon := 1/(20K\Delta)$ and let M be as above. Let $I = \{v \in V(G)|deg_G(v) \leq 9\}$. Clearly $|I| \geq 2|G|/5$. Use the algorithm from Lemma 4.3 to find an independent set $I_1 \subseteq I$ with $|I_1| \geq |G|/(5 \cdot 82 \cdot 2^{10})$. Repeat the process in $V \setminus I_1$ to find I_2 of size $(|G| - |I_1|)/(5 \cdot 82 \cdot 2^{10})$ and continue M times. Let $I_1, \dots, I_M$ be independent sets obtained by repeating the above procedure M times. Then with parameters as above every vertex $v \in I_j$ has at most nine neighbors in $V \setminus (I_1 \cup \dots \cup I_j)$ and with high probability, $\sum |I_j| \geq (1 - \epsilon)|G|$. For every j and every $v \in I_j$, v selects one edge from v to the set $V \setminus (I_1 \cup \dots \cup I_j)$ of the largest weight if it has at least one neighbor in $V \setminus (I_1 \cup \dots \cup I_j)$. The subgraph obtained in this way is a forest with trees $T_1, \dots, T_l$ each of diameter at most $2(M-1)$. Let $V_i := V(T_i)$. Note that the total weight of edges incident to $V \setminus (I_1 \cup \dots \cup I_M)$ is at most $K \cdot \Delta \cdot \epsilon|G|$ provided $\sum |I_j| \geq (1-\epsilon)|G|$. Therefore,

$$\sum \bar{\omega}(T_i) \geq (\bar{\omega}(G) - K \cdot \Delta \cdot \epsilon|G|)/9 \geq \bar{\omega}(G)/10$$

as $\epsilon = 1/(20K\Delta)$ and $\bar{\omega}(G) \geq |G|/2$. In addition, $\Delta(\tilde{G}) \leq \Delta^M$ as there are at most $(\Delta^M - 1)/(\Delta - 1)$ vertices in T_i and consequently $\max\{\bar{\omega}(e)|e \in \tilde{G}\} \leq K\Delta^M$. Finally, note that the probability that the above procedure fails is at most $O(M \exp\{-O(\epsilon|G|)\})$ by Lemma 4.3 as if in the course of computations $|G| - (|I_1| + \dots + |I_q|) \leq \epsilon|G|$ for some $q < M$ then we have $\sum |I_j| \geq (1-\epsilon)|G|$ with probability one. $\square$

Corollary 4.5 *There exists a randomized distributed algorithm which for a given $\delta > 0$ finds in the constant number of rounds (the constant depends on δ only) an independent set I in a planar graph G the size of which is with high probability larger than or equal to $(1 - \delta)\alpha(G)$.*

Proof. Let $R := 48/\delta$ and let $\bar{V} = \{v \in V(G)|deg_G(v) > R\}$. Then $|\bar{V}| \leq \delta\alpha(G)/2$ as $\alpha(G) \geq |G|/4$. Let G' be the subgraph of G induced by $V \setminus \bar{V}$ and define $\bar{\omega}(e) := 1$ when $e \in E(G')$. We apply Lemma 4.4 L times with L such that $(0.9)^L \leq \delta/16$ contracting subsets repeatedly and putting aside isolated vertices if any arise. Now we consider partition of V into $V_1, \dots, V_l$ obtained by un-contracting the vertices (some of the V_i's can be singletons). We have $diam(G[V_i]) \leq C$ where C depends on δ only and we call a vertex $v \in V_i$ *border* if it has a neighbor in $V(G) \setminus V_i$. Disregard all the border vertices to obtain $\bar{V}_i$'s and find optimal independent sets I_i's in $G[\bar{V}_i]$'s. Finally return the union of I_i's. The solution returned by the procedure has size of at least $\alpha(G) - |B|$ where B is the set of border vertices and $|B| \leq \delta\alpha(G)/2 + 2(0.9)^L|G| \leq \delta\alpha(G)$. $\square$

5 Acknowledgment

The authors are grateful to referees for many useful comments and suggestions.

References

[AGLP89] B. Awerbuch, A. V. Goldberg, M. Luby, S. A. Plotkin, Network Decomposition and Locality in Distributed Computation, Proc. 30th IEEE Symp. on Foundations of Computer Science , 1989, 364–369.

[CV86]   R. Cole, U. Vishkin, Deterministic coin tossing with applications to optimal parallel list ranking, Information and Control, 1986, 70, 32–53.

[CH06a]   A. Czygrinow, M. Hańćkowiak, Distributed almost exact approximations for minor-closed families, 14th Annual European Symposium on Algorithms (ESA), (2006), 244–255.

[CH06b]   A. Czygrinow, M. Hańćkowiak, Distributed approximation algorithms in unit disc graphs, 20th International Symposium on Distributed Computing (DISC), LNCS 4167, (2006), 385–398.

[CH06c]   A. Czygrinow, M. Hańćkowiak, Distributed algorithms for weighted problems in sparse graphs, Journal of Discrete Algorithms, Vol 4, (4), (2006), 588–607.

[CHS06]   A. Czygrinow, M. Hańćkowiak, E. Szymanska, Distributed approximation approximations for planar graphs, 6th Conference on Algorithms and Complexity (CIAC 2006), LNCS 3998, (2006), 296–307.

[D05]   R. Diestel,*Graph Theory*, 3rd Edition, Springer-Verlag, Berlin Heidelberg, (2005).

[E04]   M. Elkin, An Overview of Distributed Approximation, in ACM SIGACT News Distributed Computing Column Volume 35, Number 4, 132, (2004), 40–57.

[KMW04] F. Kuhn, T. Moscibroda, and R. Wattenhofer, What Cannot Be Computed Locally!, Proceedings of 23rd ACM Symposium on the Principles of Distributed Computing (PODC), (2004), 300–309.

[KMNW05a] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs, 19th International Symposium on Distributed Computing (DISC), Cracow, Poland, September (2005), LNCS 3724, (2005), 273–287.

[KMNW05b] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, Local Approximation Schemes for Ad Hoc and Sensor Networks, 3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), Cologne, Germany, (2005), 97–103.

[LOW08]   C. Lenzen, Y. A. Oswald, R. Wattenhofer, What Can Be Approximated Locally?, 20th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2008) Munich, Germany, (2008), 46–54.

[LW08]   C. Lenzen, R. Wattenhofer, Leveraging Linial's Locality Limit, 22nd International Symposium on Distributed Computing (DISC 2008), Arcachon, France, (2008), to appear.

[L92]   N. Linial, Locality in distributed graph algorithms, SIAM Journal on Computing, 1992, 21 (1), 193–201.

[N91]   M. Naor, A lower bound on probabilistic algorithms for distributive ring coloring, SIAM J. on Discrete Mathematics, 4(3), (1991), 409–412.

[N95]   J. Nesetril, Ramsey Theory, in Handbook of Combinatorics Vol. II (Edited by R.L. Graham, M. Groschel, L. Lovász), Elsevier (1995), 1331–1403.

[Pa99]   A. Panconesi, Distributed Algorithms Notes, (manuscript).

[Pe00]   D. Peleg, Distributed Algorithms, A Locality-Sensitive Approach, SIAM Press, (2000).

[SW08]   J. Schneider, R. Wattenhofer, A log* Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs, (PODC 2008), to appear.

[We01]   D. West, *Introduction to graph theory*, 2nd Ed., Penitence-Hall Inc., (2001).