Distributed Approximation Algorithms for Planar Graphs

Andrzej Czygrinow[1], Michał Hańćkowiak[2], and Edyta Szymańska[2,⋆]

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

[2] Faculty of Mathematics and Computer Science,
Adam Mickiewicz University, Poznań, Poland
mhanckow@amu.edu.pl, edka@amu.edu.pl

Abstract. In this paper we construct two distributed algorithms for computing approximations of a largest matching and a minimum dominating set in planar graphs on n vertices. The approximation ratio in both cases approaches one with n tending to infinity and the number of synchronous communication rounds is poly-logarithmic in n. Our algorithms are purely deterministic.

1 Introduction

The distributed model of computation has gained a lot of attention after the pioneering work by Awerbuch et. al. [AGLP89] and many others in the mid eighties of the last century. The most fundamental challenge in distributed networks is how the local structure of a network impacts its global properties. This leads to a completely different computational paradigm than the sequential model or the parallel PRAM model. Not surprisingly many problems which admit efficient sequential protocols, as maximum matching or maximal independent set, to name a few, require a completely new algorithmic approach and yield interesting open problems in discrete mathematics.

The model considered in this paper was introduced by Linial in [L92] and named $\mathcal{LOCAL}$ in [P00]. In this model, the network is represented by an undirected graph, each vertex of which corresponds to a processor, and each edge corresponds to a communication channel between two processors. The network is synchronized and computations proceed in discrete rounds. In a single round a vertex can send and receive messages to and from its neighbors, and perform some local computations. Neither the amount of local computations nor the lengths of messages are restricted in any way. In addition, we assume that vertices have unique identifiers. There are several measures of efficiency of distributed protocols but we will concentrate on its time complexity, that is, a maximum number of rounds needed to find a solution. An algorithm is efficient if its time complexity is poly-logarithmic in n.

⋆ The third author thanks the Department of Mathematics and Computer Science at Emory University for providing an office space and computer access.

T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.): CIAC 2006, LNCS 3998, pp. 296–307, 2006.

Very few classical graph-theoretic problems admit efficient distributed algorithms. For example, even the maximal independent set problem, for which an efficient deterministic PRAM algorithm exists [L86], still has an unknown distributed complexity. Another approach towards a better understanding of a computational model, is to study the approximability of problems in that model. This has motivated intensive research on approximation algorithms in the distributed model. For the state of the art of distributed approximation we refer the reader to an excellent survey by Elkin [E04].

In this paper we design distributed algorithms for planar graphs and exploit the fact the planar graphs are minor monotone. In a given graph $G = (V, E)$ every set of pairwise disjoint edges constitutes a matching. Let $\beta(G)$ denote the cardinality of a largest matching in G. The *maximum matching* problem is to find a matching M in graph G of size $\beta(G)$. A *dominating set* in a graph G is a subset D of vertices such that for every vertex $v \notin D$ a neighbor u of v belongs to D. By $\gamma(G)$ we will denote the cardinality of a smallest dominating set in G, also known as the *domination number* . The *minimum dominating set* problem is to find a dominating set D in graph G of size $\gamma(G)$. We propose two purely deterministic distributed algorithms with the poly-logarithmic time complexity. For every planar graph on n vertices, our first algorithm finds a matching M such that $|M| \geq (1 - O(\frac{1}{\log n}))\beta(G)$ (see Theorem 1). The second algorithm works for planar graphs that do not contain $K_{2,\log n}$ as a subgraph. In every such graph on n vertices it finds a dominating set D such that $|D| \leq (1+O(\frac{1}{\log n}))\gamma(G)$ (see Theorem 2). Although this technical assumption certainly restricts the applicability of the method, the subclass of $K_{2,\log n}$-free planar graphs is quite large and contains, for example, outer-planar graphs (they do not contain $K_{2,3}$).

To give an overview of previous research, let us mention that there exists no efficient distributed protocol for finding a maximum matching or a minimum dominating set even when restricted to very particular families of networks. As shown by Linial in [L92], finding a maximum matching in a cycle on n vertices requires $\Omega(n)$ rounds and the same bound holds for minimum dominating set. More recently, it has been shown in [KMW04] that the number of rounds required in order to achieve a constant or even only a poly-logarithmic approximation ratio for constructing an inclusion maximal matching and a minimum dominating set is at least $\Omega(\sqrt{\log n/ \log\log n})$ or $\Omega(\log \Delta/ \log\log \Delta)$, where Δ denotes the maximum degree of the graph.

A maximal matching problem admits a $O(\log n)$ time randomized distributed algorithm (see, for example [L86]). Later, in [HKP99] a deterministic, poly-logarithmic time algorithm for this problem was given. The techniques from [HKP99] were applied in [CHS04] and [CHSz04] to give a 2/3-approximation for maximum matching in general graphs. Moreover, based on these ideas, in [CH03] a $(1 - \epsilon)$-approximation (for any fixed $\epsilon > 0$) for bipartite graphs was derived.

For the minimum dominating set problem, Kutten and Peleg [KP95] gave an efficient distributed algorithm which finds a dominating set of size at most $n/2$ in general graphs. The first non-trivial approximation ratio, $O(\log \Delta)$ was achieved in [JRS01] by a randomized method. Further, in [KW03] a $O(k\Delta^{2/k}\log\Delta)$-approximation in constant time was obtained using the LP relaxation techniques with randomization. Similar, randomized result for the connected dominating set can be found in [DPRS03]. In contrast, our results are purely deterministic and are among only few examples of distributed protocols where the poly-logarithmic time complexity with a very good approximation ratio is achieved without the use of random bits.

Approximations of weighted versions of the maximum matching and minimum dominating set problems were recently studied in [CH04]. Our proof techniques rely on a clustering procedure introduced in [CH04]. We further develop the method in this work and hope that it might be applied for other problems. By a cluster we mean a subset of the vertex set that induces a connected subgraph. The clustering procedure partitions the nodes of the input graph into clusters. If the diameter of a cluster is poly-logarithmic in n, then, in the $\mathcal{LOCAL}$ model, we can compute every function efficiently. Therefore, having the vertices grouped into clusters we find the maximum matching in every cluster. The union of the matchings yields a matching of size approximating $\beta(G)$. The situation is similar but more complicated in the case of the minimum dominating set. We first make sure that all vertices of large degree are included in the dominating set. Then the clustering is performed and a set of vertices dominating the remaining vertices is constructed within the clusters.

In both problems the number of rounds of our algorithms is a poly-logarithmic function determined by the diameters of the clusters. At the same time, we control the number of edges connecting different clusters and based on that value the approximation ratio is derived. For both algorithms, better approximation ratios can be achieved at the expense of higher running times.

The rest of the paper is organized as follows. In Section 2, we present the clustering algorithm. Sections 3 and 4 contain the description and analysis of the approximation algorithms for the maximum matching and the minimum dominating set, respectively.

2 Clustering Algorithm

In this section, we give a clustering algorithm which will be applied to find matchings as well as dominating sets. We will use the low-degree decomposition of a planar graph from [CH04].

Definition 1. *A low-degree decomposition of a planar graph $G = (V, E)$ is a partition of V into K independent sets $V_1, \ldots, V_K$ that satisfies two conditions:*

1. $K = O(\log |V|)$.
2. *For every $i = 1, \ldots, K-1$, if $v \in V_i$ then $deg(v, \bigcup_{l=i+1}^{K} V_l) \leq 6$.*

It is not difficult to prove that every planar graph admits a low-degree decomposition. In addition, as shown in [CH04], such a decomposition can be found efficiently by a distributed algorithm.

DECOMPOSITION

Input: Planar graph G, number n such that $|V(G)| \leq n$.
Output: Low-degree decomposition $V_1, \ldots, V_{\log_k n}$ of G with $k = 36/35$.

1. Let $U := V(G)$, $i := 1$.
2. Iterate $\log_{36/35} n$ times:
   (a) Let A be the set of vertices in $G[U]$ of degree at most 6.
   (b) Use the Cole-Vishkin algorithm from [CV86] to find a maximal independent set I in the subgraph of $G[U]$ induced by A.
   (c) $V_i := I$, $i := i + 1$, $U := U \setminus I$.

Lemma 1. *[CH04] Let $G = (V, E)$ be a planar graph such that the identifiers of V are in $\{1, \ldots, n\}$. Then the procedure* DECOMPOSITION *finds a low-degree decomposition of G in $O(\log^* n \log n)$ rounds.*

Our approximation algorithms will use a similar clustering strategy as the one in [CH04]. In addition to procedure CLUSTERING we introduce a subprocedure SMALLCLUSTERS. The latter computes clusters of a constant diameter and in each cluster finds a set of vertex disjoint stars with special properties that can be used by CLUSTERING to compute "big clusters". Thanks to this approach we save on the time complexity for constructing the clusters (see Lemma 4).

SMALLCLUSTERS

Input: Planar graph $G = (V, E)$ with weights on edges $\omega : E \mapsto \mathcal{R}^+$ and number n such that $|V| \leq n$ and $ID(v) \leq n$.
Output: Set of vertex-disjoint stars in G.

1. $H := G$.
2. Iterate $\log 10 / \log \frac{12}{11}$ times:
   (a) Call DECOMPOSITION to find a partition $W_1, \ldots, W_{\log_{36} n}$ of H. In addition, let $Z_i := \bigcup_{l>i} W_l$.
   (b) For every vertex w, in parallel, if $w \in W_i$ and $N(w) \cap Z_i \neq \emptyset$ then:
      – Let $u(w)$ be a vertex in $N(w) \cap Z_i$ such that
      $$\omega(\{w, u(w)\}) = \max_{v \in N(w) \cap Z_i} \omega(\{w, v\}).$$
      – Add $\{w, u(w)\}$ to the auxiliary graph F.
   (c) Each connected component of F is a tree of diameter $O(\log n)$. For each tree T in F, in parallel, find a set of disjoint stars $S_1, S_2, \ldots,$ in T of the maximum weight.
   (d) Modify H as follows:
      – In each star, contract vertices to create a new vertex. Let $V(H)$ consist of new vertices and those vertices which were not contracted.
      – For every $v, w \in V(H)$ set the weight of $\{v, w\}$ to be the sum of weights of edges between vertices contracted to v and vertices contracted to w.

3. If $V(H) = \{v_1, ..., v_M\}$ then for each v_i let V_i be the set of vertices contracted to v_i in all of the above iterations.
4. In each $G[V_i]$, in parallel, compute a set of disjoint stars $Q_1^{(i)}, \ldots, Q_{M(i)}^{(i)}$ of the largest possible weight.
5. Return the set of stars $Q_j^{(i)}$, for $i = 1, \ldots, M$; $j = 1, \ldots, M(i)$.

Let κ be the supremum of all real numbers r such that every weighted planar graph G contains a set of vertex-disjoint stars with the total weight of at least an r fraction of the weight of G. We need the following lemma.

Lemma 2. $\kappa \geq \frac{1}{5}$.

Proof. A star forest is a forest in which every connected component is a star and the star arboricity of a graph G, $st(G)$, is the minimum number of star forests that partition $E(G)$. Hakimi et al. [HMS96] showed that if G is planar then $st(G) \leq 5$ and so there is a set of vertex-disjoint stars with weight of at least $\omega(G)/5$ where $\omega(G) = \sum_{e \in E} \omega(e)$.

Lemma 3. *Let* $Q_1, \ldots, Q_L$ *be the disjoint stars in* G *obtained from* SmallClusters. *Then*

$$\omega(\bigcup_i Q_i) \geq \frac{9}{10}\kappa\omega(G).$$

Proof. Let σ_i be the maximum diameter of a subgraph of G which corresponds to a vertex of H in the ith iteration. Then $\sigma_i \leq 3\sigma_{i-1} + 2$ with $\sigma_0 = 0$ which gives $\sigma_i < 2 \cdot 3^i$ and so $\sigma_k < 2 \cdot 3^{27}$ for $k = \log 10/\log \frac{12}{11}$. Therefore each subgraph $G[V_i]$ in step 4 has a constant diameter and its optimal set of stars can be computed in a constant number of rounds. Let P_i be the sum of weights of edges in H in the ith iteration. In the next iteration $w(F)$ is at least $P_i/6$ and the stars $S_1, S_2, \ldots,$ in each tree of F have the weight of at least $\omega(T)/2$. Consequently, the weight of the graph in the $(i+1)$st iteration, P_{i+1}, is at most $\frac{11}{12}P_i$ and $P_0 = \omega(G)$. This gives $P_k \leq \frac{1}{10}\omega(G)$ for $k = \log 10/\log \frac{12}{11}$. By Lemma 2, the weight of stars in $G[V_i]$ is larger then $\frac{9}{10}\kappa\omega(G)$.

The procedure SmallClusters is now used in Clustering given below.

Clustering

Input: Planar graph $G = (V, E)$ and number n such that $|V| \leq n$ and $ID(v) \leq n$. Number $c \geq 1$.

Output: Partition of V into L sets $V_1, \ldots, V_L$.

1. $H := G$ and let $\omega(e) := 1$ for any $e \in E(H)$.
2. Iterate $c \log\log n/\log \frac{1}{1-\frac{9}{10}\kappa}$ times:
   (a) Call SmallClusters to find set of disjoint stars $S_1, S_2, \ldots$ in H.
   (b) Modify H as in step 2(d) of SmallClusters

3. If $V(H) = \{v_1, ..., v_L\}$ then for each v_i let V_i be the set of vertices contracted to v_i in all of the above iterations.
4. Return sets $V_1, \ldots, V_L$.

We summarize the Clustering in the next lemma.

Lemma 4. *Let $V_1, \ldots, V_L$ be the clusters in G obtained from* Clustering. *Then*

1. *For every i, $G[V_i]$ is a subgraph of diameter $O(\log^d n)$, where*
$$d = c \log 3 / \log \frac{1}{1 - \frac{9}{10}\kappa} < 5.54c.$$
2. *The number of edges connecting different clusters is $O(|E(G)|/\log^c n)$.*
3. Clustering *can be performed in $O(\log\log n \log^* n \log^{1+d} n)$ rounds.*

Proof. Analogously to the proof of Lemma 3 we have $\sigma_i < 2 \cdot 3^i$ and so $\sigma_k < 2 \log^d n$ for $k = c \log\log n / \log \frac{1}{1-\kappa\frac{9}{10}}$. Then, for the second part, we have $P_{i+1} \leq (1 - \kappa\frac{9}{10})P_i$ and $P_0 = |E(G)|$, and so $P_k = O(|E(G)|/\log^c n)$. Finally, the third part of the lemma follows from the fact that we have $O(\log\log n)$ iterations of step 2 and in each iteration we invoke SmallClusters that calls the Decomposition a constant number of times. Decomposition, in turn, needs $O(\log^* n \log n)$ rounds. Since the diameter of each cluster (which corresponds to a vertex of H) is $O(\log^d n)$, Clustering needs $O(\log^* n \log^{1+d} n)$ rounds.

3 Maximum Matching

In this section, we will give a distributed algorithm which approximates a maximum matching in a planar graph G. The algorithm consists of two main parts. First we modify the graph G to obtain a new graph $\bar{G}$ and then we invoke the clustering algorithm for $\bar{G}$ and find a maximum matching locally in each cluster. Recall that the total number of edges connecting different clusters is small in comparison with the number of vertices in the graph. However, a maximum matching in a planar graph can be much smaller than the number of vertices and so if clustering is invoked in such a graph its result would not yield a good approximation. The preprocessing phase addresses this issue. It obtains from a graph G a subgraph $\bar{G}$ with the property that $\beta(G) = \beta(\bar{G}) = \Omega(|V(\bar{G})|)$.

The first phase of the algorithm, the preprocessing, eliminates (by deleting some of the vertices) two special subgraphs of G: the stars and the double-stars. We say that G contains a *k-star* if for some $v, v_1, \ldots, v_k \in V(G)$, $\{v, v_i\} \in E(G)$ for every i, and $deg_G(v_i) = 1$ for every i. In a similar way, we say that G contains a *k-double-star* if for some $u, v, v_1, \ldots, v_k \in V(G)$, $\{u, v_i\} \in E(G)$ and $\{v, v_i\} \in E(G)$ for every i, and $deg_G(v_i) = 2$ for every i. Every such structure contributes at most two edges to any maximum matching in G. In the next two lemmas we shall show that if H contains neither 2-stars nor 3-double-stars then $\beta(H) = \Omega(|V(H)|)$.

Lemma 5. *Let $H = (V, E)$ be a planar graph and let $\tau = |\{v \in V : deg(v) \geq 3\}|$. Then $\beta(H) \geq (\tau + 4)/6$.*

Proof. Let M be a matching in H with $|M| = \beta(H)$. Let V_1 be the set of M-saturated vertices. Then, since M is a maximum matching, $V \setminus V_1$ induces the empty subgraph of H. Let $V_2 := (V \setminus V_1) \cap \{v \in V : deg(v) \geq 3\}$. Consider the bipartite graph $F = H[V_1, V_2]$. As F is planar, $|E(F)| \leq 2(|V_1| + |V_2|) - 4$. On the other hand, $3|V_2| \leq |E(F)|$. Thus $|V_2| + 4 \leq 2|V_1|$. However, $|V_1| = 2\beta(H)$ and $|V_2| \geq \tau - 2\beta(H)$ yields $\beta(H) \geq (\tau + 4)/6$.

Lemma 6. *Let $G = (V, E)$ be a planar graph with $n = |V|$ and no isolated vertices. If G contains neither 2-stars nor 3-double-stars then $\beta(G) = \Omega(n)$.*

Proof (Sketch). By Lemma 5 we may concentrate only on the set $W = \{v \in V(G) : deg(v) = 2\}$ and the case when, say $|W| \geq 14n/15$. Then, for a subset $W' \subseteq W$ such that if $w \in W'$ then $deg(w, W) \geq 1$, using planarity and the assumption about the absence of 2-stars and 3-double-stars, we have $|W'| \geq |W|/2$ and $\beta(G) \geq |W'|/3 = 7n/45$.

PREPROCESS
Input: Planar graph G.
Output: Planar graph $\bar{G}$ with no 2-stars and no 3-double stars.

1. For every vertex v, in parallel, find the largest k-star $v, v_1, \ldots, v_k$ with the center in v. If $k > 1$ then delete $v_2, \ldots, v_k$.
2. For every pair of vertices u, v which are at distance two, in parallel, find the largest k-double-star $u, v, v_1, \ldots, v_k$ with centers in u and v. If $k > 2$ then delete $v_3, \ldots, v_k$.
3. Return the new graph $\bar{G}$.

Clearly $\bar{G}$ contains neither 2-stars nor 3-double-stars, as in the second step we did not create any vertices of degree one. Thus, by Lemma 6, $\beta(\bar{G}) = \Omega(|V(\bar{G})|)$. In addition, it is easy to see that

$$\beta(\bar{G}) = \beta(G). \tag{1}$$

We can now describe our approximation algorithm.

APPROXMAXMATCHING
Input: Planar graph G.
Output: Matching M in G.

1. Call PREPROCESS to obtain $\bar{G}$.
2. Call CLUSTERING with $c = 1$ to partition $V(\bar{G})$ into clusters $V_1, \ldots, V_L$.
3. For every i, in parallel, find a maximum matching M_i in $\bar{G}[V_i]$.
4. Return $M := M_1 \cup M_2 \cup \cdots \cup M_L$.

Theorem 1. ApproxMaxMatching *finds in a planar graph G on n vertices a matching M with*

$$|M| \geq (1 - O(1/\log n))\,\beta(G).$$

The algorithm runs in $O(\log\log n \log^* n \log^{1+d} n)$ rounds, where $d = 5.54$.

Proof. Consider a maximum matching M^* in $\bar{G}$ and let M_i^* be the subset of M^* which contains all edges with both endpoints in V_i. In addition, let C be the set of edges that connect different clusters. We have

$$|M^*| \leq |C| + \sum_{i=1}^{L} |M_i^*| \leq |C| + \sum_{i=1}^{L} |M_i| = |C| + |M|.$$

By Lemma 4 (part 2), $|C| \leq |V(\bar{G})|/\log n$ which in view of Lemma 6 gives $|C| \leq O(\beta(\bar{G}))/\log n$. Consequently,

$$|M| \geq \beta(\bar{G}) - |C| = \beta(\bar{G})(1 - O(1/\log n))$$

which by (1) gives $|M| \geq (1 - O(1/\log n))\beta(G)$.

4 Minimum Dominating Set

We will now turn our attention to the minimum dominating set problem. We assume that $G = (V, E)$ is a planar graph on n vertices such that for any two vertices $u, v \in V$ $|N(u) \cap N(v)| \leq \log n$. Again the algorithm has two phases. In the first phase we add to a dominating set vertices with degrees of at least $\log^2 n$. Then we consider two sets of vertices. Let V_{SN} be the set of vertices of degree smaller than $\log^2 n$ which do not have neighbors of degree at least $\log^2 n$, that is

$$V_{SN} = \{v \in V : \forall_{u \in N[v]} deg(u) < \log^2 n\}.$$

Let $V_{BN} \subset V \setminus V_{SN}$ be the set of vertices which have degree smaller than $\log^2 n$ but have a neighbor in V_{SN}, that is

$$V_{BN} = \{v \in V \setminus V_{SN} : deg(v) < \log^2 n, \exists_{u \in V_{SN}} \{u, v\} \in E\}.$$

In the second phase of the algorithm we shall find a clustering using Clustering in the graph induced by $V_{SN} \cup V_{BN}$ and locally, in each cluster V_i, we will find a set of the smallest size which dominates $V_i \cap V_{SN}$.

ApproxMinDS
Input: Planar graph $G = (V, E)$.
Output: A dominating set D in G.

1. Let $D := \emptyset$.
2. For every vertex v, in parallel, if $deg(v) \geq \log^2 n$ then add v to D.

3. Let $G' = G[V_{BN} \cup V_{SN}]$. Call CLUSTERING with constant $c = 5$ to partition $V(G')$ into clusters $V_1, \dots, V_L$. Let V_i' be the set of vertices v in $V(G') \setminus V_i$ such that for some $u \in V_i$, $\{v, u\} \in E$ and let V_i'' be the set of vertices $v \in V_i$ such that for some $u \in V_i'$, $\{v, u\} \in E$.
4. For every $i = 1, \dots, L$, in parallel, find a smallest set $D_i' \subseteq V_i$ which dominates $(V_{SN} \cap V_i) \setminus V_i''$. Let $D_i := D_i' \cup V_i'$.
5. For every $i = 1, \dots, L$, in parallel, add all vertices from D_i to D.
6. Return D.

In the lemma below we analyze the first phase of the algorithm where vertices of degree at least $\log^2 n$ are added to D.

Lemma 7. *Let $G = (V, E)$ be a planar graph such that for any two distinct vertices $u, v \in V$, $|N(u) \cap N(v)| \leq \log n$ and let $B = \{v \in V : deg(v) \geq \log^2 n\}$. If D^* is a dominating set in G then $|B \setminus D^*| = O(|D^*|/\log n)$.*

Proof. We will show that $|D^*| = \Omega(|B \setminus D^*| \log n)$. For that we first prove that there is a subset $\{w_1, \dots, w_k\} \subseteq B \setminus D^*$ of at least $k = |B \setminus D^*|/10$ vertices such that each w_i has a set $S_i \subseteq (N(w_i) \setminus \{w_1, \dots, w_k\})$ of $\frac{\log^2 n}{4}$ neighbors and $S_i \cap S_j = \emptyset$ whenever $i \neq j$. Indeed, note that as $G[B \setminus D^*]$ is planar there is an independent set I in $G[B \setminus D^*]$ of at least $2k$ vertices. Take $w_1 \in I$ arbitrarily and let S_1 be a set of $\frac{\log^2 n}{4}$ neighbors of w_1. Now suppose that $\{w_1, \dots, w_l\}$ have been selected with $l < k$. Consider the bipartite subgraph of G with bipartition $W = I \setminus \{w_1, \dots, w_l\}$ and $S = \bigcup_{i=1}^{l} S_i$. Then $G[W, S]$ is a planar graph and so

$$|E(W, S)| \leq 2(|W| + |S|) - 4 = 2\left(|W| + \frac{l \log^2 n}{4}\right) - 4 < 2\left(|W| + \frac{|W| \log^2 n}{4}\right) - 4.$$

Consequently,

$$|E(W, V \setminus S)| > \log^2 n|W| - 2\left(|W| + \frac{|W| \log^2 n}{4}\right) + 4 =$$

$$|W|\left(\frac{\log^2 n}{2} - 2\right) + 4 \geq \frac{|W| \log^2 n}{4}$$

and we can select w_{l+1} from W which is connected with at least $\log^2 n/4$ vertices from $V \setminus S$.

Let $(w_1, S_1), \dots, (w_k, S_k)$ be as above. Let D be a subset of $V \setminus \{w_1, \dots, w_k\}$ which dominates $S = \bigcup_{i=1}^{k} S_i$ in G. We claim that $|D| = \Omega(k \log n)$. Consider $D' = D \cap S$. If $|D'| \geq k \log n$ then we are done. Otherwise, let $S_i' = S_i \setminus D$. Since $|S| - |D'| \geq k\left(\frac{\log^2 n}{4} - \log n\right)$ at least $k/2$ of w_i's have $|S_i'| \geq \frac{\log^2 n}{8}$. Otherwise

$$\sum_{i=1}^{k} |S_i'| < \frac{k}{2}\left(\frac{\log^2 n}{8} + \frac{\log^2 n}{4}\right) < |S| - |D'|$$

which is not possible. Without loss of generality, we can assume that for $i = 1 \ldots, k/2$, $|S'_i| = \frac{\log^2 n}{8}$. Note that $\bigcup_{i \leq k/2} S'_i \cap D = \emptyset$. Consider the auxiliary bipartite graph $H = (V_1, V_2)$ obtained by setting $V_1 = D$ and contracting each S'_i to one vertex and adding it to V_2. Put an edge between $v \in V_1$ and $S'_i \in V_2$ if v dominates at least one vertex from S'_i in G. First observe that H is planar as all edges correspond to edges in G and so a subdivision of $K_{3,3}$ or K_5 in H will yield the subdivision of $K_{3,3}$ or K_5 in G. Thus $|E(H)| \leq 2(|V_1| + |V_2|) - 4 = 2\left(\frac{k}{2} + |D|\right) - 4$. Degree of each S'_i in H is at least $\frac{\log n}{8}$ as if for some i, there are less than $\frac{\log n}{8}$ vertices dominating S'_i then one of them has more than $8|S'_i|/\log n = \log n$ neighbors in S'_i and so more than $\log n$ common neighbors with w_i. Thus,

$$\frac{k \log n}{8} \leq |E(H)| \leq k + 2|D| - 4$$

and so

$$|D^*| \geq |D| \geq \left(\frac{\log n}{8} - 1\right)\frac{k}{2} + 2 = \Omega(k \log n) = \Omega(|B \setminus D^*| \log n).$$

In the next lemma, we analyze the second phase.

Lemma 8. *Let $G = (V, E)$ be a planar graph and let $D' = \bigcup_{i=1}^{L} D_i$ be the union of sets obtained by* ApproxMinDS *in step four. Let D^*_{SN} be a set of the smallest size which dominates V_{SN} in G . Then $|D'| \leq (1 + O(1/\log n))\,|D^*_{SN}|$.*

Proof. First note that $D^*_{SN} \subseteq V_{SN} \cup V_{BN}$. Since every vertex in $V_{SN} \cup V_{BN}$ has degree of at most $\log^2 n$ we have

$$|D^*_{SN}| \geq |V_{SN}|/\log^2 n \text{ and } |V_{BN}| \leq |V_{SN}| \log^2 n.$$

On the other hand, by Lemma 4, the number of edges connecting different clusters $e_{clusters}$ is at most

$$|E(G[V_{SN} \cup V_{BN}])|/\log^5 n < 3(\log^2 n + 1)|V_{SN}|/\log^5 n \leq$$

$$3 \log^2 n(\log^2 n + 1)|D^*_{SN}|/\log^5 n = O(|D^*_{SN}|/\log n).$$

Thus

$$e_{clusters} = O(|D^*_{SN}|/\log n). \tag{2}$$

We claim that

$$|D'| \leq |D^*_{SN}| + 2e_{clusters}. \tag{3}$$

Indeed, vertices from $D^*_{SN} \cap V_i$ dominate all vertices from $(V_{SN} \cap V_i) \setminus V''_i$ as any vertex in the latter set has all of its neighbors in V_i. Thus

$$\sum_{i=1}^{L} |D'_i| \leq \sum_{i=1}^{L} |D^*_{SN} \cap V_i| = |D^*_{SN}|,$$

and so

$$|D'| \le \sum_{i=1}^{L}(|D'_i| + |V'_i|) \le |D^*_{SN}| + \sum_{i=1}^{L}|V'_i| \le |D^*_{SN}| + 2e_{clusters}$$

which verifies (3). Finally, by (3) and (2),

$$|D'| \le (1 + O(1/\log n))\,|D^*_{SN}|.$$

We can now summarize the performance of ApproxMinDS.

Theorem 2. *Let $G = (V, E)$ be a planar graph on n vertices such that for any two distinct vertices u, v, $|N(u) \cap N(v)| \le \log n$. Then* ApproxMinDS *finds a dominating set D in G with*

$$|D| \le (1 + O(1/\log n))\gamma(G).$$

Procedure ApproxMinDS *runs in $O(\log\log n \log^* n \log^{1+d} n)$ rounds, where $d = 27.7$.*

Proof. To see that D is a dominating set note that after the second step of ApproxMinDS all vertices with degree of at least $\log^2 n$ or which have a neighbor of such a degree are dominated by D. Therefore, only vertices from V_{SN} are not dominated at this moment. However $D_i := D'_i \cup V'_i$ dominates all vertices in $V_{SN} \cap V_i$ and so D is a dominating set in G. Now let D^* be a dominating set in G of the minimum size and, as in Lemma 7, let $B = \{v : deg(v) \ge \log^2 n\}$. We have $D = B \cup D'$ where $D' = \bigcup_{i=1}^{L} D_i$ and so $|D| \le |B| + |D'|$. By virtue of Lemma 7,

$$|B| = |B \cap D^*| + |B \setminus D^*| = |D^* \cap B| + O\left(\frac{|D^*|}{\log n}\right).$$

In addition, $|D^*_{SN}| \le |D^* \cap S|$ as every vertex in V_{SN} can be dominated only by vertices of degree less than $\log^2 n$. Consequently, be Lemma 8,

$$|D'| \le (1 + O(1/\log n))\,|D^*_{SN}| \le (1 + O(1/\log n))\,|D^* \cap S|.$$

Thus

$$|D| \le |D^* \cap B| + O\left(\frac{|D^*|}{\log n}\right) + (1 + O(1/\log n))\,|D^* \cap S| = (1 + O(1/\log n))\,|D^*|.$$

Finally, by Lemma 4, the number of rounds is $O(\log\log n \log^* n \log^{1+d} n)$.

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, pp. 364-369.

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

[CH03] A. Czygrinow, M. Hańćkowiak, Distributed Algorithm for Better Approximation of the Maximum Matching, COCOON 2003 LNCS 2697 , 2003, pp. 242-251.

[CH04] A. Czygrinow, M. Hańćkowiak, Distributed algorithms for weighted problems in sparse graphs, to appear in Journal of Discrete Algorithms, in Press, (Available online 7 September 2005)

[CHS04] A. Czygrinow, M. Hańćkowiak, E. Szymańska, Distributed algorithm for approximating the maximum matching, Discrete Applied Mathematics, Volume 143, Issues 1-3, (2004), 62–71.

[CHSz04] A. Czygrinow, M. Hańćkowiak, E. Szymańska, A fast distributed algorithm for approximating the maximum matching, Algorithms - ESA 2004 LNCS 3221, (2004), 252–263.

[DPRS03] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan, Fast Distributed Algorithms for (Weakly) Connected Dominating Sets and Linear-Size Skeletons, In Proc. of the ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 717-724, 2003.

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

[HMS96] S. Hakimi, J. Mitchem, E. Schmeichel, Star arboricity of graphs, Discrete Mathematics, 149, 1-3, (1996), 93–98.

[HKP99] M. Hańćkowiak, M. Karoński, A. Panconesi, A faster distributed algorithm for computing maximal matching deterministically, Proceedings of PODC 99, the Eighteen Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, pp. 219-228.

[JRS01] L. Jia, R. Rajaraman, and R. Suel, An Efficient Distributed Algorithm for Constructing Small Dominating Sets, In Proc. of the 20th ACM Symposium on Principles of Distributed Computing (PODC), pp. 33-42, 2001.

[KP95] S. Kutten, D. Peleg, Fast distributed construction of k-dominating sets and applications, Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing, 1995, pp. 238 - 251.

[KW03] F. Kuhn, R. Wattenhofer, Constant-Time Distributed Dominating Set Approximation, 22nd ACM Symposium on the Principles of Distributed Computing (PODC), Boston, Massachusetts, USA, July 2003.

[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, pp. 300-309.

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

[L86] M. Luby, A simple parallel algorithm for the maximal independent set problem, SIAM J. Comput., vol 15(4), 1986, pp. 1036-1053.

[P00] D.Peleg, Distributed Computing: A Locality-Sensitive Approach, SIAM, 2000.