What Can Be Approximated Locally? Case Study: Dominating Sets in Planar Graphs

TIK Report Number 331

Christoph Lenzen, Yvonne Anne Oswald, Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
{lenzen, oswald, wattenhofer}@tik.ee.ethz.ch

Abstract

Whether local algorithms can compute constant approximations of NP-hard problems is of both practical and theoretical interest. So far, no algorithms achieving this goal are known, as either the approximation ratio or the running time exceed O(1), or the nodes are provided with non-trivial additional information. In this technical report, we present the first[1] distributed algorithm approximating a minimum dominating set on a planar graph within a constant factor in constant time. Moreover, the nodes do not need any additional information.

1 Introduction

Common distributed network protocols require some nodes of the network to have information about the global state of the network. As networks grow larger and become more dynamic, using such protocols becomes increasingly difficult. Indeed, nodes only being aware of their local neighborhood suffices for many problems. Such distributed algorithms are known as "localized" or "local" algorithms.

Whereas many algorithms called "localized" are not wait-free and prone to experience a *butterfly effect* due to chains of causality, the term "local" is often used rigorously: In a k-*local* algorithm nodes are allowed to gather information of their k-hop neighborhood before they make a decision. Such algorithms are very useful when tackling problems in dynamic networks. The topology of a dynamic network may change over time, thus a solution may need to be modified. In the worst case, rerunning a non-local algorithm may lead to a solution already rendered useless before the computation finishes. When facing

[1] The algorithm presented in [14] is wrong and the analysis of the algorithm in [5] is incomplete.

communication or state mistakes thwarting computational progress, correcting errors locally can lead to self-stabilizing networks.

In the past few years, k-local algorithms have attracted remarkable interest, stimulated by innovations in ad hoc and sensor networks. However, as discussed in the related work section, many proposed algorithms have a drawback. They allow nodes to gather information on an extended neighborhood, increasing with a function f of the number of nodes n; we call this model $f(n)$-locality. In this report we focus on O(1)-local algorithms, where each node knows its neighbors within a *constant* radius. Hence we use the original definition of locality coined in the seminal paper by Naor and Stockmeyer [22], omitting O(1) in the notation.

Despite the research momentum $f(n)$-local algorithms have experienced, little is known about strictly local algorithms. An exception is the work by Kuhn et al. [16, 18] proving that many classic graph optimization problems cannot be solved locally on general graphs. As the graph family to construct the lower bound is exotic, one might hope that many practically interesting graph classes still permit local algorithms. However, Linial [20] proved that even in a ring topology some problems are not solvable locally, hence one cannot hope to e.g. find a local algorithm for maximal independent sets in unit disk graphs or a coloring in a planar graph.

Positive results on local algorithms are rare. Naor and Stockmeyer [22] present non-trivial problems with a local solution, e.g. the weak 2-coloring problem, a coloring of all nodes with two colors such that each non-isolated node has at least one neighbor colored differently. However, what all their problems have in common is the fact that a simple broadcast algorithm can solve them.

So is there any hope that a difficult problem can be computed locally? Or, more specifically, are there *NP-hard problems* that permit a constant *approximation* by an algorithm depending on knowledge of the local neighborhood only? Rather surprisingly, this work answers this question affirmatively. We present a constant-time constant-approximation local algorithm for the minimum dominating set problem on planar graphs (shown to be NP-complete in [11]). To the best of our knowledge, this is presently the hardest problem solved by a local algorithm. We hope our result will help in comprehending the limitations and capabilities of local algorithms, and eventually capture the complexity of distributed algorithms.

2 Related Work

Local algorithms have been studied for more than three decades [1, 3, 12, 20, 21, 22, 23]. Recently, research on local algorithms has been thriving again, probably thanks to emerging applications in ad hoc and sensor networks (see [25] for a recent survey). In particular, the minimum dominating set (MDS) problem and related problems have caught the attention of the community, as MDS, connected MDS, or maximal independent sets (MIS) promise to provide an elegant solution to many theoretical problems in wireless multi-hop networks.

Judging by the abundance of literature on the MDS problem, it seems to be key to understanding local algorithms.

A first stab at the MDS problem was an ingenious MIS algorithm [1, 12, 21]. However, in a general graph a MIS is not necessarily a good approximation for the MDS problem. Afterwards there have been numerous proposals, unfortunately, similarly to [1, 12, 21] always either the running time or the approximation ratio were trivial. The first distributed MDS algorithm non-trivial in both locality and approximation is by Jia et al. [13]. They present a $O(\log n \log \Delta)$-local algorithm that approximates the MDS problem within a factor $O(\log \Delta)$ of the optimal in expectation, where n is the number of nodes and Δ is the largest node degree. Later, Kuhn et al. proposed the first $O(1)$-local algorithm with a non-trivial approximation ratio [19]. This result has been improved [18] to the currently best result for general graphs: The MDS problem can be approximated up to a factor of $O(\Delta^{1/\sqrt{k}} \log \Delta)$ in $O(k)$ time.

Kuhn et al. [16] showed that in general graphs local algorithms are limited, as even a polylogarithmic approximation of the MDS problem requires at least $\Omega(\sqrt{\log n / \log \log n})$ time. As the graph family that is used in the lower bound argument needs an elaborate construction unlikely to ever appear in practice, people started studying special graph classes. Of particular interest are geometric graphs, such as unit disk graphs (UDGs), since they represent wireless multi-hop networks well. In UDGs, if distance information is available, one can compute a constant approximation of the MDS problem in $O(\log^* n)$ time [17], while without distance information the best deterministic algorithm needs $O(\log \Delta \log^* n)$ time [15], the best (deterministic) algorithm runs in $O(\log^* n)$ time [24]. Interestingly all these UDG algorithms make a detour and compute a MIS, which in UDGs provides a constant approximation of the MDS problem. With respect to the approximation quality Czygrinow et al. presented the best currently known algorithm on planar graphs [4]. It yields an asymptotically optimal approximation ratio of $(1 + O(\log^{-1} n))$, but the number of rounds is in $O(\log \log n \log^* n \log^{28.7} n)$. The authors have improved on this bound in [5] with an algorithm computing a $(1+\epsilon)$-approximation in $O(\log^* n)$ time[2] among other results. In particular, they show that for any $\epsilon > 0$ there is no deterministic algorithm computing a $(5 - \epsilon)$-approximation DS in $o(\log^* n)$ rounds in planar graphs.

Thus all algorithms mentioned so far are either not local in our strict sense, as their running time is a function of the size of the network, or they do not reach an $O(1)$-approximation ratio. For several special graph classes, e.g. constant-degree graphs or trees, the MDS problem is trivial, as there are simple constant-approximation local algorithms. In fact, few local algorithms for nontrivial problems are known. Naor and Stockmeyer [22] showed that such problems exist, e.g. weak 2-coloring or a modification of the dining philosophers problem. However, these problems can be solved by simple broadcast algorithms on a global basis. More sophisticated strategies are necessary to reach a constant

[2] An algorithm computing a constant MDS approximation in $O(\log^* n)$ like the one presented in this report is a prerequisite for their algorithm.

approximation of a MDS on planar graphs.

Another class of algorithms assumes the nodes to have additional information. Algorithms for sensor networks, for instance, often allow nodes to know their position in space. Even with location information the MDS problem in unit disk graphs remains NP-complete [2] yet a folklore single round algorithm will give a constant approximation; for a PTAS in constant time see [26]. Instead of knowledge on their location, nodes could have other helpful extra information at their disposal, e.g., the maximum degrees of the network or the total number of nodes. The power of additional information was studied from a more general perspective in a series of papers [9, 8, 10]: In these papers, Fraigniaud et al. examine how many bits are necessary to allow efficient algorithms for problems such as coloring, MST, wake up and broadcast. Not surprisingly, they observe that problems become easier the more information is available. Pushing the envelope, Floréen et al. recently presented local algorithms which construct constant approximations for activity and sleep scheduling problems [6, 7], allowing each node one additional bit of information. This bit is used to break the symmetry of the original problem, essentially partitioning it into (easier) subproblems. As a matter of fact, from a radical viewpoint, additional information may push our original question into absurdity: Even a small number of bits of additional information per node is enough to compute a constant-time constant-approximation of any NP-hard problem—simply let the additional information encode the (approximate) solution!

3 Model and Notation

A distributed system is modeled as a simple undirected graph where each node represents a processor and edges correspond to bidirectional communication channels between them. Nodes are able to distinguish between their communication channels, i.e., they can designate the intended receiver of a message and they are able to identify the sender of a message. We use the classic synchronous message passing model, where in each communication round every node of the network graph can send a message to each of its direct neighbors. A local algorithm may only use a constant number of communication rounds before each node reaches a decision based on the acquired information. An algorithm is correct if the combined solution of all nodes is a valid solution to the given problem, regardless of the distribution of the identifiers.

Given a graph $G = (V, E)$, a node $w \in V$ is a neighbor of some set $A \subset V$ if $\{a, w\} \in E$ for an $a \in A$. For a set of nodes $A \subseteq V$ we define $\mathcal{N}_A^+$ to be the inclusive neighborhood of A, i.e., A and all its neighbors. By $\mathcal{N}_A := \mathcal{N}_A^+ \setminus A$ we denote the neighbors of A not in A. For subgraphs and minors H of G we define neighbors correspondingly and write $\mathcal{N}_A^+(H)$ and $\mathcal{N}_A(H)$. In cases where A consists of a single node a, we may omit the braces in the notation, e.g. $\mathcal{N}_a$ instead of $\mathcal{N}_{\{a\}}$. The two-hop neighborhood of v is denoted by $\mathcal{N}_v^{(2)}$. For two sets of nodes A and B of graph H the expression "A covers B in H" means $B \subseteq \mathcal{N}_A^+(H)$, where we may omit "in H" when clear from the context. A dominating set (DS) of G is a set $D \subseteq V$ covering V. A minimum dominating set (MDS) is a DS of minimum cardinality.

4 Algorithm and Analysis

In this section we present an algorithm computing a constant approximation of a minimum dominating set on planar graphs in constant time. Assuming maximum degree Δ and identifiers of size $O(\log n)$, the algorithm makes use of messages of size $O(\Delta \log n)$. As planar graphs exhibit unbounded degree, the algorithm is thus not suitable for practice. Moreover, the constant in the approximation ratio is 130, i.e., there is a large gap to the lower bound of $5 - \varepsilon$ (for any constant $\varepsilon > 0$) [5]. Nevertheless, we demonstrate that in planar graphs in principle it is feasible to obtain a constant MDS approximation in a constant number of distributed rounds.

4.1 Algorithm

The key idea of the algorithm is to exploit planarity in two ways. On the one hand, planar graphs have arboricity three, i.e., the number of edges of any subgraph is linear in its number of nodes. What is more, as planarity is preserved under taking minors, so does any minor of the graph. On the other hand, in a planar graph circles are barriers separating parts of the graph from others; any node enclosed in a circle cannot cover nodes on the outside. This is a very strong structural property enforcing that dominating sets are either large or exhibit a simple structure. It will become clear in the analysis how these properties are utilized by the algorithm.

The algorithm consists of two main steps. In the first step all nodes check whether their neighborhood can be covered by six or less other nodes. Note that after learning about their two-hop neighborhood in two rounds, nodes can decide this locally by means of a polynomial-time algorithm.[3] If this is not the case, they join the (future) dominating set. In the second step, any node that is not yet covered elects a neighbor of maximal residual degree (i.e., one that covers the most uncovered nodes) into the set. Algorithm 1 summarizes this scheme.

4.2 Analysis

As evident from the description, the algorithm can be executed in six rounds and computes a dominating set due to the second step. Therefore, we need to bound the number of nodes selected in each step in terms of the size of a minimum dominating set M of the planar graph G. For the purpose of our analysis, we fix some MDS M of G and assume that $n \geq 3$. By D_1 and D_2

[3] Trivially, one can try all combinations of six nodes. Note, however, that planarity permits more efficient solutions.

Algorithm 1: MDS Approximation in Planar Graphs

output: DS D of G
$D := \emptyset$
for $v \in V$ *in parallel* **do**
  if $\nexists A \subseteq \mathcal{N}_v^{(2)} \setminus \{v\}$ *such that* $\mathcal{N}_v \subseteq \mathcal{N}_A^+$ *and* $|A| \leq 6$ **then**
    $D := D \cup \{v\}$
  end
end
for $v \in V$ *in parallel* **do**
  $\bar{\delta}_v := |\mathcal{N}_v^+ \setminus \mathcal{N}_D^+|$ // residual degree
  if $v \in V \setminus \mathcal{N}_D^+$ **then**
    $\Delta_v := \max_{w \in \mathcal{N}_v^+} \{\bar{\delta}_w\}$ // maximum within one hop
    choose any $d(v) \in \{w \in \mathcal{N}_v^+ \mid \bar{\delta}_w = \Delta_v\}$
    $D := D \cup \{d(v)\}$
  end
end

we denote the nodes that enter D in the first and second step of the algorithm, respectively.

We often use the following lemma to bound the number of edges of a subgraph or minor.

Lemma 4.1. *A minor of a planar graph is planar. A planar graph of $n \geq 3$ nodes has at most $3n - 6$ edges.*

We begin by bounding the number of nodes in $D_1 \setminus M$ after the first step.

Lemma 4.2. $|D_1 \setminus M| < 3|M|$.

Proof. We construct the following subgraph $H = (V_H, E_H)$ of G (see Figure 1).

- Set $V_H := \mathcal{N}_{D_1 \setminus M}^+ \cup M$ and $E_H := \emptyset$.
- Add all edges with one endpoint in $D_1 \setminus M$ to E_H.
- Add a minimal subset of edges from E to E_H such that $V_H = \mathcal{N}_M^+(H)$, i.e., M is a DS in H.

Thus, each node $v \in V_H \setminus (D_1 \cup M)$ has exactly one neighbor $m \in M$, as we added a minimal number of edges for M to cover V_H. For all such nodes v, we contract the edge $\{v, m\}$, where we identify the resulting node with m. In other words, the star subgraph of H induced by $\mathcal{N}_m^+(H) \setminus D_1$ is collapsed into m. By Lemma 4.1, the resulting minor $\bar{H} = (V_{\bar{H}}, E_{\bar{H}})$ of G satisfies that $|E_{\bar{H}}| < 3|V_{\bar{H}}|$. Due to the same lemma, the subgraph of $\bar{H}$ induced by $D_1 \setminus M$ has fewer than $3|D_1 \setminus M|$ edges. As the neighborhood of a node from $D_1 \setminus M \subset V_{\bar{H}}$ cannot be covered by fewer than seven nodes, the performed edge contractions did not reduce the degree of such a node below seven.

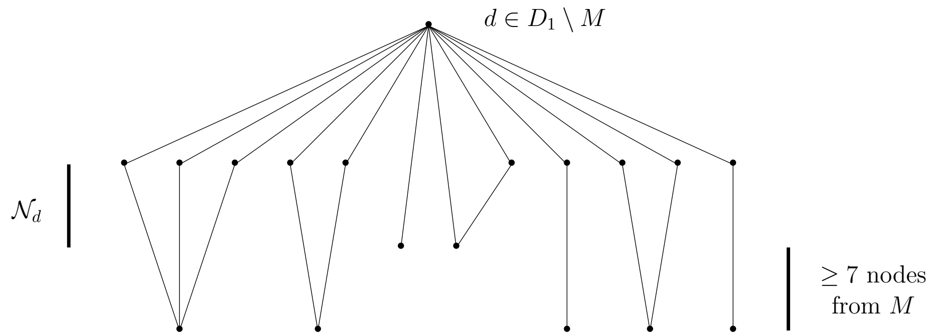

Figure 1: Part of the subgraph constructed in Lemma 4.2.

Consequently, we have

$$\begin{aligned}
& 7|D_1 \setminus M| - 3|D_1 \setminus M| \\
< & \sum_{d \in D_1 \setminus M} \delta_d(\bar{H}) - |\{\{d, d'\} \in E_{\bar{H}} \mid d, d' \in D_1 \setminus M\}| \\
\leq & |E_{\bar{H}}| < 3|V_{\bar{H}}| \\
\leq & 3(|D_1 \setminus M| + |M|),
\end{aligned}$$

which can be rearranged to yield the claimed bound. □

To bound the number of nodes $|D_2|$ that is chosen in the second step of the algorithm, more effort is required. We consider the following subgraph of G.

Definition 4.3. *We define $H = (V_H, E_H)$ to be the subgraph of G obtained by the following construction.*

- *Set $V_H := \emptyset$ and $E_H := \emptyset$.*
- *For each node $d \in D_2$ for which this is possible, add one node $v \in V \setminus M$ to V_H such that $d = d(v)$ in Line 1 of the algorithm.*
- *Add $M \setminus D_1$ to V_H and a minimal number of edges to E_H such that $\mathcal{N}^+_{M \setminus D_1}(H) = V_H$, i.e., $M \setminus D_1$ covers the nodes so far added to H (this is possible as only nodes from $V \setminus \mathcal{N}^+_{D_1}$ elect nodes into D_2).*
- *For each $m \in M \setminus D_1$, add a minimal number of nodes and edges to H such that there is a set $C_m \subseteq V_H \setminus \{m\}$ of minimal size satisfying $\mathcal{N}_m(H) \subseteq \mathcal{N}^+_{C_m}(H)$, i.e., C_m covers m's neighbors in H. We define that $C := \cup_{m \in M \setminus D_1} C_m$.*
- *Remove all nodes $v \in V_H \setminus (C \cup M)$ for which $d(v) \in M \cup C$.*
- *For each $m \in M \setminus D_1$, remove all edges to C_m.*

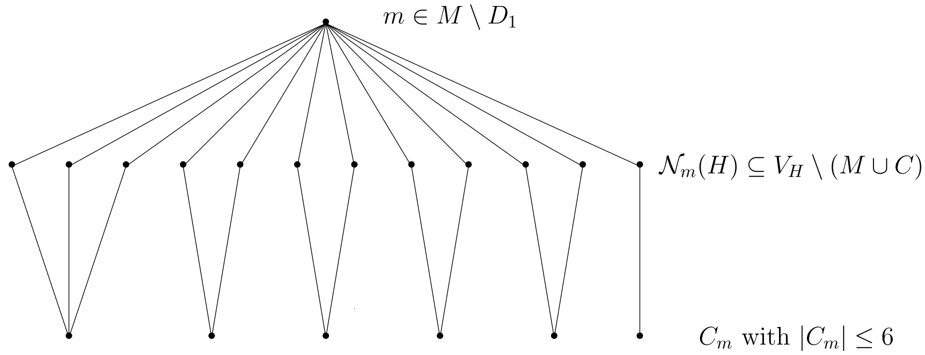

Figure 2: Part of the subgraph H from Definition 4.3.

See Figure 2 for an illustration.

In order to derive our bound on $|D_2|$, we consider a special case first.

Lemma 4.4. *Assume that for each node $m \in M \setminus D_1$*

(i) no node $m' \in M \cap C_m$ covers more than seven nodes in $\mathcal{N}_m(H)$ and

(ii) no node $v \in C_m \setminus M$ covers more than four nodes in $\mathcal{N}_m(H)$.

Then it holds that $|D_2| < 98|M|$.

Proof. Denote by $A_1 \subseteq V_H \setminus (M \cup C)$ the nodes in V_H that elect others into D_2 and have two neighbors in M, i.e., when we added C to V_H, they became covered by a node in $M \cap C$. Analogously, denote by $A_2 \subseteq V_H \setminus (M \cup C)$ the set of electing nodes for which the neighbor in C is not in M. Observe that $A := A_1 \cup A_2 = V_H \setminus (M \cup C)$ and $A_1 \cap A_2 = \emptyset$. Moreover, we claim that $|A| \geq |D_2| - 14|M|$. To see this, recall that in the first step of the construction of H, we choose for each element of $|D_2|$ that is not elected by elements of M only one voting node v, i.e., at least $|D_2| - |M|$ nodes in total. In the second last step of the construction, we remove v if $d(v) \in \{m\} \cup C_m$ for some $m \in M \setminus D_1$. As $m \in M \setminus D_1$, its neighborhood can be covered by six or less nodes from $V \setminus \{m\}$. Therefore $|C_m| \leq 6$ for all $M \setminus D_1$ and we remove at most $7|M|$ nodes in total in the second last step. Finally, in the last step we cut off at most $|C| \leq 6|M|$ voting nodes from their dominators in $M \setminus D_1$. The definition of A explicitly excludes these nodes, hence $|A| \geq |D_2| - 14|M|$.

We contract all edges from nodes $a \in A$ to the respective nodes $m \in M \setminus D_1$ covering them we added in the third step of the construction of H. Denote the resulting minor of G by $\bar{H} = (V_{\bar{H}}, E_{\bar{H}})$. For every seven nodes in A_1, there must be a pair of nodes $m, m' \in M \setminus D_1$ such that $m \in C_{m'}$ and vice versa, as by assumption no such pair shares more than seven neighbors. Thus, for every seven nodes in A_1, we have two nodes less in $V_{\bar{H}}$ than the upper bound of $|V_{\bar{H}}| \leq |M| + |C| \leq 7|M|$. By Lemma 4.1, $\bar{H}$ thus has fewer than

$$3|V_{\bar{H}}| \leq 3|M \cup C| \leq 3|M| + 3\left(6|M| - \frac{2|A_1|}{7}\right) = 21|M| - \frac{6|A_1|}{7}$$

edges.

On the other hand,

$$|E_{\bar{H}}| \geq \frac{|A_1|}{7} + \frac{|A_2|}{4},$$

as by assumption each pair of nodes from M may share at most seven neighbors in A_1 and pairs of nodes $m \in M \setminus D_1$, $v \in C_m \setminus M$ share at most four neighbors. We conclude that

$$|A_2| < 84|M| - 4|A_1|$$

and therefore

$$|D_2| \leq |A_1| + |A_2| + 14|M| < 98|M| - 3|A_1| \leq 98|M|. \qquad \square$$

In order to complete our analysis, we need to cope with the case that a node $m \in M \setminus D_1$ and an element of C_m share many neighbors. In a planar graph, this results in a considerable number of nested circles which separate their interior from their outside. This necessitates that nodes from the optimal solution M are enclosed that we may use to compensate for the increased number of nodes in A in comparison to the special case from Lemma 4.4.

Lemma 4.5. *Suppose the subgraph H from Definition 4.3 violates condition (i) or (ii) from Lemma 4.4. Fix a planar embedding of G and consider either*

(i) nodes $m \in M \setminus D_1$ and $v \in M \cap C_m$ with $|\mathcal{N}_m(H) \cap \mathcal{N}_v(H)| \geq 8$ or

(ii) nodes $m \in M \setminus D_1$ and $v \in C_m \setminus M$ with $|\mathcal{N}_m(H) \cap \mathcal{N}_v(H)| \geq 5$.

Then the outmost circle formed by m, v, and two of their common neighbors in H must enclose some node $m' \in M$ (with respect to the embedding).

Proof. Set $\tilde{A} := \mathcal{N}_m(H) \cap \mathcal{N}_v(H)$. Consider case (i) first and assume for contradiction that there is no node from M enclosed in the outmost circle. W.l.o.g., we may assume that $|\tilde{A}| = 8$ (otherwise we simply ignore some nodes from $\tilde{A}$). There are four nodes from $\tilde{A}$ that are enclosed by two nested circles consisting of v, m, and the four nodes that are the outer nodes from $\tilde{A}$ according the embedding (see Figure 3). Recall that by the second last step of the construction of H nodes $a \in \tilde{A}$ satisfy that $d(a) \notin \{m, v\} \subseteq M$. Therefore, these enclosed nodes elected (distinct) nodes into D_2 that are enclosed by the outmost circle. As the electing nodes $a \in \tilde{A}$ are connected to m and v, by Line 1 of the Algorithm the nodes $d(a)$ they elected must have at least residual degree $\bar{\delta}_{d(a)} \geq \max\{\bar{\delta}_v, \bar{\delta}_m\}$. In other words, they cover at least as many nodes from $V \setminus \mathcal{N}^+_{D_1}$ as both m and v.

Consider the subgraph S of G induced by $\tilde{A}$, v, m and L, the set of nodes that are enclosed in the outmost circle and that are neither in $\tilde{A} \subseteq V \setminus \mathcal{N}^+_{D_1}$ nor

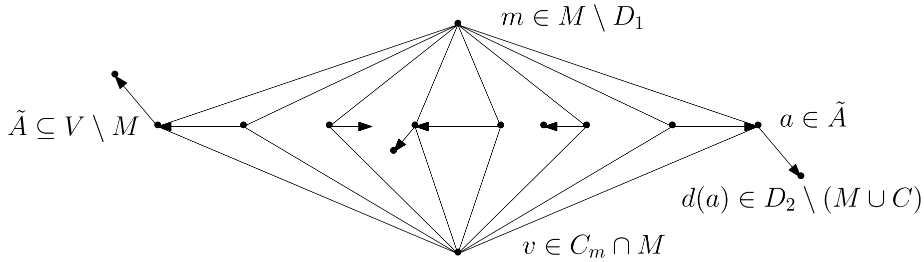

Figure 3: Example of a subgraph considered in the first case of the proof of Lemma 4.5. While the choices $d(a)$ of the two leftmost and righmost nodes $a \in \tilde{A}$ may have large degrees because of nodes outside the outer circle, the choices of the four inner nodes must have many neighbors that are not covered by D_1 on or inside the outer circle.

already covered by D_1. Let ℓ denote the cardinality of set L. Thus $S = (V_S, E_S)$ contains $|V_S| = l + |\tilde{A}| + |\{v, m\}| = \ell + 10$ nodes. Regarding the number of edges we claim that the cardinality of E_S is at least

$$\begin{aligned} |E_S| &\geq |\mathcal{N}_v(S)| + |\mathcal{N}_m(S)| + 4\max\{|\mathcal{N}_v(S)|, |\mathcal{N}_m(S)|\} - 18 \\ &> 3(|\mathcal{N}_v(S)| + |\mathcal{N}_m(S)| - 6). \end{aligned}$$

To see that this claim holds note that the subgraph S contains at least the edges to all neighbors of v and m in S and the edges incident to the four nodes from $\tilde{A}$ that are enclosed by two nested circles consisting of v, m and the four outer nodes from $\tilde{A}$ according the embedding. Remember, that the residual degree of these four nodes from $\tilde{A}$ in the second step of the algorithm is at least as large as the residual degrees of v and m, as they would not have been chosen on Line 11 otherwise. By adding $4\max\{|\mathcal{N}_v(S)|, |\mathcal{N}_m(S)|\}$ we might count some edges twice, therefore we subtract 18 edges to account for the facts that (i) there might be up to $\binom{4}{2} = 6$ edges between pairs of the four considered nodes $d(a) \in D_2$, (ii) up to 8 edges between these four nodes and $\{v, m\}$ might exist, and (iii) a chosen node might cover itself in which case no edge is necessary.

The second construction step of Definition 4.3 ensures that $\tilde{A} \cap M = \emptyset$ by only adding nodes from $V \setminus M$ to V_H. Hence, the assumption that no other node from M is enclosed by the outmost circle implies that everything inside is covered by $\{v, m\}$. Therefore, it holds that

$$|\mathcal{N}_v(S)| + |\mathcal{N}_m(S)| \geq 2|\tilde{A}| + \ell = \ell + 16.$$

However, Lemma 4.1 lower bounds $|V_S|$ in terms of $|E_S|$, giving that

$$3(\ell + 10) = 3|V_S| > |E_S| > 3(|\mathcal{N}_v(S)| + |\mathcal{N}_m(S)| - 4) \geq 3(\ell + 10),$$

a contradiction.

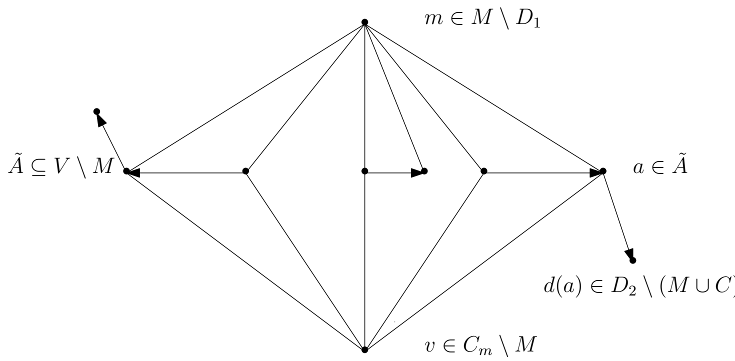

Figure 4: Example of a subgraph considered in the second case of the proof of Lemma 4.5. Supposing there is no other node $m' \in M$ inside the outer circle, apart from v all neighbors of the node chosen by the innermost node from $\tilde{A}$ must also be neighbors of m.

Case (ii) is treated similarly, but it is much simpler. This time, the assumption that no node from M is enclosed by the outmost circle implies that all the nodes inside must be covered by m alone, as M is a DS. Since v and m are connected via the (at least) five nodes in $\tilde{A}$, for the node $d(a) \notin \{m, v\}$ elected into D_2 by the innermost node $a \in \tilde{A}$, it holds that $\mathcal{N}^+_{d(a)} \setminus \mathcal{N}^+_m \subseteq \{v\}$ (see Figure 4). However, there are at least two nodes in $\tilde{A} \subseteq V \setminus \mathcal{N}^+_{D_1}$ that are not connected to $d(a)$, i.e., we get the contradiction that a would have preferred m over $d(a)$ in Line 1 of the algorithm. $\square$

Next, we repeatedly delete nodes from H until eventually the preconditions of Lemma 4.4 are met. Arguing as in the proof of Lemma 4.5, we can account for deleted nodes by allocating them to enclosed nodes from $M \cup C$. Doing this carefully, we can make sure that no nodes from $M \cup C$ need to compensate for more than four deleted nodes.

Corollary 4.6. $|D_2| < 126|M|$.

Proof. Fix an embedding of G and thus of all its subgraphs. We will argue with respect to this embedding only. We use the notation from the proof of Lemma 4.4. Starting from H, we iteratively delete nodes from A until we obtain a subgraph H' satisfying the prerequisites of the lemma. Assume that $H' := H$ violates one of the preconditions of Lemma 4.4. No matter which of the conditions (i) and (ii) from Lemma 4.4 is violated, we choose respective nodes $m \in M \setminus D_1$ and $v \in C_m$ satisfying precondition (i) or (ii) of Lemma 4.5 such that the smallest circle formed by m, v, and $a_1, a_2 \in \tilde{A} := \mathcal{N}^+_v(H') \cap \mathcal{N}_m(H')$ enclosing an element $m' \in M$ has minimal area. We delete the two elements from $\tilde{A} \subseteq A$ participating in the circle. Since the area of the circle is minimal, there is no third element from $\tilde{A}$ enclosed in the circle.

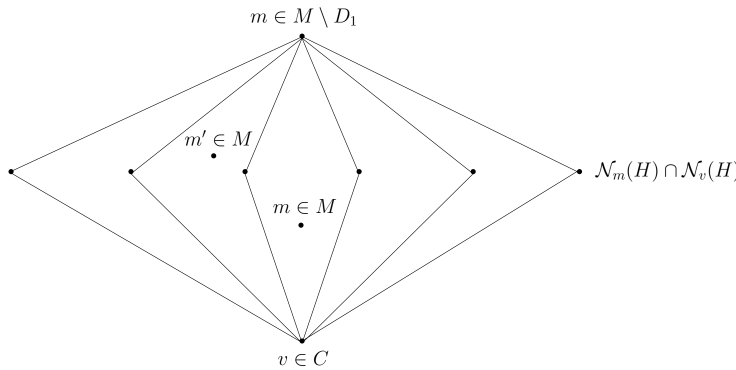

Figure 5: Example of a sequence of three nested circles as considered in Corollary 4.6. Each pair of two voting nodes involved in a circle is deleted from H' after it has been accounted for. Therefore, all neighbors of the two outmost nodes from $\mathcal{N}_m(H) \cap \mathcal{N}_v(H)$ are not adjacent to nodes inside the innermost circle.

We repeat this process until H' satisfies the preconditions of Lemma 4.4. We claim that we can account for deleted nodes in terms of nodes from $M \cup C$ in a way such that no element of $M \cup C$ needs to compensate for more than four deleted nodes. Whenever we delete a pair of nodes, we count a node from $M \cup C$ enclosed by the respective circle that has not yet been counted twice.

We need to show that this is indeed always possible. To see this, observe that the minimality of the enclosed area of a chosen circle X together with the planarity of G ensures that any subsequent circle X' either encloses this circle or its enclosed area is disjoint from the one of X. In the latter case, we obviously must find a different node from $M \cup C$ enclosed in X' than the one we used when deleting nodes from X. Hence, we need to examine the case when there are three nested circles X_1, X_2, and X_3 that occur in the construction. If the nodes $m \in M$ and $v \in C_m$ participating in each circle are not always the same, one node from the first such pair becomes enclosed by one of the subsequent circles.

Hence, the remaining difficulty is that we could have three such nested circles formed by nodes $m \in M$, $v \in C_m$, and three pairs of nodes from $\mathcal{N}_m(H) \cap \mathcal{N}_v(H)$ (see Figure 5). Any node chosen by a node $a \notin \{m, v\}$ lying on the outmost circle X_3 is separated from nodes enclosed by X_1 by X_1. Therefore, nodes $m' \in M$ enclosed by X_1 can cover only nodes that are either not adjacent to the nodes from D_2 considered in Lemma 4.5 (when applied to H' after X_1 and X_2 already have been removed) or lie on X_1. Since the nodes on X_1 are m, v, and two of their shared neighbors in H, we can thus argue analogously to the proof of Lemma 4.5 in order to find a node $m'' \in M$ enclosed by X_3, but not enclosed by X_1.

Altogether, we remove at most two times two nodes each from A, for each element of $M \cup C$ i.e., in total no more than $4|M \cup C| \leq 28|M|$ nodes. To the remaining subgraph H', we apply Lemma 4.4, yielding

$$|D_2| < (28 + 98)|M| = 126|M|.$$ □

Having determined the maximum number of nodes that enter the dominating set in each step, it remains to assemble the results and finally state the approximation ratio our algorithm achieves.

Theorem 4.7. $|D| < 130|M|$.

Proof. Combining Lemma 4.2 and Corollary 4.6, we obtain

$$|D| \leq |M| + |D_1 \setminus M| + |D_2| < (1 + 3 + 126)|M| = 130|M|.$$ □

5 Conclusions

We presented a constant-approximation MDS algorithm for planar graphs. It is deterministic and fully local, i.e., each node bases its decisions on information on a neighborhood of constant size, and no knowledge on any global properties is necessary. To our best knowledge the algorithm is the first of this kind for an *NP-hard* problem, showing that such tasks can be solved by strictly local algorithms.

As *approximating* an MDS on planar graphs is not *NP-hard*, one might ask what exactly makes this problem "harder" than e.g. the weak 2-coloring problem. In contrast to an MDS approximation problem, the weak 2-coloring problem can be solved by a simple global algorithm. After an arbitrary node in each component chooses a color, each of its neighbors may take the other color. Iterating this process leads to a valid weak 2-coloring of any graph. Having a closer look, locally computing a weak 2-coloring is basically a question of breaking symmetric decisions of nodes based on node identifiers. This can be seen by looking e.g. at a completely symmetric ring topology. On the contrary, our algorithm operates only on the structure of a constant neighborhood of the nodes. Nevertheless, the situation is more intricate for the MDS approximation problem, as illustrated in Figures 6 and 7. Our algorithm copes with these challenges by exploiting the sparsity as well as the decomposing properties of circles that planar graphs exhibit.

Since computing an MDS is a fundamental problem, this result sheds new light into the tantalizing question of the possibilities and limitations of different models in distributed computing. It remains a challenging task to find out which other graph classes permit local O(1)-approximation algorithms for the MDS problem without additional information available to the nodes. This may finally lead to a hierarchy of graph classes and approximation ratios achievable by strictly local algorithms.

6 Acknowledgements

We would like to thank Jukka Suomela for valuable comments.

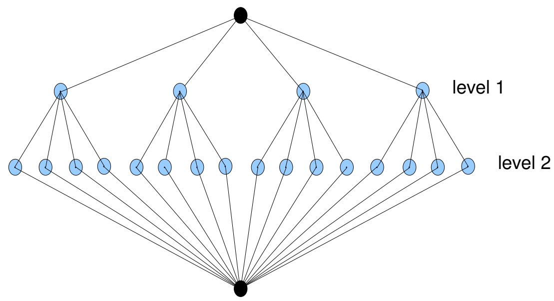

Figure 6: This graph illustrates, why a simple broadcast algorithm cannot compute a constant MDS approximation. We suppose nodes are traversed from top to bottom and from left to right. Each node knows its degree and the decisions of already visited neighbors.

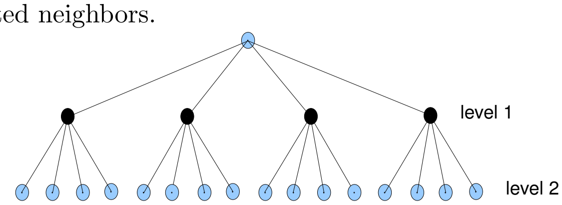

Figure 7: The situation looks identical for all level 1 nodes in the graphs displayed here and in Figure 6. Thus they must take identical decisions. Entering the DS is wrong in the graph in Figure 6. Not entering the DS leads, again due to indistinguishability, to all level 2 nodes entering the DS in the graph displayed. By scaling up node degrees we see that such an algorithm can achieve an approximation ratio of $\Omega(\sqrt{n})$ at best.

References

[1] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, 1986.

[2] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.

[3] R. Cole and U. Vishkin. Deterministic Coin Tossing with Applications to Optimal Parallel List Ranking. *Information and Control*, 70(1):32–53, 1986.

[4] A. Czygrinow, M. Hanckowiak, and E. Szymanska. Distributed Approximation Algorithms for Planar Graphs. In *Proceedings of the 6th Italian Conference on Algorithms and Complexity (CIAC)*, 2006.

[5] A. Czygrinow, M. Hanckowiak, and W. Wawrzyniak. Fast Distributed Approximations in Planar Graphs. In *Proceedings of the 22nd International Symposium on Distributed Computing (DISC)*, September, 2008.

[6] P. Floréen, P. Kaski, T. Musto, and J. Suomela. Local Approximation Algorithms for Scheduling Problems in Sensor Networks. In *Proceedings of the 3rd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors)*, 2007.

[7] P. Floréen, P. Kaski, and J. Suomela. A distributed approximation scheme for sleep scheduling in sensor networks. In *Proc. 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON, San Diego, CA, USA, June 2007)*, pages 152–161, Piscataway, NJ, USA, 2007. IEEE.

[8] P. Fraigniaud, C. Gavoille, D. Ilcinkas, and A. Pelc. Distributed computing with advice: Information sensitivity of graph coloring. In *34th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 231–242, 2007.

[9] P. Fraigniaud, D. Ilcinkas, and A. Pelc. Oracle size: a new measure of difficulty for communication tasks. In *Proceedings of the 25th annual ACM symposium on Principles of distributed computing (PODC)*, pages 179–187, 2006.

[10] P. Fraigniaud, A. Korman, and E. Lebhar. Local mst computation with short advice. In *Proceedings of the 19th annual ACM symposium on Parallel algorithms and architectures (SPAA)*, pages 154–160, 2007.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.

[12] A. Israel and A. Itai. A fast and simple randomized parallel algorithm for maximal matching. *Information Processing Letters*, 22(2):77–80, 1986.

[13] L. Jia, R. Rajaraman, and T. Suel. An Efficient Distributed Algorithm for Constructing Small Dominating Sets. *Distributed Computing*, 15(4):193–205, 2002.

[14] C. Lenzen, Y.A. Pignolet, and R. Wattenhofer. What can be Approximated Locally? Case Study: Dominating Sets in Planar Graphs. In *Proc. 20th ACM Symposium on Parallelism in Algorithms and Architecture (SPAA)*, 2008.

[15] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs. In *Proc. 19th International Symposium on Distributed Computing (DISC)*, 2005.

[16] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What Cannot Be Computed Locally! In *Proc. 23rd annual ACM symposium on Principles of distributed computing (PODC)*, 2004.

[17] F. Kuhn, T. Moscibroda, and R. Wattenhofer. On the Locality of Bounded Growth. In *Proc. 24th ACM Symposium on the Principles of Distributed Computing (PODC)*, 2005.

[18] F. Kuhn, T. Moscibroda, and R. Wattenhofer. The Price of Being Near-Sighted. In *Proc. 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

[19] F. Kuhn and R. Wattenhofer. Constant-Time Distributed Dominating Set Approximation. In *Proc. 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[20] N. Linial. Locality in Distributed Graph Algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[21] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 15(4):1036–1055, 1986.

[22] M. Naor and L. Stockmeyer. What Can Be Computed Locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.

[23] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[24] J. Schneider and R. Wattenhofer. A Log-Star Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs. In *27th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, August 2008.

[25] J. Suomela. Survey of local algorithms. Manuscript, October 2010.

[26] A. Wiese and E. Kranakis. Local PTAS for Independent Set and Vertex Cover in Location Aware Unit Disk Graphs. In *Proceedings of the 4th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2008.