What Can Be Approximated Locally?

Case Study: Dominating Sets in Planar Graphs

Christoph Lenzen
Computer Engineering and Networks Laboratory
ETH Zurich
Switzerland
lenzen@tik.ee.ethz.ch

Yvonne Anne Oswald
Computer Engineering and Networks Laboratory
ETH Zurich
Switzerland
oswald@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich
Switzerland
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 paper, we present the first 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.

Categories and Subject Descriptors

F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures; G.2.2 [**Graph Theory**]: Graph algorithms; C.2.4 [**Distributed Systems**]

General Terms

Algorithms, Theory

Keywords

approximation, distributed algorithms, local algorithms, dominating sets, planar graphs

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 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 paper 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 [21], 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. [15, 17] 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 [19] 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 [21] 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 paper 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 [10]). 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.

SPAA'08, June 14–16, 2008, Munich, Germany.

2. RELATED WORK

Local algorithms have been studied for more than three decades [1, 3, 12, 19, 20, 21, 22]. Recently, research on local algorithms has been thriving again, probably thanks to emerging applications in ad hoc and sensor networks. 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, 20]. However, in a general graph a MIS is not necessarily a good approximation for the MDS problem. Afterwards there have been numerous proposals, however, similarly to [1, 12, 20] 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 $\mathrm{O}(\log n \log \Delta)$-local algorithm that approximates the MWhether 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 $\mathrm{O}(1)$, or the nodes are provided with non-trivial additional information. In this paper, we present the first 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.DS problem within a factor $\mathrm{O}(\log \Delta)$ 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 $\mathrm{O}(1)$-local algorithm with a non-trivial approximation ratio [18]. This result has been improved [17] to the currently best result for general graphs: The MDS problem can be approximated up to a factor of $\mathrm{O}(\Delta^{1/\sqrt{k}} \log \Delta)$ in $\mathrm{O}(k)$ time.

Kuhn et al. [15] 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 $\mathrm{O}(\log^* n)$ time [16], while without distance information the best deterministic algorithm needs $\mathrm{O}(\log \Delta \log^* n)$ time [14], the best randomized algorithm runs in $\mathrm{O}(\log \log n \log^* n)$ time [11][1]. 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 $(1 + \mathrm{O}(\log^{-1} n))$, but the number of rounds is in $\mathrm{O}(\log \log n \log^* n \log^{28.7} n)$.

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 $\mathrm{O}(1)$-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 [21] 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 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 [24]. 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 [8, 7, 9]: 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 suprisingly, 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 [5, 6], 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) sub-problems. 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. In principle, those messages can be of arbitrary size; however, our algorithm will use messages of $\mathrm{O}(\log n)$ bits. 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 $N^+(v)$ to be the inclusive neighborhood of A, i.e., A and all its neighbors. By $N(A) := N^+(A) \setminus A$ we denote the neighbors of A not in A. For subgraphs or minors H of G we define neighbors correspondingly and write $N_H^+(A)$ and $N_H(A)$. In cases where A consists of a single node a, we may omit the braces in the notation, e.g. $N(a)$ instead of $N(\{a\})$. For two sets of nodes A and B of graph H the expression "A covers B in H" means $B \subseteq N_H^+(A)$, 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 con-

[1]We are aware of work in submission to PODC 2008 presenting a deterministic algorithm with running time $\mathrm{O}(\log^* n)$ [23].

stant time. Basically, the algorithm consists of three phases. In the first phase all uncovered nodes select a neighbor they consider most suited for the dominating set, namely the neighbor that dominates the most uncovered nodes. These chosen nodes then enter the dominating set. In a next step, nodes that have more than a certain number of neighbors in the dominating set enter this set themselves and tell their neighbors to leave the dominating set again. In the second phase these steps are repeated. Subsequently, in the third phase, all nodes that are still uncovered choose a neighbor again, completing the final dominating set. For ease of notation we describe the algorithm in more detail in a centralized fashion in Algorithm 1. Let $M_i,\ i \in \{2,\dots,6\}$, denote the set of nodes chosen in step i by Algorithm 1. By V_4 and V_6 we refer to the sets of nodes electing candidates for joining M in steps 4 and 6 respectively.

Algorithm 1 Local approximation of MDS on planar graphs

Input: $k, l \in \mathbb{N}$ and a graph $G = (V, E)$
Output: a dominating set M for G

1: Set $M := \emptyset$.
2: For each node $v \in V$ add an arbitrary element $c_2(v) \in N^+(v)$ of maximum degree to M.
3: Set $A := \{v \in V \,|\, |M_2 \cap N(v)| \geq k\}$. Set $M = (M \setminus N(A)) \cup A$.
4: Set $V_4 := V \setminus N^+(M)$. For each $v \in V_4$ add a node $c_4(v) \in N^+(v)$ with $|N^+(c_4(v)) \cap V_4|$ maximum to M.
5: Set $A := \{v \in V \,|\, |M_4 \cap N(v)| \geq l\}$. Set $M = (M \setminus (M_4 \cap N(A))) \cup A$.
6: Set $V_6 := V \setminus N^+(M)$. For each $v \in V_6$ add a node $c_6(v) \in N^+(v)$ with $|N^+(c_6(v)) \cap V_6|$ maximum to M.
7: Return M.

No intricate analysis is necessary to ascertain that we can implement Algorithm 1 locally, as each step takes at most two rounds of communication. Step 6 ensures that a dominating set is returned. We immediately see the following.

LEMMA 4.1 (CORRECTNESS AND LOCALITY).
Algorithm 1 is correct and local.

We will now show that Algorithm 1 yields a constant approximation for the MDS problem on planar graphs. We prove our claim by bounding the number of nodes in M chosen in each step relative to the number of nodes in an arbitrary MDS D.

We will need the following basic lemma repeatedly:

LEMMA 4.2 (NUMBER OF EDGES IN A PLANAR GRAPH).
In a simple planar graph with $n \geq 3$ vertices the number of edges is at most $3n - 6$.

REMARK 4.3.
We will not mention the trivial special case $n < 3$ in the following.

We begin by bounding the number of nodes chosen in step 2 of the algorithm that are still in M after the third step.

PROPOSITION 4.4 (BOUND FOR STEP 2).
Set $A := M_2 \setminus (N^+(M_3) \cup D)$. Then the inequality

$$|A| \leq (k-1)|D| \tag{1}$$

holds.

PROOF. As D is a dominating set, it particularly covers A. For (1) to be violated, there must be a node $d \in D$ covering at least k nodes in A, thus at least k nodes of A are in $N(d)$. This yields a contradiction, because then d would have been chosen in step 3 and $N(d)$ removed from M, so $N(d) \cap A$ must be empty. □

We proceed by constructing a subgraph of G capturing the relevant structures to bound $|M_3 \setminus D|$. First, we examine the special case where nodes in D do not cover more than three neighbors of each node in $M_3 \setminus D$ in this subgraph.

PROPOSITION 4.5 (SPECIAL CASE FOR STEP 3).
Consider the following subgraph $H = (V_H, E_H)$ of G:

- *Set $V_H := M_3 \cup D$.*
- *Add $(N(M_3 \setminus D) \cap M_2)$ to V_H and all edges from each node in $M_3 \setminus D$ to this set present in E to E_H.*
- *Remove all edges $\{\{m, c_2(m)\} \,|\, m \in M_3 \setminus D\}$ from E_H.*
- *Insert a minimum subset of edges in E such that D covers V_H.*

For each $m \in M_3 \setminus D$ fix D_m to be the minimum subset of D covering $N_H(m)$. Let k be at least 11. If for each m no element of D_m covers more than three elements of $N_H(m)$, i.e.,

$$\forall m \in M_3,\ d_m \in D_m\ \ |N_H^+(d_m) \cap N_H(m)| \leq 3, \tag{2}$$

the inequality

$$|M_3 \setminus D| < \frac{3}{\lceil \frac{k-10}{3} \rceil}|D|$$

holds.

PROOF. For each $v \in V_H \setminus (M_3 \cup D)$ contract the edge to the unique neighbor in D, identifying the resulting node with the one in D (removing duplicate edges, as we talk of simple graphs). After this operation the following holds for the resulting graph $\hat{H} = (V_{\hat{H}}, E_{\hat{H}})$:

- $\hat{H}$ is a minor of G and hence planar.
- $M_3 \setminus D \subset V_{\hat{H}}$, as these nodes never participate in a contraction.
- Each node $m \in M_3 \setminus D$ shares an edge with each element of D_m, as each element of D_m must cover at least one neighbor of m in $\hat{H}$.

This leads to an estimate for the minimum number of edges in H. The assumption that each element in D_m covers at most three of the neighbors of $m \in M_3 \setminus D$ different from $c_2(m)$ implies m to have at least $\lceil \frac{k-1}{3} \rceil$ edges to D, the -1 stemming from the exclusion of edges $\{m, c_2(m)\}$. Counting the number of edges for each node $m \in M_3 \setminus D$ yields

$$\left\lceil \frac{k-1}{3} \right\rceil |M_3 \setminus D| \leq |E_{\hat{H}}| < 3(|M_3 \setminus D| + |D|),$$

where we use Lemma 4.2. Since $k \geq 11$ we can conclude that

$$|M_3 \setminus D| < \frac{3}{\lceil \frac{k-10}{3} \rceil}|D|.$$

□

In Proposition 4.5 we required that no node in D covers more than three neighbors of a node in $M_3 \setminus D$ in the subgraph H. In the following proposition we abandon this requirement and establish a general bound for the number of nodes chosen in step 3.

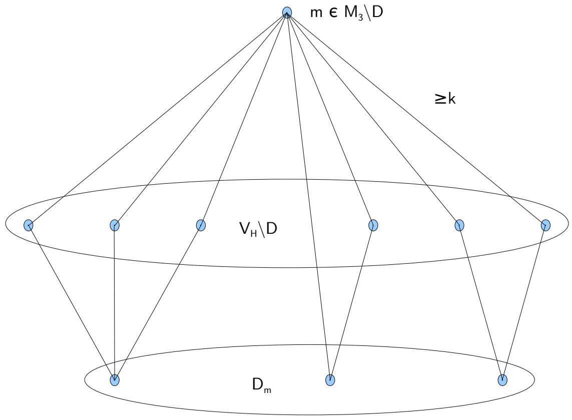

Figure 1: Construction of the subgraph of G used in Proposition 4.5 and 4.6.

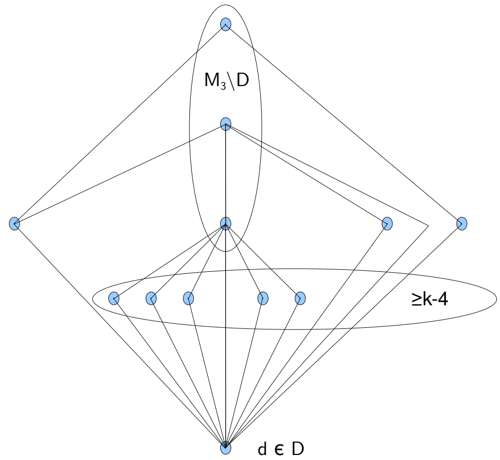

Figure 2: Example of a contradictive structure created by two occurrences of Case 1 in Proposition 4.6 using the same $d \in D$. We can see that there must be at least one node in $M_2 \setminus \{c_2(m)\}$ enclosed in the inner circle, because otherwise the enclosed node is in $M_3 \setminus D$.

PROPOSITION 4.6 (BOUND FOR STEP 3). *Let k be at least 11. Then for the number of nodes chosen in step 3 not in D the estimate*

$$|M_3 \setminus D| < \left(3 + \frac{3}{\lceil \frac{k-10}{3} \rceil}\right) |D|$$

holds.

PROOF. We fix an embedding of G and we construct the same subgraph H as in Proposition 4.5. Set $\hat{H} := H$. We will iteratively remove nodes from $\hat{H}$ until we can apply Proposition 4.5 to the remaining subgraph.

If the prerequisite (2) of Proposition 4.5 is violated, there are nodes $m \in M_3 \setminus D$ and $d \in D_m$ connected by at least four disjoint paths in $\hat{H}$. At least three of them are of length two and contain an element of $V_H \setminus D$. Two of these form a circle C enclosing the third path. Choose C such that it encloses a minimum area. We will replace at most $3|D|$ elements of $M_3 \setminus D$ by placeholder nodes until the requirements of Proposition 4.5 are satisfied. Every time we remove an element of $M_3 \setminus D$ we will count an element of D once. Showing that no element of D is counted more than three times completes the proof.

Case 1: No element of D is enclosed by C.
Remove m from $\hat{H}$ and count d once. Replace m by a node v connected only to $N_{\hat{H}}(m) \cap (M_3 \setminus D)$ and an arbitrary element of $N_H(m) \cap D$, from now on treating v with respect to Proposition 4.5 as an element of $V_H \setminus (M_3 \cup D)$. Remove all elements of $V_H \setminus (N^+_{\hat{H}}(M_3) \cup D)$, i.e., all nodes in $M_2 \setminus D$ and all placeholder nodes no longer connected to elements of M_3.

This case cannot occur more than once for a given d. Supposing the contrary, first observe that the minimum area property of C prohibits that any node $\hat{m} \in M_3 \setminus D$ inside C can cause d to be counted again after C has been used. Second, C encloses an element $v \in M_2 \setminus \{c_2(m)\}$, because otherwise the inner node of the enclosed path must either be in M_3 or replace some node in M_3. But this would imply that there are at least $k - 4$ nodes in $N_H(d)$ enclosed by C belonging to M_3. Repeating this argument leads to an infinite number of nodes in G. Thus there exists at least one node $v \in M_2 \cap V_{\hat{H}}$ distinct from $c_2(m)$ enclosed by C. This node must have been chosen by a node in step 2. All nodes inside C are connected to d, hence we must have $|N_{\hat{H}}(v)| \geq |N_{\hat{H}}(d)|$.

Under the assumption that a certain node d is counted again by an occurrence of case 1 for a node $\hat{m} \in M_3 \setminus D$, there must be another circle $\hat{C}$ for the same node d. As d must have four independent paths of length at most two in H to the node $\hat{m} \in M_3 \setminus D$ used in the construction the second time this case occurs, d must have a neighbor outside C v cannot have. Either $\hat{C}$ encloses m or d is enclosed by $(C \cup \hat{C}) \setminus \{d\}$, because otherwise d has too many neighbors outside C, violating $|N_{\hat{H}}(v)| \geq |N_{\hat{H}}(d)|$. The first implies m to be covered by d and the latter needs d to be connected to an additional node outside C to form the needed paths. Thus both possibilities lead to the contradiction $|N_{\hat{H}}(v)| < |N_{\hat{H}}(d)|$.

Case 2: C encloses at least one node of D.
We count the element $\hat{d}$ enclosed last by the sequence of circles defined by the sequence of replacements of elements of $M_3 \setminus D$. Remove m from $V_{\hat{H}}$, replace it and remove superfluous nodes $V_H \setminus (N^+_{\hat{H}}(M_3) \cup D)$ as we did in Case 1.

Suppose an element $\hat{d} \in D$ is counted by this case more than twice. Then we have a sequence of three nested circles C_i, $i \in \{1, 2, 3\}$ in H containing different elements m_i. These circles contain all the same node $d \in D$, because otherwise the circle following the last one containing d would enclose it, and d instead of $\hat{d}$ would be counted by the subsequent circle. No node of D can lie between C_1 and C_3 for the same reason, hence all nodes between C_1 and C_3 and are connected to d. The node m_2 must lie between C_1 and C_3, thus all its $k \geq 11$ neighbors lie on or between C_1 and C_3. By the same argument we used to prove the existence of a node in $M_2 \setminus (M_3 \cup D)$ in the circle C in Case 1, a node v in $M_2 \setminus (M_3 \cup D)$ between C_1 and C_3 with $N_H(v) \geq N_H(d)$ exists: At most seven neighbors may lie on C_1 or C_3 and only m_1 and m_3 might be not connected to d. For this node v one of the following two statements holds: It is either (a) enclosed in a circle of length four that encloses no further nodes in H or (b) an additional node from $M_2 \setminus (M_3 \cup D)$ not chosen by m_1 or m_3 is enclosed between C_1 and C_3, because in this case v has d and some element of M_3 as

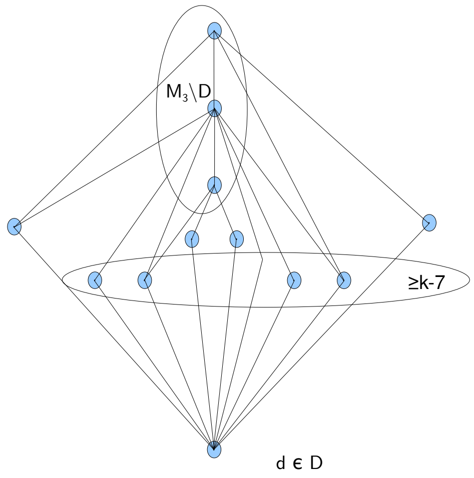

Figure 3: Example of a contradictive structure created by three occurrences of Case 2 in Proposition 4.6. Let the numbering for nodes of $M_3 \setminus D$ be m_1, m_2, m_3 start at the top. Of the at least $k-7$ marked neighbors of m_2 at least $k-9 \geq 2$ are not chosen by m_1 or m_3 in step 2.

neighbors, resulting in a smaller circle in H enclosing an element of $M_2 \setminus M_3$. As the number of nodes in G is finite, eventually (a) must occur for some node v. We have $|N_H(v)| \leq 4$, hence v can not have been chosen in step 2, because all its neighbors are connected to d which is of higher degree in H. Thus such nested circles $C_i,\ i \in \{1,2,3\}$ cannot exist and $\hat{d}$ is counted at most twice by case 2. □

To bound the number of nodes chosen in step 4 which are neither deselected in step 5 nor in D we can use the same arguments as in Proposition 4.4.

PROPOSITION 4.7 (BOUND FOR STEP 4). *Set $A := M_4 \setminus (N^+(M_5) \cup D)$. Then the inequality*

$$|A| \leq (l-1)|D|$$

holds.

PROOF. Analogous to the proof of Proposition 4.4. □

Similarly to Proposition 4.5 and 4.6 we construct a subgraph H to determine an upper bound on the number of nodes selected in step 5. This time we examine paths of length three and we begin with the special case where no node $d \in D$ covers more than five exactly characterized nodes of the two-hop neighborhood of each $m \in M_5$.

PROPOSITION 4.8 (SPECIAL CASE FOR STEP 5). *We construct a subgraph $H = (V_H, E_H)$ of G:*

- *Set $V_H := M_4$.*
- *For each element $m_4 \in M_4$ select an arbitrary element a choosing m_4 in step 4. Add a to V_H and an edge $\{m_4, a\}$ to E_H. We denote by A the set of all these added nodes.*
- *Add all nodes $m \in M_5 \setminus D$ to V_H and edges between m and $N(m) \cap M_4$ to E_H.*

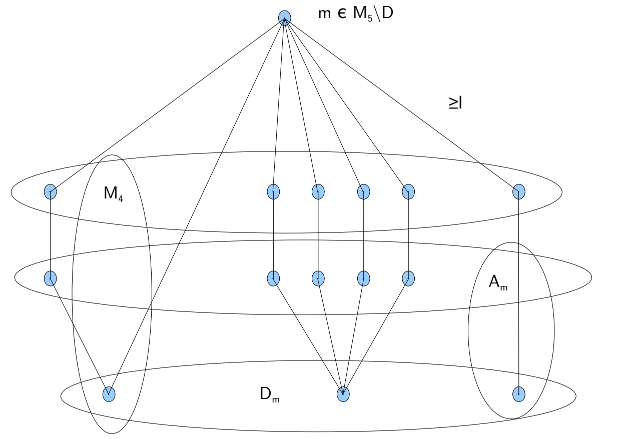

Figure 4: Construction of the subgraph H of G examined in Proposition 4.8 and 4.9. Note that nodes of M_4 or A_m can also belong to D_m.

- *Add D to V_H and a minimum subset of edges in E to E_H ensuring that D covers A in H.*

Denote for each node $m \in M_5 \setminus D$ the minimum set covering $A_m := A \cap N_H(N_H(m))$ by $D_m \subseteq D$. Let l be at least 16. Assume there is no node $m \in M_5 \setminus D$ with some node $d \in D_m$ covering more than five nodes in A_m, i.e.,

$$\forall m \in M_5 \setminus D,\ d \in D_m\ \ |N_H^+(d) \cap A_m| \leq 5. \qquad (3)$$

Then the inequality

$$|M_5 \setminus D| < \frac{3}{\lceil \frac{l-15}{5} \rceil}|D| \qquad (4)$$

holds.

PROOF. The argumentation is the same as in Proposition 4.5. This time, contracting the paths from elements $m \in M_5 \setminus D$ to D_m leads to at least $\lceil \frac{l}{5} \rceil$ edges for each m, because we do not exclude a possible choice of m itself. We have $|M_5 \cup D| \leq |M_5 \setminus D| + |D|$ many nodes in the planar minor of G resulting from the contractions, proving the assertion by Lemma 4.2. □

As in Proposition 4.6 we dispose of the restricting prerequisite of the special case and establish a constant bound for the number of nodes chosen in step 5.

PROPOSITION 4.9 (BOUND FOR STEP 5). *For $l \geq 16$ we have the estimate*

$$|M_5 \setminus D| < (1 + \frac{3}{\lceil \frac{l-15}{5} \rceil})|D|.$$

PROOF. Fix an embedding of G. Let H be the same subgraph of G as in Proposition 4.8 and assume the precondition (3) of Proposition 4.8 to be violated. In this case there exist nodes $m \in M_5 \setminus D$ and $d \in D_m$ connected by six disjoint paths in H, where all except possibly one (if $d \in A$ or $d \in M_4$) have at least one inner node. There are at least five paths of length three forming two nested circles C_1 and C_2 enclosing the innermost path, were C_1 and C_2 consist of disjoint paths from m to d in H. Without loss of generality let the area enclosed by the outer circle C_1 be minimum.

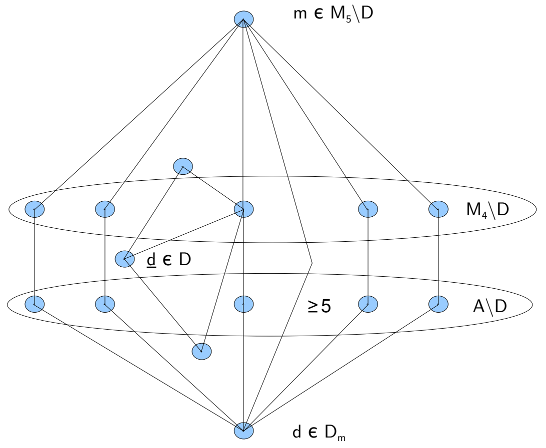

Figure 5: Example of the construction used in Proposition 4.9.

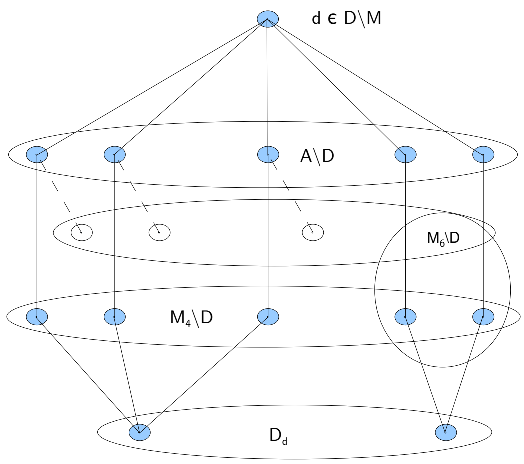

Figure 6: A part of the subgraph H studied in Proposition 4.10 and 4.11. Nodes in $M_6 \setminus D$ may coincide with their counterparts in $M_4 \setminus D$.

Let v be a node in $M_4 \setminus D$. There must be a node $\underline{d} \in D$ distinct from d inside C_2, because otherwise all nodes in $N^+(v) \cap V_4(v)$ except possibly m are neighbors of d as well. But since d covers the voting nodes of C_1, the node selecting v would have preferred d to v in step 4, yielding a contradiction.

We remove m, all its neighbors not connected to other nodes of $M_5 \setminus D$ and the nodes choosing them which are not in D from H and repeat this process, until the prerequisites of Proposition 4.8 are fullfilled. The remaining nodes of $M_5 \setminus D$ in V_H are bounded by (4), hence we must show that we repeat the procedure above at most $|D|$ times.

In each iteration we count a node $\underline{d} \in D$ enclosed by the corresponding C_2 that has not been counted before. This is possible: The area enclosed by C_1 was minimum, hence no $\underline{m} \in M_5 \setminus D$ lying inside C_1 can participate again in the construction. Thus none of the nodes enclosed by C_1 can be involved again, excluding especially nodes on or inside C_2 different from d or m, which is removed. Hence we can apply the same argument as above in each iteration to complete the proof. $\square$

In order to determine the number of nodes chosen in step 6, we need an even more complex construction of a subgraph. Analogously to the propositions above we first establish a bound for a special case.

PROPOSITION 4.10 (SPECIAL CASE FOR STEP 6). *We construct a subgraph $H := (V_H, E_H)$ of G in the following way:*

- *For each element $m \in M_6 \setminus D$ add one element a choosing it in step 6 to V_H. The inserted nodes form the set A.*
- *For each $a \in A$ add $c_4(a)$ to V_H and an edge $\{a, c_4(a)\}$. Denote the set of added nodes by $c_4(A)$.*
- *Remove all edges to nodes $a \in A \cap D$.*
- *Add D to V_H.*
- *Add a minimum subset of E to E_H such that D covers $A \cup c_4(A)$.*
- *Remove all nodes $a \in A$ with $|N_H(d)| < l$, where $d \in D$ is the unique element covering $c_4(a)$ in H.*
- *Remove all nodes $c_4(a) \in c_4(A) \setminus N_H^+(A)$, i.e., the elements of $c_4(A)$ we isolated by removing nodes or edges.*

Denote for each $d \in D \setminus M$ after step 5 by $D_d \subset D$ the minimum set of nodes covering $M_d := c_4(A) \cap N_H(N_H(d))$.

Define $D_M := \{d \in D \,|\, d \notin M$ after step $5\}$. Assume that for all $d \in D_M$ there exists no element of D_d covering more than four elements of M_d. I.e.,

$$\forall d \in D_M \; \hat{d} \in D_d \quad |N_H^+(\hat{d}) \cap M_d| \leq 4. \tag{5}$$

Then the number of nodes chosen in step 6 can be estimated by

$$|M_6 \setminus D| \leq (l + 12)|D|.$$

PROOF. By construction we have $|M_6 \setminus D| = |A|$. As $A \subset V_6$ and $c_4(A)$ is already covered by M after step 5, we have $A \cap c_4(A) = \emptyset$. Furthermore, we removed at most $l|D|$ many nodes from A in the construction. For the remaining nodes $a \in A \cap V_H$ we know that $c_4(a)$ is covered by an element of D_d that must be in M after step 5, as all elements of D_d cover at least l elements of M_4. On the other hand, no element $d \in D$ covering an element of A can be in M after step 5.

Hence contracting all paths from all $d \in D$ to D_d consisting of an element $a \in A \cap N_H(d)$, its choice $c_4(a)$ and the covering element of D_d yields a minor of G with at most $|D|$ vertices and at least $\frac{|M_6 \cap V_H|}{4}$ edges. Thus we get

$$|M_6 \setminus D| = |A| \leq l|D| + |M_6 \cap V_H| \leq (l + 12)|D|$$

by the planarity of H as a minor of G and Lemma 4.2. $\square$

PROPOSITION 4.11 (BOUND FOR STEP 6). *The number of vertices chosen by the algorithm in step 6 is bounded by*

$$|M_6| \leq (l + 22)|D|.$$

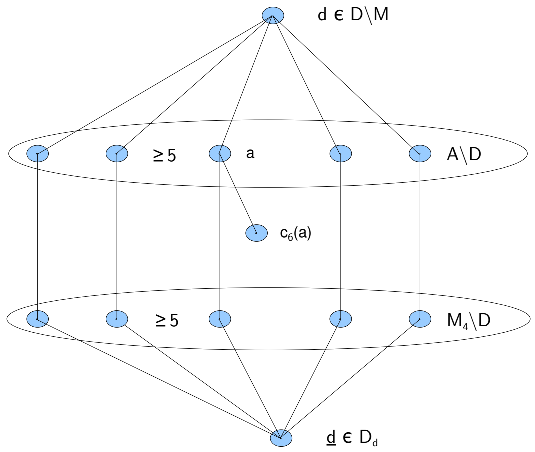

Figure 7: Substructure of H analyzed in Proposition 4.11. Observe that nodes of $M_6 \setminus D$ can also be in $M_4 \setminus D$.

PROOF. Consider again the subgraph $H = (V_H, E_H)$ of G constructed in Proposition 4.10. Suppose the condition (5) of Proposition 4.10 does not hold. In this case a node $d \in D$ not in M after step 5 and some element $\hat{d} \in D_d$ connected by five disjoint paths in H must exist. Each of these paths consists of d, a neighbor a of d in A, the element a chose in step 4 and $\hat{d}$, as $D \cap V_H \cap (c_4(A) \cup A) = \emptyset$ by construction. Fix an embedding of G (and thus H). Let C_1 denote the circle formed by the two outer paths and C_2 the circle enclosing the innermost path formed by two different paths. Let $a \in A$ be the enclosed element on the innermost path. As $c_6(a) \notin D$, it is enclosed by C_1. Without loss of generality let C_1 enclose a minimum area. There must be a node of D different from d and $\hat{d}$ inside C_1. To see this, firstly observe that d and $\hat{d}$ are by construction the only elements of D on C_1, where $\hat{d}$ is already in M after step 5. Node $c_6(a)$ covers at least as many uncovered nodes as d, otherwise a would not have selected $c_6(a)$. Assuming there is no other element of D inside C_1 means any uncovered node inside C_1 must be connected to d, so $c_6(a)$ must cover all nodes covered by d in H as well. This is impossible, as d covers at least two nodes not reachable by $c_6(a)$, taking into account that no node on or in C_2 different from d and $\hat{d}$ can have edges to the inner nodes of more than three of the five paths.

We remove all five paths from H and count a node of D lying inside C_1 not counted twice already. The node counted can not cover any element outside C_1. Hence we can repeat this procedure without counting nodes in D more than twice, as the minimum area property of C_1 together with the planarity of H ensures no path used later lies inside C_1. Thus, in the worst case, a node $d \in D$ already counted once can influence the choice $c_6(a)$ of the vertex playing the role of the enclosed $a \in A$ in the next group of five paths enclosing d. Hence, the process must stop after at most $2|D|$ iterations, each removing five elements of A from H. An application of Proposition 4.10 to the remaining subgraph finishes the proof. □

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 4.12 (BOUND FOR THE NUMBER OF NODES). *Let k be at least 11 and l be at least 16. Then the number of nodes chosen by Algorithm 1 is bounded by*

$$|M| < \left(k + \frac{3}{\lceil\frac{k-10}{3}\rceil} + 2l + \frac{3}{\lceil\frac{l-15}{5}\rceil} + 25\right)|D|. \qquad (6)$$

PROOF. We combine the Propositions 4.4, 4.6, 4.7, 4.9 and 4.11 and we obtain

$$\begin{aligned} |M| \quad &\leq \quad |D| + |M_2 \setminus (N^+(M_3) \cup D)| \\ &\quad + |M_3 \setminus D| + |M_4 \setminus (N^+(M_5) \cup D)| \\ &\quad + |M_5 \setminus D| + |M_6 \setminus D| \\ &< \quad \Bigg(1 + k - 1 + 3 + \frac{3}{\lceil\frac{k-10}{3}\rceil} \\ &\quad + l - 1 + 1 + \frac{3}{\lceil\frac{l-15}{5}\rceil} + l + 22\Bigg)|D| \\ &= \quad \left(k + \frac{3}{\lceil\frac{k-10}{3}\rceil} + 2l + \frac{3}{\lceil\frac{l-15}{5}\rceil} + 25\right)|D|. \end{aligned}$$

□

By assigning the smallest possible integers to k and l we minimize the above constant.

COROLLARY 4.13 (OPTIMUM CHOICE OF PARAMETERS). *For the values $k = 11$ and $l = 16$ the bound (6) is best, yielding*

$$|M| < 74|D|.$$

□

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. Moreover the size of the messages exchanged during the execution of the algorithm is small and the computations performed by each node do neither necessitate large space nor much time. 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 8 and 9. Our algorithm copes with these challenges by exploiting the sparsity as well as the decomposing properties of circles that planar graphs exhibit.

Though our algorithm follows a quite simple idea, astonishingly the procedure of deselecting too many elected adjacent nodes has to be repeated twice. If we stop after step 4, graphs yielding approximation ratios of $\Omega(\sqrt{n})$ can be given. Changing step 4 to electing

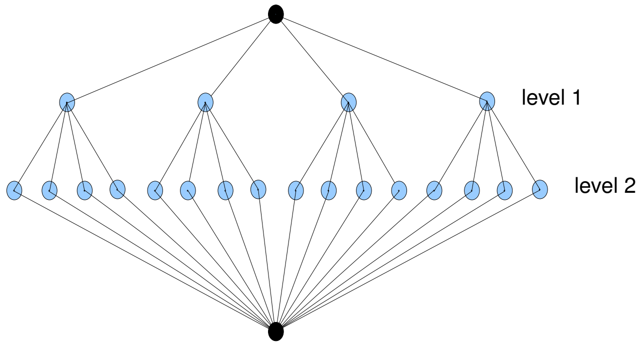

Figure 8: Why a simple broadcast algorithm cannot compute a constant MDS approximation. We suppose nodes are traversed from top to bottom and from left to right. Each node knows its degree and the decisions of already visited neighbors.

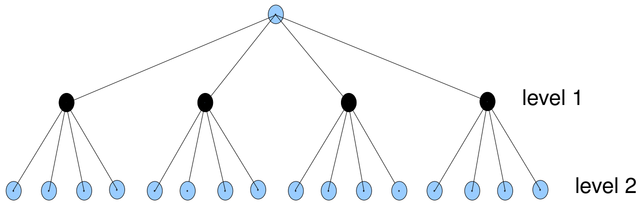

Figure 9: In the graphs displayed here and in Figure 8 the situation looks identical for all level 1 nodes. Thus they must take identical decisions. Entering the DS is wrong in the graph in Figure 8. Not entering the DS leads, again due to indistinguishability, to all level 2 nodes entering the DS in the graph displayed. By scaling up node degrees we see that such an algorithm can achieve at best an approximation ratio of $\Omega(\sqrt{n})$.

the neighbor of highest degree again leads to the same lower bound for the approximation ratio. The algorithm will also fail to give a good approximation quality when applied to sparse graphs. Basically, Propositions 4.6, 4.9 and 4.11 do not apply, as circles do not separate nodes in sparse graphs. On the one hand, this poses the question if MDS approximation on sparse graphs is more difficult. On the other hand, Propositions 4.5, 4.8 and 4.10 rely on the sparsity of a planar graph. Hence one might hope that a nontrivial lower bound only holds for non-sparse graphs. 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 $\mathrm{O}(1)$-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.

7. 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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] B. Gfeller and E. Vicari. A Randomized Distributed Algorithm for the Maximal Independent Set Problem in Growth-Bounded Graphs. In *Proc. 26th annual ACM symposium on Principles of distributed computing (PODC)*, 2007.

[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] 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.

[15] 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.

[16] 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.

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

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

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

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

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

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

[23] 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.

[24] 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.