The Power of Orientation in Symmetry-Breaking

Satya Krishna Pindiproli and Kishore Kothapalli
Center for Security, Theory and Algorithmic Research (CSTAR),
International Institute of Information Technology, Hyderabad
Hyderabad, India 500 032.
Email: {satyakrishna@research., kkishore@}iiit.ac.in

***Abstract*—Symmetry breaking is a fundamental operation in distributed computing. It has applications to important problems such as graph vertex and edge coloring, maximal independent sets, and the like. Deterministic algorithms for symmetry breaking that run in a polylogarithmic number of rounds are not known. However, randomized algorithms that run in polylogarithmic number of rounds are known starting from Luby's algorithm [17].**

Recently, orientation on edges was considered and it was shown that an $O(\Delta)$ coloring of the vertices of a given oriented graph can be arrived at using essentially $O(\log \Delta + \sqrt{\log n})$ bits of communication.

In this paper we further demonstrate the power of orientation on edges in symmetry-breaking. We present efficient algorithms to construct fractional independent sets in constant degree graphs using very low order communication between the vertices. For instance, we show that in bounded degree graphs and planar graphs, it is possible to construct a fractional independent set by exchanging $O(1)$ bits. Further, we present algorithms to construct maximal independent sets in bounded degree graphs and oriented trees. Our algorithm for constructing an MIS of an oriented tree uses only $O(\log n)$ bits of communication.

***Index Terms*—Symmetry Breaking, Orientation, Fractional Independent Sets, Ruling Sets, Maximal Independent Sets.**

I. INTRODUCTION

Consider a distributed system modeled as a graph $G = (V, E)$ with the nodes denoting the set of processors and edges representing the communication links in the distributed system. Computation is achieved by message passing and one can view the computation in rounds where each round involves some local computation and communication. This is often called as distributed computing.

Symmetry breaking is a fundamental problem in distributed computing. It has applications to leader election, graph algorithms such as independent sets and coloring, and scheduling problems in wireless networks. The problem of symmetry breaking can be explained as follows. Often, it is the case that during each round of the algorithm there is a large number of possible operations. But, the interdependencies between them prevents their simultaneous execution. In this case, a quick way to break the symmetry in execution is needed to arrive at algorithms that finish within a small number of rounds. For example to 3-color an n–node directed cycle in the distributed setting, it is difficult to assign colors to many vertices simultaneously because the same color might be assigned to adjacent vertices which makes the coloring invalid. We somehow have to distinguish a set of vertices with the same color from the remaining vertices, but all the vertices look alike and thus the problem is difficult to solve. However, a distributed algorithm that uses symmetry-breaking to solve the problem in $O(\log^* n)$ rounds is presented by Goldberg et al. in [6].

Another example is the problem of finding a maximal independent set (MIS) of a graph. In any step of a distributed MIS algorithm, (cf. [5], [14], [16]), there are many candidate vertices which can be added to the independent set but not all of these candidate vertices can be added. A symmetry-breaking technique is thus essential to find a large set of vertices which can be added to the MIS under construction. Distributed algorithms which present such techniques are discussed in, for example [5], [14], [16].

The standard measure of efficiency of a distributed algorithm is the number of rounds required by the algorithm assuming that in each round a reasonable amount of local computation can be performed. For example, the algorithm of Luby [16] works in $O(\log n)$ rounds where in each round, each yet uncolored node has to pick a color from the set of available colors, and depending on the nodes that are successful in this round, update the set of available colors.

However, while the number of rounds is an important measure of the efficiency of a distributed algorithm, another important measure is the amount of communication in each round. Especially for present generation networks such as wireless sensor networks, communication is a major consumer of energy [19] and algorithms that reduce the amount of communication are of interest. While focusing on graph algorithms, a large number of distributed graph algorithms take the approach of minimizing the number of communication rounds assuming that in each round a reasonable number of bits can be communicated. Thus, studying the exact number of bits that have to be exchanged is a natural and appropriate measure of efficiency of distributed algorithms.

In a recent work, the authors of [12] consider graphs equipped with a sense of orientation on the edges of the graph. It is shown that, if the underlying graph G is provided with an orientation on its edges such that the orientation does not induce oriented cycles of length at most $\sqrt{\log n}$, then a vertex coloring with $(1 + \epsilon)\Delta$ colors, for a constant $\epsilon > 0$, can be obtained by exchanging essentially $O(\log \Delta + \sqrt{\log n})$ bits, with high probability.

While the work of [12] addresses distributed graph coloring using essentially $O(\log \Delta + \sqrt{\log n})$ bits of communication, it is to be seen whether for problems involving symmetry breaking, it is possible to reduce the number of bits exchanged

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.98

by focusing on special classes of graphs.

Towards this end, in this work, we consider the problem of finding fractional independent sets in bounded degree graphs and planar graphs, and, maximal independent sets and k-ruling sets in bounded degree graphs and trees, with orientation on the edges. Interestingly, we show that in the case of planar graphs and bounded degree graphs, we can arrive at a fractional independent set using essentially $O(1)$ bits of communication. The result is possible only because of orientation. Moreover, while a $\Delta + 1$ coloring of bounded degree graphs using only $O(\sqrt{\log \Delta})$ bits of communication [12] can be used to arrive at a fractional independent set of such a graph, the algorithms we present in this work require only $O(1)$ bits of communication.

Our results assume significance in light of the fact that many networks have planar topologies and algorithms such as geometric routing algorithms in wireless networks [13], information gathering trees in wireless sensor networks [8] work on planar graphs. Further, a fractional independent set can be used as a subroutine in several important distributed algorithms.

We finally note that oriented graphs are different from, and should not be confused with, directed graphs. Unlike directed graphs where communication is possible only in the direction of the edge, in oriented graphs, communication can proceed in both directions. Further, we assume that nodes have a reasonable mechanism to store their neighbors along with the orientation of the edges incident at that node. For example, when one stores the neighbors in an adjacency lists, node v can store $+w$ when the edge vw is oriented from v to w and $-w$ otherwise.

A. *Model and Definitions*

In this section, we review the model from [12]. Let us model a distributed system as a graph $G = (V, E)$ with V representing the set of computing entities, or processors, and $E \subseteq V \times V$ representing all the available communication links. We assume that all the communication links are undirected and hence bidirectional. All the processors start at the same time and time proceeds in synchronized rounds. We let $n = |V|$. The degree of node u is denoted d_u and by Δ we denote the maximum degree of G, i.e., $\Delta = \max_{v \in V} d_v$. We do not require that the nodes in V have unique labels of any kind. For our algorithms to work, it is enough that each node has knowledge of n and Δ apart from its own degree and neighbors. When we consider graphs of bounded degree, no global knowledge is required for our algorithm and it suffices that each node knows its own degree.

In the model[12], the measure of efficiency is the number of bits exchanged. We also refer to this as the *bit complexity*. We view each round of the algorithm as consisting of 1 or more *bit rounds*. In each bit round each node can send/receive at most 1 bit from each of its neighbors. We assume that the rounds of the algorithm are synchronized. Figure 1 shows three rounds with round 1 having 4 bit rounds, round 2 having 3 bit rounds, and round 3 having 4 bit rounds. The bit complexity of algorithm A is then defined as the number of bit rounds required by algorithm A. We note that, since the nodes are synchronized, each round of the algorithm requires as many bit rounds as the maximum number of bit rounds needed by any node in this round. Local computation performed by the nodes is not counted, which is reasonable because in our algorithms nodes perform only simple local computation.

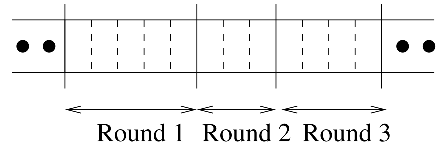

Fig. 1. Figure showing the notion of a round and a bit round. The communication in a round can consist of multiple bit rounds. In the figure, round 1 has 4 bit rounds, round 2 has 3 bit rounds, and round 3 has 4 bit rounds. Bit rounds are shown via dashed edges.

It is assumed that the edges in E have an orientation associated with them. That is, for any two neighbors v, w exactly one of the following holds for the edge $\{v, w\}$: $\{v, w\}$ is oriented either $v \rightarrow w$ or as $w \rightarrow v$. In the former we also call v *superior* to w and vice-versa in the latter. Having orientation on the edges is a property that has not been studied extensively in the context of symmetry breaking. It is however a natural property since networks usually evolve and for every connection there is usually a node that initiated it. We show that algorithms for symmetry breaking can be greatly improved provided that the underlying graph is oriented.

The exact way in which orientation is used for symmetry breaking is explained in Figure 2. As shown, if nodes v and w choose the same action during any round of the algorithm, for example opting to enter the MIS under construction, in the existing algorithms, both nodes stay out of the MIS as in Figure 2(b) and have to try in a later round. With orientation, if the edge $\{v, w\}$ is oriented as $v \rightarrow w$ as shown in Figure 2(c), then node v can retain its choice and enter the MIS provided that there is no edge $\{u, v\}$ oriented $u \rightarrow v$ and u also chooses the enter the MIS.

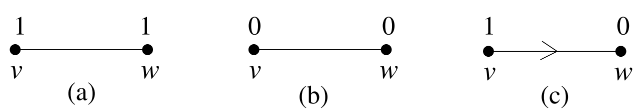

Fig. 2. Orientation helps in symmetry breaking. In Figure (a) both v and w have the same priority (state of 1 shown above the node labels in the figure) in case of conflict. In (b), for existing algorithms both remain unchosen (state of 0) whereas in (c), when using orientation, node v is superior to w and hence may be chosen.

One parameter that will be important for our investigations is the length of the shortest cycle in the orientation. We formalize this notion in the following definition.

Definition 1.1 (ℓ–acyclic Orientation): An orientation of the edges of a graph is said to be ℓ–acyclic if and only if the minimum length of a directed cycle induced by the orientation is at least $\sqrt{\log n}$. Note that this is not the girth of the given graph.

B. Related Work

Symmetry-breaking in distributed computing is a fundamental problem with a rich literature. Here, we focus on the literature corresponding to vertex coloring and maximal independent sets as these are more pertinent for our discussion. (It is known that an MIS can be obtained from a coloring and vice-versa).

It is an open problem whether deterministic poly-logarithmic time distributed algorithms exist for the problem of $(\Delta+1)$-vertex coloring [20]. The best known deterministic algorithm to date is presented in [20] and requires $n^{O(1/\sqrt{\log n})}$ rounds. Hence, most of the algorithms presented are randomized algorithms.

Karp and Widgerson [14] have shown that an MIS can be found in $O(\log^3 n)$ rounds w.h.p. and Luby [16] presents algorithms to find MIS in arbitrary graphs in $O(\log n)$ rounds with high probability. Luby [17] and Johansson [10] present parallel algorithms that can be interpreted as distributed algorithms that provide a $(\Delta+1)$–coloring of a graph G in $O(\log n)$ rounds, with high probability. Algorithms for vertex coloring are also presented in [5], [11] in the PRAM model of computation.

Focusing on special classes of graphs, Cole and Vishkin [1] and Goldberg et. al. [5] have shown that a $(\Delta+1)$–coloring of the cycle graph on n nodes can be achieved in $O(\log^* n)$ communication rounds. This was shown to be optimal in Linial [15] by establishing that 3-coloring an n–node cycle graph cannot be achieved in less than $(\log^* n - 1)/2$ rounds. The result of [5] extends to also 3-coloring of rooted trees, and 5-coloring of planar graphs. When arbitrary amount of local computation is allowed [15], De Marco and Pelc [18] show that an $O(\Delta)$ coloring can be achieved in $O(\log^*(n/\Delta))$ rounds improving the results of Linial [15] in this model. When unlimited local computation is available Linial [15] shows how to obtain an $O(\Delta^2)$ coloring in $O(\log^* n)$ rounds. This was later improved by De Marco and Pelc [18] to show that an $O(\Delta)$ coloring can be achieved in $O(\log^*(n/\Delta))$ rounds.

Linial's work [15] was extended to study the trade-off between the amount of advice and the number of communication rounds required to color (oriented) cycles and trees using 3 colors. It was shown by Fraigniaud et. al. [4] that to color oriented cycles and trees in less than $O(\log^* n)$ communication rounds requires $\Omega(n/\log^{(k)} n)$ bits of advice where $\log^{(k)} n$ denotes the k iterations of $\log n$. In the above, advice refers to the amount of information given to the nodes for example about the topology of the network. However, in this model, the amount of bits transfered during each communication round is not limited.

Distributed algorithms with the underlying graph equipped with sense of direction have been studied in [21], [3]. Sense of direction is a similar notion to that of orientation on edges. Singh [21] shows that leader election in an n-node complete graph equipped with sense of direction can be performed in a distributed setting via exchange of $O(n)$ messages. In [3], the authors show that having sense of direction reduces the communication complexity of several distributed graph algorithms such as leader election, spanning tree construction, and depth-first traversal. More recently, the authors in [12] show that an $O(\Delta)$ vertex coloring of a given graph G with its edges oriented can be found in essentially $\tilde{O}(\sqrt{\log n})$ bits of communication. Empirical analysis of the algorithm presented in [12] is presented in [7].

We note that some of our algorithms in this work have resemblance to the algorithms in JaJa [9].

C. Our Results

In this work we consider the problem of fractional independent sets (FIS) and maximal independent sets. We focus on classes of graphs for which such independent sets can be constructed with very small bit complexity. Specifically, we show the following results.

- For constant degree graphs equipped with a random orientation on the edges, we show that a fractional independent set can be found with NO communication.
- For oriented planar graphs, with a random orientation between the edges, we construct fractional independent sets using only a $O(1)$ bits of communication between the vertices of the graphs.
- We give an algorithm to construct a maximal independent set of an oriented tree in $O(\log n)$ rounds and using $O(\log n)$ bits of communication.
- We construct a maximal independent set in a bounded degree graph using $O(\sqrt{\log n})$ bits of communication.

While for bounded degree graphs, the coloring algorithm presented in [12] can achieve fractional and maximal independent sets using only $O(\sqrt{\log n})$ bits of communication, it is interesting to note that for example, an FIS can be achieved with no communication.

D. Rest of the Paper

The rest of the paper is organized as follows. In Section II we describe an algorithm to construct a fractional independent sets in bounded degree oriented graphs and planar oriented graphs. In Section III, we present an algorithm to construct an MIS in bounded degree graphs and oriented trees.

II. Fractional Independent Sets

Given a graph $G = (V, E)$, a *fractional independent set* is a large independent set of G consisting exclusively of low-degree vertices. More precisely, a fractional independent set is a set $X \subseteq V$ such that

1) the degree of each vertex $v \in X$ is less than or equal to some constant d,
2) the set X is independent, i.e., no two vertices of X are connected by an edge, and
3) the size of X satisfies $|X| \geq c|V|$ for some positive constant c.

In the following, we first discuss our algorithms for bounded degree oriented graphs. In Section II-B we consider FIS in oriented planar graphs.

A. FIS in Bounded Degree Graphs

We now present an algorithm to construct a fractional independent set in a bounded degree graph where the edges are equipped with a random orientation.

1) Graphs with random orientation: Let $G = (V, E)$ be a bounded degree graph. Let each edge $e = vw$ in E be oriented independently and from v to w, or from w to v with equal probability. We then say that G is randomly oriented, or G has a random orientation.

In bounded degree graphs with a random orientation, communication between the vertices is NOT required to form an FIS. Let $d = \Delta(G)$ be the degree of the graph. For all the edges from a particular vertex v to be oriented outwards from it, the probability is $1/2^d$. Thus, the probability that all the edges from or to a low-degree vertex are oriented in a particular direction is $1/2^d$. Thus, the expected number of such vertices in $n/2^d$. Since d is constant, this is $n/O(1)$. All such vertices will definitely form an independent set and thus they form a fractional independent set.

Since no communication is necessary to find such vertices, no exchange of bits takes place between the vertices.

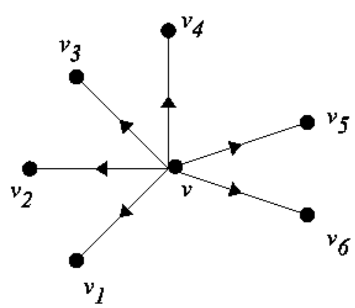

Fig. 3. Random Orientation between vertices with all outgoing edges. All such vertices can directly be taken into the Fractional Independent Set.

The above observations lead us to the following theorem.

Theorem 2.1: For a bounded degree graph G with its edges oriented randomly, one can find a fractional independent set in G using no communication, with high probability.

Proof: From the above discussion, if each node is taken to be in the FIS if it has no edge oriented inward, then there are $n/2^d$ vertices in expectation that are part of the FIS.

Let X_v denote a random variable that takes a value of 1 if v is in the FIS and 0 otherwise. Let $X := \sum_{v \in V} X_v$ counts the number of vertices in the FIS. Further, $E[X] = n/2^d$.

Since X is a sum of independent Bernoulli random variables, one can use Chernoff bounds on the lower tail of X to bound X from the below. Hence,

$$\Pr[X \leq (1-\epsilon)E[X]] \leq e^{-E[X]\epsilon^2/3} = e^{-n\epsilon^2/3 \cdot 2^d}.$$

The above probability is polynomially small even for a constant ϵ. Hence, $|X| = n/c'$ for some constant c' with high probability.

Since no bits are exchanged in the process, the algorithm has zero bit complexity. ■

The above result is possible only due to the random orientation on the edges of the graph. However, for special cases where the graph is oriented in a linear fashion, for example the graph P_n or C_n with all edges oriented in the same direction, the above result does not hold. In the following subsection, we handle these special cases. Our algorithm for this case has a bit complexity of $O(\sqrt{\log n})$.

2) Graphs with a Linear Orientation: Consider the line graph shown in Figure 4 with all edges oriented in the same direction. For such a graph, we show that an FIS can be constructed using essentially $O(\sqrt{\log n})$ bits of communication. The same technique extends to also oriented cycles where all edges are oriented in the clockwise (or anticlockwise) direction. For simplicity, we say that for a node v, its neighbor w such that vw is oriented towards(away from) w is called its right (left) neighbor. Our algorithm and the analysis has some similarity to that of [12].

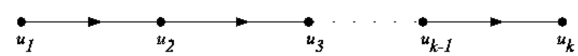

Fig. 4. Linear Orientation between the nodes.

Our algorithm for the line graph works in two phases as follows. In Phase I, nodes choose a label in $\{0, 1\}$ independently and uniformly at random. A node v be added to the FIS if its choice of the label is 1 and its left neighbor (i.e., the node u such that uv is oriented from u to v) did not choose 1. In this case, we will argue that in $O(\sqrt{\log n})$ rounds of Phase I, every oriented path of length $\sqrt{\log n}$ has a node in the FIS with high probability. Phase II uses this to complete the construction of the FIS using orientation on the edges.

The detailed algorithm is given in Figure 5. The algorithm is executed by every node till its status is known. A status of 1 indicates that the node is in FIS, and 0 indicates that the node is not in the FIS.

We now analyze the above algorithm and show the following theorem.

Theorem 2.2: Given the line graph G with all its edges oriented in the same direction, the above algorithm constructs an FIS in $O(\sqrt{\log n})$ rounds with high probability. Further, the bit complexity of the algorithm is $O(\sqrt{\log n})$.

Proof: The proof for Phase I shows that in every oriented path of length $\sqrt{\log n}$ there is at least one node v such that status$(v) = 1$. The proof for Phase II then shows that a further $\sqrt{\log n}$ rounds are enough to set the status of all nodes correctly.

a) Analysis for Phase I:: Let P be an oriented path of length $\ell = \sqrt{\log n}$. Let $E_{P,i}$ denote the event that at the end of round i, all nodes in P still have their status as -1. Then,

$$\Pr(E_{P,i}) = (1/c)^{\ell}, \text{ for some constant } c.$$

Let the event E_P denote the event that $E_{P,i}$ occurs for for $r = 4\sqrt{\log n}$ consecutive rounds. Then,

$$\Pr(E_P) = (1/c)^{\ell r} = (1/c)^{4 \log n}.$$

Finally, let E denote the event that for some path P of length ℓ, the event E_P occurs. We have that $\Pr(E) =$

Algorithm FIS-Linear(G)
  Initialize status(v) $= -1$
  Comment: Phase I
  for $4\sqrt{\log n}$ rounds do
    1. Choose label(v) independently and uniformly at random between $\{0, 1\}$.
    2. Communicate the choice of the label to the right neighbor
    3. If v receives 1 from its left neighbor then
      set label(v) $= 0$.
    Otherwise, set status(v) $= 1$.
  end-for.
  Comment: Phase II
  repeat
    1. If status(v) $= 1$ then
      Communicate 0 to the right neighbor.
    2. else
      Communicate 1 to the right neighbor of v.
    3. If v receives 0 from its left neighbor then
      set status(v) $= 0$.
  until status(v) $\neq -1$
End-Algorithm

Fig. 5. Algorithm for constructing FIS in a linear graph.

$\Pr(\cup_P E_P) \leq \sum_P \Pr(E_P) \leq n \cdot \frac{1}{c^{4 \log n}}$ which is at most $1/n^3$.

So, with high probability, at least one node in every path P of length $\sqrt{\log n}$ has a node v with status(v) $= 1$. This completes the analysis of Phase I.

At the end of Phase I, the graph is broken into connected components so that each component has nodes with status still being -1. These components are separated by nodes in the FIS. Such nodes in the FIS initiate a marking process that sets the status of the nodes in these components correctly. This happens in a way that in each round, one node in each component has its status set correctly. Since the number of nodes in any component is at most ℓ, with high probability, at the end of ℓ rounds, all nodes have their status set correctly.

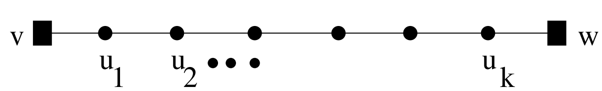

Fig. 6. A path of k nodes $u_1, u_2, \cdots, u_k$ such that $k \leq \ell$ with no nodes in FIS, but bounded by nodes v and w in FIS. Node v can initiate the marking process in Phase II. The orientation on the edges, not shown in the Figure, is as follows. The edge vu_1 is oriented from v to u_1, the edge u_iu_{i+1} is oriented from u_i to u_{i+1} for $1 \leq i < k$ and the edge u_kw is oriented from u_k to w.

Since every path of length $\sqrt{\log n}$ contributes at least $\sqrt{\log n}/3$ nodes to the FIS, the FIS constructed has a size of at least $n/3$.

Further, in each round in Phase I and in Phase II, nodes communicate at most 1 bit. So, the overall bit complexity of the algorithm is $O(\sqrt{\log n})$. ■

B. FIS in Planar Graphs

If $G = (V, E)$ is a planar graph with at least three vertices then using Euler's formula, it can be proved that $|E| \leq 3|V| - 6$. This indicates that planar graphs are quite sparse. Hence, we expect to find many vertices of low degree, since $\sum_{v \in V} deg(v) = 2|E| \leq 6|V| - 12$. The following theorem gives a constructive proof of the fact that a fractional independent set exists in any planar graph.

Theorem 2.3: Given a planar graph $G = (V, E)$, a fractional independent set X for G can be constructed in linear sequential time.

Proof: Let d be a constant such that $d \geq 6$. Let V_d denote the set of vertices of G with degree at most d. It can be shown that $|V_d| \geq c.|V|$ for a constant c using Euler's formula. To actually use V_d to construct a fractional independent set, one can proceed as follows.

A fractional independent set S can be constructed as follows. Initially, let v be any vertex in V_d chosen arbitrarily. Place v in S and remove from consideration all neighbors of v. Continue this process by picking another vertex from V_d, adding it to S, and removing the neighbors of the chosen vertex from consideration. When there are no more vertices in V_d available for consideration, we can see that S is a fractional independent set as follows.

The number of vertices in S is at least $|V_d|/(d+1) \geq (d-5)|V|/(d+1)^2$, which is a constant fraction of $|V|$ (since $d \geq 6$ and is constant). ■

The scheme suggested in the proof of Theorem 2.3 is a highly sequential scheme and is not amenable to a polylogarithmic time distributed algorithm with any bit complexity. To arrive at a distributed scheme, with a low bit complexity, we proceed as follows.

Let $G = (V, E)$ be a planar graph whose edges are oriented uniformly at random. We thus assume that any edge $e = uv$ is oriented from u to v or from v to u with equal probability. Under this assumption, we now show that a fractional independent set can be constructed with only $O(1)$ bits of communication. Our algorithms runs also in a single round.

Define a *low-degree vertex* as a vertex of degree less than or equal to 6. As seen in the proof of Theorem 2.3, if we set $d = 6$, there are at least $|V|/7$ such vertices. Let us denote by V_ℓ the set of low degree vertices.

The basic technique used is **randomized symmetry breaking** that consists of the following steps. Our algorithm runs in one round and each low degree node chooses a label in $\{0, 1\}$ uniformly and independently at random. The choice of the label is exchanged between neighbors. If no superior low degree neighbor also chose 1, then a node which chose 1, enters the FIS.

We now show the following theorem.

Theorem 2.4: Given a planar graph $G = (V, E)$, a fractional independent set S can be found in one round and the bit complexity of the algorithm is $O(1)$.

Proof: We note that the set of vertices whose labels are equal to 1 does not necessarily form an independent set. For each edge $(u, v) \in E$, we relabel vertices u and v whenever u

Algorithm FIS-Planar(G)

1. Choose label(v) independently and uniformly at random between $\{0, 1\}$.
2. Communicate the choice of the label to its low degree superior neighbors
3. If v receives 1 from any of its low degree neighbors then set label(v) $= 0$.
Otherwise, set status(v) $= 1$.

End-Algorithm.

Fig. 7. Algorithm for FIS in planar graphs.

and v were assigned the label 1. The remaining vertices (which are of *low-degree*) with label 1 constitute an independent set whose size is a constant fraction of $|V|$ with high probability and this is the fractional independent set S.

To show that the size of the independent set thus obtained is at least n/c' for a constant c' with high probability, we proceed as follows. Let X_v be a random variable which takes a value of 1 when v is in FIS and 0 otherwise, where v is a low degree node. Unlike the proof of Theorem 2.1, the random variable $X := \sum_{v \in V_\ell} X_v$ is not a sum of independent Bernoulli random variables. However, one can manufacture independence by choosing only a subset of the low degree vertices (cf. [9]). In this direction, let $V_{\geq 3} \subseteq V$ be the set of low degree neighbors which are at a distance of at least 3 from each other. Then, $X_{\geq 3} := \sum_{v \in V_{\geq 3}} X_v$ is a sum of independence Bernoulli random variables. Moreover, since $|X_{\geq 3} \geq |V_\ell|/36 \geq |V|/36 \cdot 7$. Hence, $E[X] \geq n/252$.

One can now use Chernoff bounds on $X_{\geq 3}$ and show that the event that $|X_{\geq 3}| \leq n/c'$ for a suitable constant c' has a polynomially small probability. Hence, the independent set obtained is a fractional independent set.

It can be seen that each node communicates 1 bit in the algorithm and hence the algorithm has $O(1)$ bit complexity. ■

Remark: At this point, one may think that since we are working with the subgraph of G induced by vertices of low degree, the algorithm in Section II-A can be used in this context also. To use the approach of Section II-A, in the case of a planar oriented graph, a node has to know the orientation of only its edges incident to low degree neighbors. For this, a node has to know not only its degree but also the degree of its neighbors. In a planar graph, this may require $\Omega(\log n)$ bits of communication as nodes may have a degree of $\Theta(n)$. Hence, a new approach is required.

III. MIS IN ORIENTED GRAPHS

A Maximal Independent Set (MIS) in a graph $G = (V, E)$ is a maximal subset $U \subseteq V$ so that no two nodes in U are neighbors in G. In this section, we show that in oriented trees and oriented bounded degree graphs an MIS can be found using very few bits of communication.

A. Constructing an MIS of an Oriented Tree

We define an oriented tree as an oriented graph $G = (V, E)$ that is connected and is acyclic. It is different from a rooted tree [5] where all edges are oriented towards the root of the tree [1].

In this section, we show how to construct an MIS in an oriented tree of arbitrary degree Δ. Notice that the 3-coloring algorithm of [5] can be used to arrive at an MIS. The algorithm of [5] requires $O(\log^* n)$ rounds and has a bit complexity of $O(\log n)$. While our algorithm also shall have a bit complexity of $O(\log n)$ to arrive at an MIS, we note that our algorithm works with any orientation on the tree and not necessarily a rooted tree, still having the same bit complexity. Further, the algorithm of [5] requires nodes to have unique numbers. Our approach does not need this assumption, and hence can work for wireless sensor networks which may not have unique identifiers. Our algorithm is described below in Figure 8.

Algorithm MIS-Tree(T)

Comment: Phase I
1. for $c \log n$ rounds do
    $V_i = \text{FIS(T)}$
    $T = T\ V_i$
end-for
Comment: Phase II
2. for each node $v \in V(T)$ set status = -1;
3. for $i = 1$ to $c \log n$ rounds do
    4.If status(v) $\neq 0$
        add node v in V_i to the MIS
        set status(v) $= 1$;
end-for
5. For each node v with status(v) $= 1$ do
    6. Send 1 to all the neighbors of v
end-for
7. If node v receives 1 from any of its neighbors
    set status(v) $= 0$
end-for

End-Algorithm.

Fig. 8. Algorithm for MIS in an oriented tree.

The idea of the above algorithm is that $O(\log n)$ FIS sets $V_1, V_2, \cdots$, are constructed for the tree T. Now, the nodes in set V_i are added to the MIS (status = 1) provided that no neighbor of such a node in V_{i-1} is already added to the MIS. We show the following theorem.

Theorem 3.1: Given an oriented tree $T = (V, E)$, Algorithm MIS-Tree constructs an MIS of T in $O(\log n)$ rounds. Further, the algorithm has a bit complexity of $O(\log n)$.

Proof: Let us consider Phase I of the algorithm. Since T is a planar graph, it holds from Theorem 2.4 that it has a fractional independent set. So there exists a constant c such that all nodes in T can be partitioned into $c \log n$ fractional independent sets $V_1, V_2, \cdots, V_{c \log n}$. During Phase I, since each node communicates 1 bit in every iteration of the loop in line 1, the bit complexity of Phase I is $O(\log n)$.

It can be seen that also in Phase II, nodes communicate 1 bit. So, the bit complexity of Phase II is $O(\log n)$. The correctness

[1]A rooted tree is one in which a specific node is designated as the *root* of the tree. The edges in a rooted tree have a natural orientation, towards or away from the root. In a rooted tree, every vertex except the root has a unique parent.

Algorithm MIS-BoundedDegree(G)

1. Every vertex v chooses 1 or 0 independently with probability $\frac{1}{2\Delta}$ and $(1-\frac{1}{2\Delta})$ respectively.
2. Node v communicates its choice with its neighbors.
3. If no superior of v has chosen 1, v retains its choice and is added to the MIS.
4. If v is added to MIS, then v sends a dropout message to neighbors of the MIS nodes.

End-Algorithm.

Fig. 9. Algorithm for MIS in bounded degree graphs.

of the construction follows from the fact that nodes are added to MIS only when none of its lower colored neighbors are in the MIS.

So the overall bit complexity of the algorithm is $O(\log n)$. ■

B. MIS in Bounded Degree Graphs

We now present our algorithm for MIS in bounded degree graphs. In the algorithm shown in Figure 9, nodes choose a label, either 0 or 1, independently and uniformly at random. A choice of 1 indicates that the node is willing to enter the MIS under construction. Nodes have to resolve conflicts with their superior neighbors by exchanging their choices. Running this algorithm for $O(\sqrt{\log n})$ rounds results in an MIS as we will show in Theorem 3.2.

Theorem 3.2: Let G be a bounded degree graph provided with an $\sqrt{\log n}$ acyclic orientation. Then, an MIS of G can be found in $O(\sqrt{\log n})$ rounds using essentially $O(\sqrt{\log n})$ bits of communication.

Proof: We divide the proof into two phases. Phase I ends when every path of length $\ell = \sqrt{\log n})$ has at least one node in MIS. Phase II then uses the orientation to finish off in another $\sqrt{\log n}$ rounds.

Let us consider Phase I. A vertex v will be in the maximal independent set if it chooses 1 and all its neighbors choose 0. Let us call this event v *wins*. The complementary of this event that v is not in the MIS (i.e., v *loses*) is a sub-event of v losing out in some path of the graph. Lets denote this event as E. The probability E occurs in a path P of length $\ell = \sqrt{\log n}$ in the ith round is

$$Pr(E_{P,i}) \leq (\frac{1}{c})^{l} \quad (1)$$

Let E_P denote the event that E_{P_i} occurs for $r = 4\sqrt{\log n}$ rounds consecutively. Then,

$$Pr(E_P) \leq (\frac{1}{c})^{lr} \quad (2)$$

Let E denote the event that there exists some path P of length ℓ the event E_P holds. Then,

$$\Pr(E) = \Pr(\cup_P E_P) \leq n\Delta)^{l}.(\frac{1}{c})^{lr} = \frac{1}{n^4}. \quad (3)$$

Let us now consider Phase II. After the first r rounds, the graph G is separated into connected components of diameter at most $\sqrt{\log n}$ so that nodes in MIS separate the components. In each component, any oriented path has length at most $\sqrt{\log n}$. One can now use the fact that the graph G is provided with a $\sqrt{\log n}$–acyclic orientation, the components are acyclic. This implies that in each component, nodes can be organized in to layers so that nodes in layer i are superior to nodes in layer $i+1$ and that there are at most $\sqrt{\log n}$ layers.

If each node runs the same algorithm, then at the end of i further rounds, nodes in layer i can decide whether they are in the MIS or not. Hence, Phase II finishes in a further $\sqrt{\log n}$ rounds.

Since in each round of Phase I and Phase II, nodes exchange at most 1 bit with its neighbors, the bit complexity of the algorithm is $O(\sqrt{\log n})$. ■

We note that with small modifications, the above approach can be extended to find a k-ruling set in a bounded degree graph with essentially $O(\sqrt{\log n})$ bits of communication.

IV. Conclusion and Future Work

In this paper we discussed efficient algorithms to construct fractional independent sets in constant degree oriented graphs and planar oriented graphs. We also presented algorithms to construct MIS in bounded degree graphs and oriented trees.

As future work, we plan to extend our MIS algorithm of an oriented tree to color an oriented tree using only $O(1)$ colors. This requires a different approach as the degree of a tree can be $\Theta(n)$. Similarly, it would be interesting to see how orientation helps in other problems involving symmetry breaking.

References

[1] R. Cole and U. Vishkin. Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms. In *Proceedings of the 18th ACM symposium on Theory of computing(STOC)*, pages 206–219, 1986.

[2] D. P. Dubhashi and A. Panconesi. Near-optimal distributed edge coloring. In *European Symposium on Algorithms (ESA)*, pages 448–459, 1995.

[3] P. Flocchini, B. Mans, and N. Santoro. On the impact of sense of direction on message complexity. *Information Processing Letters*, 63(1):23–31, 1997.

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

[5] A. Goldberg, S. Plotkin, and G. Shannon. Parallel symmetry breaking in sparse graphs. *SIAM J. Discrete Math.*, 1:434–446, 1989.

[6] D. A. Grable and A. Panconesi. Nearly optimal distributed edge colouring in $O(\log\log n)$ rounds. *Random Structures and Algorithms*, 10(3):385–405, 1997.

[7] Satya Krishna Pindiproli, Kishore Kothapalli, *Experimental Analysis of Distributed Coloring Algorithms,* IACC 2009, pp. 147–152.

[8] Kothapalli K., Onus M., Richa A., Scheideler C., *Efficient Broadcasting and Gathering in Wireless Ad-Hoc Networks,* Proc. of IEEE International Symposium on Parallel Architectures, Algorithms and Networks, 2005, pp. 346–351.

[9] Joseph JaJa, *An Introduction to Parallel Algorithms*, Addison-Wesley Publishing Company, 1997.

[10] O. Johansson. Simpled distributed $\Delta+1$ coloring of graphs. *Information Processing Letters*, 70:229–232, 1999.

[11] M. Karchmer and J. Naor. A fast parallel algorithm to color a graph with Δ colors. *J. Algorithms*, 9:83–91, 1988.

[12] Kothapalli, K., Onus, M., Scheideler, C. and Schindelhauer C., *Distributed Coloring in O $(\sqrt{\log n})$-bits*. Proceedings of IPDPS, 2006.

[13] Fabian Kuhn, Roger Wattenhoffer and Aaron Zollinger, *Asymptotically optimal geometric mobile ad-hoc routing,* Proc. of International workshop on Discrete algorithms and methods for mobile computing and communications, 2002, pp. 24–33.

[14] R. Karp and A. Widgerson. A fast parallel algorithm for the maximal independent set problem. *J. Assoc. for Computing Machinery*, 32(4):762–773, 1985.

[15] N. Linial. Locality in distributed graph algorithms. *SIAM Journal of Computing*, 21:193–201, 1992.

[16] M. Luby. A simple parallel algorithm for the maximal independent set problem. In *Proc. of ACM Symposium on Theory of Computing*, pages 1–10, 1985.

[17] M. Luby. Removing randomness in parallel without processor penalty. *Journal of Computer and System Sciences*, 47(2):250286, 1993.

[18] G. De Marco and A. Pelc. Fast distributed graph coloring with $O(\Delta)$ colors. In *Proc. of ACM Symposium on Discrete Algorithms*, pages 630–635, 2001.

[19] G. Pottie and W. Kaiser. Wireless Integrated Network Sensors. Communications of the ACM, 43:51-58, 2000.

[20] A. Panconesi and A. Srinivasan. On the complexity of distributed network decomposition. *J. Algorithms*, 20(2):356–374, 1996.

[21] G. Singh. Efficient leader election using sense of direction. *Distributed Computing*, 10(3):159–165, 1997.