Distributed computing with advice: information sensitivity of graph coloring

Pierre Fraigniaud, Cyril Gavoille, David Ilcinkas, Andrzej Pelc

Distributed Computing with Advice: Information Sensitivity of Graph Coloring[§]

Pierre Fraigniaud[*] Cyril Gavoille[†] David Ilcinkas[†] Andrzej Pelc[‡]

Abstract

We study the problem of the amount of information (advice) about a graph that must be given to its nodes in order to achieve fast distributed computations. The required size of the advice enables to measure the information sensitivity of a network problem. A problem is information sensitive if little advice is enough to solve the problem rapidly (i.e., much faster than in the absence of any advice), whereas it is information insensitive if it requires giving a lot of information to the nodes in order to ensure fast computation of the solution. In this paper, we study the information sensitivity of distributed graph coloring.

Keywords: network algorithm, graph coloring, distributed computing.

[§]A preliminary version of this paper appeared in the proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP), July 2007.

[*]CNRS, Laboratoire d'Informatique Algorithmique: Fondements et Applications (LIAFA), Université Denis Diderot, Paris, France. E-mail: pierre.fraigniaud@liafa.jussieu.fr. Additional support from the ANR project ALADDIN.

[†]CNRS, Laboratoire Bordelais de Recherche en Informatique (LaBRI), Université Bordeaux, France. E-mail: gavoille@labri.fr, David.Ilcinkas@labri.fr. A part of this work was done during the stay of David Ilcinkas at the Research Chair in Distributed Computing of the Université du Québec en Outaouais, as a postdoctoral fellow.

[‡]Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mail: pelc@uqo.ca. Andrzej Pelc was supported in part by NSERC discovery grant and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

1 Introduction

This work is a part of a recent project aiming at studying the quantitative impact of *knowledge* on the *efficiency* when computing with distributed entities (nodes of a distributed system, mobile users in ad hoc networks, etc.). Indeed, as observed by Linial [18], "within the various computational models for parallel computers, the limitations that follow from the local nature of the computation are specific to the distributed context". Two frameworks have been considered for analyzing the limitations incurred by the local nature of distributed computations. One aims at identifying which tasks can or cannot be computed locally, i.e., when every node can acquire knowledge only about the nodes that are at constant distance from it. Surprisingly, non-trivial tasks can be achieved locally [23]. This is for instance the case for weak-coloring, a basis for a solution to some resource allocation problems. However, many important problems in distributed computing do not have a local solution [16]. This is the case of computing an approximate minimum vertex cover or an approximate minimum dominating set.

The other framework that has been considered is distributed computing *with advice*. In this framework, the computing entities can be given information about the instance of the considered problem. The traditional approach is actually *qualitative* in the sense that algorithms are designed or impossibility results are proved under the assumption that the nodes are aware of specific parameters, e.g., the size of the network. It was proved that the impact of knowledge concerning the environment is significant in many areas of distributed computing, as witnessed by [8, 20] where a lot of impossibility results and lower bounds are surveyed, many of them depending on whether or not the nodes are provided with partial knowledge of the topology of the network. A *quantitative* approach was recently introduced in [9], in which limitations of local computation can be estimated by establishing tradeoffs between the efficiency of the computation (number of steps, number of messages, etc.) and the amount of information provided to the nodes about their environment, independently of what kind of information they receive.

More precisely, we consider network computing with advice in the following context. A network is modeled as an undirected graph where links represent communication channels between nodes. Nodes of n-node networks have distinct IDs from $\{1, \ldots, n\}$, and communication ports at a node of degree d are labeled by distinct integers from $\{1, \ldots, d\}$. A priori, every node knows only its own ID and the labels of its ports. All additional knowledge available to the nodes of the graph (in particular knowledge concerning the topology and the labels of the rest of the graph) is modeled by an *oracle* providing *advice*. An oracle is a function $\mathcal{O}$ whose arguments are networks. The value $\mathcal{O}(G)$ for a network $G = (V, E)$, called the advice provided by the oracle to this graph, is a function $f : V \to \{0, 1\}^*$. This function assigns a finite binary string to every node v of the graph. Intuitively, the oracle looks at the entire labeled graph with IDs and assigns to every node some information, coded as a string of bits. A node v is *informed* by the oracle if the string $f(v)$ is non-empty. The *size* of the advice given by the oracle to a given graph G is the sum of the lengths of all strings it assigns to nodes. Hence this size is a measure of the amount of information about the graph, available to its nodes. Clearly, the size of advice is not smaller than the number of informed nodes. The objective is to establish tradeoffs between the size of the advice and the computational efficiency of the network.

Specifically, we focus on the distributed graph coloring problem, one of the most challenging problems in network computing for its practical applications, e.g., in radio networks [21], and for its relation with many other problems such as maximal independent set (MIS) [16, 26] and symmetry breaking [13]. Initially, each node knows its ID from $\{1, \ldots, n\}$. The c-coloring problem requires each node to compute a color in $\{1, \ldots, c\}$, under the constraint that any two adjacent nodes have different colors. Computation proceeds in rounds following Linial's model defined in [18] (a.k.a., $\mathcal{LOCAL}$ model [28]). In each round, a node sends a message to each of its neighbors, receives messages from each of its neighbors, and performs some local computations. The $\mathcal{LOCAL}$ model does not put any limit on the message size and any restrictions on local computations because it is designed to estimate limitations of locality. The complexity of c-coloring a graph

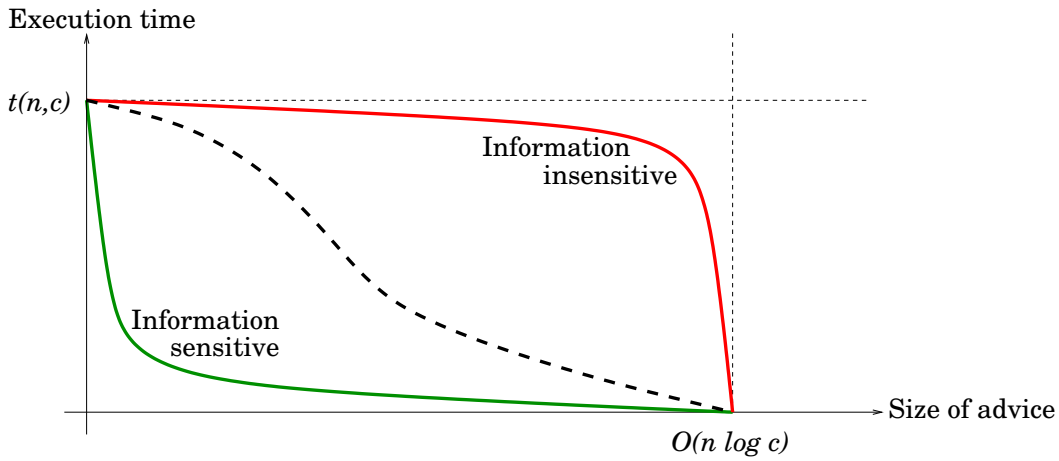

Figure 1: Tradeoff between the execution time and the size of advice.

G is measured by the number of rounds required to compute a proper c-coloring. There is an obvious relation between the complexity of c-coloring and the maximum distance between two nodes that exchange information during the computation.

Coloring graphs using advice provided by an oracle $\mathcal{O}$ consists in designing an algorithm that is *unaware* of the graph G at hand but colors it distributively, as long as every node v of the graph G is provided with the string of bits $f(v)$, where $f = \mathcal{O}(G)$. Trivially, an advice of size $O(n \log c)$ bits that provides the appropriate color to each node yields a coloring algorithm working in 0 rounds. On the other hand, an advice of size 0, i.e., providing no information, yields an algorithm running in $t(n, c)$ rounds where $t(n, c)$ is the complexity of the coloring problem in the usual distributed setting (i.e., with no advice).

The main objective of studying network computations with advice is to establish tradeoffs between these two extreme cases. Different forms of tradeoffs are illustrated in Figure 1. This figure plots the execution time as a function of the size of advice (i.e., the amount of information given to the nodes). The execution time decreases as the size of advice increases, like, e.g., the dashed curve. Depending on how quickly the curve drops down enables to roughly classify problems as "sensitive" or "insensitive" to information. A problem is *information sensitive* if few bits of information given to the nodes enable to decrease drastically the execution time. Conversely, a problem is *information insensitive*

if the oracle must give a lot of information to the nodes for the execution time to decrease significantly. In this paper, we study the information sensitivity of graph coloring.

1.1 Our results

To study the information sensitivity of graph coloring, we focus on lower bounds on the size of advice necessary for fast distributed coloring of cycles and trees, two important cases analyzed in depth by Linial in his seminal paper [18] (cf. also [12]).

We show that coloring a cycle is information insensitive. Precisely, we show that, for any constant k, $\Omega(n/\log^{(k)} n)$ bits of advice are needed in order to beat the $\Theta(\log^* n)$ time of 3-coloring a cycle, where $\log^{(k)} n$ denotes k iterations of $\log n$. This shows a huge gap between 3-coloring in time $\Theta(\log^* n)$ and 3-coloring below this time: while the first can be done without any advice [6], the second requires almost as much information as if colors were explicitly assigned to nodes (which would take $O(n)$ bits).

The result for cycles easily extends to oriented trees (i.e., rooted trees in which every node in the tree knows its parent in the tree), proving that, for any constant k, $\Omega(n/\log^{(k)} n)$ bits of advice are needed in order to beat the $O(\log^* n)$ time of 3-coloring an oriented tree [12]. Coloring an oriented tree is thus also information insensitive.

The power of orienting a tree (i.e., giving an orientation of its edges toward a root), from the point of view of distributed coloring, was known since Linial [18] proved that no algorithm can color the d-regular unoriented tree of radius r in time at most $\frac{2}{3}r$ by fewer that $\frac{1}{2}\sqrt{d}$ colors. Hence 3-coloring unoriented trees essentially requires $\Theta(D)$ rounds, where D is the diameter of the tree. Therefore, informing every node of the port leading to its parent in the tree results in decreasing the time of 3-coloring from $\Omega(D)$ to $O(\log^* n)$. We revisit this result using our quantitative approach. Precisely, we aim at computing the amount of advice required to reach the $O(\log^* n)$ time bound. It is known that $O(n \log\log n)$ bits of advice enable to orient a tree (i.e., to select a root, and to give to every node the port number of the edge leading to its parent) with an algorithm working in 0 rounds [5], and $O(n)$ bits of advice enable to orient a tree with

an algorithm working in 1 round [4]. However, 3-coloring a tree in time $\Theta(\log^* n)$ does not necessarily require to orient the tree. Nevertheless, we show that, for any constant k, $\Omega(n/\log^{(k)} n)$ bits of advice are needed in order to 3-color all n-node unoriented trees in time $\Theta(\log^* n)$. Thus, while for oriented trees 3-coloring in time $O(\log^* n)$ can be done without any additional information [12], achieving the same efficiency for arbitrary trees requires almost as much information as if colors were explicitly assigned to nodes.

Finally, both for cycles and trees, even if oriented, we also show that $\Omega(n)$ bits of advice are needed for 3-coloring in constant time (i.e., for 3-coloring to become a locally solvable problem). Thus constant-time coloring requires essentially as much information as if colors were explicitly assigned to nodes. In fact, our lower bounds do not hold only for the total number of bits of advice given to nodes but also for the number of nodes that must be informed (i.e., the number of nodes that are given at least one bit of advice).

Although we formulate our results for the task of 3-coloring, they remain true for coloring with any constant number of colors, by slight technical modification of the proofs.

While our lower bound proofs present different technical challenges in the case of the cycle and that of trees, the underlying idea is similar in both cases. Linial [18] constructed the *neighborhood graph* $N[G]$ of a graph G in order to estimate the time of coloring G using the chromatic number of $N[G]$. Since in our case there is an oracle giving advice to nodes, we have to use a more complex tool in the lower bound argument. We also argue about the chromatic number of a suitably chosen graph H in order to bound coloring time of G. However, in our case, this graph depends on the oracle as well as on the number of communication rounds, and hence on the graph G. This makes it very irregularly structured. We show that, if the number of nodes of G informed by the oracle is not too large, then H has a large chromatic number, and thus forces large coloring time of G. (Equivalently, if G can be colored fast then the advice must be large.) The main difficulty in our argument is to show the existence of a regularly structured subgraph (whose chromatic number can be bounded from below) in the highly irregularly structured graph H.

1.2 Related work

Because of the intrinsic difficulty of computing the chromatic number of a graph in the sequential setting [14], or even to approximate it [3, 7], the distributed computing literature dealing with graph coloring mostly focuses on the $(\Delta+1)$-coloring problem, where Δ denotes the maximum degree of the graph. In fact, the interest expressed for the $(\Delta+1)$-coloring problem is also due to its intriguing relation with the maximal independent set (MIS) problem, already underlined by Linial in [18]. In particular, combining the best known algorithms for MIS [1, 19] with the reduction from $(\Delta+1)$-coloring to MIS by Linial yields a randomized $(\Delta+1)$-coloring algorithm working in expected time $O(\log n)$. Using techniques described in [2] and [27], one can compute a $(\Delta+1)$-coloring (as well as a MIS) of arbitrary graphs in deterministic time $O(n^{1/\sqrt{\log n}})$. For graphs of maximum degree bounded by Δ, $(\Delta+1)$-coloring can be achieved in time $O(\Delta \log n)$ (see [2]). [6] described a PRAM algorithm that can be easily transformed into an algorithm working in the $\mathcal{LOCAL}$ model, computing a 3-coloring of oriented cycles in $O(\log^* n)$ rounds. This bound is tight as proved by Linial [18] (see also [22] for a generalization to randomized algorithms). Similarly, [12] described a 3-coloring of oriented trees working in $O(\log^* n)$ rounds. The $O(\Delta^2)$-coloring algorithm in [18], working in $O(\log^* n)$ rounds, can be easily converted into a $(\Delta+1)$-coloring algorithm working in $O(\Delta^2 + \log^* n)$ rounds, reaching the same complexity as the algorithm in [13]. [17] analyses what can be achieved in one round, and proves that no algorithm based on iterations of the application of a 1-round algorithm can achieve $O(\Delta)$-coloring in less than $\Omega(\Delta/\log^2 \Delta + \log^* n)$ rounds. On the other hand, [17] presents a $(\Delta+1)$-coloring algorithm working in $O(\Delta \log \Delta + \log^* n)$ rounds, thus improving [2, 13, 18]. Recently, the power of orienting the network was also demonstrated in terms of bit complexity in [15].

One can rephrase many recent results of the literature in the framework of advising schemes. For instance, a 0-round algorithm with maximum advice length $O(\log n)$ bits and average advice length $O(\log \log n)$ is described in [4] for computing a spanning tree. It is also easy to extract an $O(1)$-round algorithm for spanning trees with maximum advice

length 2 bits from the proof of the main result in [5]. [11] have designed distributed algorithms with advice for computing minimum spanning trees (MST), and [24] have designed distributed algorithms with advice for solving the graph searching problem, a.k.a. the cops-and-robbers problem. Finally, [10] considers the competitive ratio of the exploration time of a robot unaware of the topology compared to a robot knowing the map of the graph.

2 Coloring cycles with advice

In order to prove the lower bounds listed in Section 1.1 on the size of advice needed for fast 3-coloring of all cycles, we prove the following result.

Theorem 2.1 *Suppose that an oracle $\mathcal{O}$ informs at most q nodes in any n-node cycle. Then the time of 3-coloring of n-node cycles using oracle $\mathcal{O}$ is $\Omega(\log^*(n/q))$. This result holds even if the cycle is oriented, i.e., even if the nodes have a consistent notion of clockwise and counterclockwise directions.*

Before proving Theorem 2.1, we make the following observation. A lower bound argument for a scenario in which an oracle is giving advice as a function of the instance cannot rely on arguments based solely on the existence of large sets of nodes without advice. Indeed, these specific large sets may not need any advice. A lower bound proof must combine arguments demonstrating the existence of large sets receiving no advice, with arguments about the difficulty of coloring these sets. The following example should help understanding this fact.

Example. Let K_2 be the graph consisting in two nodes linked by an edge. If nodes are labeled by IDs in $\{1, ..., 5\}$, then 3-coloring K_2 in zero rounds (i.e., without communication) is impossible. Indeed, for any function f mapping $\{1, ..., 5\}$ to $\{0, 1, 2\}$, there is a pair x and y of distinct IDs such that $f(x) = f(y)$, and thus there is an assignment of the IDs that causes the two neighboring nodes of K_2 to be mapped to the same

color. Now, consider a cycle with five nodes, C_5, with distinct node IDs in $\{1, ..., 5\}$. If an oracle informs at most two nodes, then there are two adjacent nodes of C_5 that receive no information from the oracle. Using the fact that coloring K_2 in zero rounds is impossible, it may seem at a first glance that 3-coloring a cycle of length 5 with an oracle that informs at most 2 nodes is impossible because there is a copy of K_2 in C_5 that receives no advice, and coloring such a K_2 is impossible. There is a flaw in this reasoning because the 2 adjacent nodes that receive no information may actually be easy to color. In fact, 3-coloring C_5 without communication, using an oracle that informs only 2 nodes is possible. Here is the algorithm:

1. Nodes receiving no advice are colored 0 or 1, depending on the parity of their IDs;
2. Nodes receiving advice are colored 2.

And here is the oracle strategy for assigning advices:

1. select two adjacent nodes x and y that have IDs with different parity;
2. let x' (resp., y') be the neighbors of x (resp., y) that is different from y (resp., x);
3. Nodes x' and y' are the two nodes that receive advice.

This algorithm with advice does 3-color the cycle. Indeed, (1) nodes x and y get different colors 0 and 1 because they have different parity; (2) nodes x' and y' get color 2; and (3) the common neighbor z of x' and y' gets a color 0 or 1 depending on its parity.

The above example illustrates the fact that even if there exists a path P of length n/q in C_n that receives no advice, this is not sufficient to conclude that $\Omega(\log^*(n/q))$ rounds are required to 3-color C_n. Indeed, 3-coloring this specific path P may actually be quite easy. We now give a complete proof of Theorem 2.1.

Proof. Recall the definition of the directed graph $B_{t,n}$ from [18]. Let $s = 2t+1 < n-1$. The nodes of the graph are sequences of length s of distinct integers from $\{1, \ldots, n\}$.

Intuitively, node $(x_1, x_2, \ldots, x_s)$ of the graph $B_{t,n}$ represents the information acquired in time t by node x_{t+1} of a labeled directed cycle containing a segment $(x_1, x_2, \ldots, x_s)$. Out-neighbors of node $(x_1, x_2, \ldots, x_s)$ are all nodes $(x_2, x_3, \ldots, x_s, y)$, where $y \neq x_1$. Note that the chromatic number $\chi(B_{t,n})$ is a lower bound on the number of colors with which an n-node cycle may be colored distributively in time t. Thus, by restricting attention to 3-coloring algorithms, this yields a lower bound on the time of 3-coloring.

It was proved in [18] that $\chi(B_{t,n}) \geq \log^{(2t)} n$.

For any set $X \subseteq \{1, \ldots, n\}$ of size $> s+1$, we define $B_{t,n}(X)$ as the subgraph of $B_{t,n}$ induced by all nodes $(x_1, x_2, \ldots, x_s)$ with $x_i \in X$, for all $1 \leq i \leq s$. The graph $B_{t,n}(X)$ is isomorphic to $B_{t,|X|}$. To see why, just sort the elements of X in, say, increasing order, and consider the mapping $\rho : X \to \{1, \ldots, |X|\}$ defined by $\rho(x) = \mathrm{rank}_X(x)$. By extension, ρ induces an isomorphism $\hat{\rho}$ between $B_{t,n}(X)$ and $B_{t,|X|}$, defined by $\hat{\rho}(x_1, x_2, \ldots, x_s) = (\rho(x_1), \rho(x_2), \ldots, \rho(x_s))$.

Fix an oracle $\mathcal{O}$ giving advice to all cycles of length n. Let q be the maximum number of nodes informed by oracle $\mathcal{O}$ in any of these cycles. Without loss of generality we may assume that the number of bits given to any node is not more than needed to code all directed labeled cycles of length n, i.e., $\lceil \log(n-1)! \rceil$. Consider a 3-coloring algorithm for cycles of length n using oracle $\mathcal{O}$ and running in time t. If $t \geq n/(2q) - 1$, we are done. Hence suppose that $t < n/(2q) - 1$ which implies $s < n/q$. We define the directed graph $B_{t,n,\mathcal{O}}$ that will be crucial in our argument. The nodes of the graph are sequences $((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_s, \alpha_s))$, where x_i are distinct integers from $\{1, \ldots, n\}$ and α_i are binary strings of length at most $\lceil \log(n-1)! \rceil$. Intuitively, node $((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_s, \alpha_s))$ represents the total information acquired in time t by node x_{t+1} of a labeled directed cycle containing a segment $(x_1, x_2, \ldots, x_s)$, including labels of nodes at distance at most t and advice given to them by the oracle. There exists a (directed) edge from node $v = ((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_s, \alpha_s))$ to a node w, if $w = ((x_2, \alpha_2), \ldots, (x_s, \alpha_s), (y, \beta))$ and if there exists a labeled directed cycle of length n containing the segment $(x_1, x_2, \ldots, x_s, y)$, such that oracle $\mathcal{O}$ applied to this cycle gives

advice $\alpha_1, \alpha_2, \ldots, \alpha_s, \beta$ to nodes $x_1, x_2, \ldots, x_s, y$, respectively. We will say that the segment $(x_1, x_2, \ldots, x_s, y)$ of such a cycle *induces* this directed edge. Similarly as above, the chromatic number $\chi(B_{t,n,\mathcal{O}})$ is a lower bound on the number of colors with which the cycle may be colored distributively in time t, using oracle $\mathcal{O}$. Note that a coloring algorithm using oracle $\mathcal{O}$ does not need to assign a color to all nodes $((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_s, \alpha_s))$ of $B_{t,n,\mathcal{O}}$. Indeed, it is possible that there is no cycle containing the segment $(x_1, x_2, \ldots, x_s)$, such that oracle $\mathcal{O}$ applied to this cycle gives advice $\alpha_1, \alpha_2, \ldots, \alpha_s$ to nodes $x_1, x_2, \ldots x_s$, respectively. However, by definition, such "non-legitimate" nodes are isolated in the graph $B_{t,n,\mathcal{O}}$ and hence they do not affect its chromatic number.

We will establish a lower bound on the chromatic number of $B_{t,n,\mathcal{O}}$, and then show how to deduce from it a lower bound on the time of 3-coloring with oracle $\mathcal{O}$. To this end it is sufficient to focus on the subgraph $\tilde{B}_{t,n,\mathcal{O}}$ of $B_{t,n,\mathcal{O}}$ induced by the nodes $((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_s, \alpha_s))$, with all α_i being empty strings. By definition, the graph $\tilde{B}_{t,n,\mathcal{O}}$ is isomorphic to a subgraph of $B_{t,n}$ and has the same number of nodes as $B_{t,n}$. By a slight abuse of notation we will identify $\tilde{B}_{t,n,\mathcal{O}}$ with this subgraph of $B_{t,n}$.

Claim 2.1 *For n/q sufficiently large, there exists a set X of size $k = \left\lfloor \left(\frac{n}{q(s+1)}\right)^{1/(s+1)} \right\rfloor$ such that $B_{t,n}(X)$ is a subgraph of $\tilde{B}_{t,n,\mathcal{O}}$.*

We will establish an upper bound on the number of edges from the graph $B_{t,n}$ missing in $\tilde{B}_{t,n,\mathcal{O}}$. This upper bound will allow us to prove that $\tilde{B}_{t,n,\mathcal{O}}$ contains a subgraph $B_{t,n}(X)$, for some set X of size k. Fix a directed labeled cycle of length n. When the oracle $\mathcal{O}$ informs a node of this cycle, exactly $s+1$ of its segments (those containing the node) induce $s+1$ edges in $B_{t,n,\mathcal{O}}$ that are different than in $B_{t,n,\mathcal{O}'}$, where oracle $\mathcal{O}'$ differs from $\mathcal{O}$ by not informing this node. Moreover, these $s+1$ edges in $B_{t,n,\mathcal{O}}$ are outside $\tilde{B}_{t,n,\mathcal{O}}$. For a given cycle, at most $q(s+1)$ of the edges induced by all the n possible segments of the cycle are outside $\tilde{B}_{t,n,\mathcal{O}}$. There are $(n-1)!$ directed labeled cycles of length n. For a given edge e of $B_{t,n}$ not to appear in $\tilde{B}_{t,n,\mathcal{O}}$, each of the $(n-s-1)!$ cycles that induces e in $B_{t,n}$ must not induce e in $\tilde{B}_{t,n,\mathcal{O}}$. That is, the oracle must give advice to each of these

many cycles in the segment corresponding to edge e. Let μ be the number of edges in $B_{t,n}$ that do not appear in $\tilde{B}_{t,n,\mathcal{O}}$. Then

$$\mu \leq \frac{q(s+1)\cdot(n-1)!}{(n-s-1)!} \leq q\cdot(s+1)\cdot n^s.$$

Consider all graphs $B_{t,n}(X)$, for X of size $k > s+1$. Every edge

$$((x_1, x_2, \ldots, x_s), (x_2, \ldots, x_s, x_{s+1}))$$

of $B_{t,n}$ belongs to at most $\binom{n-s-1}{k-s-1}$ such graphs $B_{t,n}(X)$, where all x_i are in X. Thus there exist at most $q\cdot(s+1)\cdot n^s\cdot\binom{n-s-1}{k-s-1}$ graphs $B_{t,n}(X)$, for X of size k, such that at least one of their edges does not appear in $\tilde{B}_{t,n,\mathcal{O}}$. We will now prove that this number of graphs is strictly smaller than the total number $\binom{n}{k}$ of graphs $B_{t,n}(X)$, for X of a suitably chosen size k. Indeed,

$$\frac{\binom{n}{k}}{\binom{n-s-1}{k-s-1}} = \frac{n(n-1)\cdots(n-s)}{k(k-1)\cdots(k-s)} > \left(\frac{n}{k}\right)^{s+1}.$$

Let

$$k = \left\lfloor \left(\frac{n}{q(s+1)}\right)^{1/(s+1)} \right\rfloor.$$

Note that we have $k > s+1$, for n/q sufficiently large. Hence $\left(\frac{n}{k}\right)^{s+1} \geq q\cdot(s+1)\cdot n^s$. Hence there exists a graph $B_{t,n}(X)$ all of whose edges appear in $\tilde{B}_{t,n,\mathcal{O}}$. This proves Claim 2.1.

In view of Claim 2.1, the chromatic number of $B_{t,n,\mathcal{O}}$ can be bounded as follows (for n/q sufficiently large):

$$\chi(B_{t,n,\mathcal{O}}) \geq \log^{(s-1)} k = \log^{(s-1)}\left(\frac{n}{q(s+1)}\right)^{1/(s+1)}.$$

Since t is the running time of a 3-coloring algorithm for cycles of length n using oracle $\mathcal{O}$, we have $\chi(B_{t,n,\mathcal{O}}) \leq 3$, which implies $\log^{(s-1)}\left(\frac{n}{q(s+1)}\right)^{1/(s+1)} \leq 3$. In order to finish the argument, it is enough to prove that $s \geq \frac{1}{5}\log^*(n/q)$. Suppose not. Thus $n/q \geq 2^{2^{16}}$.

For such large n/q we have

$$\log \frac{n}{q(s+1)} > \log \frac{n}{q} - \log \log^* \frac{n}{q} \geq \frac{1}{2} \log \frac{n}{q}.$$

Hence

$$\frac{1}{s+1} \log \frac{n}{q(s+1)} > \frac{1}{2(s+1)} \log \frac{n}{q} \geq \frac{1}{2 \log^* \frac{n}{q}} \log \frac{n}{q} \geq \log \log \frac{n}{q}.$$

This implies

$$\left(\frac{n}{q(s+1)} \right)^{1/(s+1)} > \log \frac{n}{q},$$

and

$$3 \geq \log^{(s-1)} \left(\frac{n}{q(s+1)} \right)^{1/(s+1)} > \log^{(s)} \frac{n}{q}.$$

Thus $s \geq \log^* \frac{n}{q} - 2$, which contradicts the assumption $s < \frac{1}{5} \log^*(n/q)$. ■

Theorem 2.1 has several interesting consequences. The following corollary proves that transforming the 3-coloring problem into a locally solvable problem (in the sense of [23]) essentially requires to give the solution to the nodes.

Corollary 2.1 *Any distributed algorithm that produces a 3-coloring of all cycles of length n in constant time requires advice for $\Omega(n)$ nodes.*

The next corollary proves that 3-coloring of cycles is information insensitive.

Corollary 2.2 *Any distributed algorithm that produces a 3-coloring of all cycles of length n in time $o(\log^* n)$ requires advice for $\Omega(n/\log^{(k)} n)$ nodes, for any constant k.*

3 Coloring trees with advice

Theorem 2.1 concerning cycles has an interesting consequence concerning trees, that proves that 3-coloring is information insensitive in *oriented* trees. Recall that a tree is oriented if it is rooted, and every node is aware of which of its incident edges leads to its parent in the tree. If there exists an oracle $\mathcal{O}$ informing at most q nodes in any n-node

oriented tree, and a 3-coloring algorithm $\mathcal{A}$ using $\mathcal{O}$ and working in $t(n)$ rounds, then there exists an oracle $\mathcal{O}'$ informing at most $q+2$ nodes in any n-node oriented cycle, and a 3-coloring algorithm $\mathcal{A}'$ using $\mathcal{O}'$ and working in $t(n)+1$ rounds. $\mathcal{O}'$ picks arbitrarily two neighboring nodes x and y in the cycle. Assume that y is the neighbor of x in the counterclockwise direction. $\mathcal{O}'$ gives the advice (tail) to x, and the advice $(t(n), \text{root})$ to y. The ith node v in the cycle, counting counterclockwise from x, receives from $\mathcal{O}'$ the advice $f(v_i)$ given by $\mathcal{O}$ to the node v_i at distance i from the root of the oriented path P rooted at one of its two extremities, where $f = \mathcal{O}(P)$. $\mathcal{A}'$ proceeds in $t(n)+1$ rounds. During rounds 1 to $t(n)$, $\mathcal{A}'$ simply executes $\mathcal{A}$, for which nodes x and y just act as if they would be respectively the tail and the root of a directed path from x to y. At round $t(n)+1$ of $\mathcal{A}'$, the root node y checks if its color is different from x. If not, it takes a color distinct from the colors if its two neighbors. This simple reduction enables to establish the following corollary of Theorem 2.1 proving that 3-coloring oriented trees is information insensitive.

Corollary 3.1 *Suppose that an oracle $\mathcal{O}$ informs at most q nodes in any n-node oriented tree. Then the time of 3-coloring of n-node oriented trees using oracle $\mathcal{O}$ is $\Omega(\log^*(n/q))$. Thus in particular any distributed algorithm that produces a 3-coloring of all n-node oriented trees in time $o(\log^* n)$ requires advice for $\Omega(n/\log^{(k)} n)$ nodes, for any constant k.*

The main result of this section is a lower bound on the size of advice necessary for fast coloring of all n-node *unoriented* trees. In fact we will show that this bound holds already for the class of all unoriented complete d-regular trees. These are rooted trees $T_{d,r}$ such that each leaf is at distance r from the root, and each internal node (i.e., a node that is not a leaf) has degree d. Thus the root has d children, and all other internal nodes have $d-1$ children. It should be stressed that the notion of root and children is brought up only to facilitate the definition. From the point of view of nodes, the tree is not rooted (a node does not have information which neighbor is its parent).

Theorem 3.1 *Fix $d \geq 37$. Any 3-coloring algorithm working in time t for the class of n-node unoriented complete d-regular trees requires advice for at least $\frac{n}{d^{d^{2t}}}$ nodes.*

Proof. Fix $d \geq 37$, $t > 0$, and $r > 2t + 3$. Consider any node v of the tree $T_{d,r}$ at distance at least $t+1$ from all leaves. The number of nodes at distance at most t from v will be denoted by $\alpha(t)$. We have $\alpha(t) = d \cdot \sum_{i=0}^{t-1}(d-1)^i \leq 2(d-1)^t - 1$. Consider an edge e of the tree $T_{d,r}$ whose both extremities are at distance at least $t+1$ from all leaves. The subtree induced by the set of nodes at distance at most t from one of these extremities will be called the *bow-tie* of $T_{d,r}$ based on edge e, denoted by $\mathrm{BT}(e)$. The number of nodes in this bow-tie will be denoted by $\beta(t)$. We have $\beta(t) = \alpha(t) + 1 + (d-1)^t \leq 3(d-1)^t$.

Consider the tree $T_{d,r}$ with a labeling Φ of nodes and ports. Φ labels all nodes by distinct integers from $\{1, \ldots, n\}$, where $n = 1 + \alpha(r)$, and labels all ports at internal nodes by integers from $\{1, \ldots, d\}$. For any such labeled tree, consider its subtrees of the form $N(v, t, \Phi)$, where t is a positive integer and v is a node of $T_{d,r}$ at distance at least $t+1$ from any leaf of $T_{d,r}$. $N(v, t, \Phi)$ is defined as the labeled subtree of $T_{d,r}$ induced by all nodes at distance at most t from v. Note that if restrictions of labelings Φ and Φ' to the subtree of $T_{d,r}$ induced by all nodes at distance at most t from v are identical, then $N(v, t, \Phi) = N(v, t, \Phi')$.

Consider the following graph, denoted by $G_t(T_{d,r})$. The nodes of $G_t(T_{d,r})$ are all subtrees $N(v, t, \Phi)$ of $T_{d,r}$ for all possible nodes v, and all possible labelings Φ of nodes and ports of $T_{d,r}$. Two nodes $N(v, t, \Phi)$ and $N(v', t, \Phi')$ of $G_t(T_{d,r})$ are adjacent, if and only if there exist two adjacent nodes w and w' in $T_{d,r}$, and a labeling Ψ, such that $N(v, t, \Phi) = N(w, t, \Psi)$ and $N(v', t, \Phi') = N(w', t, \Psi)$.

Note that the graph $G_t(T_{d,r})$ is a subgraph of the t-neighborhood graph of $T_{d,r}$, defined in [18]. Moreover, it follows from [18] that the chromatic number $\chi(G_t(T_{d,r}))$ is a lower bound on the number of colors with which the tree $T_{d,r}$ may be colored distributively in time t, and that $\chi(G_t(T_{d,r})) \geq \frac{1}{2}\sqrt{d}$, if $t < 2r/3$. Also, for any set $X \subseteq \{1, \ldots, n\}$, we define the graph $G(X)$ as the subgraph of $G_t(T_{d,r})$ induced by nodes with labels from the set X. For $|X| = 1 + \alpha(s)$, for some positive integer $s \leq r$, the graph $G(X)$ is isomorphic

to $G_t(T_{d,s})$. To see why, observe first that there are precisely $1+\alpha(s)$ nodes in $T_{d,s}$, labeled from 1 to $1+\alpha(s)$. To set the isomorphism between $G(X)$ and $G_t(T_{d,s})$, just sort the elements of X in, say, increasing order, and consider the mapping $\rho : X \to \{1, \ldots, |X|\}$ defined by $\rho(x) = \text{rank}_X(x)$. By extension, ρ induces an isomorphism $\hat{\rho}$ between $G(X)$ and $G_t(T_{d,s})$, defined by $\hat{\rho}(x_1, x_2, \ldots, x_s) = (\rho(x_1), \rho(x_2), \ldots, \rho(x_s))$.

Fix an oracle $\mathcal{O}$ giving advice to all n-node labeled trees $T_{d,r}$. Let q be the maximum number of nodes informed by oracle $\mathcal{O}$ in any of these trees. Without loss of generality we may assume that the number of bits given to any node is not more than needed to code all n-node labeled trees $T_{d,r}$. There are $d^{\alpha(r-1)}$ port labelings of $T_{d,r}$, and for each such labeling there are $n!$ ways to label nodes. Hence the number of bits needed to code these trees is at most $\lceil \log(d^{\alpha(r-1)}n!) \rceil$. Consider a 3-coloring algorithm for n-node labeled trees $T_{d,r}$ using oracle $\mathcal{O}$ and running in time t. We define the following graph $G_{t,\mathcal{O}}(T_{d,r})$. Nodes of this graph are pairs of the form $(N(v,t,\Phi), f)$, where $N(v,t,\Phi)$ is the tree defined above and f is a function from nodes of this tree into the set of binary strings of length at most $\lceil \log(d^{\alpha(r-1)}n!) \rceil$. Intuitively, the value $f(w)$ is the advice given to node w of $N(v,t,\Phi)$ by the oracle, and the entire pair $(N(v,t,\Phi), f)$ represents the total information acquired in time t by node v, including advice given to nodes of $N(v,t,\Phi)$ by the oracle. Edges of the graph $G_{t,\mathcal{O}}(T_{d,r})$ are defined as follows. There is an (undirected) edge between two nodes of $G_{t,\mathcal{O}}(T_{d,r})$ if these nodes are of the form $(N(v,t,\Phi), f)$ and $N(v',t,\Phi), f')$, for some labeling Φ, where v and v' are adjacent in $T_{d,r}$ and for all nodes w of $N(v,t,\Phi)$ and w' of $N(v',t,\Phi)$, the values $f(w)$ and $f'(w')$ are advice strings given to nodes w and w', respectively, by oracle $\mathcal{O}$ for the tree $T_{d,r}$ labeled by Φ. We will say that this edge of $G_{t,\mathcal{O}}(T_{d,r})$ is induced by the bow-tie $\text{BT}(\{v,v'\})$ based on edge $\{v,v'\}$ of the tree $T_{d,r}$ labeled by Φ.

The chromatic number $\chi(G_{t,\mathcal{O}}(T_{d,r}))$ is a lower bound on the number of colors with which the tree $T_{d,r}$ may be colored distributively in time t, using oracle $\mathcal{O}$. Similarly as in the case of the cycle, there may be "non-legitimate" nodes in $G_{t,\mathcal{O}}(T_{d,r})$ but they are isolated and thus do not affect the chromatic number.

In order to establish a lower bound on the chromatic number of $G_{t,\mathcal{O}}(T_{d,r})$, it is sufficient to focus on the subgraph $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$ induced by the nodes $(N(v,t,\Phi),f)$ with f being a function giving the empty string to all nodes. By definition, the graph $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$ is isomorphic to a subgraph of $G_t(T_{d,r})$ and has the same number of nodes as $G_t(T_{d,r})$. Similarly as before we will identify $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$ with this subgraph of $G_t(T_{d,r})$.

Claim 3.1 *Let $\nu(k)$ be the number of sets X of size k, such that the graph $G(X)$ is not a subgraph of $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. Then*

$$\nu(k) \leq \frac{2 \cdot q \cdot n! \cdot d^{4d^t}}{n \cdot \big(n-\beta(t)\big)!} \cdot \binom{n-\beta(t)}{k-\beta(t)}.$$

In order to prove Claim 3.1, consider an edge of $G_{t,\mathcal{O}}(T_{d,r})$. Let λ be the number of labeled trees $T_{d,r}$ that contain a bow-tie B inducing this edge. Let b be the node of B closest to the root of $T_{d,r}$. Consider two cases.

Case 1. b is the root of $T_{d,r}$.

There are $\beta(t)$ ways of choosing node b in the bow-tie B. For each such choice there are $d^{\alpha(r-1)-(\beta(t)-1)}$ ways of fixing port numbers in $T_{d,r}$ because for every internal node other than the root the port leading to its parent has to be chosen and this has already been done for these nodes that appear in the bow-tie B. Finally for each such choice there are $\big(n-\beta(t)\big)!$ ways of labeling all nodes outside B. Hence in Case 1, there are $\beta(t) \cdot d^{\alpha(r-1)-\beta(t)+1} \cdot \big(n-\beta(t)\big)!$ labeled trees $T_{d,r}$ that contain B.

Case 2. b is not the root of $T_{d,r}$.

In this case b must be a leaf of B. The number of leaves of B is $2(d-1)^t$. For any choice of b there are $d^{\alpha(r-1)-\beta(t)}$ ways of fixing the port number leading to the parent, for all internal nodes in $T_{d,r}$ other than the root and outside B. For any such choice there are $d \cdot \sum_{i=0}^{r-(2t+3)}(d-1)^i$ ways of choosing the port numbers on the (unique) path from the root to b (index i corresponds to the distance between the root and node b). Finally, we have to consider again the $\big(n-\beta(t)\big)!$ ways of labeling all nodes outside B. Hence in

Case 2, there are $2(d-1)^t \cdot d \cdot \sum_{i=0}^{r-(2t+3)} (d-1)^i \cdot d^{\alpha(r-1)-\beta(t)} \cdot \big(n-\beta(t)\big)!$ labeled trees $T_{d,r}$ that contain B.

Consequently we have

$$\begin{aligned}
\lambda &= \left(\beta(t) \cdot d^{\alpha(r-1)-\beta(t)+1} + 2(d-1)^t \cdot d \cdot \sum_{i=0}^{r-(2t+3)} (d-1)^i \cdot d^{\alpha(r-1)-\beta(t)}\right) \cdot \big(n-\beta(t)\big)! \\
&= \left(\beta(t) + 2(d-1)^t \cdot \sum_{i=0}^{r-(2t+3)} (d-1)^i\right) d^{\alpha(r-1)-\beta(t)+1} \cdot \big(n-\beta(t)\big)! \\
&\geq 2(d-1)^t \cdot (d-1)^{r-(2t+3)} \cdot d^{\alpha(r-1)-\beta(t)} \cdot \big(n-\beta(t)\big)! \,.
\end{aligned}$$

Fix an n-node labeled tree $T_{d,r}$. When the oracle $\mathcal{O}$ informs a node of this tree, exactly $\alpha(t+1)$ bow-ties (those containing the node) induce $\alpha(t+1)$ edges in $G_{t,\mathcal{O}}(T_{d,r})$ that are different than in $G_{t,\mathcal{O}'}(T_{d,r})$, where oracle $\mathcal{O}'$ differs from $\mathcal{O}$ by not informing this node. Moreover, these $\alpha(t+1)$ edges in $G_{t,\mathcal{O}}(T_{d,r})$ are outside $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. For a given tree, at most $q \cdot (\alpha(t+1)$ of the edges induced by all possible bow-ties are outside $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. There are $d^{\alpha(r-1)} \cdot n!$ n-node labeled trees. For a given edge e of $G_t(T_{d,r})$ not to appear in $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$, each of the λ trees that induces e in $G_t(T_{d,r})$ must not induce e in $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. That is, the oracle must give advice to each of these many trees in the bow-tie corresponding to edge e. Let μ be the number of edges in $G_t(T_{d,r})$ that do not appear in $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. Then, recalling the bounds on λ and $\alpha(t)$,

$$\begin{aligned}
\mu &\leq \frac{q \cdot \alpha(t+1) \cdot d^{\alpha(r-1)} \cdot n!}{2(d-1)^t \cdot (d-1)^{r-(2t+3)} \cdot d^{\alpha(r-1)-\beta(t)} \cdot \big(n-\beta(t)\big)!} \\
&\leq \frac{q \cdot n! \cdot d^{\beta(t)}}{(d-1)^{r-(2t+3)} \cdot \big(n-\beta(t)\big)!} \\
&\leq \frac{2 \cdot q \cdot n! \cdot d^{4d^t}}{n \cdot \big(n-\beta(t)\big)!} \,.
\end{aligned}$$

The last inequality follows from $n \leq (d-1)^r$ and $d^{\beta(t)+(2t+3)} \leq d^{4d^t}$. Consider all graphs $G(X)$, for X of size $k = \alpha(\lfloor \frac{3}{2}t+1 \rfloor)$. Every edge of $G_t(T_{d,r})$ belongs to at most $\binom{n-\beta(t)}{k-\beta(t)}$

such graphs $G(X)$. Thus there exist

$$\nu(k) \leq \frac{2 \cdot q \cdot n! \cdot d^{4d^t}}{n \cdot \big(n-\beta(t)\big)!} \cdot \binom{n-\beta(t)}{k-\beta(t)}$$

sets X of size k, such that the graph $G(X)$ is not a subgraph of $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. This proves Claim 3.1.

Suppose that $\nu(k) < \binom{n}{k}$. Then there exists a set X of size k for which $G(X)$ is a subgraph of $\tilde{G}_{t,\mathcal{O}}(T_{d,r})$. Since $k = \alpha(s)$ for $s > 3t/2$, it follows from [18] that the chromatic number of the graph $G(X)$ (and thus also of the graph $G_{t,\mathcal{O}}(T_{d,r})$) is at least $\frac{1}{2}\sqrt{d}$, which is larger than 3 for $d \geq 37$. This contradicts the fact that we consider a 3-coloring algorithm running in time t. Hence we may assume $\nu(k) \geq \binom{n}{k}$. From Claim 3.1, this implies

$$\frac{2 \cdot q \cdot n! \cdot d^{4d^t}}{n \cdot \big(n-\beta(t)\big)!} \cdot \binom{n-\beta(t)}{k-\beta(t)} \geq \binom{n}{k}$$

and hence the number q of informed nodes satisfies

$$q \geq \frac{n \cdot \big(k-\beta(t)\big)!}{2 \cdot d^{4d^t} \cdot k!} \geq \frac{n}{2 \cdot d^{4d^t} \cdot k^{\beta(t)}} \,.$$

Since $k = \alpha(\lfloor \frac{3}{2}t + 1 \rfloor) \leq d^{\frac{7}{2}t}$ and $\beta(t) \leq 3(d-1)^t$, we have

$$q \geq \frac{n}{2 \cdot d^{4d^t} \cdot d^{\frac{7}{2}t \cdot 3(d-1)^t}} \geq \frac{n}{d^{d^{2t}}}$$

which completes the proof. ■

Remark. By considering trees of a sufficiently large constant degree (instead of just degree $d \geq 37$) we can generalize the above result to the case of c-coloring, for any constant c.

Theorem 3.1 has several interesting consequences. The following corollary proves that lack of cycles does not help in coloring a network since transforming the 3-coloring problem in trees into a locally solvable problem essentially requires, as for cycles, to give

the solution to the nodes.

Corollary 3.2 *Any distributed algorithm that produces a 3-coloring of all n-node trees in constant time requires advice for $\Omega(n)$ nodes.*

The next corollary proves that reaching the $O(\log^* n)$ bound in unoriented trees requires lot of advice. This should be contrasted with the fact that $O(\log^* n)$ is the complexity of 3-coloring of *oriented* trees, without advice.

Corollary 3.3 *Any distributed algorithm that produces a 3-coloring of all n-node unoriented trees in time $O(\log^* n)$ requires advice for $\Omega(n/\log^{(k)} n)$ nodes, for any constant k.*

4 Conclusion

We presented lower bounds on the amount of advice that has to be given to nodes of cycles and of trees in order to produce distributively a fast 3-coloring of these networks. Although our lower bounds are very close to the obvious upper bound $O(n)$, some interesting detailed questions concerning the trade-offs between the size of advice and the time of coloring remain open, even for cycles and trees. In particular, what is the minimum number of bits of advice to produce a 3-coloring of every n-node cycle or tree in a given time $t = o(\log^* n)$? More generally, what is the information sensitivity of coloring arbitrary graphs? For arbitrary graphs, it is natural to consider the maximum degree Δ as a parameter, and seek distributed $(\Delta + 1)$-coloring. It was proved in [17] that a $(\Delta+1)$-coloring can be produced in time $O\Delta \log \Delta + \log^* n)$. What is the minimum number of bits of advice to produce a $(\Delta + 1)$-coloring in time $O(\log^* n)$? And in constant time? We conjecture that for the former task $O(n)$ bits of advice are sufficient, and for the latter $\Omega(n \log \Delta)$ bits of advice are needed. Finally, an intriguing question is whether the notion of oracle can be generalized to randomized algorithms. In particular, it would be interesting to generalized the lower bound in [22] to a context in which advices are given to nodes.

References

[1] N. Alon, L. Babai, and A. Itai. A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem. J. Algorithms 7(4): 567-583 (1986).

[2] B. Awerbuch, A. Goldberg, M. Luby, and S. Plotkin. Network Decomposition and Locality in Distributed Computation. In 30th Symp. on Foundations of Computer Science(FOCS), pp. 364-369, 1989.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free Bits, PCPs, and Nonapproximability – Towards Tight Results. SIAM Journal on Computing 27(3): 804-915 (1998).

[4] R. Cohen, P. Fraigniaud, D. Ilcinkas, A. Korman, D. Peleg. Label-Guided Graph Exploration by a Finite Automaton. In 32nd Int. Colloquium on Automata, Languages and Programming (ICALP), LNCS 3580, pp. 335-346, 2005.

[5] R. Cohen, P. Fraigniaud, D. Ilcinkas, A. Korman, D. Peleg. Labeling Schemes for Tree Representation. In 7th Int. Workshop on Distributed Computing (IWDC), LNCS 3741, pp. 13-24, 2005.

[6] R. Cole and U. Vishkin. Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms. In 18th ACM Symp. on Theory of Computing (STOC), pp. 206-219, 1986.

[7] U. Feige and J. Kilian. Zero Knowledge and the Chromatic Number. J. Comput. Syst. Sci. 57(2):187-199 (1998).

[8] F. Fich and E. Ruppert, Hundreds of impossibility results for distributed computing, Distributed Computing, 16: 121-163 (2003).

[9] P. Fraigniaud, D. Ilcinkas, and A. Pelc. Oracle size: a new measure of difficulty for communication tasks. In 25th ACM Symp. on Principles of Distributed Computing (PODC), pp. 179-187, 2006.

[10] P. Fraigniaud, D. Ilcinkas, and A. Pelc. Tree Exploration with an Oracle. 31st Int. Symp. on Mathematical Foundations of Computer Science (MFCS), LNCS 4162, Springer, pp. 24-37, 2006

[11] P. Fraigniaud, A. Korman, and E. Lebhar. Local MST Computation with Short Advice. In 19th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2007

[12] A. Goldberg and S. Plotkin. Efficient parallel algorithms for $(\Delta + 1)$-coloring and maximal independent set problems. In 19th ACM Symp. on Theory of Computing (STOC), pp. 315-324, 1987.

[13] A. Goldberg, S. Plotkin and G. Shannon. Parallel symmetry-breaking in sparse graphs. In 19th ACM Symp. on Theory of Computing (STOC), pp. 315-324, 1987.

[14] R. Karp. Reducibility Among Combinatorial Problems. In Complexity of Computer Computations, pp. 85-103. 1972.

[15] K. Kothapalli, M. Onus, C. Scheideler, and C. Schindelhauer. Distributed coloring in $O(\sqrt{\log n})$ bit rounds. In 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2006.

[16] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed Locally! In 23th ACM Symp. on Principles of Distributed Computing, (PODC), pp. 300-309, 2004.

[17] F. Kuhn and R. Wattenhofer. On the complexity of distributed graph coloring. In 25th ACM Symp. on Principles of Distributed Computing (PODC), pp. 7-15, 2006.

[18] N. Linial. Locality in distributed graph algorithms. SIAM J. on Computing 21(1): 193-201 (1992).

[19] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. SIAM J. Comput. 15(4): 1036-1053 (1986).

[20] N. Lynch. A hundred impossibility proofs for distributed computing. In 8th ACM Symp. on Principles of Distributed Computing (PODC), pp. 1-28, 1989.

[21] T. Moscibroda and R. Wattenhofer. Coloring unstructured radio networks. In 17th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA), pp. 39-48, 2005.

[22] M. Naor. A Lower Bound on Probabilistic Algorithms for Distributive Ring Coloring SIAM J. Discrete Math. 4(3):409-412 (1991)

[23] M. Naor and L. Stockmeyer. What can be computed locally? In 25th ACM Symposium on Theory of Computing (STOC), pp. 184–193, 1993.

[24] N. Nisse and D. Soguet. Graph searching with advice. In 14th International Colloquium on Structural Information and Communication Complexity (SIROCCO), June 2007.

[25] A. Panconesi and R. Rizzi. Some simple distributed algorithms for sparse networks. Distributed Computing 14: 97-100 (2001).

[26] A. Panconesi and A. Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In 24th ACM Symp. on Theory of Computing (STOC), pp. 581-592, 1992.

[27] A. Panconesi and A. Srinivasan. On the complexity of distributed network decomposition. Journal of Algorithms 20(2): 356-374 (1996).

[28] D. Peleg. Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Mathematics, 2000.