What Makes a Distributed Problem Truly Local?

or: why might "Coloring" just possibly be easier than "MIS"?

Adrian Kosowski

IRIF and Inria Paris

Includes results of work with: Pierre Fraigniaud, Cyril Gavoille, Marc Heinrich, and Marcin Markiewicz

Outline

- What problems do **we** consider local?
  - The LOCAL model
  - MIS and Coloring
  - A Constraint Satisfaction framework
- What problems do **others** consider local?
  - Some insights from QCA and tiling communities
  - Non-signaling and its implications
  - What does this all mean for us?

What problems do we consider local?

The LOCAL model

Assumptions of the LOCAL model

- The distributed system consists of a set of processors V, $|V|=n$.
- The system operates in synchronous rounds, with no faults.
- The system input is encoded as a labeled graph $G=(V,E)$
  - node labels (inputs) are given as $x(v)$, for $v \in V$.
- The result of computations is given through local variables $y(v)$, for $v \in V$.
- *Messages exchanged in each round may have unbounded size.*
- *The computational capabilities of each node are unbounded.*
- As a rule, we will assume that nodes have unique identifiers.

The LOCAL model

Assumptions of the LOCAL model

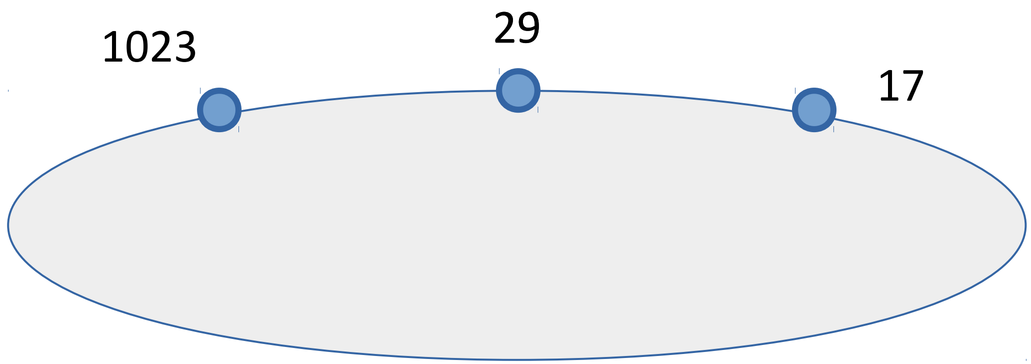
- The distributed system consists of a set of processors V, $|V|=n$.
- The system operates in synchronous rounds, with no faults.
- The system input is encoded as a labeled graph $G=(V,E)$
  - node labels (inputs) are given as $x(v)$, for $v \in V$.
- The result of computations is given through local variables $y(v)$, for $v \in V$.
- *Messages exchanged in each round may have unbounded size.*
- *The computational capabilities of each node are unbounded.*
- As a rule, we will assume that nodes have unique identifiers.

Motivation? Understanding limits of locality in distributed computing.

Sandbox for running simple greedy/distributed algorithms (auctions/pricing, load balancing, LLL,...)

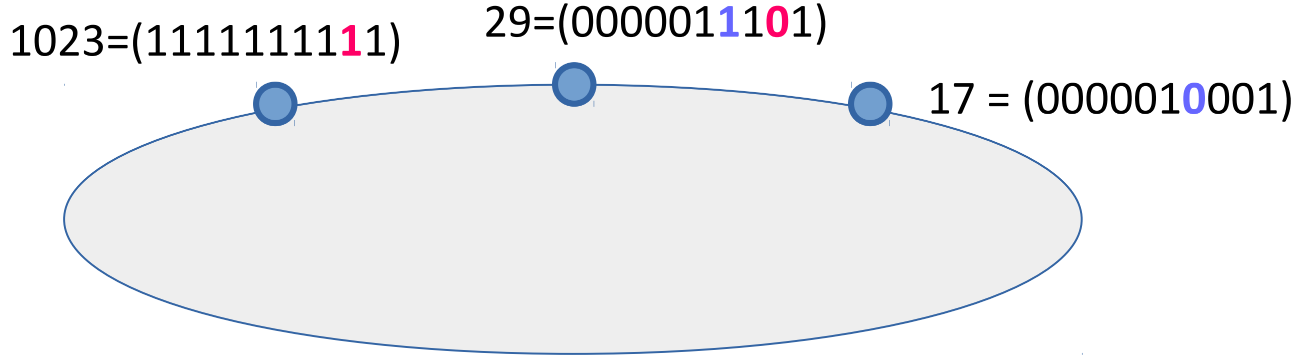
Warm-up: A simple local setting

- The most constrained local setting:
  - *G* has constant maximum degree
  - Algorithms are allowed to run for O(1) rounds
- In this setting, deterministic approaches make the most sense.
- **Example:** recoloring a ring to use fewer colors [Cole-Vishkin 1986]



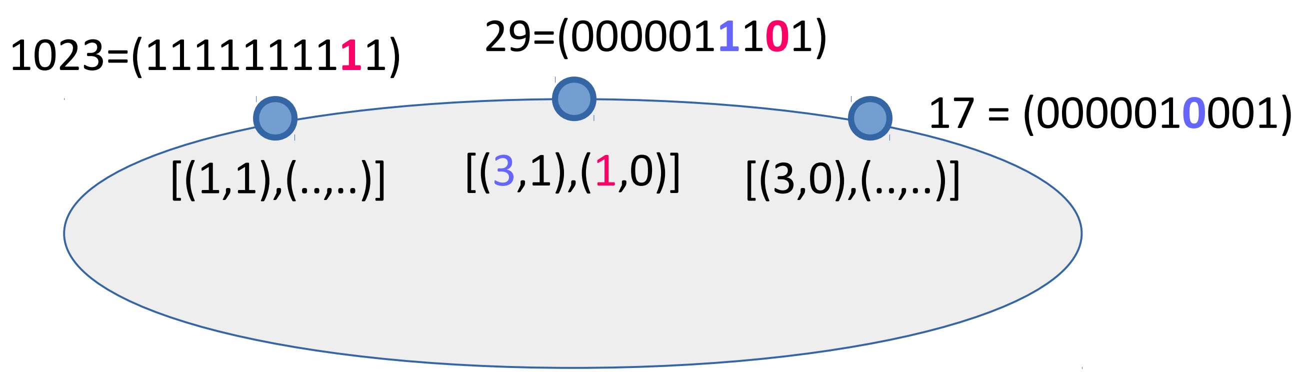
Warm-up: A simple local setting

- The most constrained local setting:
  - *G* has constant maximum degree
  - Algorithms are allowed to run for O(1) rounds
- In this setting, deterministic approaches make the most sense.
- **Example:** recoloring a ring to use fewer colors [Cole-Vishkin 1986]



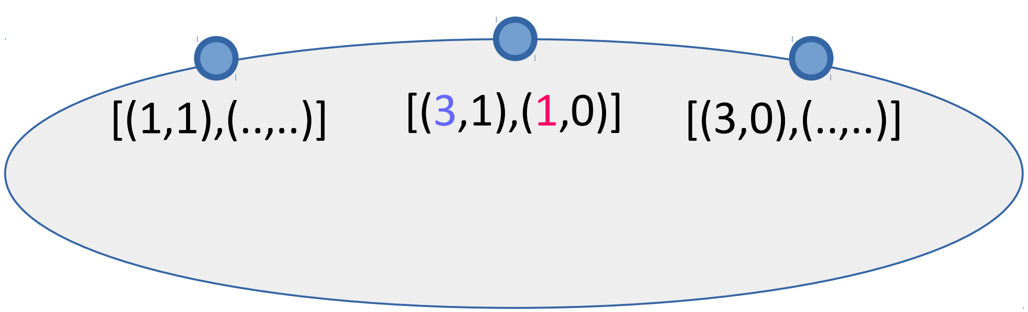
Warm-up: A simple local setting

- The most constrained local setting:
  - *G* has constant maximum degree
  - Algorithms are allowed to run for O(1) rounds
- In this setting, deterministic approaches make the most sense.
- **Example:** recoloring a ring to use fewer colors [Cole-Vishkin 1986]



Warm-up: A simple local setting

- The most constrained local setting:
  - *G* has constant maximum degree
  - Algorithms are allowed to run for O(1) rounds
- In this setting, deterministic approaches make the most sense.
- **Example:** recoloring a ring to use fewer colors [Cole-Vishkin 1986]



Warm-up: A simple local setting

- The most constrained local setting:
  - *G* has constant maximum degree
  - Algorithms are allowed to run for O(1) rounds
- In this setting, deterministic approaches make the most sense.
- **Example:** recoloring a ring to use fewer colors [Cole-Vishkin 1986]
  - We can reduce a ***c*-coloring** to a ***O*(log *c*)-coloring** of a ring in a single communication round.
  - Same approach can be applied for any graph of constant maximum degree.
- What can we compute in O(1) rounds? survey [Suomela, 2013]

Fast distributed algorithms in LOCAL

- More parameters:
  - Number of rounds depends on the number of nodes *n*
  - Number of rounds depends on maximum degree Δ
- Randomization can make a difference
- Considered problems: local validity of a solution can be checked by each node by looking at the states of its neighbors (1-LCA)
- Two basic benchmark problems:
  - **“Easier”**: $(\Delta+1)$-coloring
  - **“Harder”**: Maximal Independent Set (MIS)

LOCAL: Coloring and MIS

	Deterministic	Randomized
(Δ+1)-coloring:	$2^{O(\sqrt{\log n})}$ [PS92] $\tilde{O}(\sqrt{\Delta}) + \log^* n$ [FHK16]	$O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$ [HSS16] $\Omega(\log^* n)$ for $\Delta = 2$ [L92]
MIS:	$2^{O(\sqrt{\log n})}$ [PS92] $O(\Delta) + \log^* n$ [BE09]	$O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ [BEPS12] $\Omega(\sqrt{\log n} / \sqrt{\log \log n})$ [KMW04] for $\Delta = 2^{O(\sqrt{(\log n \ \log \log n)})}$

[Linial 1992] [Panconesi & Srinivasan 1992] [Kuhn, Moscibroda, Wattenhoffer 2004] [Barenboim & Elkin 2009] [Barenboim, Elkin, Pettie, Schneider 2012] [Fraigniaud, Heinrich, K. 2016] [Harris, Schneider, Su 2016]

LOCAL: Coloring and MIS

	Deterministic	Randomized
$(\Delta+1)$-coloring:	$2^{O(\sqrt{\log n})}$ [PS92] $\tilde{O}(\sqrt{\Delta}) + \log^* n$ [FHK16]	$O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log\log n})}$ [HSS16] $\Omega(\log^* n)$ for $\Delta = 2$ [L92]
MIS:	$2^{O(\sqrt{\log n})}$ [PS92] $O(\Delta) + \log^* n$ [BE09]	$O(\log \Delta) + 2^{O(\sqrt{\log\log n})}$ [BEPS12] $\Omega(\sqrt{\log n} / \sqrt{\log\log n})$ [KMW04] for $\Delta = 2^{O(\sqrt{(\log n \log\log n)})}$

Question 1: Is MIS harder than coloring?

- Yes, in the randomized model.

LOCAL: Coloring and MIS

	Deterministic	Randomized
(Δ+1)-coloring:	$2^{O(\sqrt{\log n})}$ [PS92] $\tilde{O}(\sqrt{\Delta}) + \log^* n$ [FHK16]	$O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$ [HSS16] $\Omega(\log^* n)$ for $\Delta = 2$ [L92]
MIS:	$2^{O(\sqrt{\log n})}$ [PS92] $O(\Delta) + \log^* n$ [BE09]	$O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ [BEPS12] $\Omega(\sqrt{\log n} / \sqrt{\log \log n})$ [KMW04] for $\Delta = 2^{O(\sqrt{(\log n \ \log \log n)})}$

Question 1: Is MIS harder than coloring?

- Yes, in the randomized model.
- Possibly, in the deterministic model.

LOCAL: Coloring and MIS

	Deterministic	Randomized
(Δ+1)-coloring:	$2^{O(\sqrt{\log n})}$ [PS92] $\tilde{O}(\sqrt{\Delta}) + \log^* n$ [FHK16]	$O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$ [HSS16] $\Omega(\log^* n)$ for $\Delta = 2$ [L92]
MIS:	$2^{O(\sqrt{\log n})}$ [PS92] $O(\Delta) + \log^* n$ [BE09]	$O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ [BEPS12] $\Omega(\sqrt{\log n} / \sqrt{\log \log n})$ [KMW04] for $\Delta = 2^{O(\sqrt{(\log n \ \log \log n)})}$

Question 2: Does randomization help in the LOCAL model?

(Yes, but this is not apparent from the above table.)

LOCAL: Time of coloring with different pallettes

	Deterministic	Randomized
2 colors: (path) [Linial 1992]	$\Theta(n)$	
O(Δ/log Δ) colors: (triangle-free) [Pettie & Su, 2013]		$O(\log n)$ (roughly)
Δ colors: (tree, Δ>54) [Chang, Kopelowitz, Pettie 2016]	$\Theta(\log_\Delta n)$	$\Theta(\log_\Delta \log n)$
O(Δ²) colors: [Linial 1992]	$\Theta(\log^* n)$	

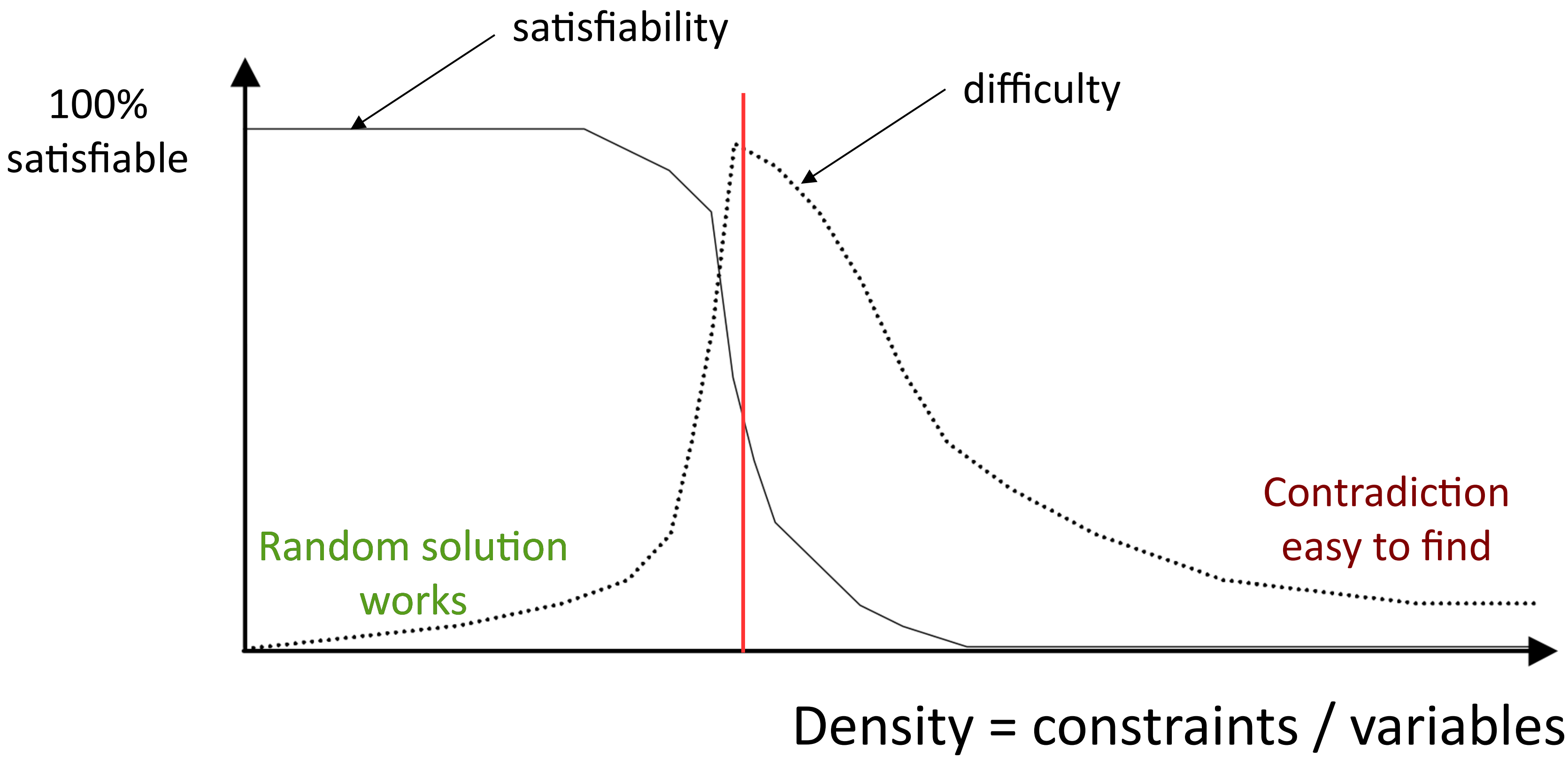
LOCAL: Time of coloring with different pallettes

	Deterministic	Randomized
2 colors: (path) [Linial 1992]	$\Theta(n)$	
O(Δ/log Δ) colors: (triangle-free) [Pettie & Su, 2013]		O(log *n*) (roughly)
Δ colors: (tree, Δ>54) [Chang, Kopelowitz, Pettie 2016]	$\Theta(\log_\Delta n)$	$\Theta(\log_\Delta \log n)$
O(Δ²) colors: [Linial 1992]	$\Theta(\log^* n)$	

Constraint Satisfaction in the LOCAL model

Constraint density vs. hardness

The picture in the centralized world:
Centralized SAT on random instances



Encoding problems through edge constraints

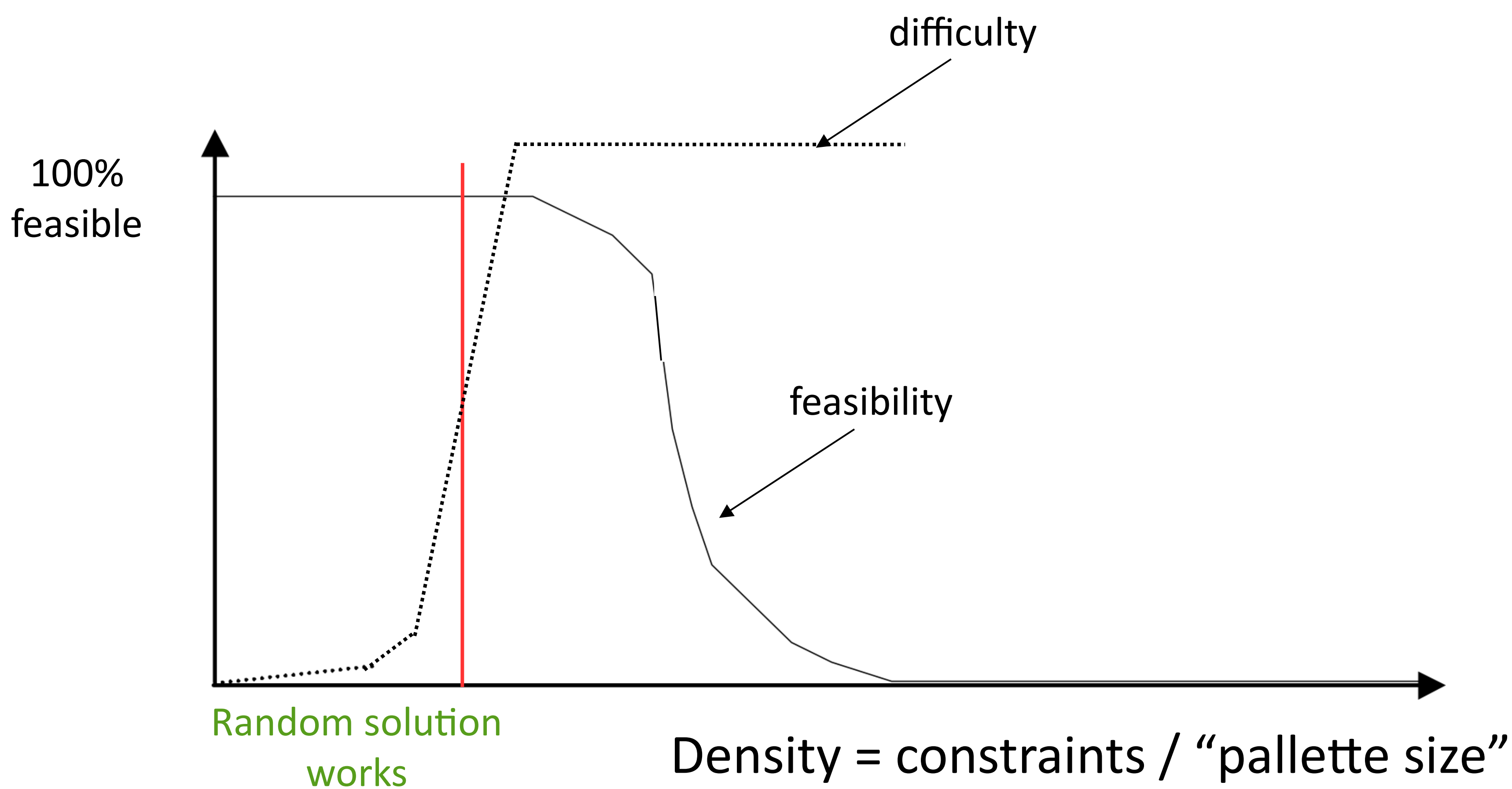
Setting:

- We are given a simple (1-round) algorithm or routine which tries to do something meaningful
  - "obtain a partial coloring of the graph with a given pallette"
  - "extend an IS towards a MIS by including new nodes"
- The routine assigns an output state $y(v) \in L(v)$ to each node v
- **Assumption:** for any pair of neighbors u, v, we can locally tell if the states $y(u)$ and $y(v)$ are compatible just by looking at the edge $\{u, v\}$
  - coloring fails locally if $y(u) = y(v)$.
  - Independent Set fails locally if $y(u) = 1$ and $y(v) = 1$.

How constraining is the problem we are considering?

Constraint density vs. hardness

Our prediction for the LOCAL model



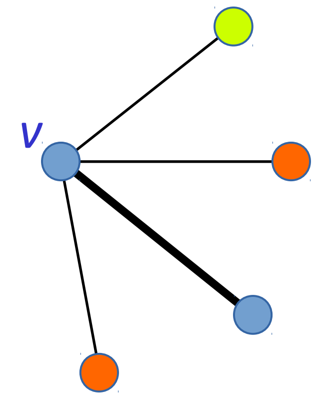
Edge constraint formulation

Probability of local failure:

- Suppose output state $y(v) \in L(v)$ is chosen by each node v *i.u.a.r.*
- What is the max. probability that y(v) violates some local constraint with respect to some neighbour?

Example 1:

- Graph coloring problem with color pallette $L = \{1, 2, \ldots, l\}$.
- Fix color $y(v)$ arbitrarily.
- Pr [v conflicts with an arbitrary neighbor u] = $1/l$.
- Expected number of conflicts of v is at most Δ/l.



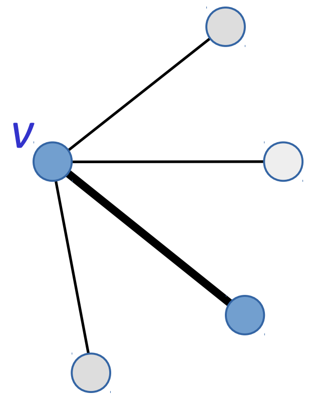
Edge constraint formulation

Probability of local failure:

- Suppose output state $y(v) \in L(v)$ is chosen by each node v *i.u.a.r.*
- What is the max. probability that $y(v)$ violates some local constraint with respect to some neighbour?

Example 2:

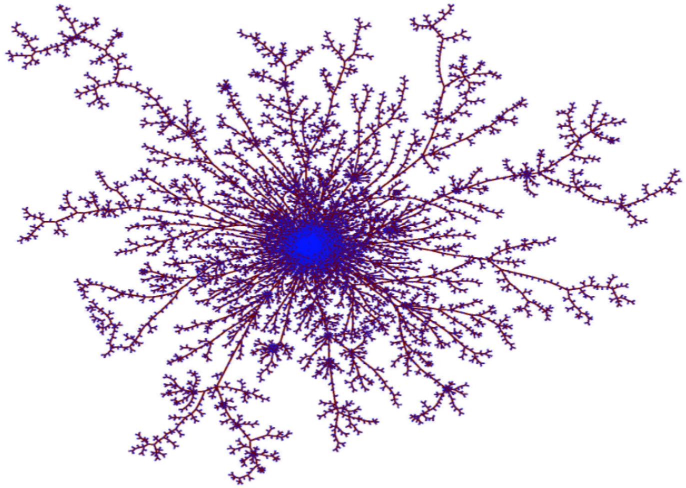
- Independent set problem, $L = \{0_1, 0_2, \ldots, 0_\Delta, 1\}$.
- Suppose $y(v)=1$.
- Pr [v conflicts with an arbitrary neighbor u] = $1/(\Delta+1)$.
- Expected number of conflicts of v is less than 1.



Density thresholds

(1) Threshold of randomized progress: Pr[conflict {u,v}] << 1/Δ

- Expected number of a node's conflicts with its neighbors is less than 1 (regardless of the choice made by the node)
- Basic idea of **shatterring method** [Schneider et al, 2012]
  - Perform random choice of values $y(v)$
  - **Connected components induced by conflicting nodes are small:** Galton-Watson-type process with extinction
  - Solve problem deterministically within these components



- Caveats: random choice not applied to original problem; dependencies.
- Approach separates: randomized LOCAL from deterministic LOCAL; (Δ+1)-coloring from MIS [randomized model].

Density thresholds

(2) Threshold of deterministic progress: Pr[conflict {*u*,*v*}] < ??

- Precise formulation of **conflict coloring**: [Fraigniaud, Heinrich, K. 2016]
  - each node v must pick some color value $y(v) \in L(v)$, where $|L(v)| \geq l$
  - conflicting pairs of color values are known (e.g., globally)
  - each color conflicts with at most d other colors
  - **goal:** choose values $y(v)$ so that there are **no conflicts**, deterministically.

- We work with the ratio $\boldsymbol{d / l}$ *(= conflict degree / list length)*
- Intuition: Pr[**conflict {*u*,*v*}**] $\sim$ $\boldsymbol{d / l}$

Generalizes: vertex coloring, edge coloring, list coloring, precoloring extension, coloring with forbidden color sets,...

Density thresholds

(2) Threshold of deterministic progress: Pr[conflict {*u*,*v*}] < ??

- Precise formulation of **conflict coloring**: [Fraigniaud, Heinrich, K. 2016]
  - each node v must pick some color value $y(v) \in L(v)$, where $|L(v)| \geq l$
  - conflicting pairs of color values are known (e.g., globally)
  - each color conflicts with at most d other colors
  - **goal:** choose values $y(v)$ so that there are **no conflicts**, deterministically.

- We work with the ratio $\boldsymbol{d / l}$ *(= conflict degree / list length)*
- Intuition: Pr[**conflict {*u*,*v*}**] $\sim$ $\boldsymbol{d / l}$

Results:

- $d / l = \tilde{O}(1/\Delta^2)$: deterministic solution in time $\mathbf{\log^*} \boldsymbol{n}$
- $d / l < 1/\Delta$: deterministic solution in time $\tilde{O}(\Delta^{1/2}) + \log^* n$

Easy case: Δ^2-vertex-coloring

Theorem [Linial 1992]. Given a graph G colored with $\boldsymbol{k}$ colors,
it is possible to obtain a coloring of G with $k' = \mathbf{5\Delta^2 \log \boldsymbol{k}}$ colors, in one round.

Proof idea:

– For each vertex v, treat its original color $i \in \{1,\ldots k\}$ as an index i of some set F_i in a special selective set family $\{F_1,\ldots, F_k\}$, $F_i \subseteq \{1,\ldots,k'\}$.

– Sets F_i have the property that for any choice of $j_1,\ldots,j_\Delta \neq i$,

$$F_i \setminus \{F_{j1} \cup \ldots \cup F_{j\Delta}\} \neq \varnothing.$$

– Assuming $j_1,\ldots,j_\Delta$ were the colors of the neighbors of v, one can pick an arbitrary element of set $F_i \setminus \{F_{j1} \cup \ldots \cup F_{j\Delta}\}$ as the new color of v.

From Δ^2-coloring to conflict coloring

- Linial's reduction mechanism gives O($\boldsymbol{\Delta^2}$ **log** $\boldsymbol{\Delta}$)-coloring in log* *n* rounds. (Slight tweak allows us to have O($\boldsymbol{\Delta^2}$)-coloring; going further is hard.)

- O($\boldsymbol{\Delta^2}$ **log** $\boldsymbol{\Delta}$)-coloring can be phrased within the conflict coloring framework with $l = \boldsymbol{\Delta^2}$ **log** $\boldsymbol{\Delta}$, d=1.
  - So, we cope with at least one instance such that $d / l = \tilde{O}(1/\Delta^2)$.
  - **But:** Linial's solution exploits the very special form of the color lists $\{1,\ldots,l\}$.
  - **Not applicable to:** list coloring, precoloring extension (!).

- When adapting the approach to work for any other reasonable coloring problem (e.g., precoloring extension), we encounter technical difficulties.

Conflict coloring simplification mechanism

Conflict coloring techniques

[Fraigniaud, Heinrich, K. 2016]

- **Lemma.** Given a conflict coloring instance $\boldsymbol{P_i}$ with parameters $(\boldsymbol{l_i}, \boldsymbol{d_i})$, there exists a one-round algorithm which for each node computes its input in a new conflict coloring instance $\boldsymbol{P_{i+1}}$ with parameters $(\boldsymbol{l_{i+1}}, \boldsymbol{d_{i+1}})$, such that:
  - A solution to $\boldsymbol{P_{i+1}}$ allows us to solve $\boldsymbol{P_i}$ in one round. (Linial-type argument)
  - The new problem $\boldsymbol{P_{i+1}}$ has an exponentially smaller conflict probability:

$$\boldsymbol{l_{i+1} / d_{i+1}} > 1/\Delta \cdot \mathbf{exp}\,[c/\Delta^2 \cdot \boldsymbol{l_i / d_i}]$$

- **Lemma.** For sufficiently large ratio l/d, a conflict coloring problem whose input is based only on information contained in a relatively small ball around each node, can be solved without communication.

The general message

- **Conflict coloring problems** admitting natural formulations through edge constraints seem to be roughly as hard computationally as the vertex coloring problem with corresponding density.
  - E.g. current best $(\Delta+1)$-list-coloring algorithms as fast as current best $(\Delta+1)$-coloring algorithms.

- **Controling conflicts on edges** is at the heart of the currently best algorithms for deterministic and randomized $(\Delta+1)$-coloring and for randomized MIS.

- One reason why "coloring is easier than MIS" may be that: **edge constraints are easier to handle in the LOCAL model** than vertex constraints.
  - Note: MIS/coloring separation not yet shown for deterministic algorithms.

What problems do *others* consider local?

The LOCAL model is so much fun...

… that other communities have their own versions of it, too :)

- **Statistical physics – bounds on rate of interaction in network models**
  - **Localized Quantum Operator Algebras** [Robinson, Bratteli 1979]
- **Theory of Cellular Automata**
- **Theory of Tilings**

How relevant is their work to what we are doing?

Note: for those of you who prefer CONGEST, the good news is that Physicists have a couple version of that as well.

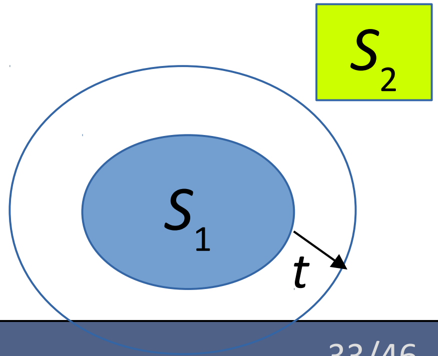
Non-signaling (=causality)

A meta-principle with a formalization for LOCAL

- Given a system evolving in discrete rounds in which information spreads at the rate of 1 unit of distance per round,
- Given a pair of nodes *u*, *v* located at distance t from each other
- The actions of *u* may only be affected by the actions of *v* taken at least *t* steps in the past.

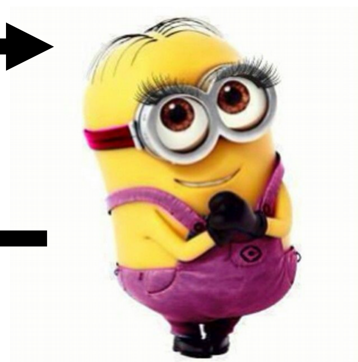
Non-signaling property: Given two subsets of nodes $\boldsymbol{S_1}$, $\boldsymbol{S_2}$ of V such that **dist(S_1,S_2) > t**, then in any t-round LOCAL algorithm, the output of nodes from S_1 must be independent of the input of nodes from S_2.

– Independence is understood in a probabilistic sense.



Example 1: Two-party non-signaling box (XOR)

Alice | Bob

$x_a \in \{0,1\} \rightarrow$   $\leftarrow x_b \in \{0,1\}$

$y_a \in \{0,1\} \leftarrow$ $\rightarrow y_b \in \{0,1\}$

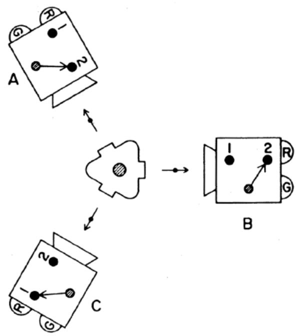
Goal: $y_a \oplus y_b = x_a \wedge x_b$

- Zero-round fictional (oracle-based) protocol for two parties.
- If $x_a \wedge x_b = 0$, the parties output $(y_a, y_b) = (0,0)$ or $(1,1)$, each with Pr=1/2.
- If $x_a \wedge x_b = 1$, the parties output $(y_a, y_b) = (0,1)$ or $(1,0)$, each with Pr=1/2.
- Non-signaling is preserved.
- But: no solution in LOCAL, without communication or access to a box oracle.

Mod 4 problem ("GHZ experiment")

- Graph G is an empty graph with 3 nodes $\{v_1, v_2, v_3\}$, whereas E is empty.
- Each node has an input label $x_i \in \{0,2\}$.
- **Goal:** output labels $y_i \in \{0,1\}$ such that:

$$2(y_1 + y_2 + y_3) \equiv (x_1 + x_2 + x_3) \bmod 4.$$



This problem cannot be solved with Pr > ¾ in LOCAL (in any time).

The problem can be solved under non-signaling, and also by extending LOCAL to include quantum information.

Example 2: the modulo 4 problem

Why is the "Mod 4" problem non-signaling?

Input (x_1, x_2, x_3)	Probability p^i	Output (y_1^i, y_2^i, y_3^i)
(0, 0, 0)	1/4	(0, 0, 0)
	1/4	(0, 1, 1)
	1/4	(1, 0, 1)
	1/4	(1, 1, 0)

Input (x_1, x_2, x_3)	Probability p^i	Output (y_1^i, y_2^i, y_3^i)
(0, 1, 1) or (1, 0, 1) or (1, 1, 0)	1/4	(1, 1, 1)
	1/4	(1, 0, 0)
	1/4	(0, 1, 0)
	1/4	(0, 0, 1)

Lower time bounds under non-signaling

Observation. [Gavoille, K., Markiewicz 2009] In any non-signaling world:

- The MIS problem requires $\Omega(\sqrt{(\log n / \log\log n)})$ rounds to solve [Kuhn, Moscibroda, Wattenhofer, 2004]
- The problem of finding a locally minimal (greedy) coloring of the system graph requires $\Omega(\log n / \log\log n)$ rounds to solve [Gavoille, Klasing, K., Navarra, Kuszner, 2009]
- The problem of finding a spanner with $O(n^{1+1/k})$ edges requires $\Omega(k)$ rounds to solve [Derbel, Gavoille, Peleg, Viennot, 2008; Elkin 2007]

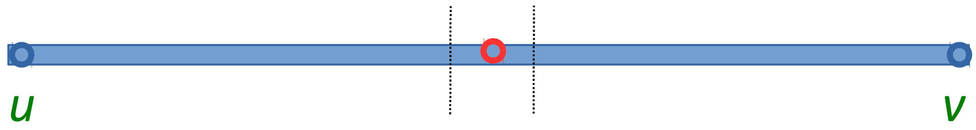
What about Linial's $\Omega(\log^* n)$ bound on $(\Delta+1)$-coloring?

- **Linial's neighbourhood-graph technique relies on many more properties of LOCAL than just non-signaling!**

Example of non-signaling lower bounds

$\lceil n\text{-}2 \rceil / 2$ rounds equired to 2-color the path

- In any non-signaling-world, **$\lceil n\text{-}2 \rceil / 2$** rounds are required.
  - let $t < \lceil n\text{-}2 \rceil / 2$, there will be two extremal nodes *u* and *v* of the path whose views are still disjoint
  - let $S = \{u,v\}$;
  - the color values of *u* and *v* are necessarily the same if these vertices are at an even distance, and odd otherwise
    - there exist corresponding input paths $G^{(1)}$ and $G^{(2)}$ with odd and even distance between *u* and *v*, respectively
  - but the difference cannot be detected based on the local views of *u* and *v.*



Non-signaling c-coloring of the path? (c>2)

Non-signaling coloring of a path – preliminaries

- It's OK to forget about node identifiers – we can use random ID's in the model.
- We identify the c-colored n-node path with a sequence of random variables $(X_1,\ldots,X_n)$, with $X_i \in \{1,2,\ldots,c\}$.
- **Question (t-non-signaling c-coloring)**: [Benjamini, Holroyd, Weiss 2008]
  Can we define the joint distribution of random variables (X_i), so that, for all i:
  - $X_i \neq X_{i+1}$,
  - $(X_1,\ldots,X_i)$ and $(X_{i+t+1},\ldots,X_n)$ are independent?

Non-signaling c-coloring of the path? ($c>2$)

Non-signaling coloring of a path – preliminaries

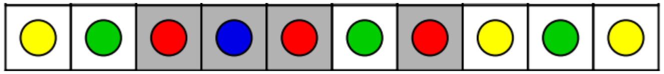
- It's OK to forget about node identifiers – we can use random ID's in the model.
- We identify the c-colored n-node path with a sequence of random variables $(X_1,\ldots,X_n)$, with $X_i \in \{1,2,\ldots,c\}$.
- **Question (t-non-signaling c-coloring)**: [Benjamini, Holroyd, Weiss 2008]
  Can we define the joint distribution of random variables (X_i), so that, for all i:
  - $X_i \neq X_{i+1}$,
  - $(X_1,\ldots,X_i)$ and $(X_{i+t+1},\ldots,X_n)$ are independent?

First attempt:

- Is it enough to pick, for successive i, X_{i+1} from $\{1,2,\ldots,c\} \setminus X_i$, *u.i.a.r.*?
- Not really...

1-non-signaling 4-coloring is possible!

"Surprisingly, this can be done. It is achieved by a family of beautiful and mysterious random colourings that seemingly have no right to exist."
– *A. Holroyd*

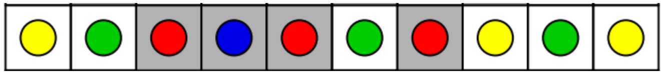


details in [Holroyd & Liggett 2015], and follow-up papers.

- The construction of any *t*-non-signaling coloring, for t=*o*(log* n), cannot have bounded block support, i.e., the random variables must be in some sense defined "globally" over the whole path.

1-non-signaling 4-coloring is possible!

"Surprisingly, this can be done. It is achieved by a family of beautiful and mysterious random colourings that seemingly have no right to exist."
– *A. Holroyd*

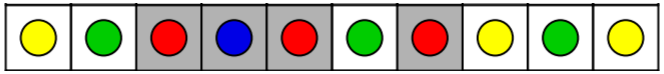


details in [Holroyd & Liggett 2015], and follow-up papers.

- The construction of any *t*-non-signaling coloring, for t=*o*(log* n), cannot have bounded block support, i.e., the random variables must be in some sense defined "globally" over the whole path.
- **1-non-signaling 4-coloring** is obtained using the following algorithm:
  - Nodes arrive on the path according to a random time ordering (enumeration of {1,…,*n*} according to a random permutation).
  - Each node picks a free color which is not used by the closest nodes on its left and right, which have already arrived.
  - The free color picked is fixed deterministically according to a private color preference ordering of each node (we omit the details).

1-non-signaling 4-coloring is possible!

"Surprisingly, this can be done. It is achieved by a family of beautiful and mysterious random colourings that seemingly have no right to exist."
– *A. Holroyd*



details in [Holroyd & Liggett 2015], and follow-up papers.

- The construction of any *t*-non-signaling coloring, for t=*o*(log* n), cannot have bounded block support, i.e., the random variables must be in some sense defined "globally" over the whole path.
- **1-non-signaling 4-coloring** exists.
- **1-non-signaling 3-coloring** does not exist.
  – However, since we can reduce the number of colors in a 4-coloring in the standard way in the LOCAL model, **2-non-signaling 3-coloring** exists.

- **Question**: if (Δ+1)-coloring is not hard because of non-signaling, then why is it hard?

Answer: Localizability

- Non-signaling solutions may, in general, be given by a global computation the system graph and all of its inputs (=a global circuit).
- Not every non-signaling computation can be converted into a combination of circuits acting only on local views.
  - In short, not every non-signaling box can be implemented in LOCAL.

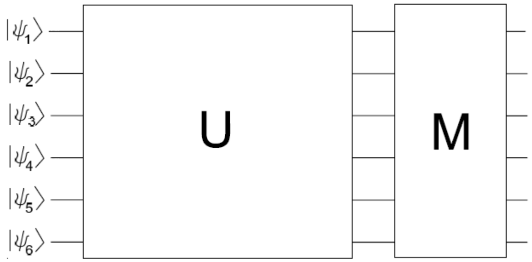
- **Question**: can we impose some **global** property on a non-signaling solution to guarantee that it can be implemented in the LOCAL model?
  - Interesting open problem, though most likely with a negative answer.
  - If we extend the LOCAL model to allow for quantum communication, then an (almost complete) answer exists.

Non signaling + Unitarity => Quantum Localizability

- **Theorem** [Arrighi, Nesme, and Werner 2011].

  If a *t*-non-signaling computation on the system is obtained using a global unitary operator on the quantum state spaces of all nodes of the system, then it can also be implemented by means of a *t*-round algorithm in the LOCAL model using quantum communication channels.

Unitary = preserving structure
(basis, inner product)
of the underlying product Hilbert space
on the state spaces of the nodes.



The message, continued

- In LOCAL, lower bounds on MIS are more fundamental than those for (Δ+1)-coloring:
  - **MIS is hard because of non-signaling** (”speed-of-information”)
  - **(Δ+1)-coloring is only known to be hard because of localizability** of distributed decision**.**
  - Open problem: decide the complexity of (Δ+1)-coloring under non-signaling in general graphs.

- The algebraic structure of the LOCAL model with quantum communication appears much more appealing than for classical communication.
  - It's not clear how much quantum links help us to solve MIS/coloring.

Thank you.