University of St Andrews | FOUNDED 1413

Distributed Weak MIS on Hypergraphs

Duncan Adamson, Will Rosenbaum and Paul G. Spirakis

May 6, 2025

Networks

- Normally, in Computer Science, we write algorithms for one computer to run on its own.

Networks

- Normally, in Computer Science, we write algorithms for one computer to run on its own.
- This works well when there is one (central) authority that can coordinate the computational power, but falls apart if we have several computers without any clear leader.

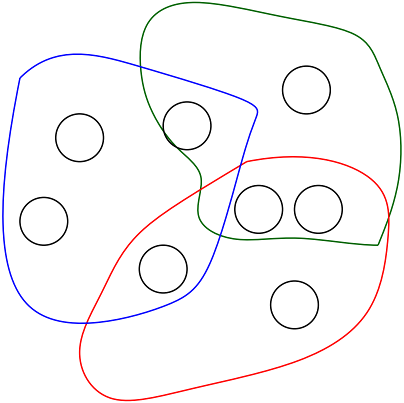
Networks

- Normally, in Computer Science, we write algorithms for one computer to run on its own.
- This works well when there is one (central) authority that can coordinate the computational power, but falls apart if we have several computers without any clear leader.
- In **distributed computing**, we design algorithms to be run on **multiple** computers in a network **simultaneously**, without a clear leader.

Representation, Communication, and Computation

- We represent a distributed network by a **graph**, where each computer is represented by a vertex.
- Two computers can **communicate** if they are connected by an edge in the graph.
- **Computation** is split into a set of discrete rounds. At each round, each computer can perform an arbitrary amount of internal computation, then send a single message to each neighbouring computer.
  - Here, we follow the LOCAL model, where there are no restrictions on the size of messages.
- The type of problems we study here involve each computer labelling itself in some way that is **locally verifiable**, i.e. a computer can look at the labels of its neighbours to determine if its own label is correct.
- A distributed algorithm is "good" if it can reach the correct labelling in the fewest number of communication rounds.

Hypergraphs (informally)

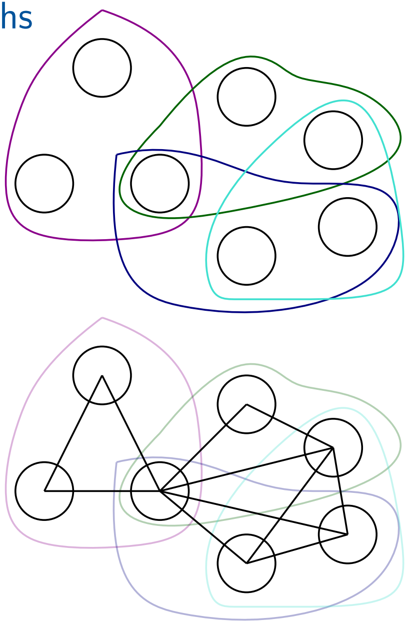


- In many situations, nodes are not connected one-to-one, but rather through shared spaces.
- Social networks are the classic example of an application of a hypergraph.
- Here, each person is represented by a **node** in the graph, and each social network by a **hyperedge**.

Hypergraphs (formally)

- We define a hypergraph H by a set of vertices V and **hyperedges** E.
- Each hyperedge in E is a (non-empty) subset of V.
- The **degree** of a vertex v (written $deg(v)$) is the number of edges in E containing v.
- The **rank** of a hyperedge $e \in E$ is the number of vertices in the edge, denoted $|e|$.
- The **rank** of a hypergraph is $\max_{e \in E} |e|$.
- An r-**rank hypergraph** is a hypergraph where every edge has rank (at most) r.
- A hypergraph is r-**uniform** if **every** edge has rank r.

Communication in Hypergraphs

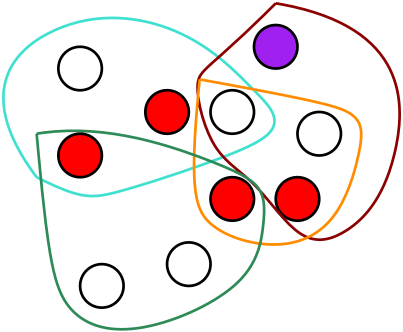


- Here, we use a model analogous to the LOCAL model for regular graphs.
- Therefore, each node can send, at each round, a message of arbitrary length to every vertex sharing a hyperedge with it.
- Thus, if $\exists e \in E$ s.t. $v, u \in e$, v can send to u a message of any length each round.

Symmetries in Networks

- When there are multiple computers asking for access to the same resources, there is a question of who has precedence.
- Solving this on a global level takes time, and a lot of coordination.
- **Symmetry Breaking** algorithms seek to distinguish vertices locally, with some identity, normally by assigning to an independent set or via a colouring.

k-Weak Maximal Independent Sets



With just the red vertices, this is a 2-weak IS, with the purple vertex, it is a 2-weak MIS.

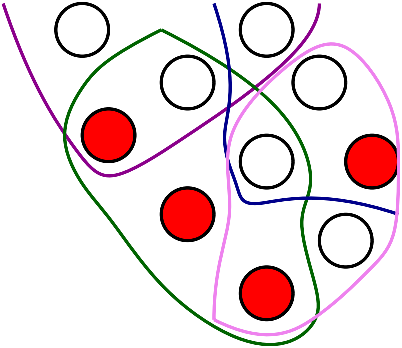
- A k-**weak independent set** (k-weak IS) I in a hypergraph $H = (V, E)$ is a subset of V such that no edge in E contains more than k vertices in I.
  - Formally, $\forall e \in E$, $|e \cap I| \leq k$.
- A k-weak independent set I is **maximal** (a k-weak MIS) if every vertex in V is either in I or in an edge containing k vertices in I.
  - Formally, $\forall v \in V \setminus I$, $\exists e \in E$ s.t. $v \in e$ and $|e \cap I| = k$.

(α, β)-Independent Sets

Generalised Maximal Independent Sets

(α, β)-Independent Sets

- An (α, β)-Independent Set (an (α, β)-IS) is a generalisation of an MIS.
- Informally, a set I is an (α, β)-IS if **no** edge contains **more than** β vertices in I, and **every** vertex that is not in I is in some edge containing at least α vertices.
- Formally, $\forall e \in E$, $|e \cap I| \leq \beta$ and $\forall v \in V \setminus I$, $\exists e \in E$ s.t. $v \in e$ and $|e \cap I| \geq \alpha$.



Both the red vertices and the white vertices form $(1, 3)$-ISs.

Lovasz Local Lemma

Lemma ([1], informal.)

Given a sequence of ("bad") events $B_1, B_2, \ldots, B_k$, each with probability p and dependent on at most d other events, then, if $ep(d+1) \leq 1$, there is a non-zero probability that none of the events occur.

[1] Paul Erdos et al. "Problems and results on 3-chromatic hypergraphs and some related questions". In: *Infinite and finite sets* 10.2 (1975), pp. 609–627.

Formulating with the LLL

- Given a hypergraph $H = (V, E)$ of rank r, our goal is to construct a randomised algorithm where each vertex adds itself to the independent set I with some probability p, such that the bad events, by LLL, have a non-zero probability of none occurring.
- We have two types of bad events:
  - A bad event B_e, for every edge $e \in E$. B_e occurs if β vertices in e add themselves to I.
  - A bad event B_v, for every vertex $v \in V$. B_v occurs if $v \notin I$ and, $\nexists e \in E$ s.t. $v \in e$ and $|e \cap I| \geq \alpha$.
- Letting $\Delta = \max_{v \in V} deg(v)$, observe that every event is independent of, at most $r\Delta + r + \Delta \leq 2r\Delta - 1$ other events.
  - So, let $d = 2r\Delta - 1$.
- Now, we need to work out the probability, p, such that each event may occur.

Formulating with the LLL

Lemma

Suppose H is a hypergraph with rank r and maximum degree Δ. Suppose α, β with $0 < \alpha \leq \beta \leq r - 1$ satisfy

$$\frac{(\beta - \alpha)^2}{\beta + \alpha} \geq 6 \log(16 r \Delta). \tag{1}$$

Then there is an LLL formulation of finding (α, β)-independent sets in H.

Formulating with the LLL

Proof (Sketch)

- Let $\mu = (\alpha + \beta)/2$ and consider the probability measure determined by each vertex v adding itself to I with probability $q = \mu/r$, represented by setting v to 1, otherwise v sets itself to 0.
- Let $X_e = \sum_{v \in e} v$ with $\mathbb{E}(X_e) = \mu$.
- To ensure that neither bad event occurs, we need $X_e \leq \beta$, $\forall e \in E$, and $\forall v \in V \setminus I$, $\exists e \in E$ such that $v \in e$ and $X_e \geq \alpha$.
- Note neither happen if $|X_e - \mu| \leq (\beta - \alpha)/2$.
- As X_e is the sum of independent 0-1 variables with expected value μ, then, for $\delta \in (0, 1)$, by the Chernoff bound we have:

$$Pr(|X_e - \mu| > \delta\mu) \leq 2\exp(-\mu\delta^2/3).$$

Formulating with the LLL

- If $\delta = (\beta - \alpha)/(\beta + \alpha)$, we get:

$$Pr(|X - \mu| > (\beta - \alpha)/2) \leq 2\exp(-(\beta - \alpha)^2/6(\beta + \alpha)).$$

- As this is an upper bound on the probability, p, of an bad event occurring, recalling that $\frac{(\beta-\alpha)^2}{\beta+\alpha} \geq 6\log(16r\Delta)$, we have that:

$$ep(d + 1) < 4 \cdot 2\exp(-(\beta - \alpha)^2/6(\beta + \alpha)) \cdot (2\Delta r) \leq$$

$$16\Delta r \exp(-\ln(16\Delta r)) = 1.$$

- So, by LLL, there is a non-zero probability of this occurring.

$\square$

Solving LLL Problems

Lemma ([2], informal.)

Given a distributed network with n vertices where each vertex represents either a bad event or a variable, and two vertices are connected iff the corresponding events are not independent, then, an assignment of variables such that no bad event occurs can be found in $O(\log^2 n)$ rounds.

[2] Robin A. Moser et al. "A constructive proof of the general lovász local lemma". In: *Journal of the ACM* 57.2 (Jan. 2010), pp. 1–15.

Solving LLL Problems

Corollary

There exists a deterministic distributed algorithm that finds an (α, β)-IS in an hypergraph $H = (V, E)$ with rank r and maximum degree Δ in $O(\log^2 |V|)$ rounds of the LOCAL model for any α, β satisfying $\frac{(\beta-\alpha)^2}{\beta+\alpha} \geq 6\log(16r\Delta)$. In particular, these conditions are satisfied for:

- $\alpha = 1$ *and* $\beta \geq 18\log(16r\Delta)$, *and*
- $\alpha = cr$ *and* $\beta \geq \alpha + c'\sqrt{r\log(r\Delta)}$.

A zero round generalisation

- We can extend this same argument further using concentration bounds to obtain an algorithm finding an (α, β)-IS with high probability.

Corollary

Suppose α and β satisfy

$$\frac{(\beta - \alpha)^2}{\beta + \alpha} \geq 6c \log |V| + 6. \tag{2}$$

Then there exists a zero-round randomized distributed algorithm that finds an (α, β)-IS with probability at least $1 - \frac{1}{|V|^c}$. In particular, this condition is satisfied for:

- $\alpha = 1$ *and* $\beta \geq 18c \log |V| + 18$
- $\alpha = c' r$ *and* $\beta \geq \alpha + c'' \sqrt{r \log |V|}$

(α, β)-IS in $O(\Delta r/(\beta - \alpha))$ rounds

Theorem

There exists a deterministic algorithm constructing an (α, β)-IS in $O(\Delta r/(\beta - \alpha + 1))$ rounds.

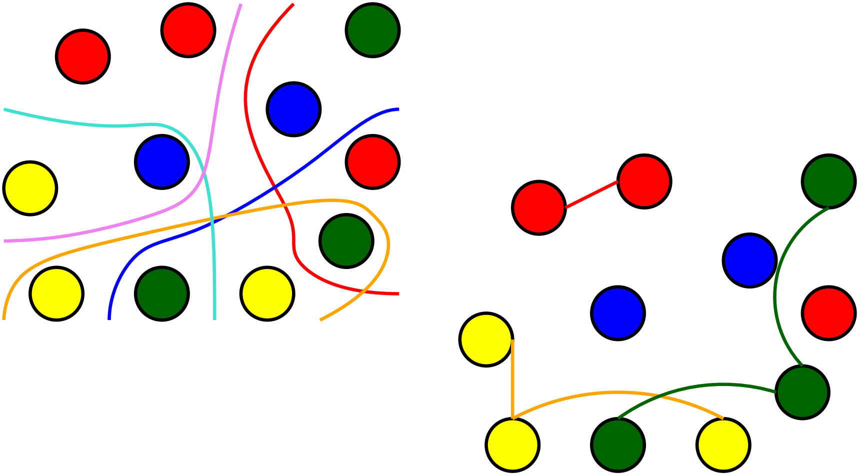
Approach

- At a high-level, the goal is to find a δ-**edge defective vertex colouring**, then iterate through each colour class, activating all vertices with that colour.
- When a vertex in the colour class is active, it can either add itself to the IS, or pass.
- By choosing δ appropriately, we can ensure that the vertices will only add themselves if, in doing so, no edge would end up with more than β colours.

Definition

A vertex colouring is δ-**edge defective** if no edge contains more than δ vertices with the same colour.

Defective Colouring Example

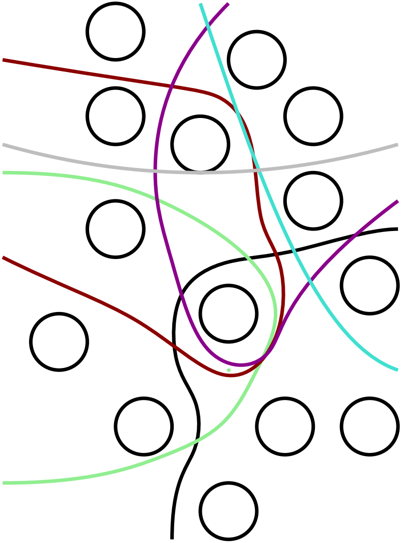


Finding a defective colouring

- Let $\delta = (\beta + 1 - \alpha)$.
- For each edge $e \in E$, we construct δ sets of size (at most) r/δ, $e_1, e_2, \ldots, e_\delta$.
- Let $H' = (V, E')$ be the hypergraph formed by replacing each edge e with the sets $e_1, e_2, \ldots, e_\delta$.
    - Formally, $E' = \bigcup_{e \in E}\{e_1, e_2, \ldots, e_\delta\}$.
- Now, let $G' = (V, E'')$ be the **underlying graph** of H', i.e. the graph formed by replacing each edge in H' with the clique containing the corresponding vertices.
    - Formally, $E'' = \bigcup_{e \in E'}\{(v, u) \mid v, u \in e\}$.
- By finding a colouring on G', we find a δ-edge defective vertex colouring.
- Using state-of-the-art colouring algorithms, this can be done in $O(\Delta r/\delta)$ rounds.

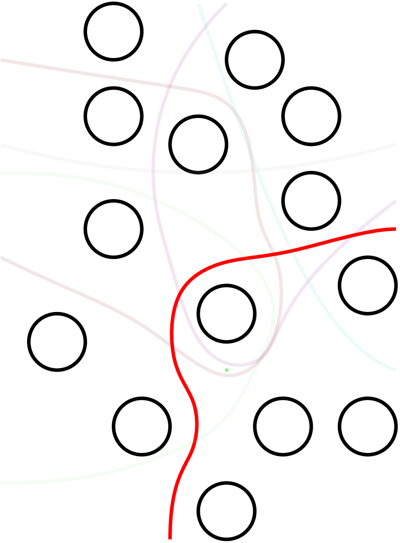
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E}\{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\mathbf{\Psi} \leftarrow (\Delta r/\delta)\text{-}\textsc{color}(G', v)$
5: **Return** $\mathbf{\Psi}$.



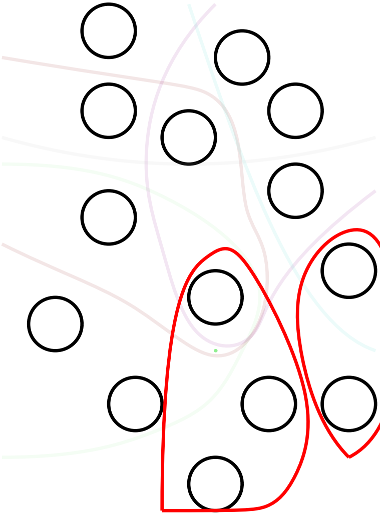
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E} \{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\boldsymbol{\Psi} \leftarrow (\Delta r/\delta)\text{-}\textsc{Color}(G', v)$
5: **Return** $\boldsymbol{\Psi}$.



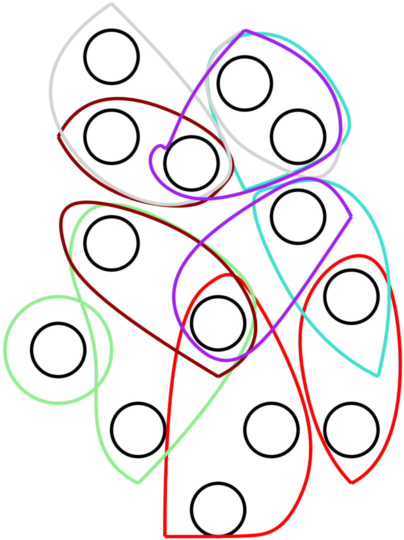
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E} \{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\mathbf{\Psi} \leftarrow (\Delta r/\delta)\text{-}\textsc{Color}(G', v)$
5: **Return** $\mathbf{\Psi}$.



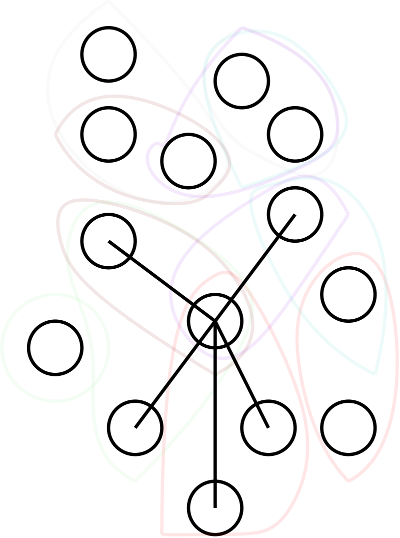
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E} \{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\boldsymbol{\Psi} \leftarrow (\Delta r/\delta)\text{-}\textsc{color}(G', v)$
5: **Return** $\boldsymbol{\Psi}$.



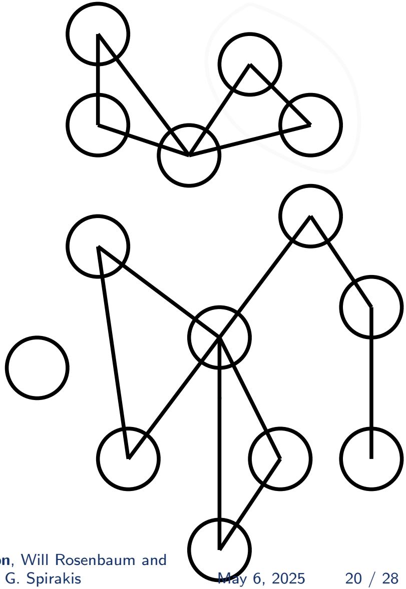
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E} \{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\mathbf{\Psi} \leftarrow (\Delta r/\delta)\text{-}\mathrm{COLOR}(G', v)$
5: **Return** $\mathbf{\Psi}$.



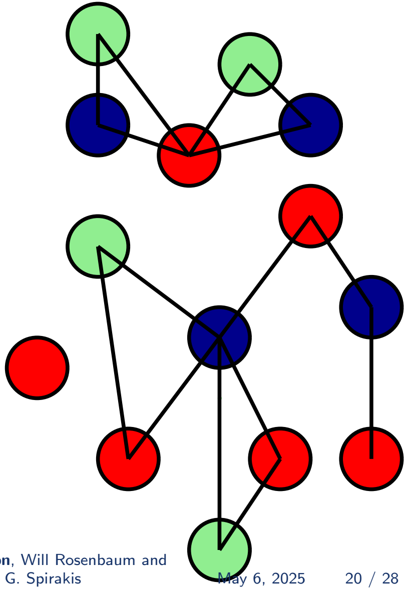
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E}\{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\mathbf{\Psi} \leftarrow (\Delta r/\delta)\text{-}\mathrm{COLOR}(G', v)$
5: **Return $\mathbf{\Psi}$.**



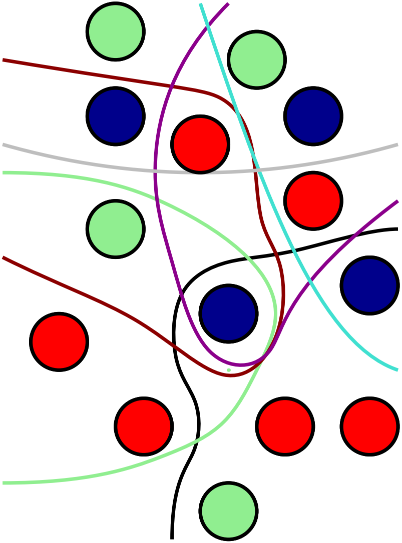
Finding a 2-defective colouring

1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E}\{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\boldsymbol{\Psi} \leftarrow (\Delta r/\delta)\text{-COLOR}(G', v)$
5: **Return $\boldsymbol{\Psi}$.**



Finding a 2-defective colouring

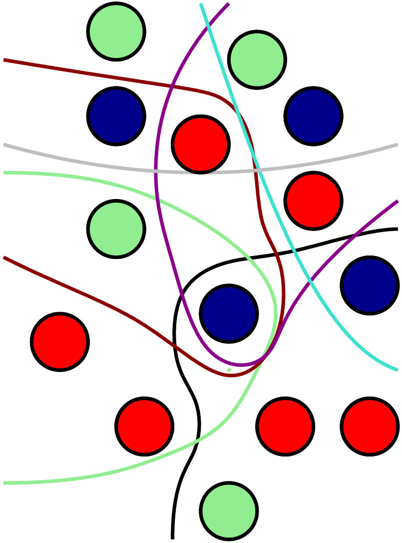
1: Split each edge e into c subsets $e_1, e_2, \ldots, e_\delta$, with v in e_i.
2: $H' \leftarrow (V, \bigcup_{e \in E} \{e_1, e_2, \ldots, e_\delta\})$
3: $G' \leftarrow$ Underlying Graph of H'
4: $\mathbf{\Psi} \leftarrow (\Delta r/\delta)\text{-}\textsc{color}(G', v)$
5: **Return $\mathbf{\Psi}$**.



Forming an (α, β)-IS

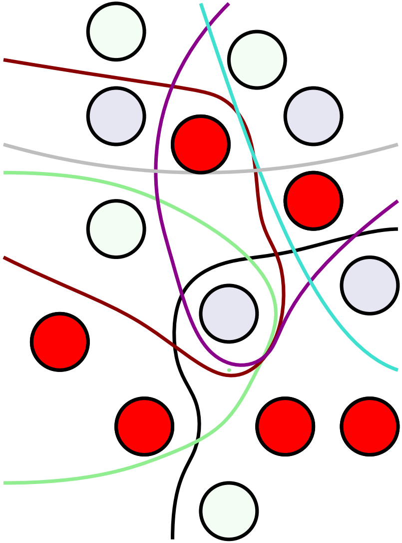
- Let $\Psi(v) : V \mapsto [1, \Delta r/\delta]$ be the δ-edge defective vertex colouring.
- For each $i \in [1, \Delta r/\delta]$ we have a round in which all vertices in the set $C_i = \{v \in V \mid \Psi(v) = i\}$ are active.
- A vertex $v \in C_i$ will, in round i, add itself to the IS iff it is incident to no edge containing more than α vertices in the IS.
- Therefore, after v has been active, it is either in the IS, or is incident to at least one edge with α vertices in the IS.
- As at most δ vertices in any edge e can add themselves to the IS at the same time, and will do so only if e has at most $\alpha - 1$ vertices in the IS, no edge will contain more than $\alpha + \delta - 1 = \alpha + \beta + 1 - \alpha - 1 = \beta$ vertices in the IS.
- Hence, once this is completed, we have an (α, β)-IS.

Forming a (2,4)-IS



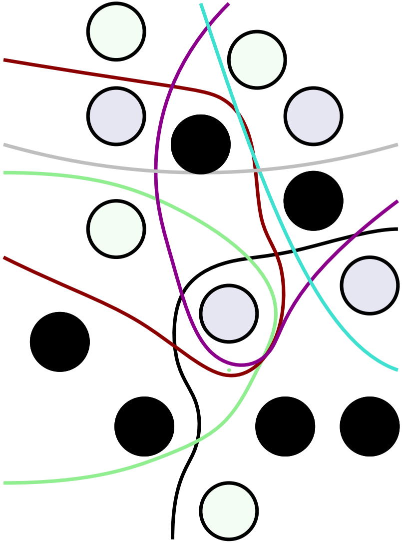
for $i \in [\psi]$ **do**
    if $\Psi(v) = i$ **then**
        if $\forall e \in \{v \in e \mid e \in E\}, |e \cap A| \leq \alpha$ **then**
            Mark as in the independent set
        else
            Mark as not in the independent set
        end if
    end if
end for

Forming a (2,4)-IS



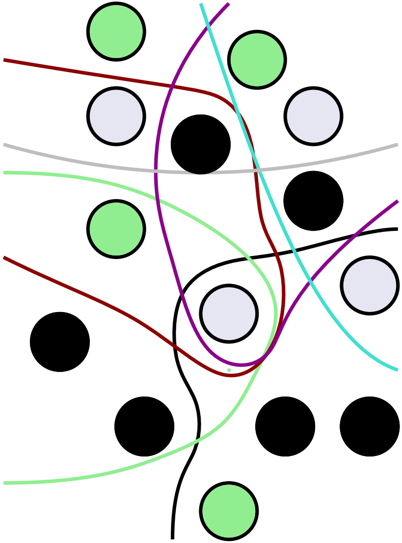
1: **for** $i \in [\psi]$ **do**
2: **if** $\Psi(v) = i$ **then**
3: **if** $\forall e \in \{v \in e \mid e \in E\}, |e \cap A| \leq \alpha$ **then**
4: Mark as in the independent set
5: **else**
6: Mark as not in the independent set
7: **end if**
8: **end if**
9: **end for**

Forming a (2,4)-IS



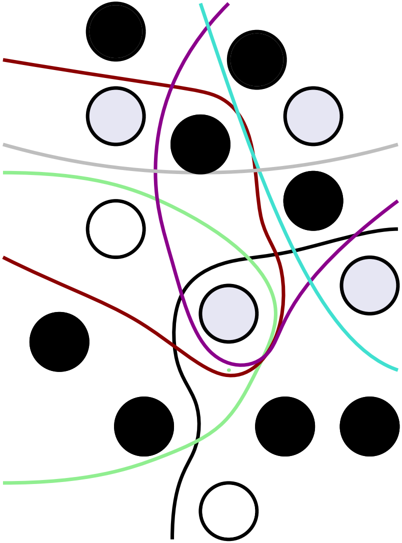
for $i \in [\psi]$ **do**
  if $\Psi(v) = i$ **then**
    if $\forall e \in \{v \in e \mid e \in E\}$, $|e \cap A| \leq \alpha$ **then**
      Mark as in the independent set
    else
      Mark as not in the independent set
    end if
  end if
end for

Forming a (2,4)-IS



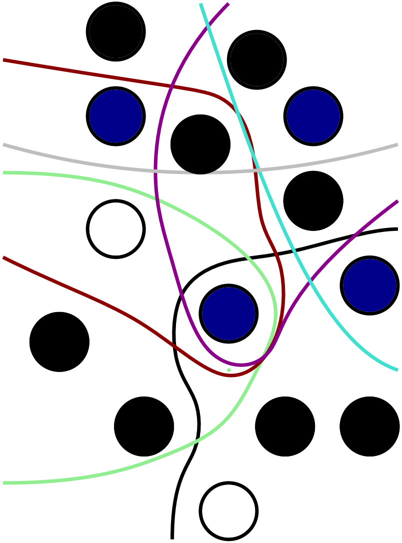
```
for i ∈ [ψ] do
    if Ψ(v) = i then
        if ∀e ∈ {v ∈ e | e ∈ E}, |e ∩ A| ≤ α then
            Mark as in the independent set
        else
            Mark as not in the independent set
        end if
    end if
end for
```

Forming a (2,4)-IS



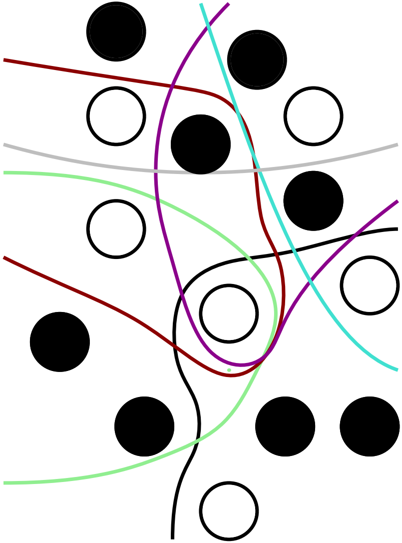
for $i \in [\psi]$ **do**
  if $\Psi(v) = i$ **then**
    if $\forall e \in \{v \in e \mid e \in E\}$, $|e \cap A| \leq \alpha$ **then**
      Mark as in the independent set
    else
      Mark as not in the independent set
    end if
  end if
end for

Forming a (2,4)-IS



for $i \in [\psi]$ **do**
  if $\boldsymbol{\Psi}(v) = i$ **then**
    if $\forall e \in \{v \in e \mid e \in E\}$, $|e \cap A| \leq \alpha$ **then**
      Mark as in the independent set
    else
      Mark as not in the independent set
    end if
  end if
end for

Forming a (2,4)-IS



for $i \in [\psi]$ **do**
  if $\Psi(v) = i$ **then**
    if $\forall e \in \{v \in e \mid e \in E\}, |e \cap A| \leq \alpha$ **then**
      Mark as in the independent set
    else
      Mark as not in the independent set
    end if
  end if
end for

Finding a *k*-Weak MIS

Maximal Independent Sets in Hypergraphs.

Lower Bounds

Theorem

Computing a 1-weak MIS on a r-uniform hypergraphs require $\Omega(r)$ rounds for any even r.

Theorem

Computing a k-weak MIS on a $2k$-uniform hypergraphs require $\Omega(\Delta)$ rounds for any odd k.

Lower Bounds Sketch

- The first lower bound come from the result by Balliu et al.[3] that finding a **maximal matching** requires $\Omega(r\Delta)$ rounds.
- By finding a 1-weak MIS each vertex in the independent set can, in 1-round, add some edge to the matching. Doing this $O(\Delta)$ tmes gives a maximal matching.
- The second case come from the $\Omega(\Delta)$ round lower bound for MIS on graphs by Balliu et al.[4]
- Any graph can simulate a hypergraph by replacing each vertex with a set of k vertices, and each edge with the hyperedge containing the induced sets.
- If more than half of the new vertices are added to the k-weak MIS, then the original vertex is added to the MIS, giving the lower bound of $\Omega(\Delta)$.

[3] Alkida Balliu et al. "Distributed Maximal Matching and Maximal Independent Set on Hypergraphs". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. pp. 2632–2676.

[4] Alkida Balliu et al. "Lower bounds for maximal matchings and maximal

Finding a k-weak MIS

We consider two “trivial” algorithms, based on existing results:

- First, there is an $O(\Delta r)$ round algorithm by finding proper $\Delta r + 1$ colouring on the **underlying graph** $G = (V, \bigcup_{e \in E}\{\ (v, u) \subset e\})$ (i.e. the graph formed by replacing each edge with the clique connecting every vertex in the graph), then iterating through each colour class, with vertices adding themselves to the IS iff it is not already locally maximal.
- Second, using network decompositions, we can also get a $O(\log \Delta r \cdot \log \log n + \log^9 \log n)$ round randomised algorithm, and $O(\log^2 n)$ deterministically.

Finding k-MIS with large k

Theorem

There exists an $O(\Delta^2(r-k)\log r + \Delta \log r \log^* r)$ round algorithm finding a k-weak MIS.

Finding k-MIS with large k

- **Idea:** We use our (α, β)-IS algorithm to iteratively partition the graph into increasingly small instances.
- First, we find a $(k/2, k)$-IS, add all vertices in this set to the final set, and remove any vertex incident to some edge with k vertices in the set.
- We repeat this on the graph induced by removing these vertices, finding $(k/2^{i+1}, k/2^i)$ independent sets for increasing value of i (with slight modification).
- The final time complexity is due to a logarithmic (in r) number of rounds of finding a $(k/2^{i+1}, k/2^i)$ independent set.

Conclusion and Overview

Final Slide

Setting	Lower Bound	Upper Bound
k-weak MIS	$\Omega(\Delta)$, $\Omega(r)$	$O(\Delta r)$, $O(\log^2 n)$, $O(\Delta^2(r-k)\log r + \Delta \log r \log^* r)$
(α, β)-IS	**?**	$O(\Delta r/(\beta + 1 - \alpha))$

- There are still many open questions for independent sets on hypergraphs.
- The most obvious:
  - Can we get a lower bound for finding an (α, β)-IS when $\alpha \neq \beta$?
  - Can we close the gap between the upper and lower bounds? for k-weak MIS?