Distributed Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs

MFCS 2019

Christian Konrad and Victor Zamaraev

University of Bristol

Durham University

26.08.2019

Distributed Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs

Outline:

1. Introduction: LOCAL Model and Vertex Coloring
2. Results: Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs
3. Discussion: Tree Decomposition and Distributed Computing

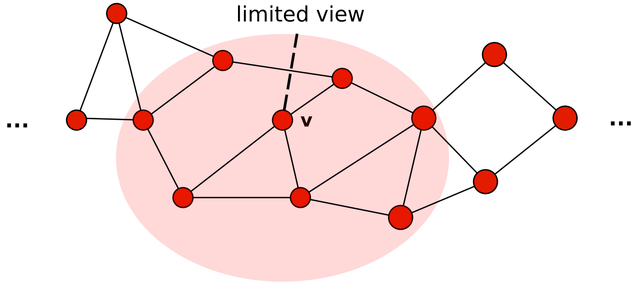
Distributed Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs

Outline:

1. **Introduction:** LOCAL **Model and Vertex Coloring**
2. Results: Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs
3. Discussion: Tree Decomposition and Distributed Computing

The LOCAL Model

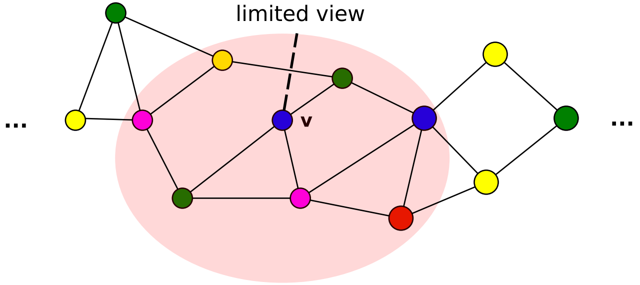
Input: Network $G = (V, E)$, $n = |V|$, max degree Δ



- Nodes host processors and have unique IDs
- Synchronous communication along edges, individual messages of unbounded sizes
- **Running time:** Number of communication rounds
- r rounds $\Leftrightarrow$ compute output from distance-r neighborhood

The LOCAL Model

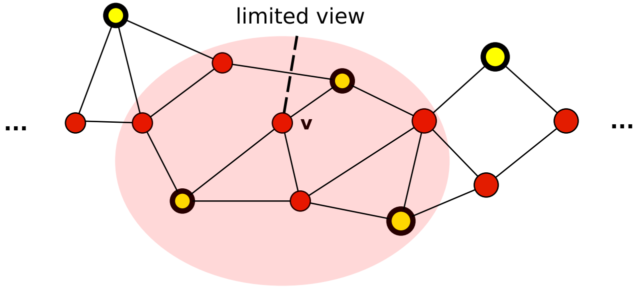
Input: Network $G = (V, E)$, $n = |V|$, max degree Δ



- Nodes host processors and have unique IDs
- Synchronous communication along edges, individual messages of unbounded sizes
- **Running time:** Number of communication rounds
- r rounds $\Leftrightarrow$ compute output from distance-r neighborhood

The LOCAL Model

Input: Network $G = (V, E)$, $n = |V|$, max degree Δ



- Nodes host processors and have unique IDs
- Synchronous communication along edges, individual messages of unbounded sizes
- **Running time:** Number of communication rounds
- r rounds $\Leftrightarrow$ compute output from distance-r neighborhood

Vertex Coloring Problems

$(\Delta + 1)$-**coloring:**

$2^{O(\sqrt{\log \log n})}$ rounds **[Chang, Li, Pettie, 2018]**

Δ-**coloring:** (assuming no $\Delta + 1$ clique, $\Delta \geq 3$)

$O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ rounds **[Ghaffari et al., 2018]**

Fewer colors:

- Arboricity a: $O(a)$-coloring in $O(a \log n)$ rounds **[Barenb., Elkin, 2010]**
- 3-coloring trees, 6-coloring planar graphs, ...

Minimum Vertex Coloring

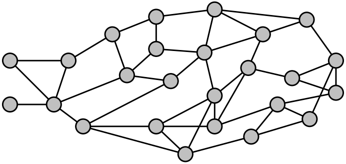
Chromatic number $\chi(G)$: smallest c such that there is a c-coloring

Minimum Vertex Coloring (MVC): find $\chi(G)$-coloring

- NP-hard **[Karp, 1972]**
- Hard to approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

Distributed MVC: Network-decomposition **[Linial, Saks, 1993]**

- Partition vertices $V = V_1 \dot\cup \cdots \dot\cup V_k$ into clusters, $\mathrm{O}(\log^2 n)$ rounds
- Each cluster $G[V_i]$ has diameter $\mathrm{O}(\log n)$
- Cluster graph colored with $\mathrm{O}(\log n)$ colors:



- $\mathrm{O}(\log n)$-approximation in $\mathrm{O}(\log^2 n)$ rounds to MVC
- Poly-time if graph class admits poly-time approximations

Minimum Vertex Coloring

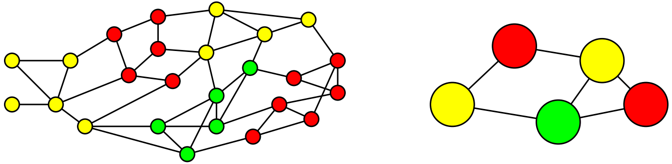
Chromatic number $\chi(G)$**:** smallest c such that there is a c-coloring

Minimum Vertex Coloring (MVC): find $\chi(G)$-coloring

- NP-hard **[Karp, 1972]**
- Hard to approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

Distributed MVC: Network-decomposition **[Linial, Saks, 1993]**

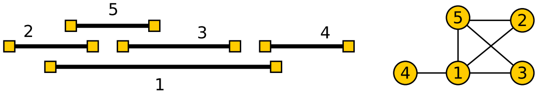
- Partition vertices $V = V_1 \dot\cup \cdots \dot\cup V_k$ into clusters, $\mathrm{O}(\log^2 n)$ rounds
- Each cluster $G[V_i]$ has diameter $\mathrm{O}(\log n)$
- Cluster graph colored with $\mathrm{O}(\log n)$ colors:



- $\mathrm{O}(\log n)$-approximation in $\mathrm{O}(\log^2 n)$ rounds to MVC
- Poly-time if graph class admits poly-time approximations

MVC on Interval Graphs

Interval Graphs: Intersection graph of intervals on the line



[Halldórsson, Konrad, 2014,2017] :

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds (for $\epsilon > \frac{2}{\chi(G)}$)
- Lower Bound: $\Omega(\frac{1}{\epsilon} + \log^* n)$ rounds

Research Questions: Can we...

- Improve approximation factor $\mathrm{O}(\log n)$ on general graphs?
- Get $\mathrm{O}(1)$ or $(1+\epsilon)$-approximations on other graph classes?

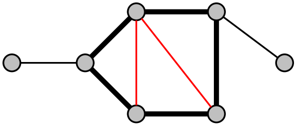
Distributed Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs

Outline:

1. Introduction: LOCAL Model and Vertex Coloring
2. **Results: Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs**
3. Discussion: Tree Decomposition and Distributed Computing

MVC on Chordal Graphs

Chordal Graphs: Every cycle of at least 4 vertices contains a chord:



[Konrad, Zamaraev, MFCS 2019] : **MVC**

- $(1+\epsilon)$-approximation in $O(\frac{1}{\epsilon} \log n)$ rounds (for $\epsilon > \frac{2}{\chi(G)}$)
- *Lower Bound:* $\Omega(\frac{1}{\epsilon} + \log n)$ *rounds (known results)*

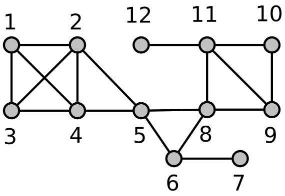
Chordal Graphs vs. Interval Graphs:

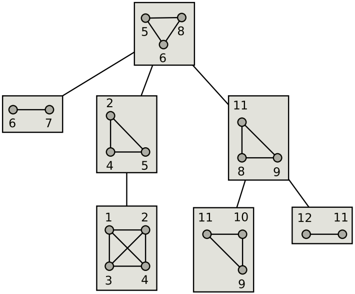
- Chordal graphs contain trees, interval graphs don't
- Linial's tree coloring LB applies: coloring trees with $O(1)$ colors requires $\Omega(\log n)$ rounds **[Linial, 1992]**

Technique: Tree Decomposition

Tree Decomposition of Chordal Graphs

Chordal Graph: Clique Tree





1. Set of bags = set of maximal cliques
2. Bags containing any vertex v induces a subtree

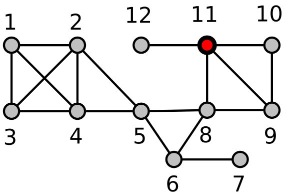
Distributed Processing:

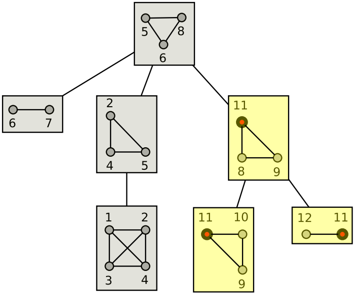
- Nodes compute local view of (global) clique tree
- Locality property: Diameter of each bag is 1

Interval Graph: Clique tree is a path

Tree Decomposition of Chordal Graphs

Chordal Graph: Clique Tree





1. Set of bags = set of maximal cliques
2. Bags containing any vertex v induces a subtree

Distributed Processing:

- Nodes compute local view of (global) clique tree
- Locality property: Diameter of each bag is 1

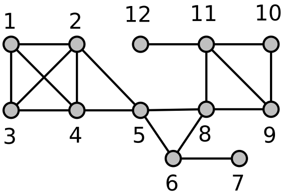
Interval Graph: Clique tree is a path

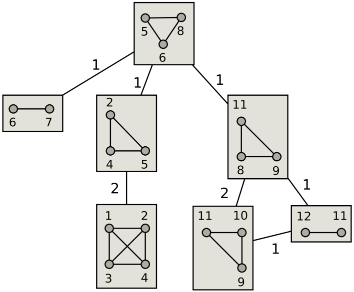
Local View of Clique Tree

Weighted Clique Intersection Graph:

- Let $\mathcal{C}$ be the maximal cliques in chordal graph G
- Let $W_G = (\mathcal{C}, \mathcal{E})$ be the *weighted clique intersection graph* of G, i.e., there is an edge of weight k ($k \geq 1$) between cliques C_i, C_j if $|C_i \cap C_j| = k$

$\mathcal{T}$ is a clique tree $\Leftrightarrow$ $\mathcal{T}$ is a maximum weight spanning tree in W_G



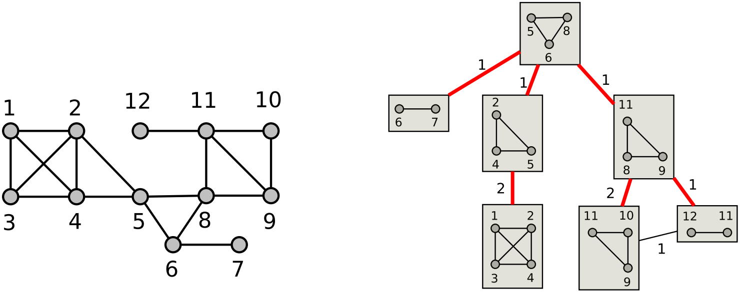


Local View of Clique Tree

Weighted Clique Intersection Graph:

- Let $\mathcal{C}$ be the maximal cliques in chordal graph G
- Let $W_G = (\mathcal{C}, \mathcal{E})$ be the *weighted clique intersection graph* of G, i.e., there is an edge of weight k ($k \geq 1$) between cliques C_i, C_j if $|C_i \cap C_j| = k$

$\mathcal{T}$ is a clique tree $\Leftrightarrow$ $\mathcal{T}$ is a maximum weight spanning tree in W_G

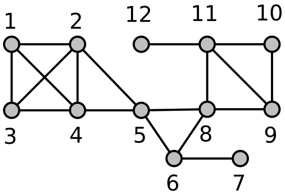


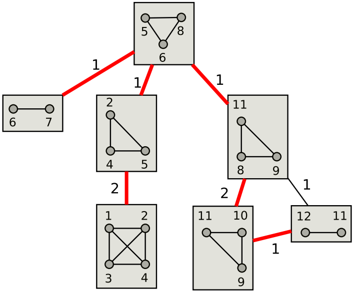
Local View of Clique Tree

Weighted Clique Intersection Graph:

- Let $\mathcal{C}$ be the maximal cliques in chordal graph G
- Let $W_G = (\mathcal{C}, \mathcal{E})$ be the *weighted clique intersection graph* of G, i.e., there is an edge of weight k ($k \geq 1$) between cliques C_i, C_j if $|C_i \cap C_j| = k$

$\mathcal{T}$ is a clique tree $\Leftrightarrow$ $\mathcal{T}$ is a maximum weight spanning tree in W_G





Local View of Clique Tree (2)

Important Property:

Nodes agree on the same maximum weight spanning tree

- Use node identifiers to distinguish between maximum weight spanning trees
- Local View of v: For each $u \in \Gamma^r(v)$: r : desired distance

Algorithm:

1. Compute maximal cliques that u is contained in;
2. Compute maximum weight spanning tree $\mathcal{T}_u$ in clique intersection graph of these cliques;
3. Add $\mathcal{T}_u$ to local view of global spanning tree .

Distributed Algorithm

$(1+\epsilon)$-**approximation Algorithm for MVC:**

1. **Peeling Phase:** Partition vertex set V into layers $V_1, V_2, \ldots, V_{\log n}$ such that $G[V_i]$ is an interval graph in $\mathrm{O}(\frac{1}{\epsilon} \log n)$ rounds
2. **Coloring Phase:** Color each interval graph $G[V_i]$ independently and separately (compute a $(1+\epsilon)$-approximation to MVC) in $\mathrm{O}(\frac{1}{\epsilon} \log^* n)$ rounds using **[Halldórsson, Konrad, 2017]**
3. **Color Correction Phase:** Resolve coloring conflicts between the layers in $\mathrm{O}(\frac{1}{\epsilon} \log n)$ rounds

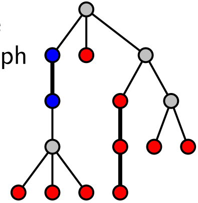
Overall Runtime: $\mathrm{O}(\frac{1}{\epsilon} \log n)$ rounds

Peeling Phase

Definition: Let $\mathcal{T}$ be the clique tree of G

- **Pendant Path:** incident to a leaf, degrees at most 2
- **Internal Path:** not incident to a leaf, degrees at most 2

Lemma: Graph induced by vertices whose subtrees are contained in pendant or internal path is an interval graph



Peeling Process: Let $\mathcal{T}_1 = \mathcal{T}$. For $i = 1 \ldots \log n$ do:

- Remove all pendant paths, and all "long enough" internal paths from $\mathcal{T}_i$.
  (nodes can decide this in $\mathrm{O}(\frac{1}{\epsilon})$ rounds)
- Let V_i be all vertices whose corresponding subtree in $\mathcal{T}_i$ is included in a pendant/long enough internal path
- Let $\mathcal{T}_{i+1}$ be clique tree of residual graph (can be obtained by removing pendant/internal paths)

Lemma: Peeling process terminates after $\log n$ rounds.
(each step number of nodes of degree ≥ 3 halves)

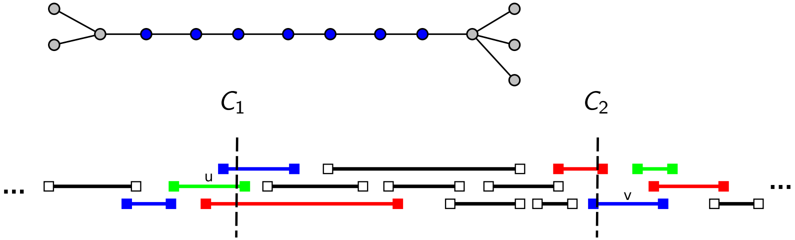
Color Correction Phase

Algorithm:

1. Leave colors of layer $V_{\log n}$ unchanged
2. Correct colors layer by layer from $V_{\log n - 1}$ downwards to layer V_1

Correcting layer i**:**

Layer i corresponds to pendant and internal paths in $\mathcal{T}_i$



Lemma:[Halldórsson, Konrad, 2017] : Only intervals at distance $O(\frac{1}{\epsilon})$ from boundary cliques need to change colors to resolve all coloring conflicts

Algorithm: Summary

Three phases:

1. Peeling Phase: $\log n$ iterations, each requiring $\mathrm{O}(\frac{1}{\epsilon})$ rounds
2. Coloring Phase: $\mathrm{O}(\frac{1}{\epsilon} \log^* n)$
3. Color Correction Phase: $\log n$ iterations, each requiring $\mathrm{O}(\frac{1}{\epsilon})$ rounds

[Konrad, Zamaraev, MFCS 2019] : **MVC**
$(1+\epsilon)$-approximation in chordal graphs in $O(\frac{1}{\epsilon} \log n)$ rounds

Maximum Independent Set

Adapt Technique to Maximum Independent Set: (MaxIS)

[Konrad, Zamaraev, MFCS 2019] : **MaxIS**

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log^* n)$ rounds
- On the way: $(1+\epsilon)$-approx. on interval graphs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds
- Lower Bound: $\Omega(\frac{1}{\epsilon})$ rounds

Intuition:

- **Lemma:** Layers $1 \ldots O(\log(\frac{1}{\epsilon}))$ contain $1(+\epsilon)$-approximate independent set
- Develop maximum independent set algorithm for interval graph
- Apply algorithm on these layers and make sure that not much is lost at intersections

Lower Bound for Maximum Independent Set

Indistinguishability Argument: Consider a path P_n

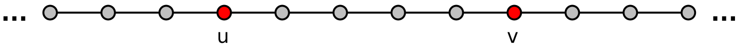
- Assume every vertex is assigned a unique label from $\{1, 2, \ldots, n\}$
- Vertices far enough from boundary have same local views (in expectation over labellings), same probability to be chosen

- Local neighborhoods of u and v are disjoint, therefore whether u and v are chosen is independent

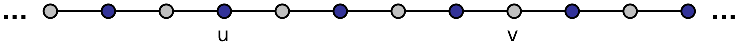
Lower Bound for Maximum Independent Set

Indistinguishability Argument: Consider a path P_n

- Assume every vertex is assigned a unique label from $\{1, 2, \ldots, n\}$
- Vertices far enough from boundary have same local views (in expectation over labellings), same probability to be chosen



- Local neighborhoods of u and v are disjoint, therefore whether u and v are chosen is independent
- If both u and v are chosen then solution is suboptimal

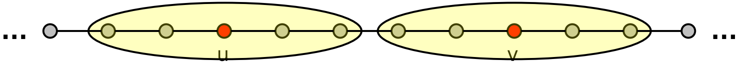


[Konrad, Zamaraev, MFCS 2019] : **MaxIS** Computing a $(1+\epsilon)$-approximation to MaxIS on a path requires $\Omega(\frac{1}{\epsilon})$ rounds

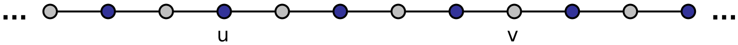
Lower Bound for Maximum Independent Set

Indistinguishability Argument: Consider a path P_n

- Assume every vertex is assigned a unique label from $\{1, 2, \ldots, n\}$
- Vertices far enough from boundary have same local views (in expectation over labellings), same probability to be chosen



- Local neighborhoods of u and v are disjoint, therefore whether u and v are chosen is independent
- If both u and v are chosen then solution is suboptimal



[Konrad, Zamaraev, MFCS 2019] : **MaxIS** Computing a $(1+\epsilon)$-approximation to MaxIS on a path requires $\Omega(\frac{1}{\epsilon})$ rounds

Distributed Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs

Outline:

1. Introduction: LOCAL Model and Vertex Coloring
2. Results: Minimum Vertex Coloring and Maximum Independent Set in Chordal Graphs
3. **Discussion: Tree Decomposition and Distributed Computing**

Tree Decomposition in Distributed Computing

How useful are Tree Decompositions for Distributed Algorithms?

- Only few papers make use of tree decompositions
- Perfect tool for chordal graphs
- Can we handle other graph classes as well using tree decompositions?

Obstacle:

- Tree decomposition of cycle of length k contains bags that are at distance $\Omega(k)$ in the original graph
- Impossible for nodes to obtain coherent local views of global tree decomposition in $o(k)$ rounds

Outlook: Tree Length

- Graph of tree length k has tree decomposition where diameter of every bag is at most k
- Contains k-chordal graphs

Our Results

Minimum Vertex Coloring:

[Konrad, Zamaraev, MFCS 2019] : **MVC**

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log n)$ rounds (for $\epsilon > \frac{2}{\chi(G)}$)
- Lower Bound: $\Omega(\frac{1}{\epsilon} + \log n)$ rounds

Maximum Independent Set:

[Konrad, Zamaraev, MFCS 2019] : **MaxIS**

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log^* n)$ rounds
- On the way: $(1+\epsilon)$-approx. on interval graphs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds
- Lower Bound: $\Omega(\frac{1}{\epsilon})$ rounds

Our Results

Minimum Vertex Coloring:

[Konrad, Zamaraev, MFCS 2019] : **MVC**

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log n)$ rounds (for $\epsilon > \frac{2}{\chi(G)}$)
- Lower Bound: $\Omega(\frac{1}{\epsilon} + \log n)$ rounds

Maximum Independent Set:

[Konrad, Zamaraev, MFCS 2019] : **MaxIS**

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log^* n)$ rounds
- On the way: $(1+\epsilon)$-approx. on interval graphs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds
- Lower Bound: $\Omega(\frac{1}{\epsilon})$ rounds

Thank you very much.