Treewidth Computation and Courcelle's Theorem in the CONGEST Model

Benjamín Jáuregui[1], Jason Li, Pedro Montealegre, Ioan Todinca

IRIF, Université Paris Cité & Universidad de Chile

Algorithmic Meta Theorems

Families of properties, often defined by logical conditions, can be efficiently solved on certain classes of graphs.

Algorithmic Meta Theorems

*Families of **properties, often defined by logical conditions**, can be efficiently solved on certain classes of graphs.*

✦ **We focus in properties expressible in Monadic Second Order Logic.**

Algorithmic Meta Theorems

*Families of **properties, often defined by logical conditions**, can be efficiently solved on **certain classes of graphs.***

- ✦ **We focus in properties expressible in Monadic Second Order Logic.**
- ✦ **We focus in graphs of bounded treewidth**

Algorithmic Meta Theorems

*Families of **properties, often defined by logical conditions**, can be **efficiently solved** on **certain classes of graphs.***

- **We focus in properties expressible in Monadic Second Order Logic**
- **We focus in graphs of bounded treewidth**
- $\tilde{O}(D)$ **rounds**

Monadic Second-Order Logic (MSO)

A property $\mathscr{P}$ is MSO-expressible if it can be defined using quantification over **vertices/edges** and over monadic **sets of vertices/edges.**

2-VERTEX COVER

Given a graph $G = (V, E)$ a vertex cover is a set $U \subseteq V$ such each each $uv \in E$ satisfies that $u \in S$ or $v \in S$

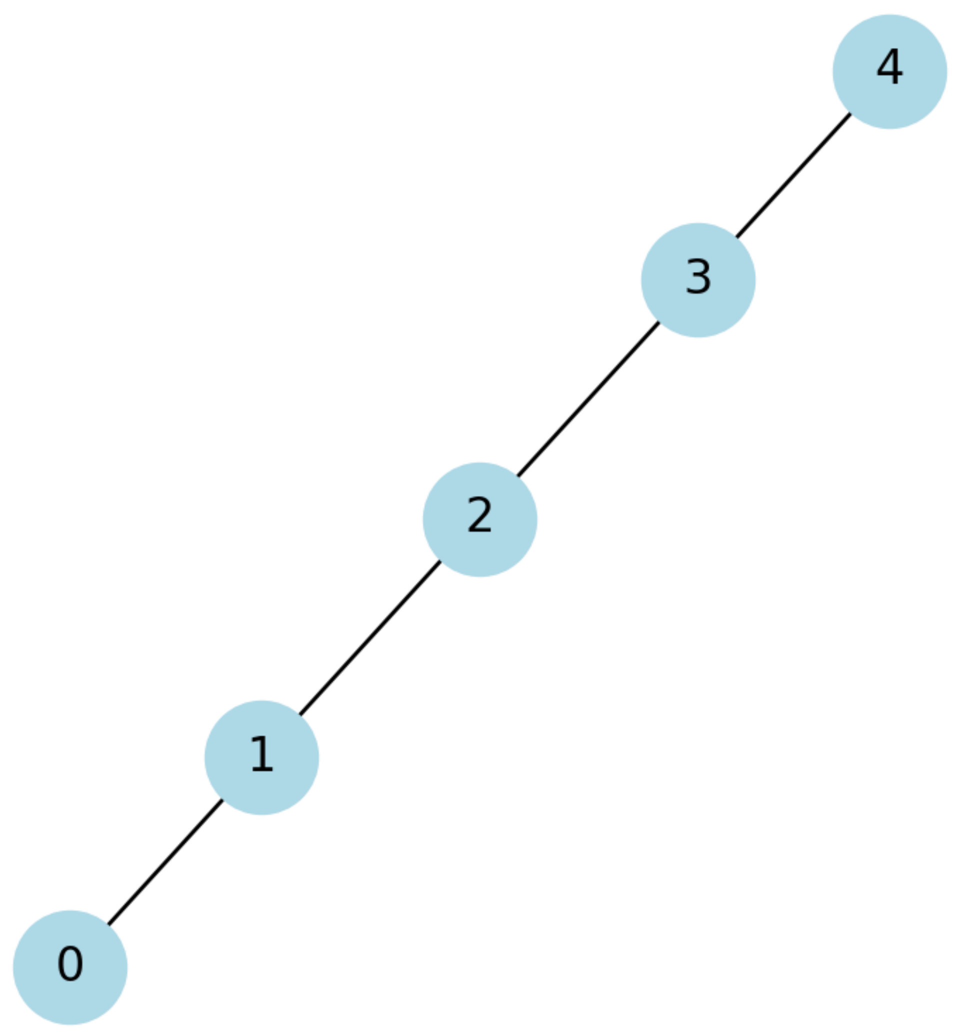
$$\text{(MSO)}\ \exists X \subseteq V, \exists x_1 \in V, \exists x_2 \in V, \forall y \in S, (y = x_1 \vee y = x_2)$$

$$\forall u \in V, \forall v \in V, (uv \in E \implies u \in X \vee v \in X)$$

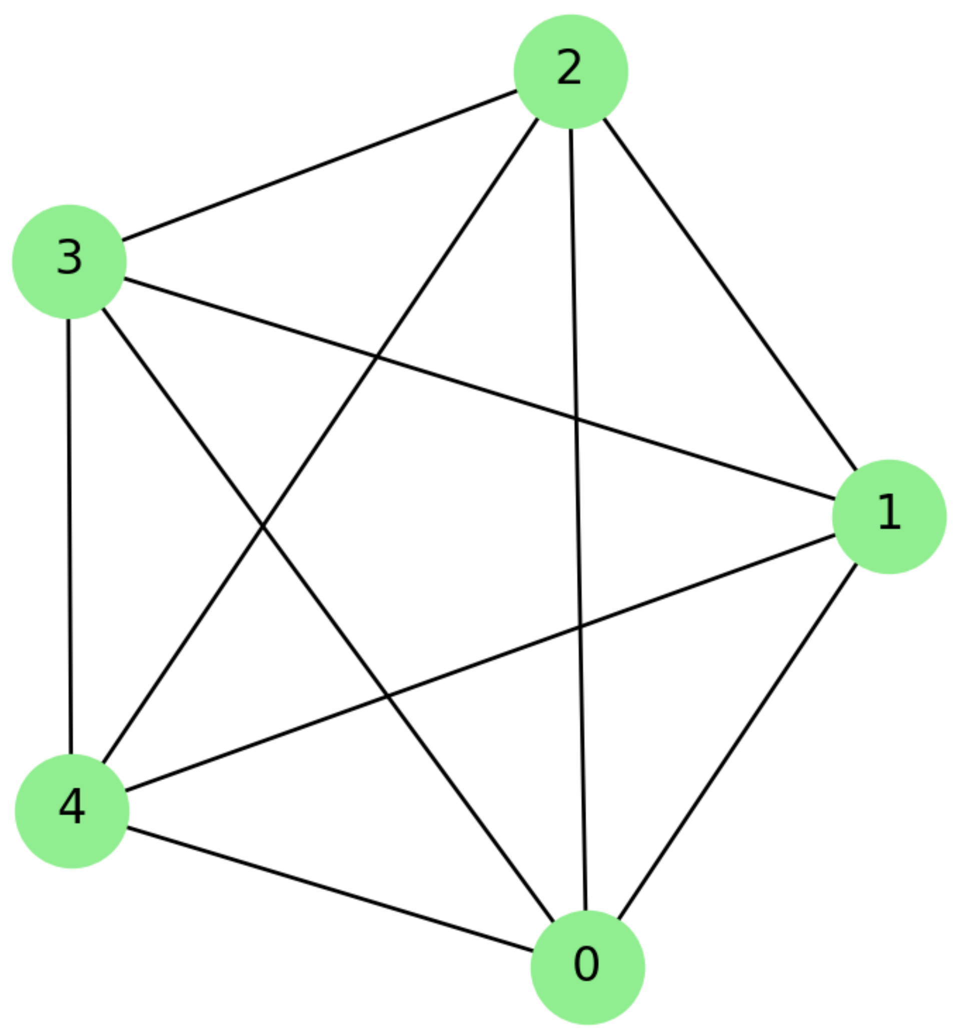
Optimization problem: Given G, find a vertex cover S of *minimum size*

Treewidth of a graph

- **Treewidth** is a **parameter** of a graph
- Characterizes how close a graph is to being a tree.



Treewidth 1



Treewidth $|V| - 1$

Class of k-bounded treewidth

$\downarrow$

Graphs with treewidth $\leq k$

Courcelle's Theorem

Courcelle's Theorem (Courcelle, 90'. Informal). Let φ be an MSO property and $k \in \mathbb{N}$. There exists an algorithm such that, for any graph G with $tw(G) \leq k$, decides if $G \models \varphi$ in linear time in the size of G.

Optimization version

Theorem (Borie, Parker and Tovey, 92'; Courcelle and Mosbah, 93'. Informal).

Given a graph optimization problem $\mathscr{P}$ of an MSO property and a class $\mathscr{G}$ of k-bounded treewidth graphs, there exists an algorithm such that, for any $G \in \mathscr{G}$, solves $\mathscr{P}(G)$ in linear time

Optimization version

Theorem (Borie, Parker and Tovey, 92'; Courcelle and Mosbah, 93'. Informal).

Given a graph optimization problem $\mathscr{P}$ of an MSO property and a class $\mathscr{G}$ of k-bounded treewidth graphs, there exists an algorithm such that, for any $G \in \mathscr{G}$, solves $\mathscr{P}(G)$ in linear time

➡ **Optimization version of many NP problems solved in linear time in k-bounded treewidth graphs.**

Maximum Independent Set, Minimum Vertex Cover, etc...

➡ **Linear in $|G|$ but super-exponential dependance in k**

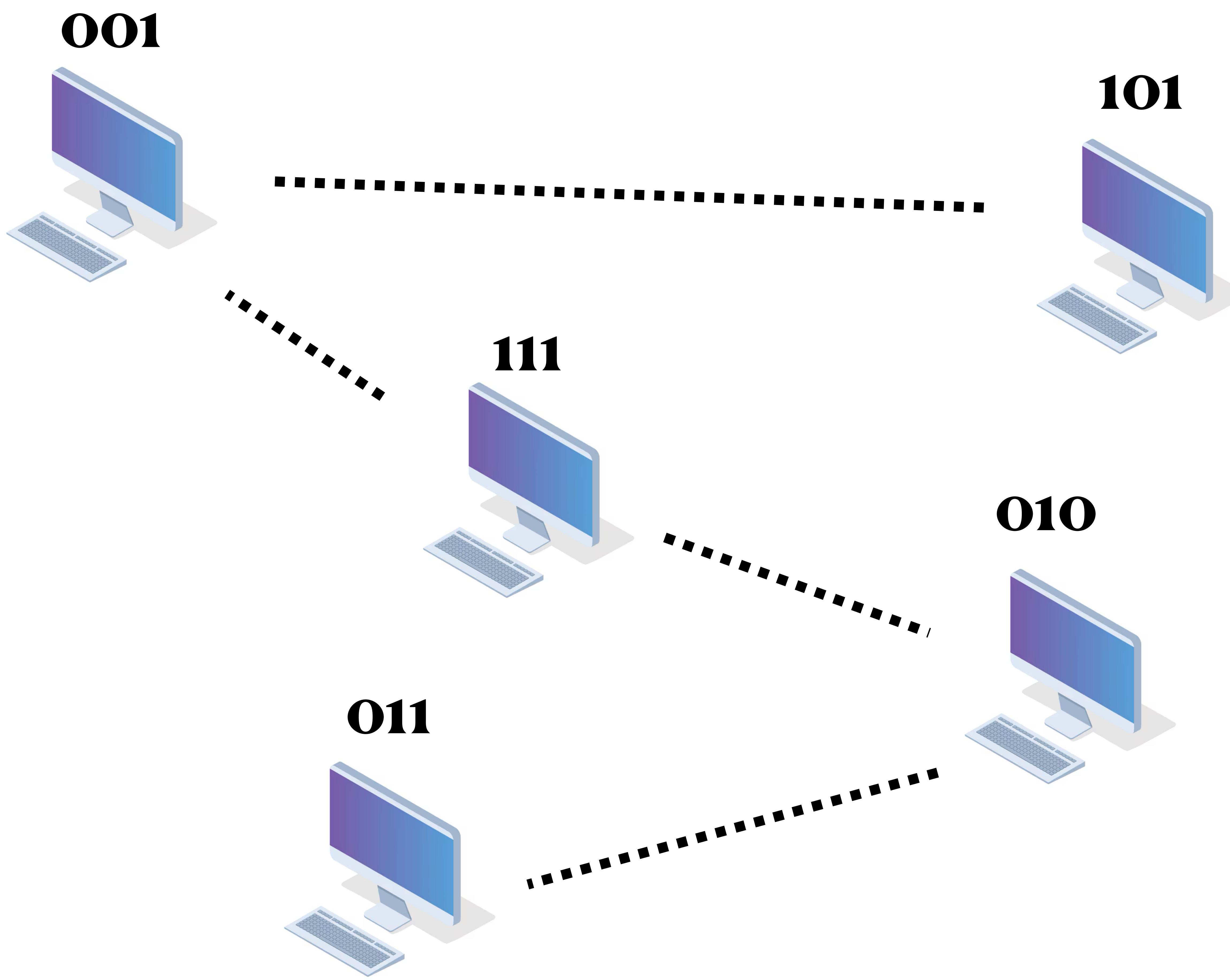
Distributed Computing

CONGEST Model

Distributed network represented as n-node graph $G = (V, E)$

- Each node has a unique **ID**.
- Nodes interact with their neighbors in **synchronous rounds.**
- Message size of $\mathcal{O}(\log n)$ bits.

- **Round complexity:** Maximum number of rounds a node in G sends a message to a neighbor.



Recent years

- Special interest in **sparse classes of graphs**
- Mainly **planar graphs**

Problem	General	Planar Graphs	Reference
Reachability	$\Omega(D+\sqrt{n}/\log n)$	$\tilde{O}(D)$	Parter, 2020.
MST	$\Omega(D+\sqrt{n})$	$\tilde{O}(D)$	Ghaffari and Haeupler, 2016.
Max-Flow	$\Omega(D+\sqrt{n})$	$\tilde{O}(D)$	Y. Abd-Elhaleem et al., 2025

MSO properties?

- Many of them are **hard to solve** in CONGEST.

Problem	General	Reference
Maximum IS	$\Omega(n^2/\log^2 n)$	Censor-Hillel, 2017.
Maximum Matching	$\tilde{\mathcal{O}}(\mu(G)^2)$	Ben-Basat et al et al. 2018
Minimum Vertex Cover	$\mathcal{O}(n^2/\log n)$	Folklore

MSO properties?

- Many of them are **hard to solve** in CONGEST.

Problem	General	Reference
Maximum IS	$\Omega(n^2/\log^2 n)$	Censor-Hillel, 2017.
Maximum Matching	$\tilde{\mathcal{O}}(\mu(G)^2)$	Ben-Basat et al et al. 2018
Minimum Vertex Cover	$\mathcal{O}(n^2/\log n)$	Folklore

Can we improve them in bounded treewidth graph?

Our results

(Decision version)

Theorem (J. Li, P. Montealegre, I. Todinca, B.J.). Given a fixed integer k and an MSO-property φ, there is a deterministic CONGEST algorithm that, given an n-node graph with diameter D and treewidth at most k, decides $G \models \varphi$ in $\tilde{\mathcal{O}}(D)$ rounds.

(Optimization version)

Theorem (J. Li, P. Montealegre, I. Todinca, B.J.). Given a fixed integer k and an optimization problem $\mathcal{P}$ from an MSO-property φ, there is a deterministic CONGEST algorithm that, given an n-node graph with diameter D and treewidth at most k, computes $\mathcal{P}(G)$ in $\tilde{\mathcal{O}}(D)$ rounds.

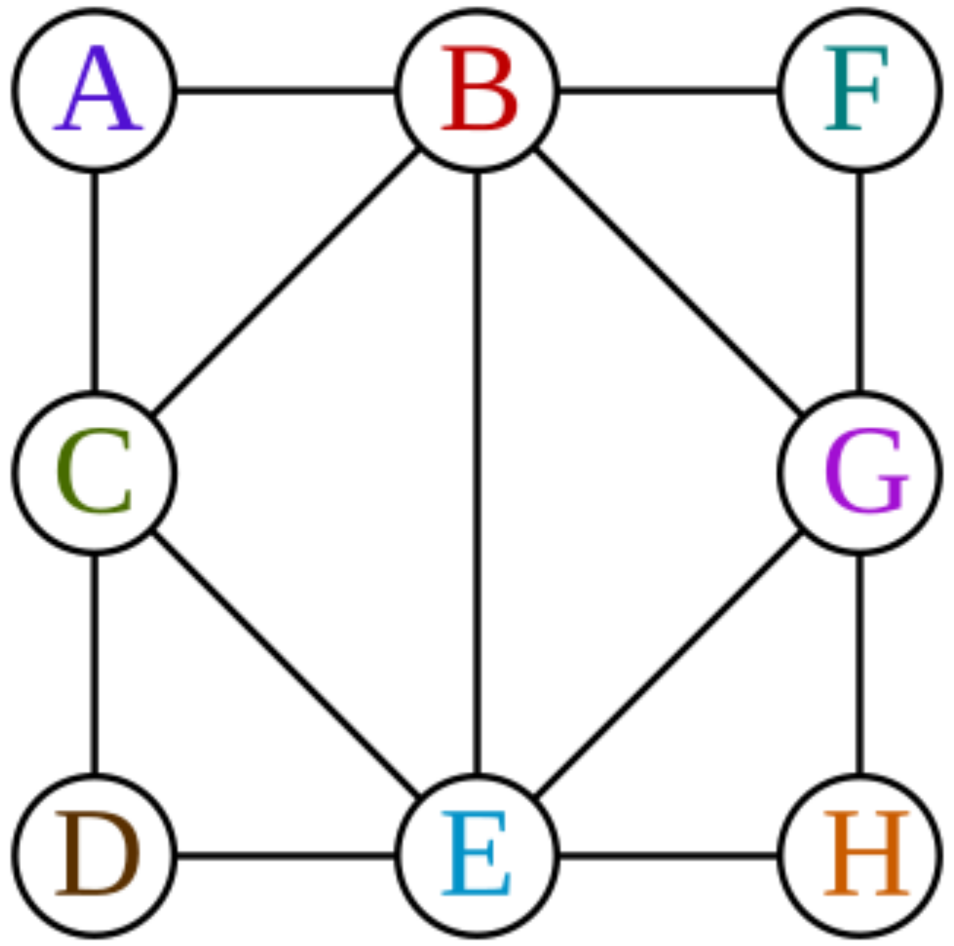
Applications

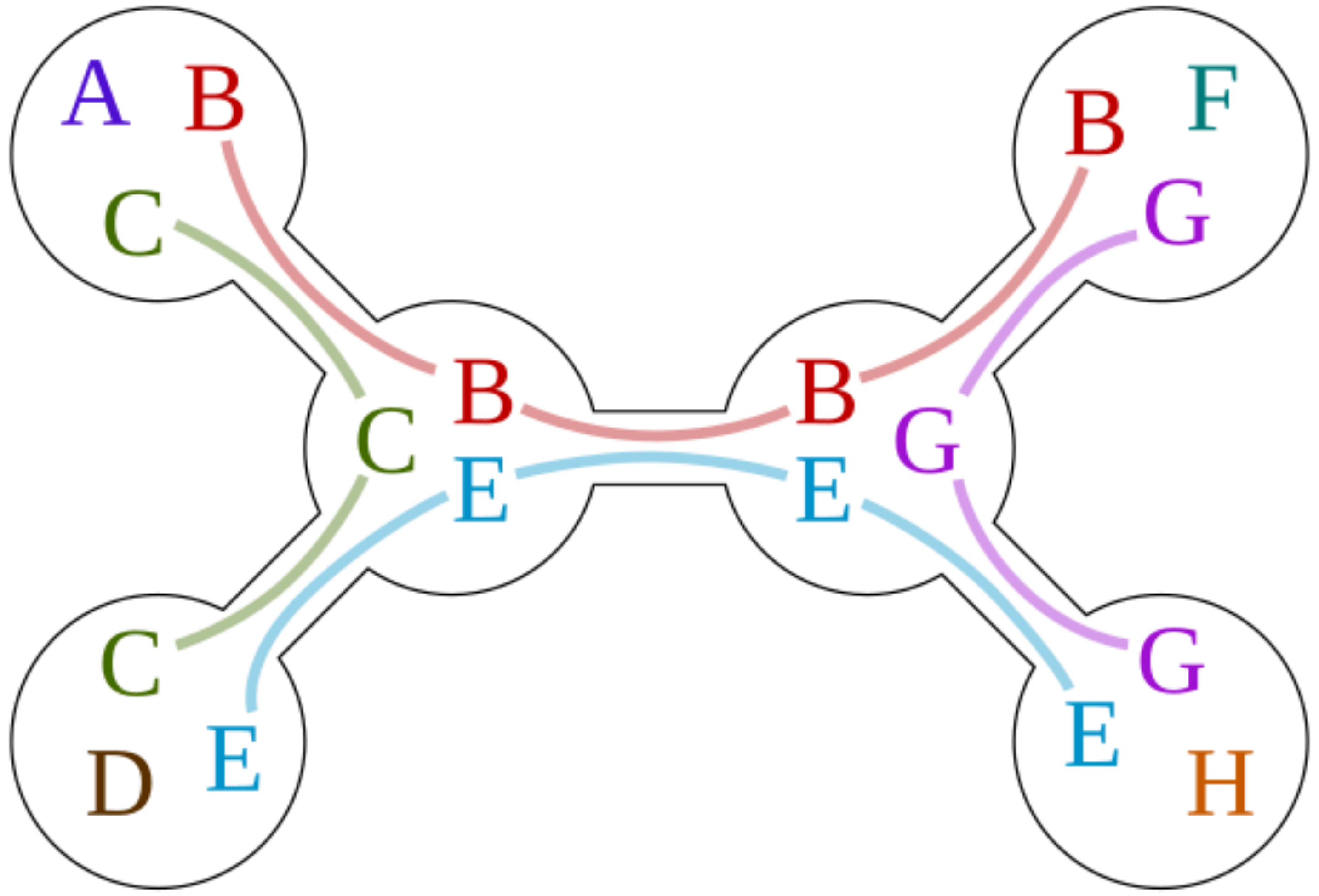
Problem	General	Bounded treewidth
Maximum IS	$\Omega(n^2/\log^2 n)$	$\tilde{\mathcal{O}}(D)$
Maximum Matching	$\tilde{\mathcal{O}}(\mu(G)^2)$	$\tilde{\mathcal{O}}(D)$
Minimum Vertex Cover	$\mathcal{O}(n^2/\log n)$	$\tilde{\mathcal{O}}(D)$

… Minimum Dominating Set, Maximum Clique, etc…

High Level Idea

TREE DECOMPOSITION: Pair (X,T). T is a tree, V(T)=X and $X \subseteq \mathscr{P}(V)$





(1) Union of bags equals V

(2) Every edge is contained in a bag

(3) For every node v, graph induced by bags containing v is connected

Courcelle's idea

Theorem. Let G be an n-node graph with $tw(G) \leq k$. Then G admits a tree decomposition (X, T) such that

- T has depth $O(\log n)$ and,
- For every node $t \in X$, the bag B_t has size $O(k)$.

ALGORITHM

(I) Compute a tree decomposition (X,T)

(II) Apply dynamic programming over the tree decomposition

Distributed implementation

(I) Compute a tree decomposition (X,T)

(II) Apply dynamic programming over the tree decomposition

Distributed implementation

(I) Compute a tree decomposition (X,T)

How to represent a tree decomposition?

How to build a tree decomposition?

(II) Apply dynamic programming over the tree decomposition

Distributed implementation

(I) Compute a tree decomposition (X,T)

How to represent a tree decomposition?

How to build a tree decomposition?

(II) Apply dynamic programming over the tree decomposition

How to apply dynamic programming

In $\tilde{\mathcal{O}}(D)$ rounds?

Computing a tree decomposition

Definition. Given an n-node graph G a separator set S of G is a subset of nodes such that the size of each connected component of $G - S$ is at most $2n/3$

One Separator

Lemma (J. Li, P. Montealegre, I. Todinca, B.J.) There exists a CONGEST algorithm such that given a graph G with $tw(G) = k$ and any connected subgraph H, computes in $\tilde{\mathcal{O}}(D)$ rounds, a separator set of size at most k

Computing a tree decomposition

Definition. Given an n-node graph G a separator set S of G is a subset of nodes such that the size of each connected component of $G - S$ is at most $2n/3$

Disjoint Separators

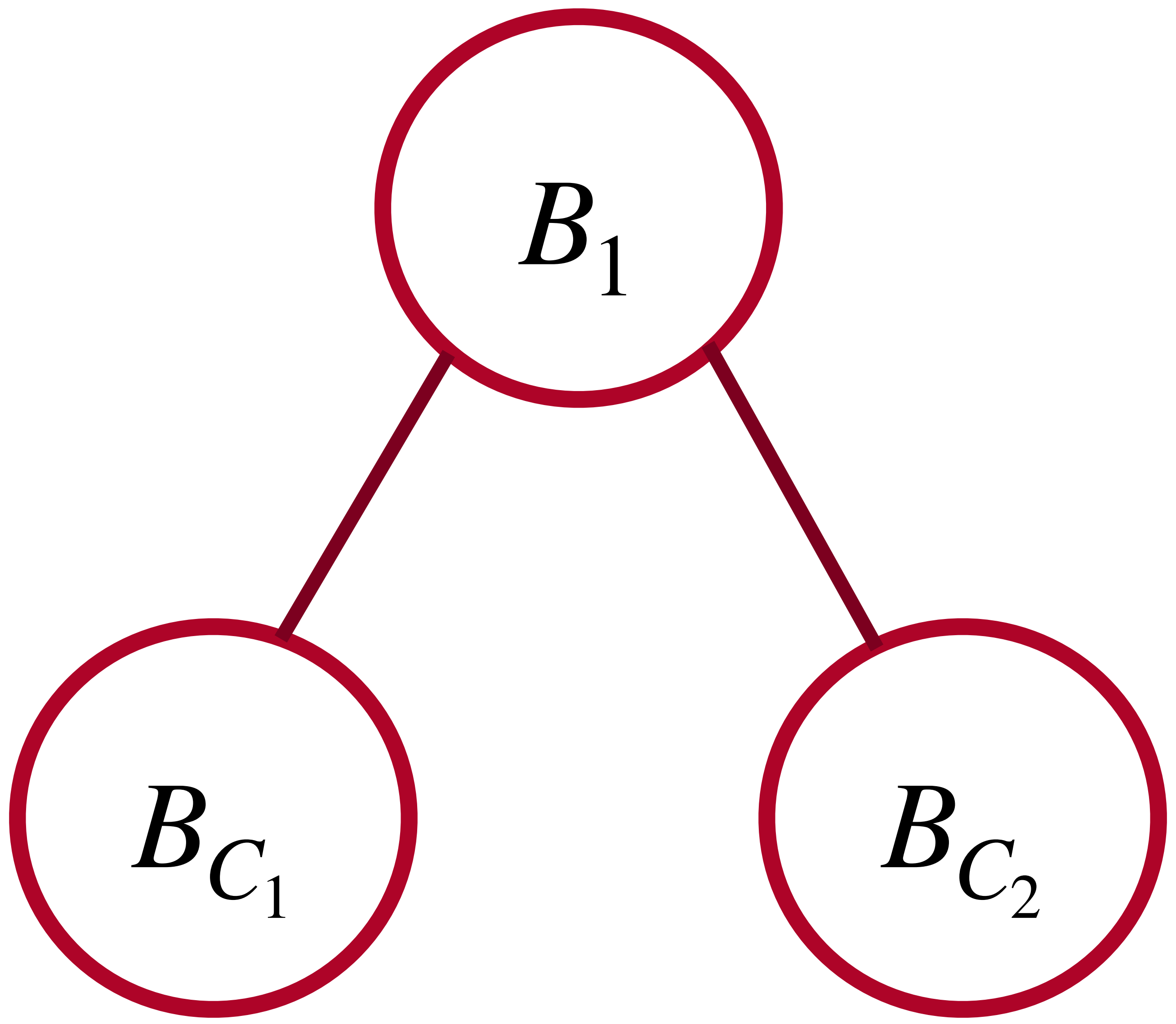
Lemma (J. Li, P. Montealegre, I. Todinca, B.J.) There exists a CONGEST algorithm such that given a graph G with $tw(G) = k$ and any collection $H_1, \ldots, H_j$ of disjoint and connected subgraphs, computes in $\tilde{\mathcal{O}}(D)$ rounds, in parallel, a separator set of size at most k

Distributed computation

Recurse in instances (U, X) with $U \subseteq V, X = N(U)$.

Start with $(V, \emptyset)$

- $S_V = separator(G)$
- $B_1 = S_V$ **(root bag)**
- **In parallel in each c.c.** C_i **of** $G - B_1$
  - $S_{C_i} = separator(C_i)$
  - $B_{C_i} = S_{C_i} \cup N\left(C_i\right)$ **(**)**
  - **Continue recursively in c.c. of** $C_i - S_{C_i}$

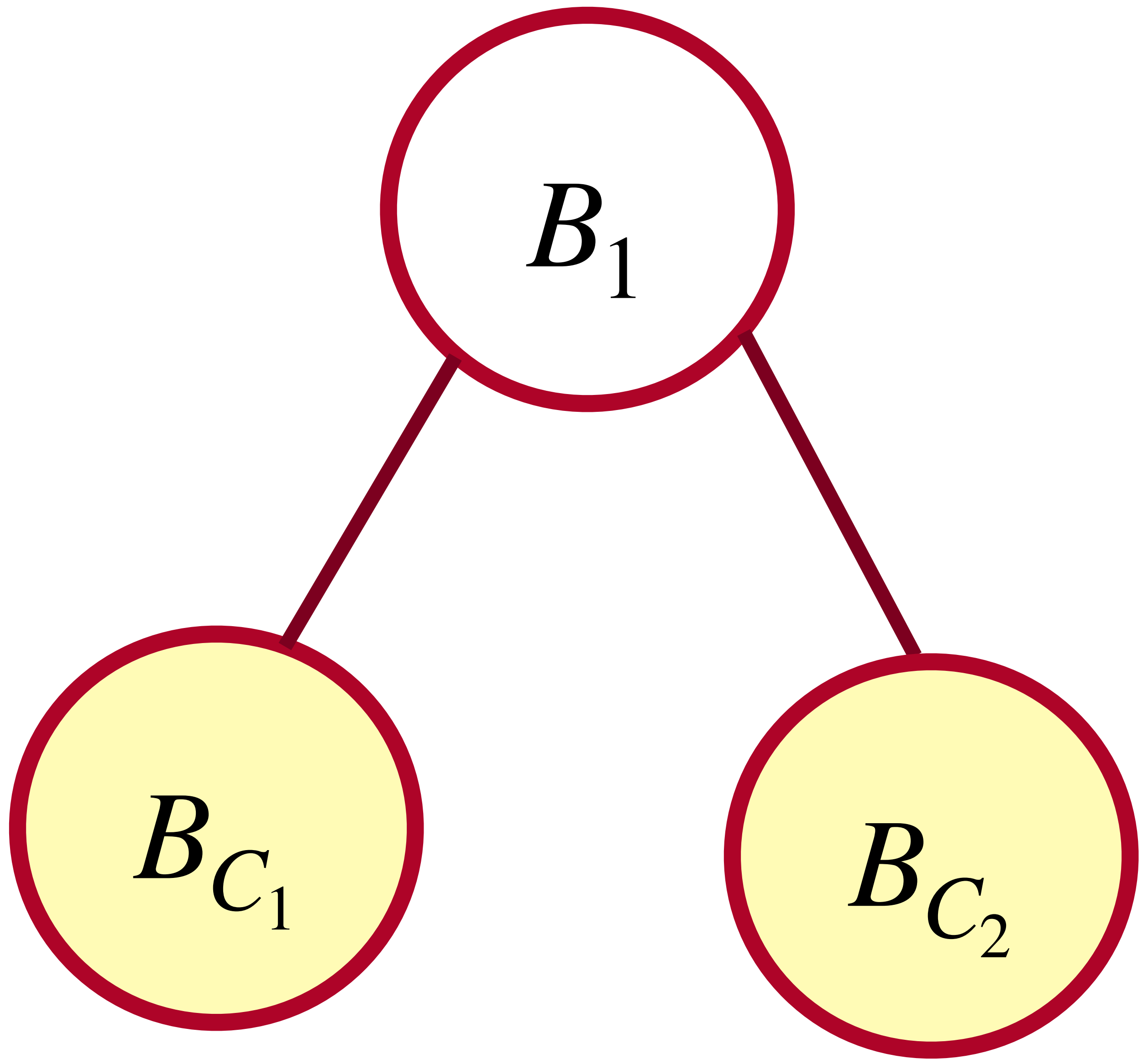


Properties

- **Size $\mathcal{O}(k)$ of each bag**
- **Depth $\mathcal{O}(\log n)$**
- **Inherently we obtain a distributed representation in each node**
- **Each bag has a leader**

Dynamic programming: Maximum Independet Set

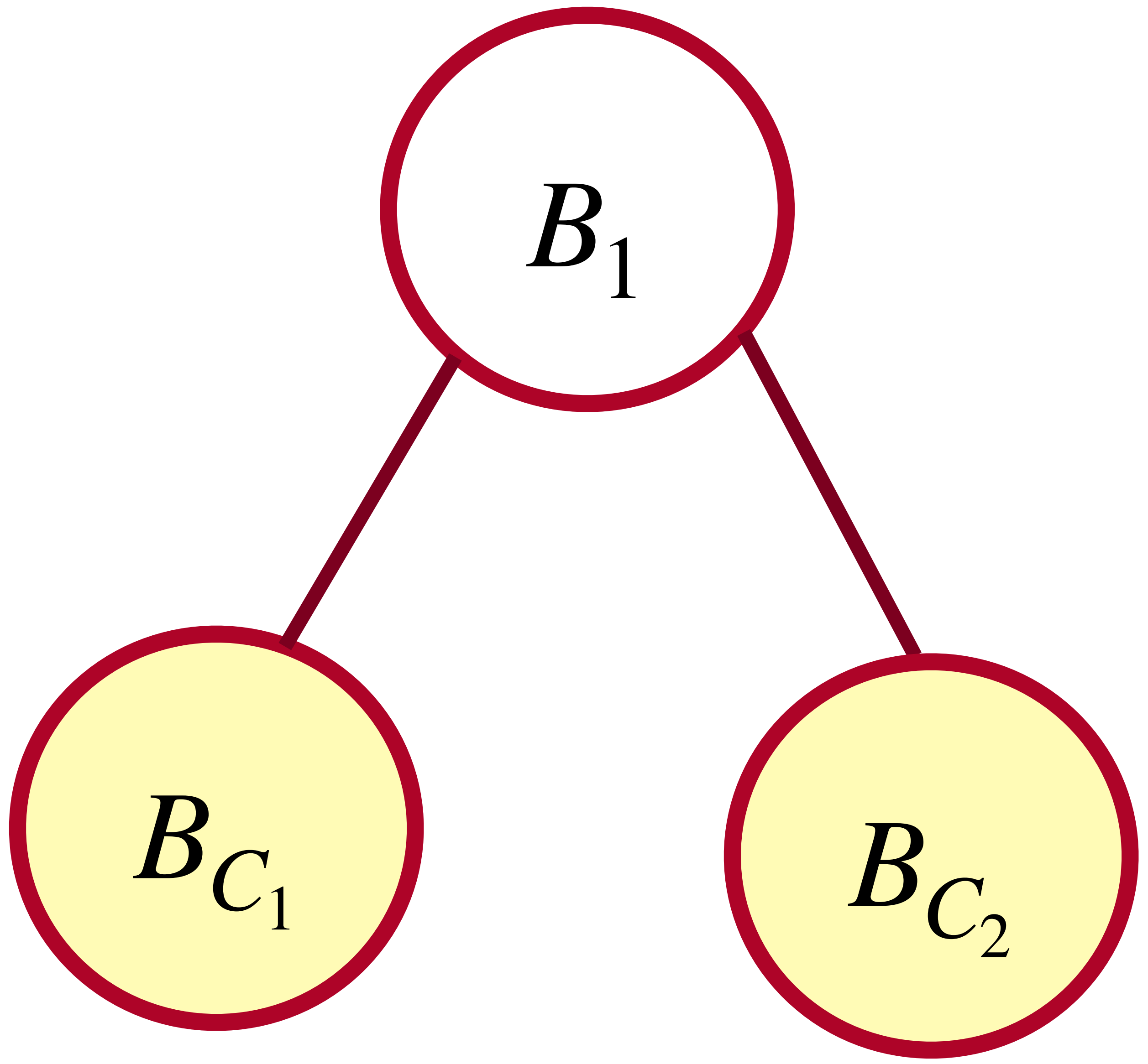
(1) **Leaf bags compute all possible solutions greedly.**



Dynamic programming: Maximum Independet Set

(1) Leaf bags compute all possible solutions greedly.

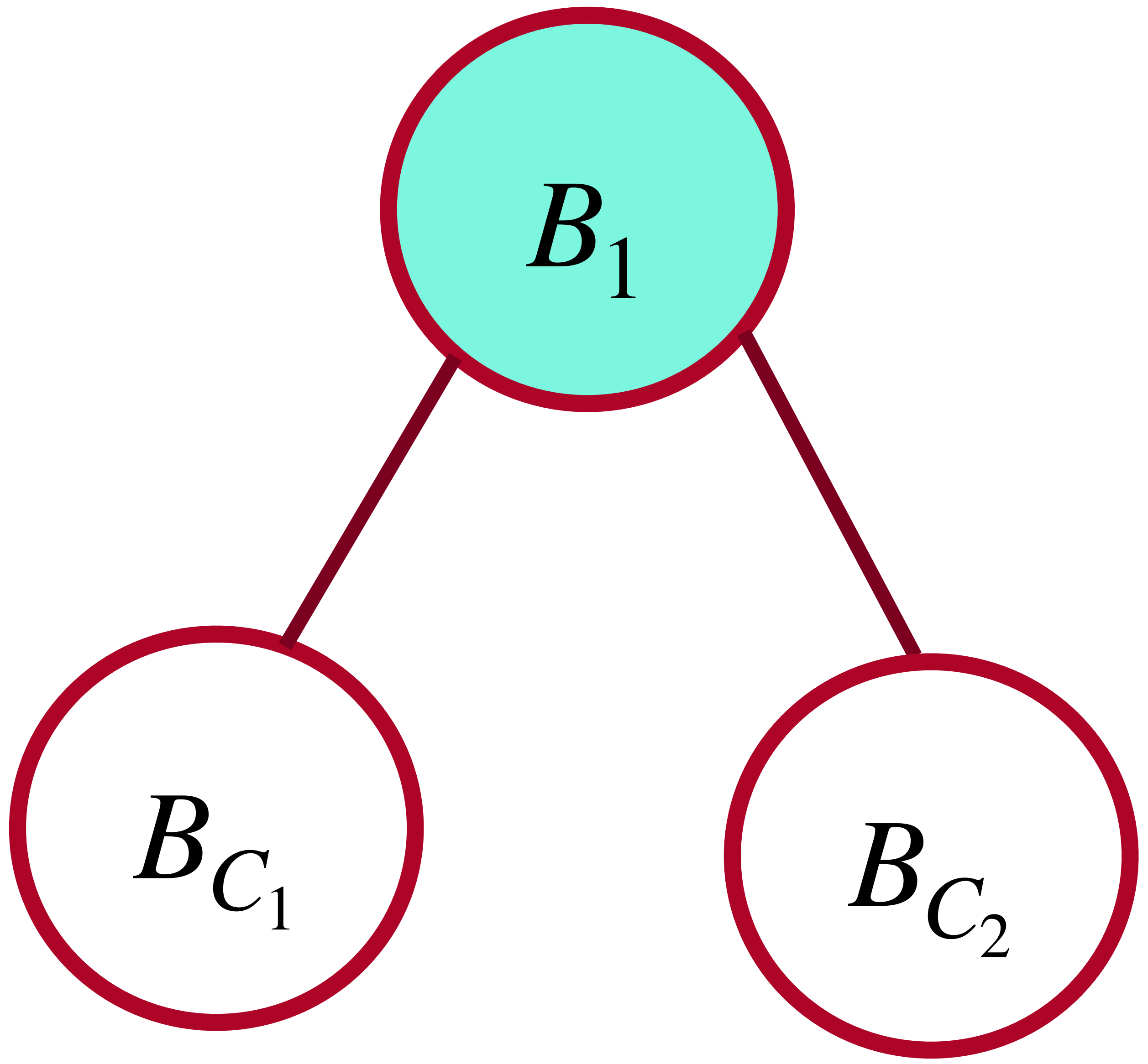
$$OPT(X_{C_i}) = \left| MaxIS\left(G\left[X_{C_i}\right]\right) \right|$$



$\boxed{2^{O(k)} \text{ rounds}}$

Dynamic programming: Maximum Independet Set

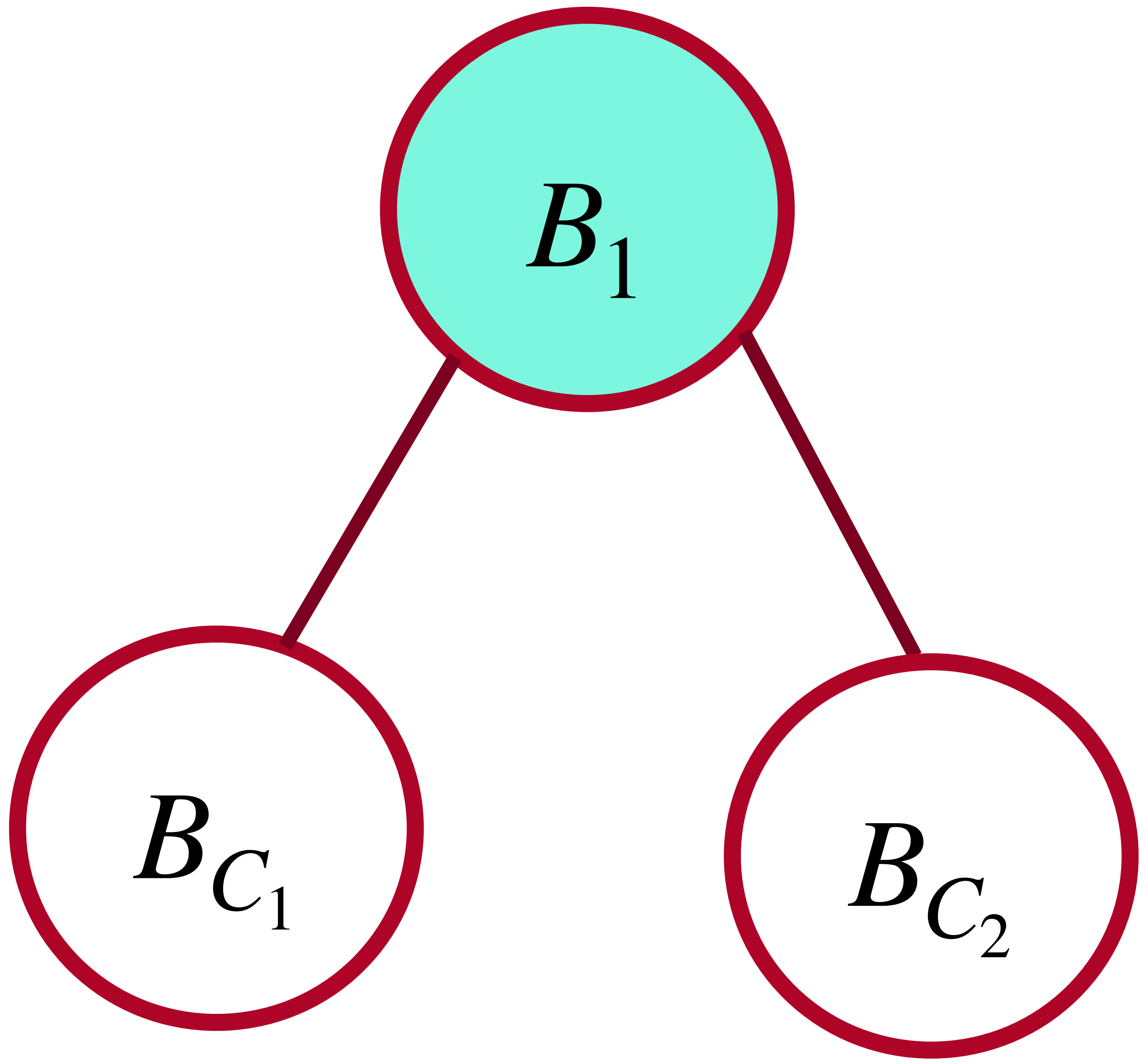
(2) Non-leaf bag: Extend the solution of its children



Dynamic programming: Maximum Indepedent Set

(2) Non-leaf bag: Extend the solution of its children

"Compute commutative and associative function of local input"



Great tool: Partwise aggregation problems

$\boxed{\tilde{O}(k \cdot D) \text{ rounds}}$

Summary

➡ Deterministic implementation of Courcelle's Theorem in CONGEST.

➡Improve to $\tilde{\mathcal{O}}(D)$ rounds several problems **with the same algorithm**

➡ Decision, optimization and **counting** version implemented.


MISAOTRA
KIITOS
MAHALO
THANK YOU
CHOKRANE
MATONDO
AŠANTE
DANKE
MÈSI
VINAKA
TAKK
MERCI
OBRIGADO
ARIGATO
DZIĘKUJĘ
MULTUMESC
ŠALAMAT
HVALA
TAK
ΕΥΧΑΡΙΣΤΩ
GRÀCIES
KÖSZÖNÖM
BEDANKT
TEŞEKKÜRLER
PALDIES
DAKUJEM
GRACIAS
ありがとう
WELALIN
GRAZIE
TACK
DĚKUJI
XIE XIE
DĚKUJI
תודה
धन्यवाद
謝謝
شكراً
GRAZIAS
NA GODE
BARKA
СПАСИБО
ขอบคุณ
TERIMA KASIH
SPASIBO
ধন্যবাদ
DANK JE
CAM ON BAN
OBRIGADA
SHUKRAN
GRATIÄS AGO
TENKI
NANDRI
SHUKRIYA
شكريه
CHOUKRANE
DHANYAVAD
MAMNUN
MATUR NUWUN
RAHMAT

Paper