COMP 691: Online Algorithms and Competitive Analysis

Lecture 7

Denis Pankratov

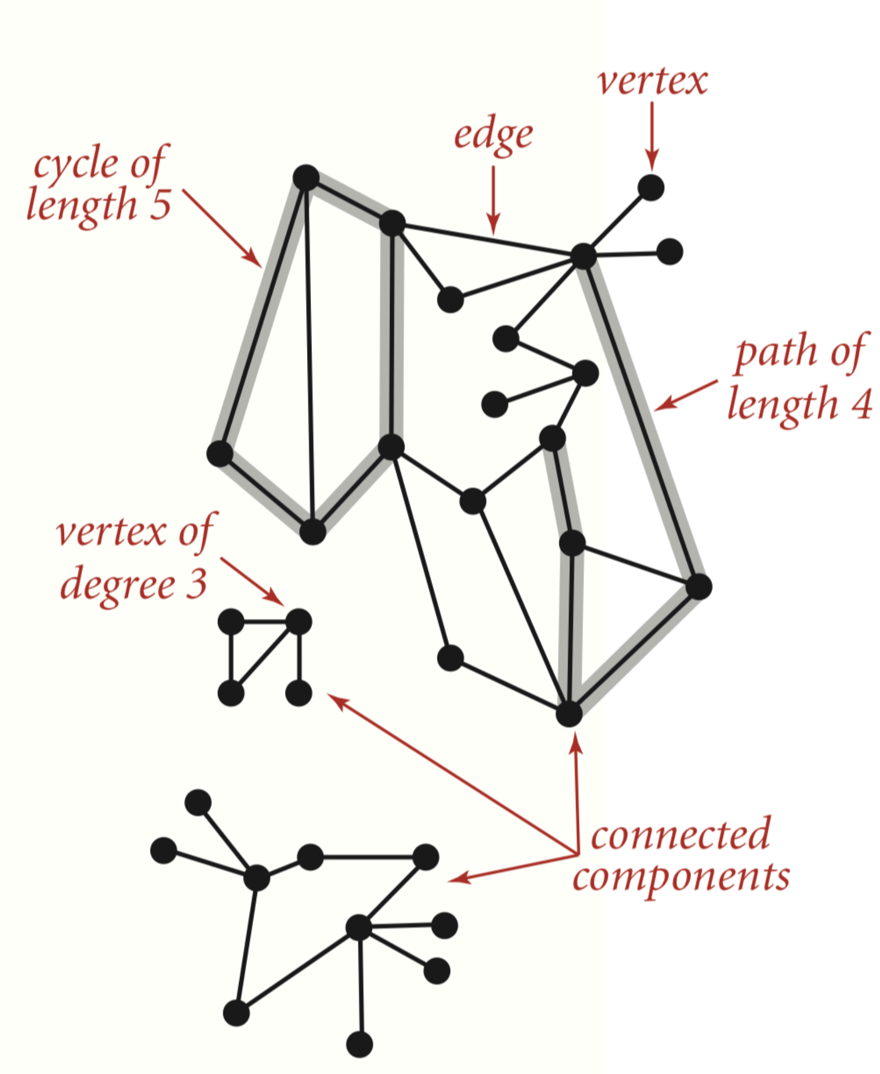
Office Hour: Th 1:00PM – 2:00PM @ EV 3.127

Or: appointments by email

Last time

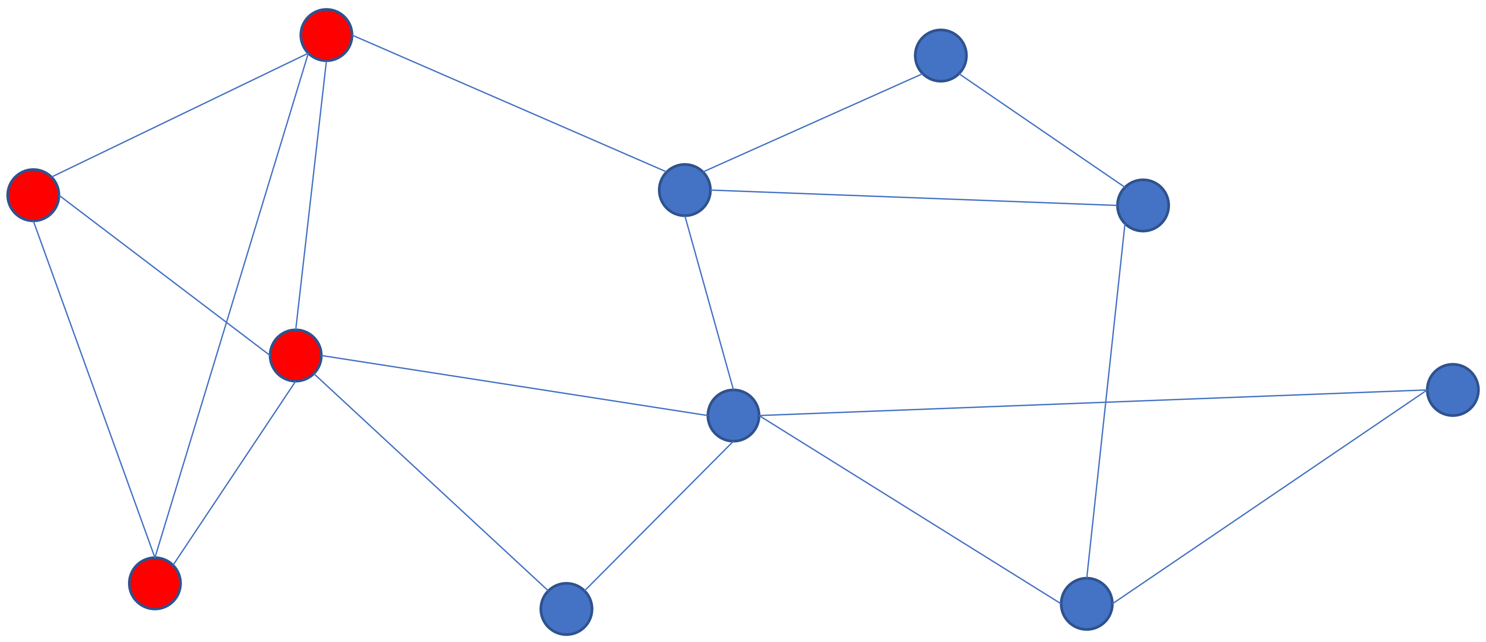
- Metric Spaces
- k-Server problem
  - Informal statement
  - Formal statement, configurations
  - Greedy algorithm is not competitive
  - Deterministic and randomized conjectures
  - Deterministic lower bound
  - Line metric: DoubleCoverage algorithm
  - Work Function and Work Function Algorithm
- Graph Theory Primer & Intro to Online Graph Problems
- **Announcement**: A2 due on Feb ~~23rd~~ **26th** at 23:55 on Moodle
- **Announcement:** next week is a break, after that we have a midterm

Graph anatomy: summary so far



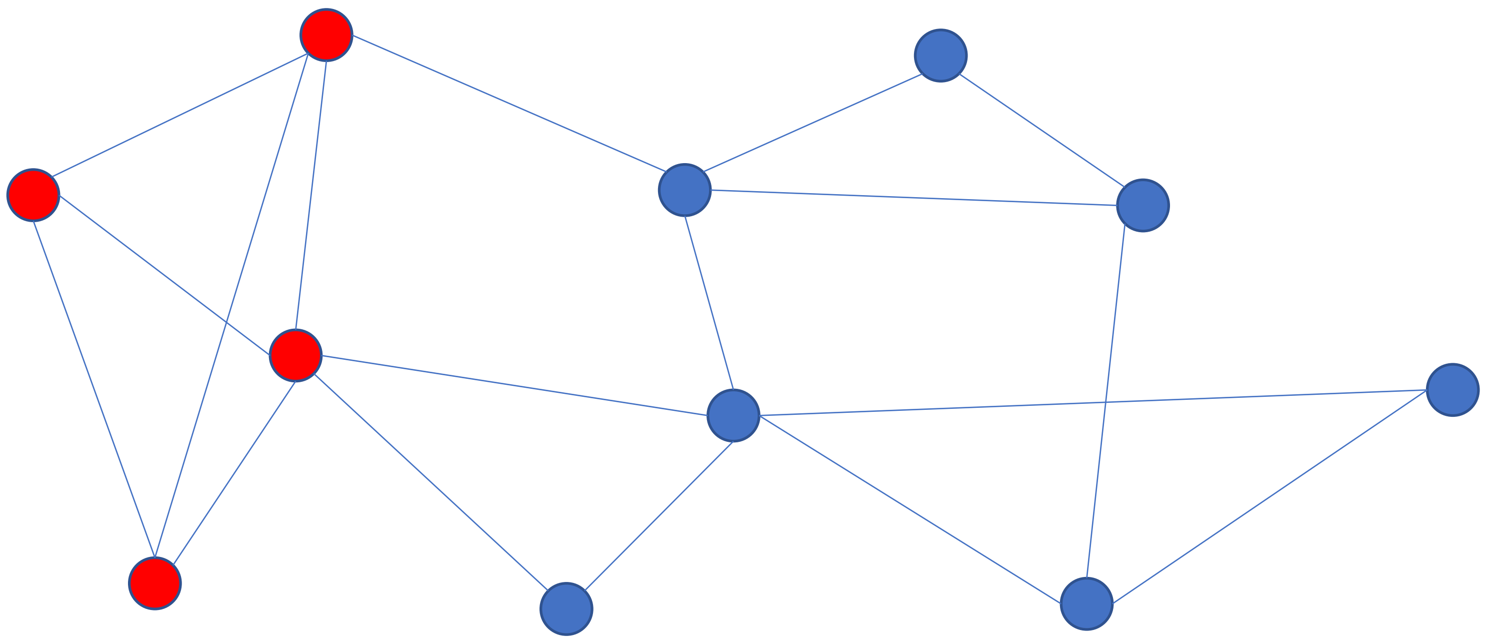
Famous Problems

A set $S \subseteq V$ is a **clique** if for all $u, v \in S$ we have $\{u, v\} \in E$



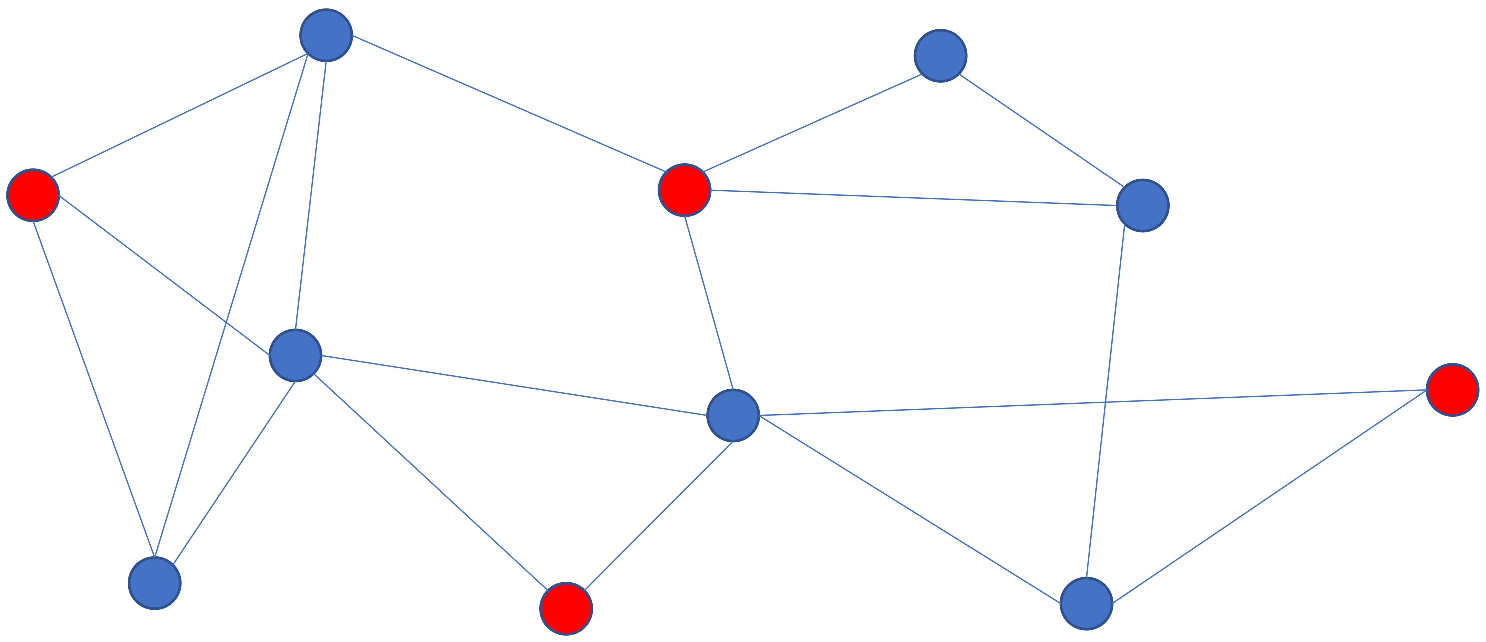
Famous Problems

Maximum clique: find a clique of maximum size



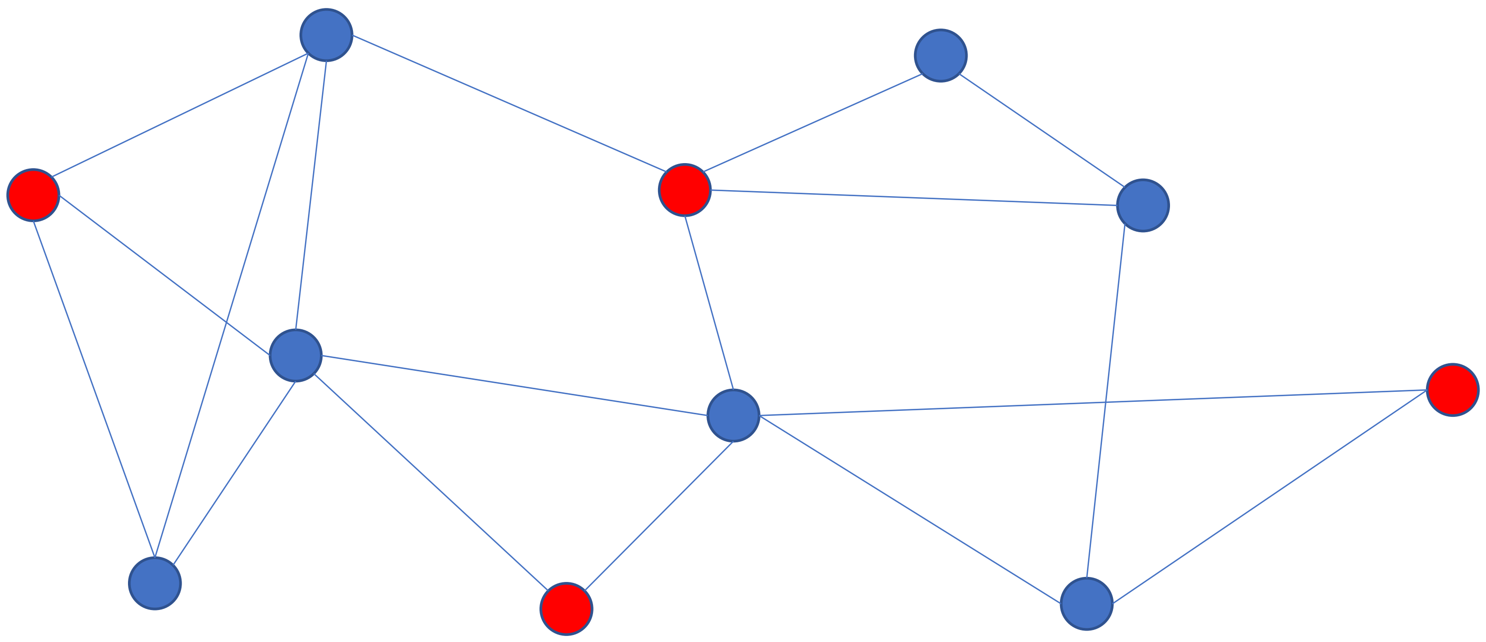
Famous Problems

A set $S \subseteq V$ is an **independent set** if for all $u, v \in S$ we have $\{u, v\} \notin E$



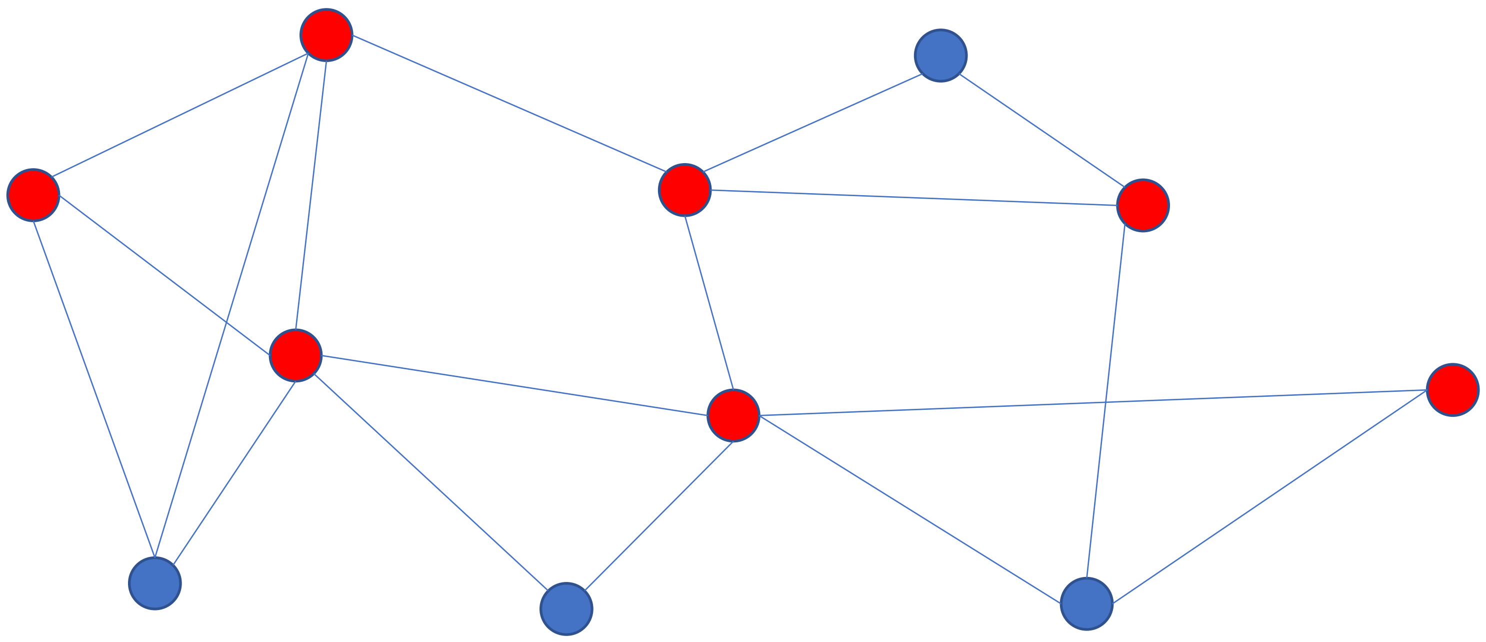
Famous Problems

Maximum independent set: find an independent set of maximum size



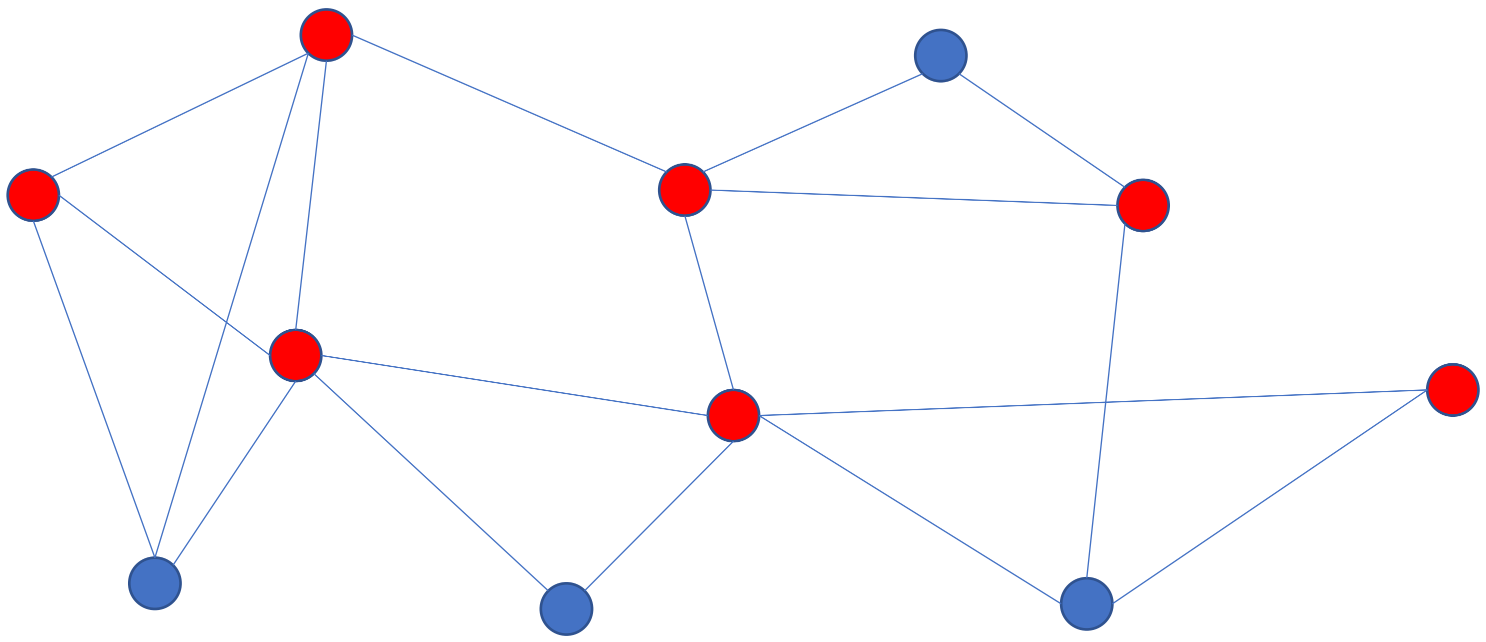
Famous Problems

A set $S \subseteq V$ is a **vertex cover** if every edge is incident on a vertex in S, i.e., for all $e \in E$ we have $e \cap S \neq \emptyset$



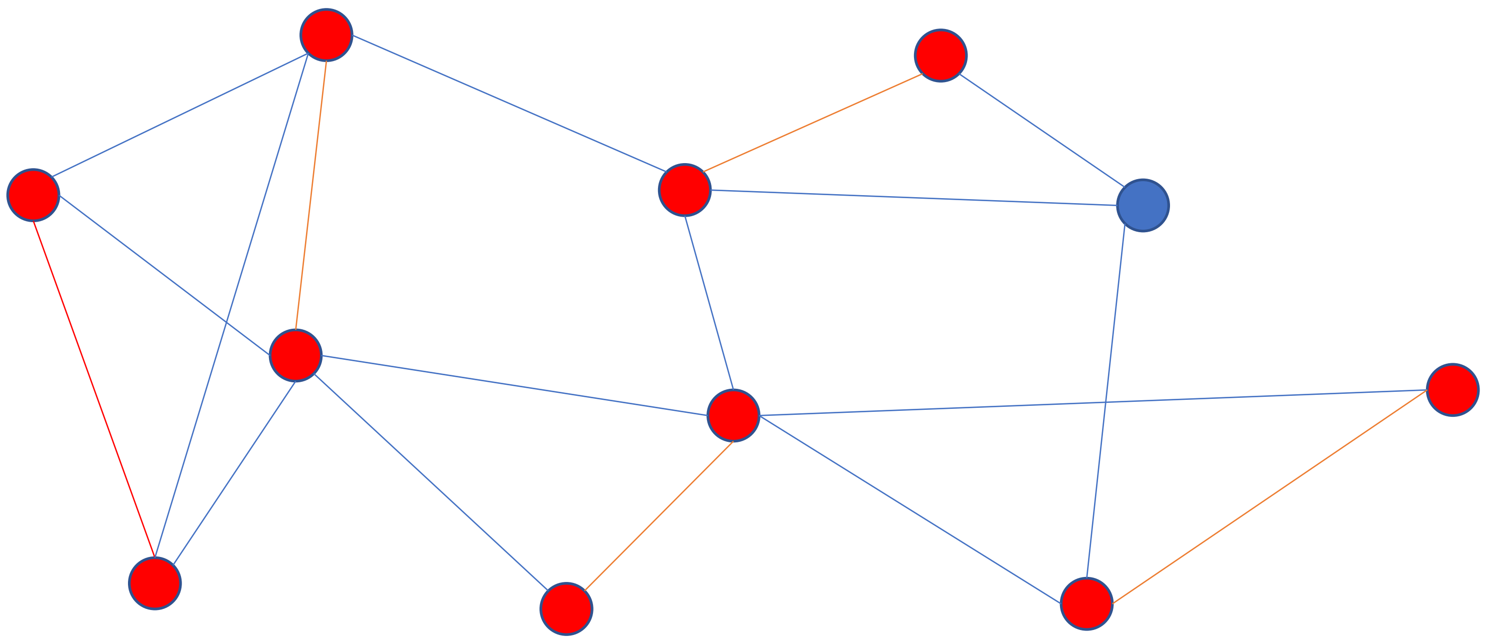
Famous Problems

Minimum vertex cover: find a vertex cover of minimum size



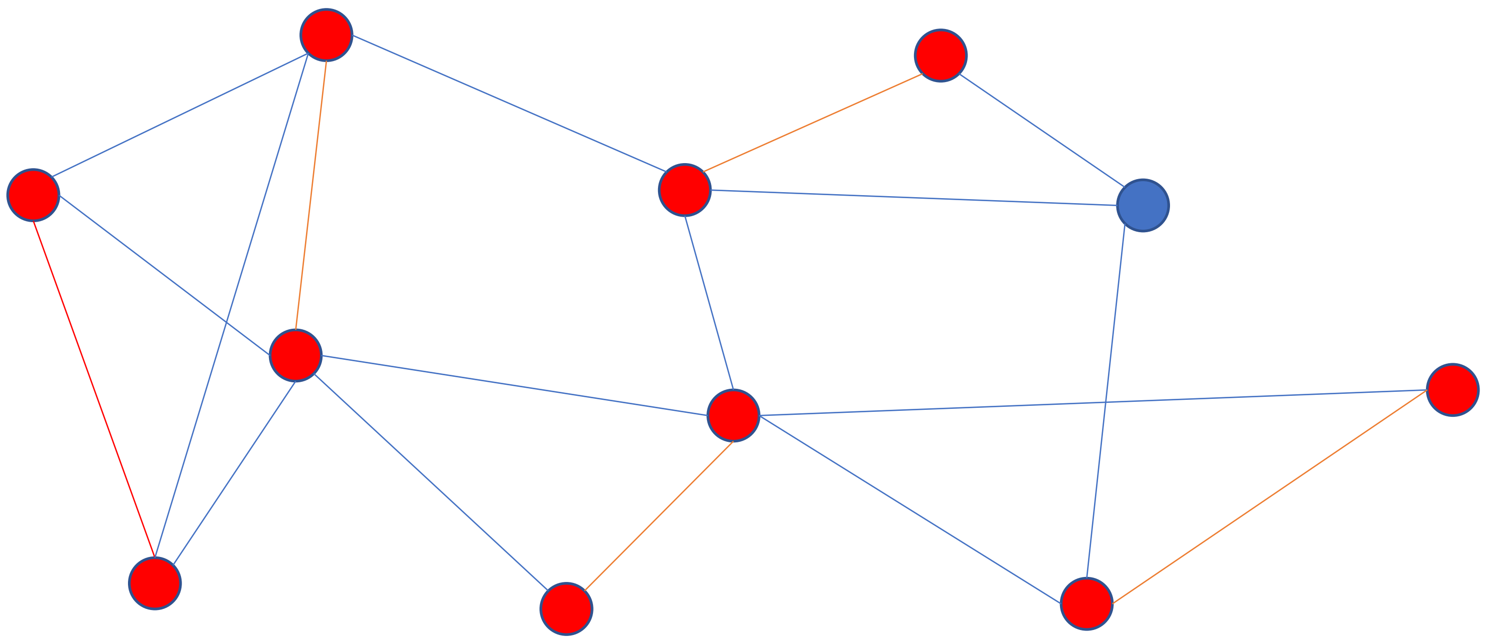
Famous Problems

A subset of edges $M \subseteq E$ is a matching if all edges of M are disjoint, i.e., for all $e_1, e_2 \in M$ we have $e_1 \cap e_2 = \emptyset$



Famous Problems

Maximum matching: find a matching of maximum possible size



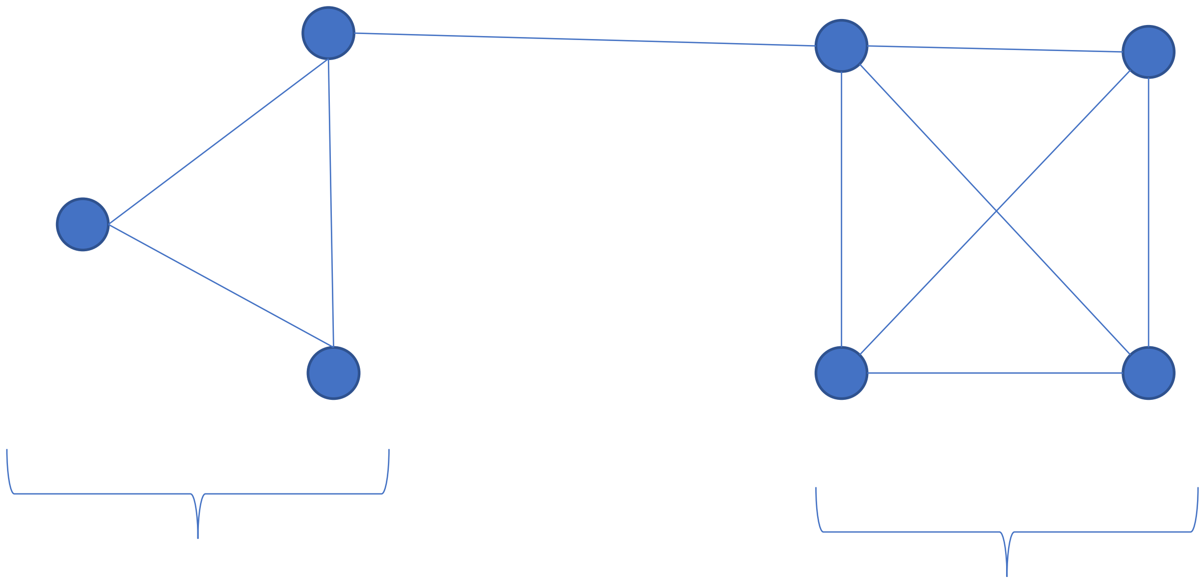
Maximum vs maximal

Very important distinction!!! Comes up a lot!!! Memorize it!!!

Maxim**um**: means absolutely maximum, there is no object that satisfies the constraints and can be bigger

Maxim**al**: means cannot be extended without violating the constraints

Maximum vs maximal example: clique



This clique is maximal, because no other vertex can be added to it

This clique is maximum, because there is no larger clique in the graph

Online graph problems

Online graph problems

Online problems where a part (or the entirety) of input consists of a graph that **is not known in its entirety** in advance

The input graph is revealed in an online fashion

Not to be confused with online problems where a graph is known in advance, e.g., the k-Server problem on graph metrics

Online input models

How is graph revealed?

There are at least 4 different models!

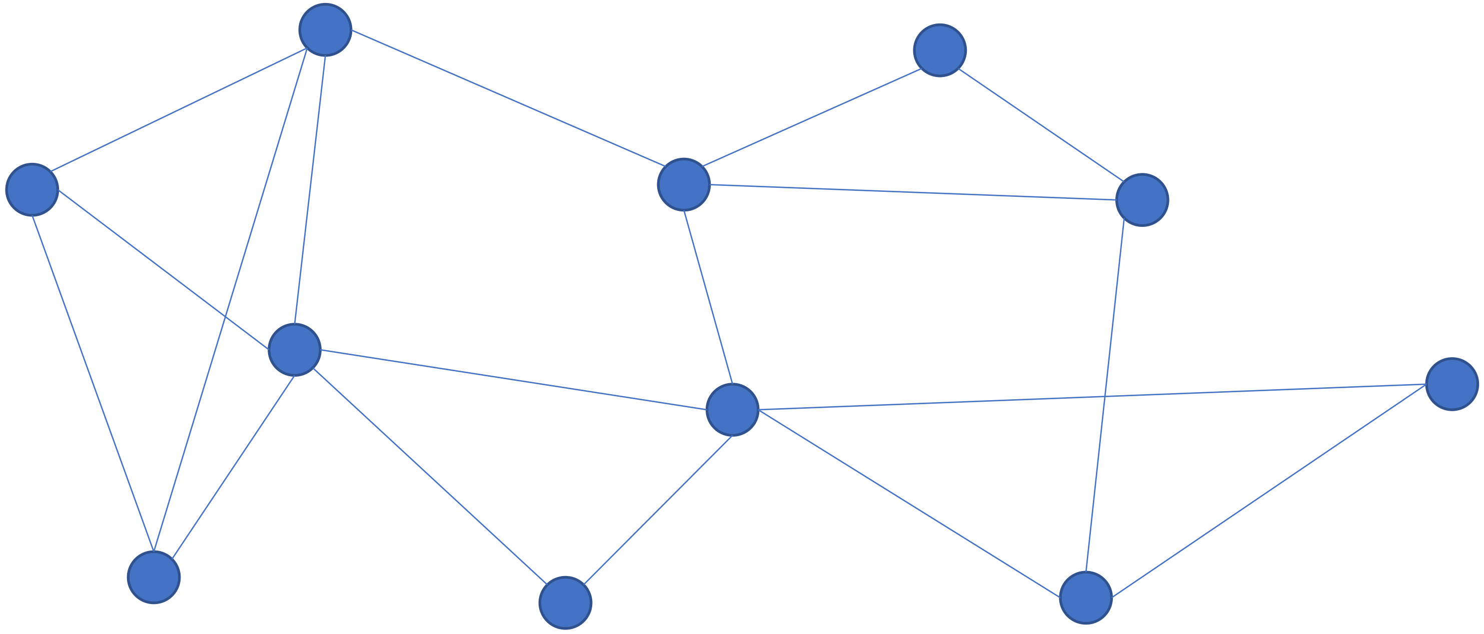
1. The Edge Model (EM)
2. The Vertex Adjacency Model, Past History (VAM-PH)
3. The Edge Adjacency Model (EAM)
4. The Vertex Adjacency Model, Full Information (VAM-FI)
5. + an extra model specific to bipartite graphs: Bipartite Vertex Arrival Model (BVAM)

The Edge Model (EM)

Each input item is an edge given by its two endpoints

Input: $\{u_1, v_1\}, \{u_2, v_2\}, \dots, \{u_m, v_m\}$

Vertex set V is not known in advance

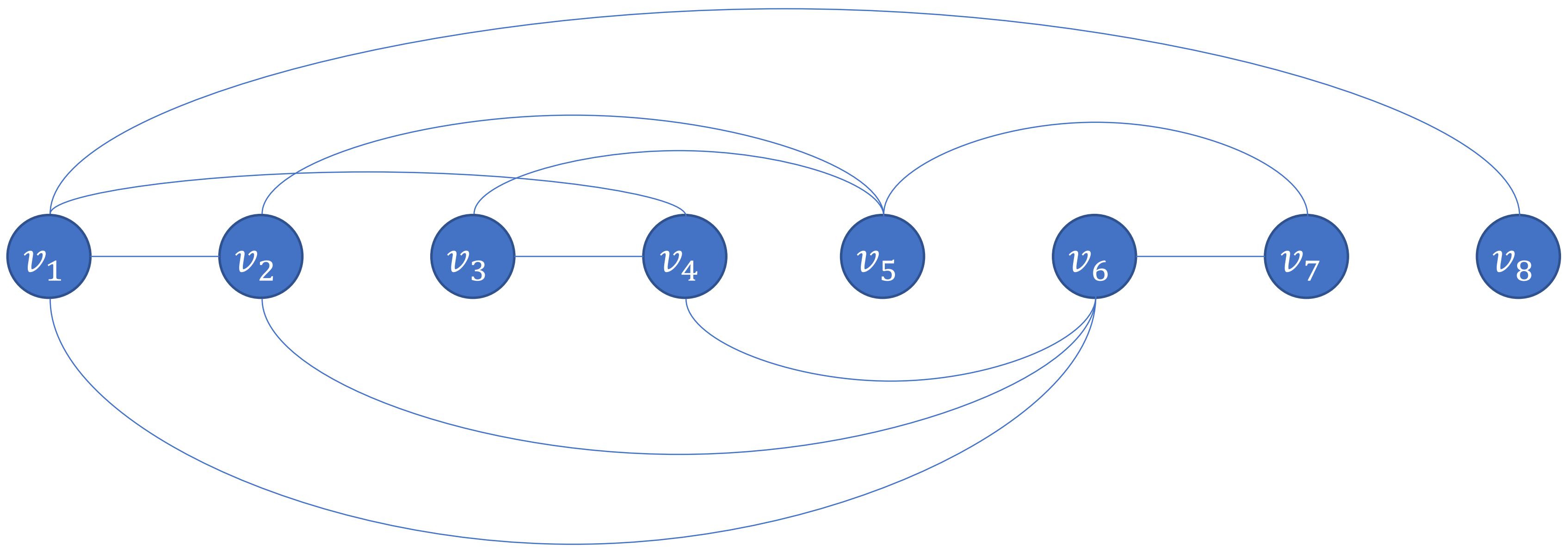


Vertex Adjacency Model, Past History (VAM-PH)

$\prec$ - order in which vertices are revealed

Each input item is a vertex together with its neighbors among the vertices that appeared before:

$$\left(v, \left(N(v) \cap \{u : u \prec v\}\right)\right)$$



Edge Adjacency Model (EAM)

Each edge e has a label ℓ_e

Knowing one vertex $v \in e$ and the label ℓ_e it is impossible to say what the other vertex in the edge is

Each input item is a vertex together with labels of all edges incident on it:

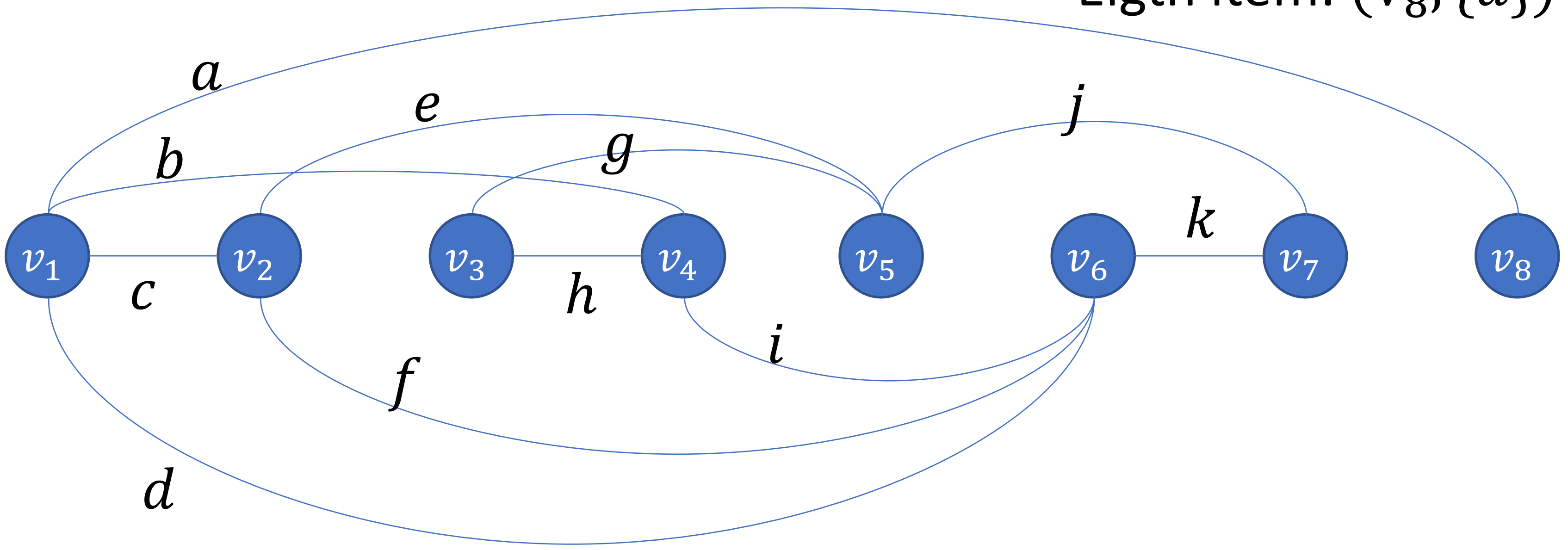
$$(v, \{\ell_e : v \in e\})$$

Suppose $u \sim v$ and $u \prec v$, both vertices will have ℓ_e label in the list of labels of edges, but we only learn that $u \sim v$ when v arrives

EAM example

Seventh item: $(\mathrm{v}_7, \{j, k\})$ learn that…

Eigth item: $(\mathrm{v}_8, \{a\})$ learn that…



First item: $(\mathrm{v}_1, \{a, b, c, d\})$

Second item: $(\mathrm{v}_2, \{c, e, f\})$ learn that $v_1 \sim v_2$

Third item: $(\mathrm{v}_3, \{g, h\})$

Fourth item: $(\mathrm{v}_4, \{b, h, i\})$ learn that $v_3 \sim v_4, v_1 \sim v_4$

Fifth item: $(\mathrm{v}_5, \{e, g, j\})$ learn that $v_5 \sim v_3, v_5 \sim v_2$

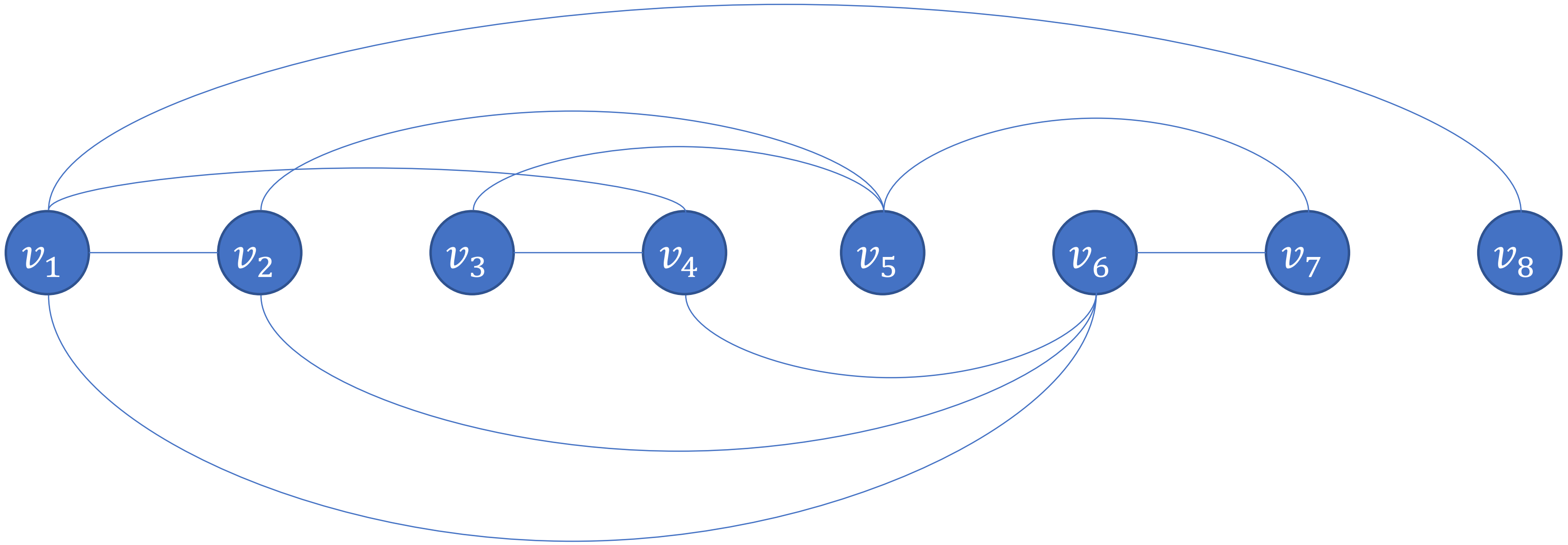
Sixth item: $(\mathrm{v}_6, \{i, f, d, k\})$ learn that…

Vertex Arrival Model, Full Information (VAM-FI)

$\prec$ - order in which vertices are revealed

Each input item is a vertex together with its **entire** neighborhood:

$$(v, N(v))$$



Relationships between models

For two models M1 and M2, we write M2≤M1 to indicate that any algorithm that works in M1 can be transformed to work in M2 without deterioration of performance.

Theorem (informal)	VAM-FI ≤ EAM ≤ VAM-PH ≤ EM

Claim: VAM-PH $\leq$ EM

Pf:

ALG – algorithm in EM

In VAM-PH model, item $(v, \{u_1, \dots, u_k\})$ arrives

Represent this item as a sequence of edges:

$$\{v, u_1\}, \{v, u_2\}, \dots, \{v, u_k\}$$

Feed these edges into ALG, make the same decisions as ALG does.

QED

Claim: EAM $\leq$ VAM-PH

Pf:

ALG – algorithm in VAM-PH

In EAM model, item $(v, \{\ell_1, \ell_2, \ldots, \ell_k\})$ arrives

Explore previous items $u \prec v$ and see if any of the ℓ_i appear in u's list

Thus, we can construct $\{u : u \prec v \; and \; u \sim v\}$

Feed this item into ALG, make the same decisions as ALG does.

QED

Claim: VAM-FI $\leq$ EAM

Pf:

ALG – algorithm in EAM

In VAM-FI model, item $(v, \{u_1, u_2, \dots, u_k\})$ arrives

If edge $\{v, u_i\}$ hasn't appeared before, assign it a new label ℓ_i

If edge $\{v, u_i\}$ has appeared before, use the previously assigned label ℓ_i

Thus, we can construct item $(v, \{\ell_1, \ell_2, \dots, \ell_k\})$

Feed this item into ALG, make the same decisions as ALG does.

QED

Discussion of models

EM:

typically decisions about edges

VAM-PH:

typically decisions about vertices; natural for dynamically growing graphs, e.g., social networks

EAM, VAM-FI:

typically decisions about vertices; not very natural;

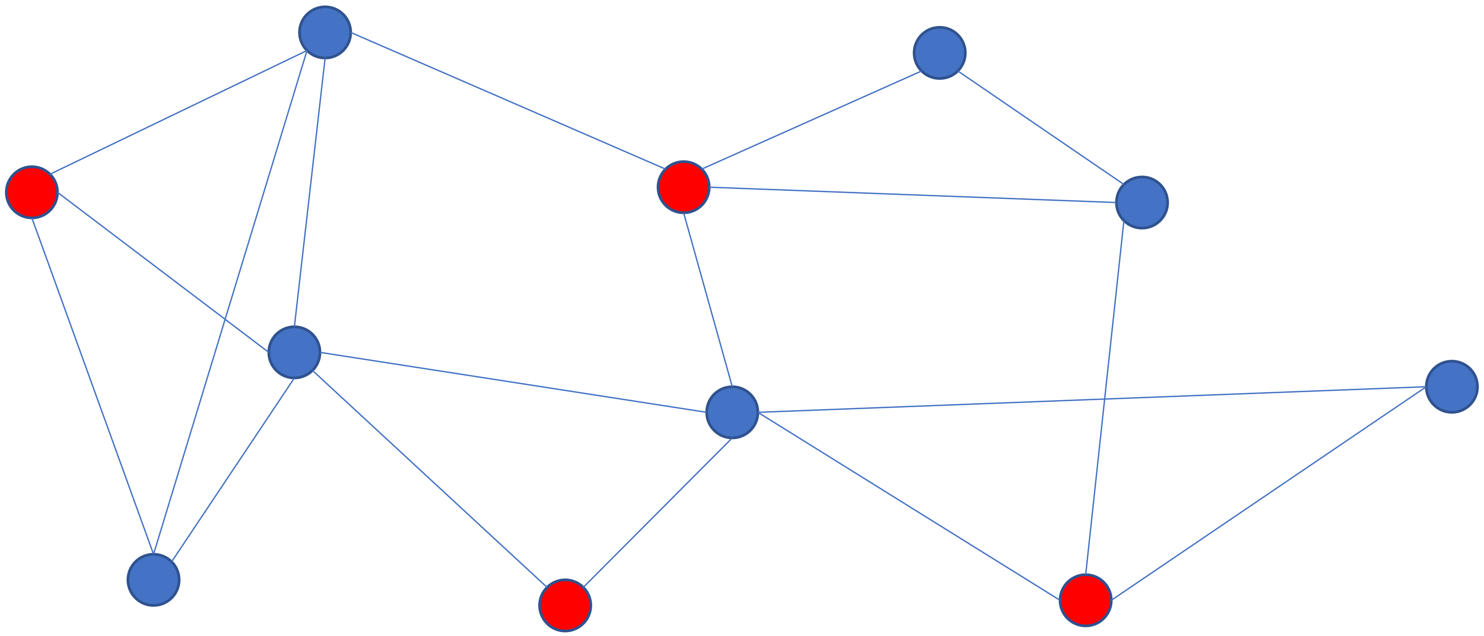
can be considered to be a strengthening of degree information

Hard Online Graph Problems

Independent Set

Set of nodes such that there are no edges between any pair of nodes

In social networks: set of people such that none of them are friends



Maximum Independent Set

Input: $G = (V, E, \prec)$ unweighted undirected graph

$\prec$ is a total order on V

$(v_1, N_1), (v_2, N_2), \dots, (v_n, N_n)$ input sequence, where

$v_1 \prec v_2 \prec \cdots \prec v_n$ and

$N_i = N(v_i) \cap \{v_j : j < i\}$

Output: $d_1, \dots, d_n$ where $d_i \in \{0,1\}$ indicates whether to include v_i in the independent set or not

Objective: maximize the size of $S = \{v_i : d_i = 1\}$ subject to S being an independent set

Theorem	Let ALG be a deterministic algorithm for Maximum Independent Set problem in the VAM-PH input model. $\rho(ALG) \geq n - 1$

Pf:

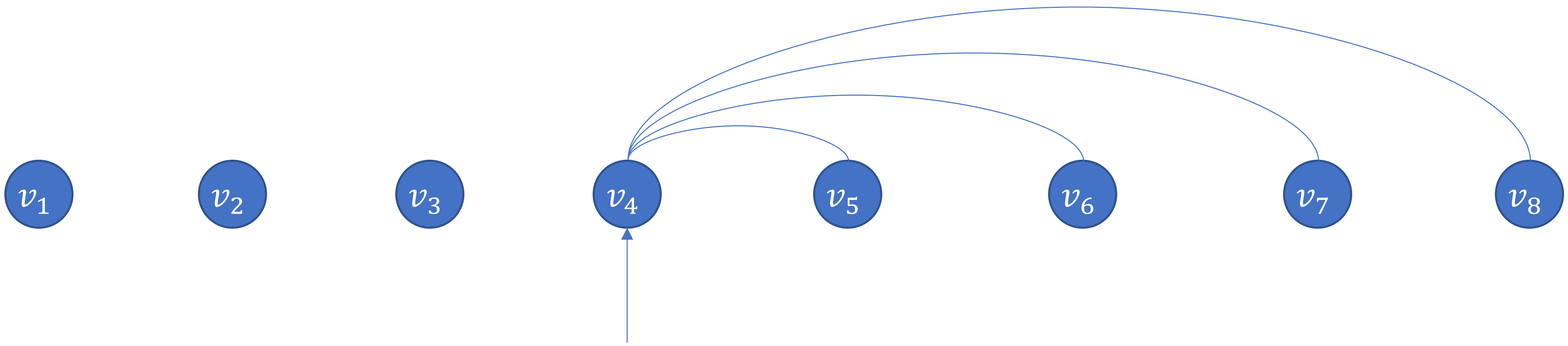
Adversary presents isolated vertices until ALG accepts one such vertex

After that adversary presents vertices that have a single neighbor – the vertex previously accepted by ALG

Theorem

Let ALG be a deterministic algorithm for Maximum Independent Set problem in the VAM-PH input model.

$$\rho(ALG) \geq n - 1$$

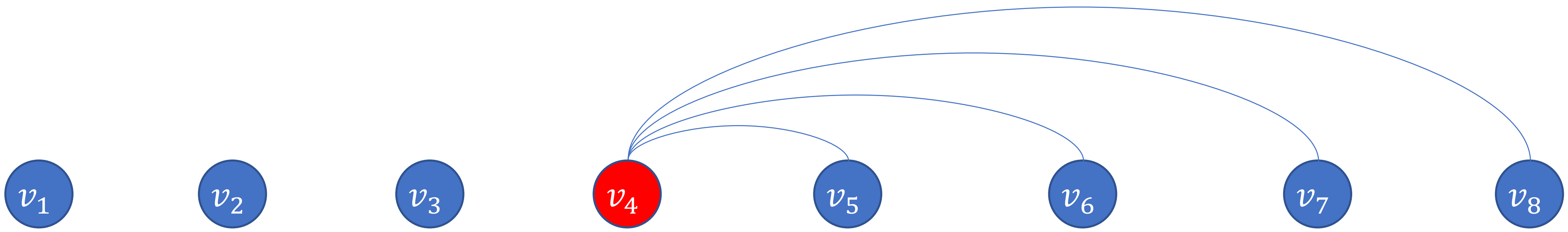


ALG accepts

Theorem

Let ALG be a deterministic algorithm for Maximum Independent Set problem in the VAM-PH input model.

$$\rho(ALG) \geq n - 1$$

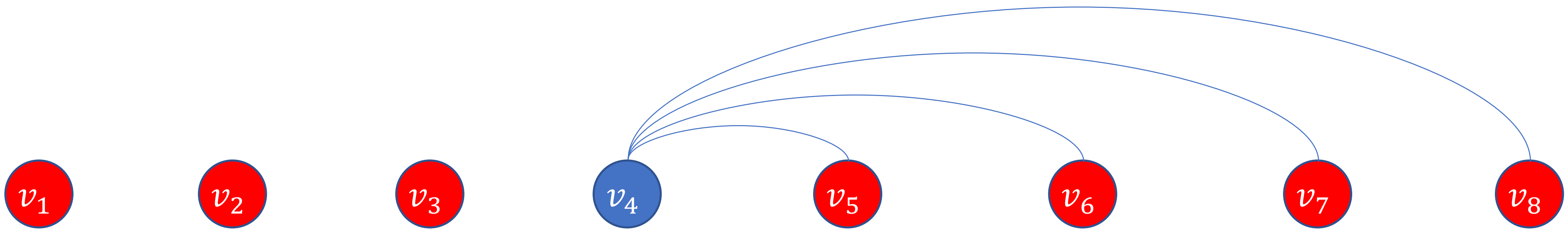


$$ALG = 1$$

Theorem

Let ALG be a deterministic algorithm for Maximum Independent Set problem in the VAM-PH input model.

$$\rho(ALG) \geq n - 1$$



$$OPT = n - 1$$

Theorem	Let ALG be a **randomized** algorithm for Maximum Independent Set problem in the VAM-PH input model. $\rho_{OBL}(ALG) = \Omega(n)$

Pf:

Use Yao's minimax principle

Find **random input**, on which every deterministic algorithm has competitive ratio $\Omega(n)$ in **expectation**

Theorem

Let ALG be a **randomized** algorithm for Maximum Independent Set problem in the VAM-PH input model.

$$\rho_{OBL}(ALG) = \Omega(n)$$

Pf:

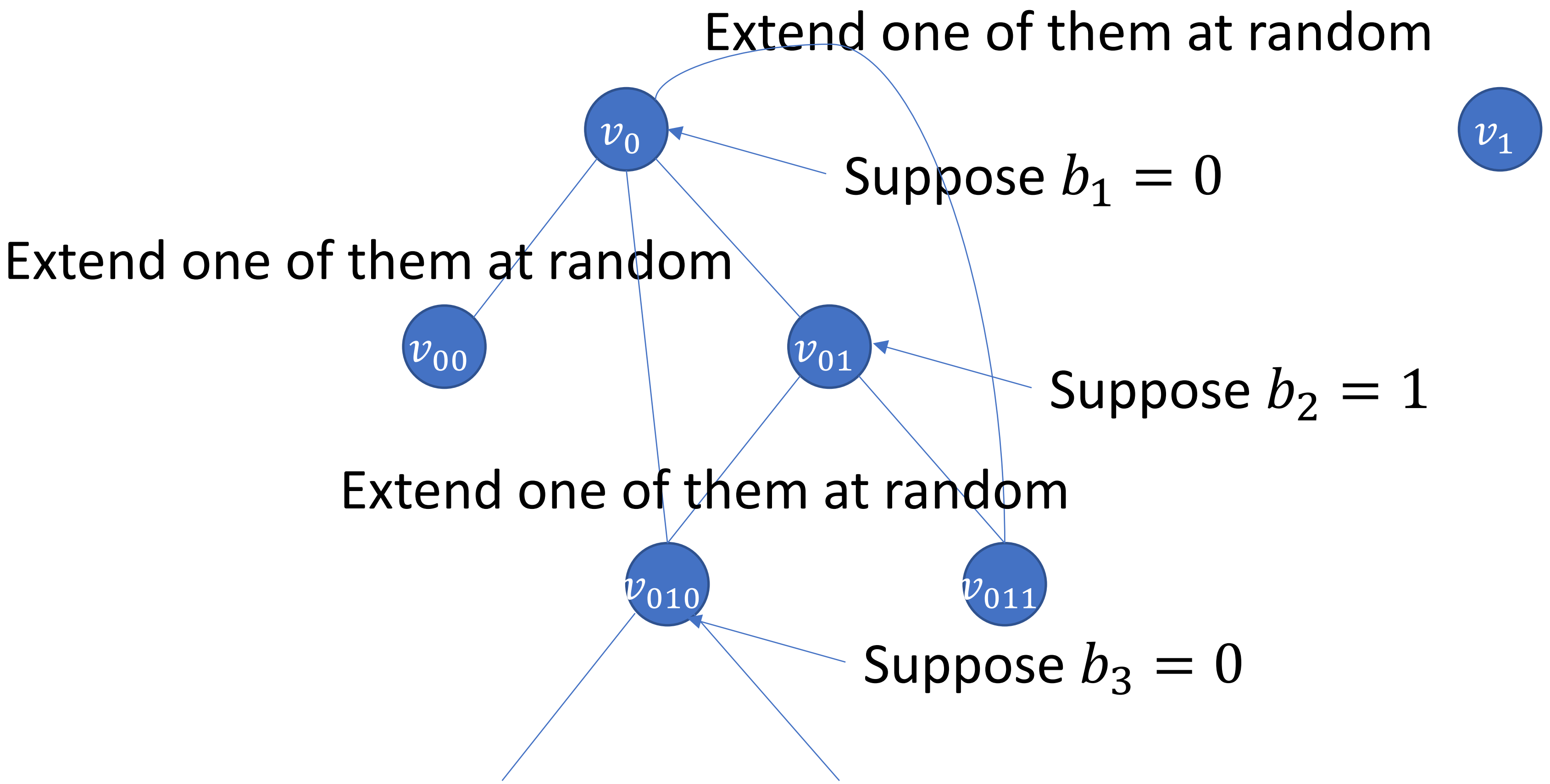
Random input procedure:

1. Present two isolated vertices v_0 and v_1
2. Pick a random bit $b_1 \in \{0,1\}$
3. Present two vertices $v_{b_1 0}$ and $v_{b_1 1}$. They have v_{b_1} as the neighbor
4. Pick a random bit $b_2 \in \{0,1\}$
5. Present two vertices $v_{b_1 b_2 0}$ and $v_{b_1 b_2 1}$. They have v_{b_1} and $v_{b_1 b_2}$ as neighbors
6. So on…

Theorem

Let ALG be a **randomized** algorithm for Maximum Independent Set problem in the VAM-PH input model.

$$\rho_{OBL}(ALG) = \Omega(n)$$



Theorem	Let ALG be a **randomized** algorithm for Maximum Independent Set problem in the VAM-PH input model. $\rho_{OBL}(ALG) = \Omega(n)$

Pf:

If we select **non-extended** node at each step, then we get an independent set. Size = $\frac{n}{2}$ therefore $OPT \geq \frac{n}{2}$

If we select **extended** node, then we cannot select any of the following nodes, since they are all connected to the extended node.

ALG has 50-50 chance of guessing the right non-extended node in each round.

On average ALG will make 2 correct guesses.

QED

Proof technique: online reduction

Two online problems: P1 and P2

We say P1 **reduces to** P2 if any online algorithm for P2 can be transformed into an online algorithm for P1 while preserving competitive ratio guarantees

P1 $\leq$ P2 : P1 reduces to P2, mnemonic “P1 is easier than P2”

P1 $\leq$ P2 means two things:

(1) Upper bound for P2 implies an upper bound for P1

(2) Lower bound for P1 implies a lower bound for P2

Reduction steps:

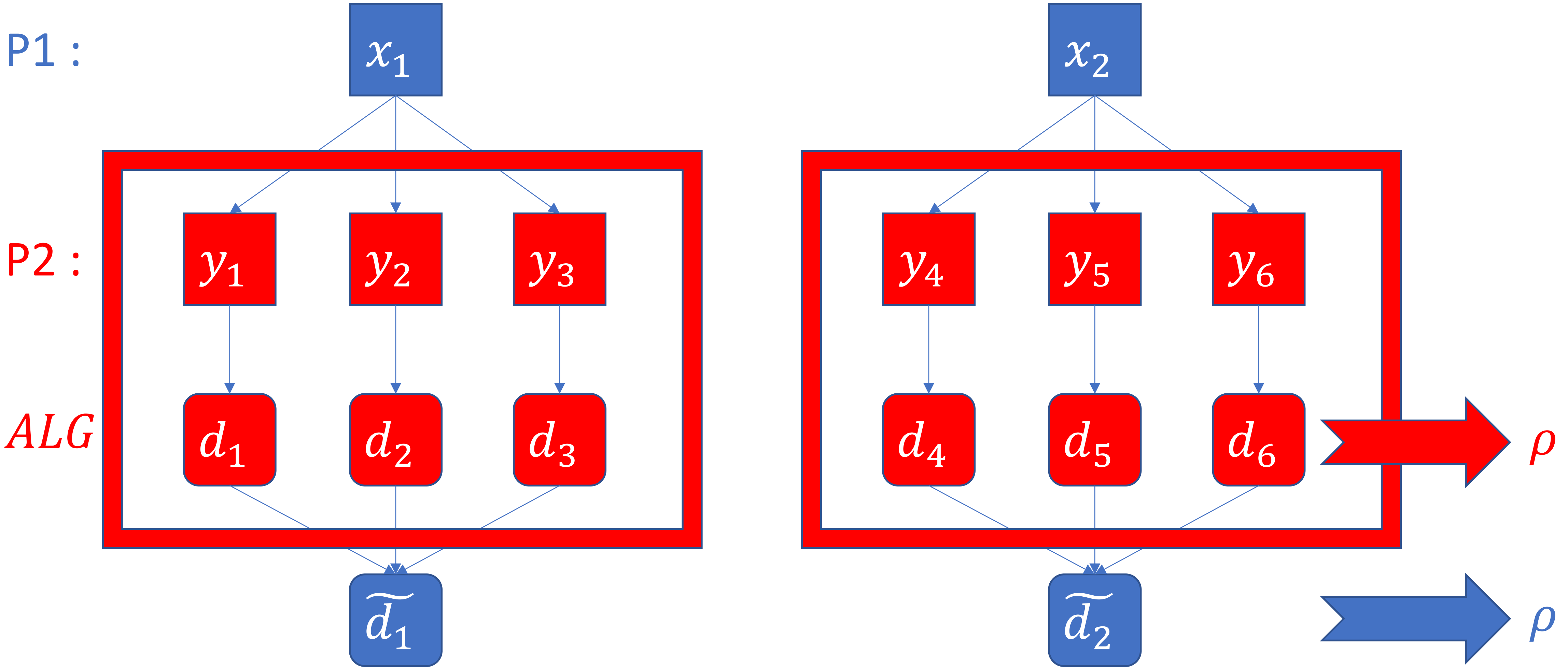
(1) Start with an arbitrary algorithm ALG for P2

(2) Transform input $x_1, \dots, x_n$ for P1 into input $y_1, \dots, y_m$ for P2 in an **online fashion**

(3) Transform decisions $d_1, \dots, d_m$ of ALG for P2 into decisions $\widetilde{d_1}, \dots, \widetilde{d_n}$ for P1 in an **online fashion**

(4) Prove that if decisions $d_1, \dots, d_m$ guarantee competitive ratio ρ for P2 then $\widetilde{d_1}, \dots, \widetilde{d_n}$ also guarantee C.R. ρ for P1

P1 reduces to P2 : P1 $\leq$ P2

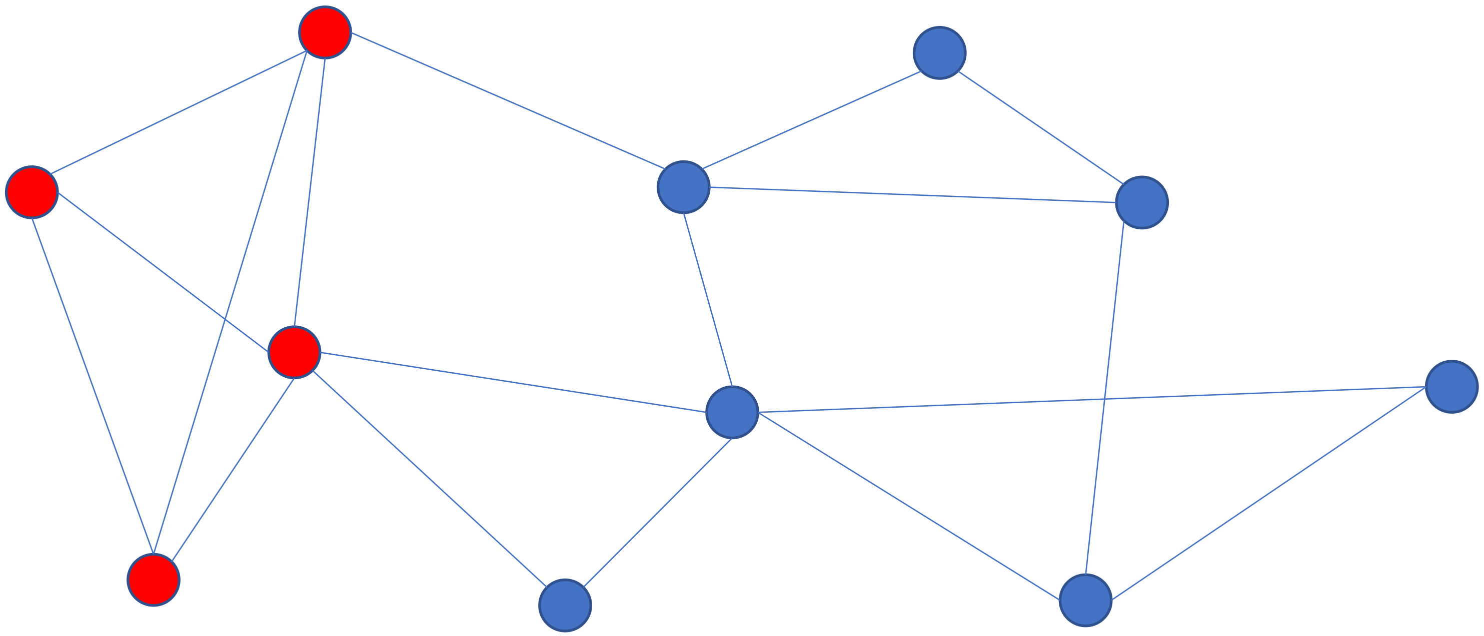


Reduction example: Maximum Clique

Clique

Set of nodes such that every pair of nodes forms an edge

In social networks: group of people where everybody is friends with each other



Maximum Clique

Input: $G = (V, E, \prec)$ unweighted undirected graph

$\prec$ is a total order on V

$(v_1, N_1), (v_2, N_2), \ldots, (v_n, N_n)$ input sequence, where

$v_1 \prec v_2 \prec \cdots \prec v_n$ and

$N_i = N(v_i) \cap \{v_j : j < i\}$

Output: $d_1, \ldots, d_n$ where $d_i \in \{0,1\}$ indicates whether to include v_i in the clique or not

Objective: maximize the size of $S = \{v_i : d_i = 1\}$ subject to S being a clique

Maximum Independent Set $\leq$ Maximum Clique

Idea:

complement graph

construct it online in VAM-PH input model

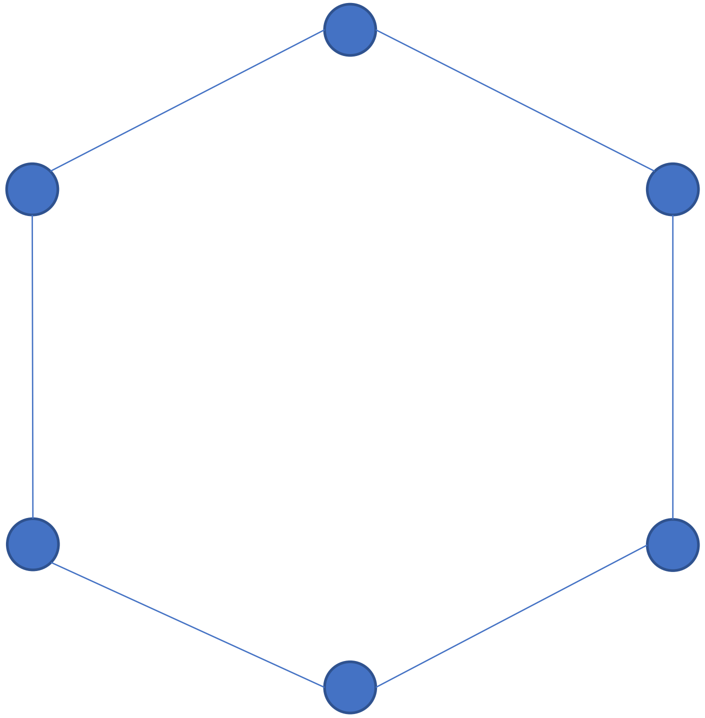
Graph $G = (V, E)$

Complement graph $G^c = \left(V, \binom{V}{2} \setminus E\right)$

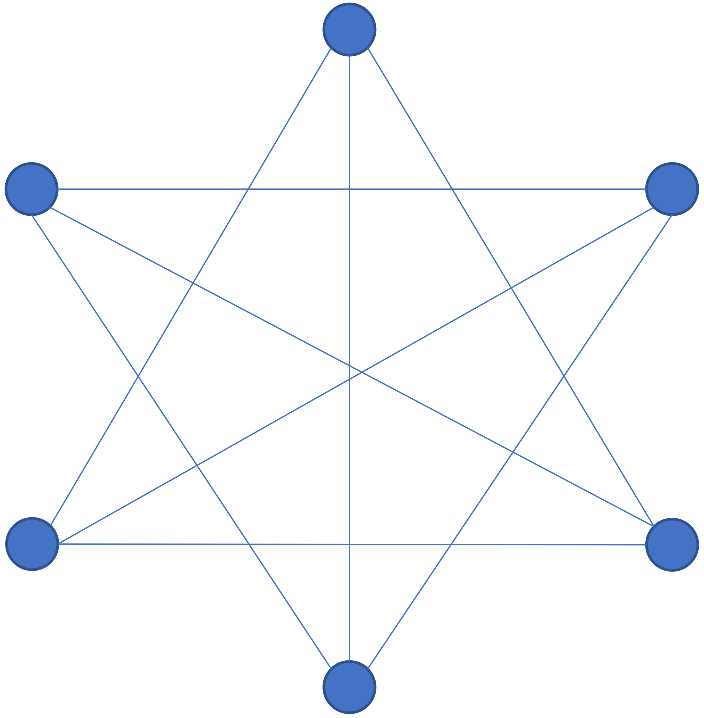
$S \subseteq V$ is an **independent set** in $G \Leftrightarrow S$ is a **clique** in G^c

S is a **clique** in $G \Leftrightarrow S$ is an **independent set** in G^c

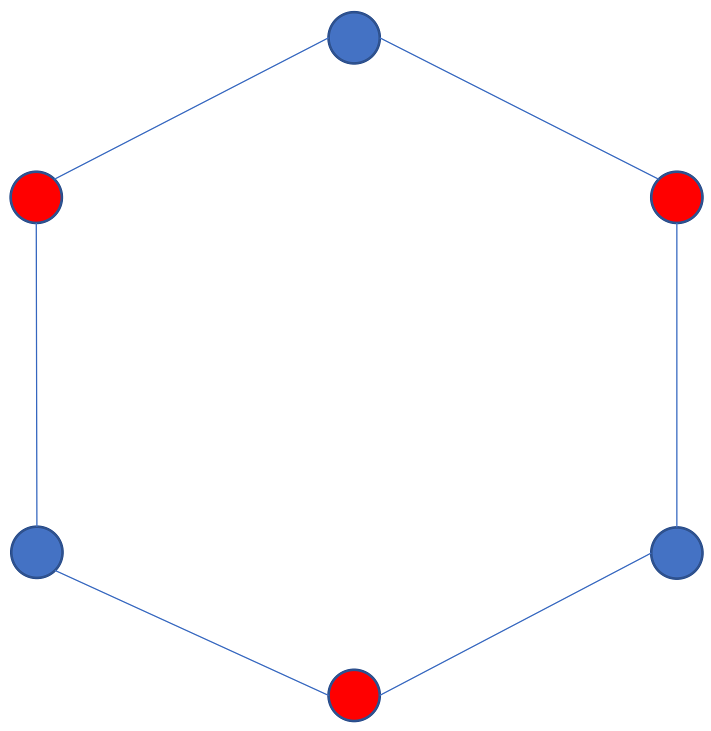
G



G^c

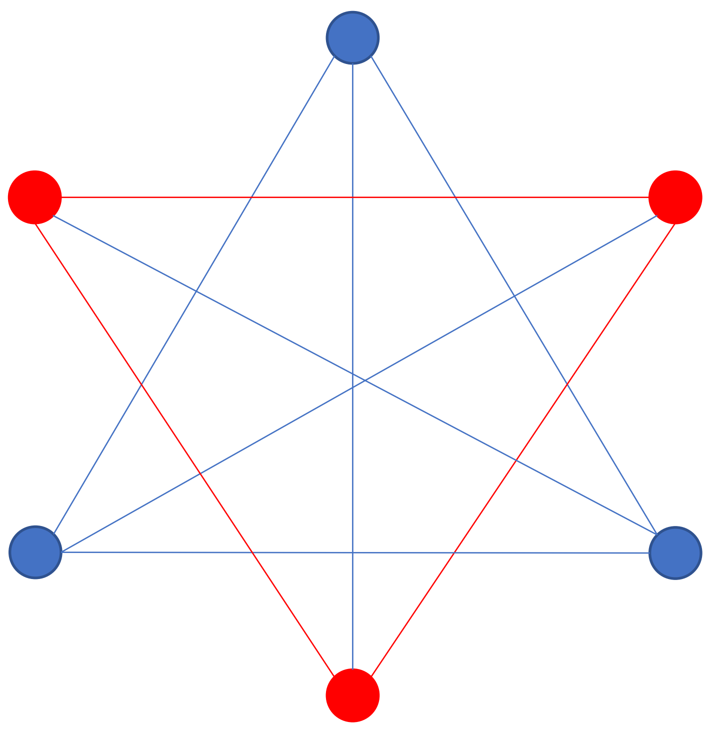


G



Independent set

G^c



Clique

$MIS(G)$ – maximum independent set in G

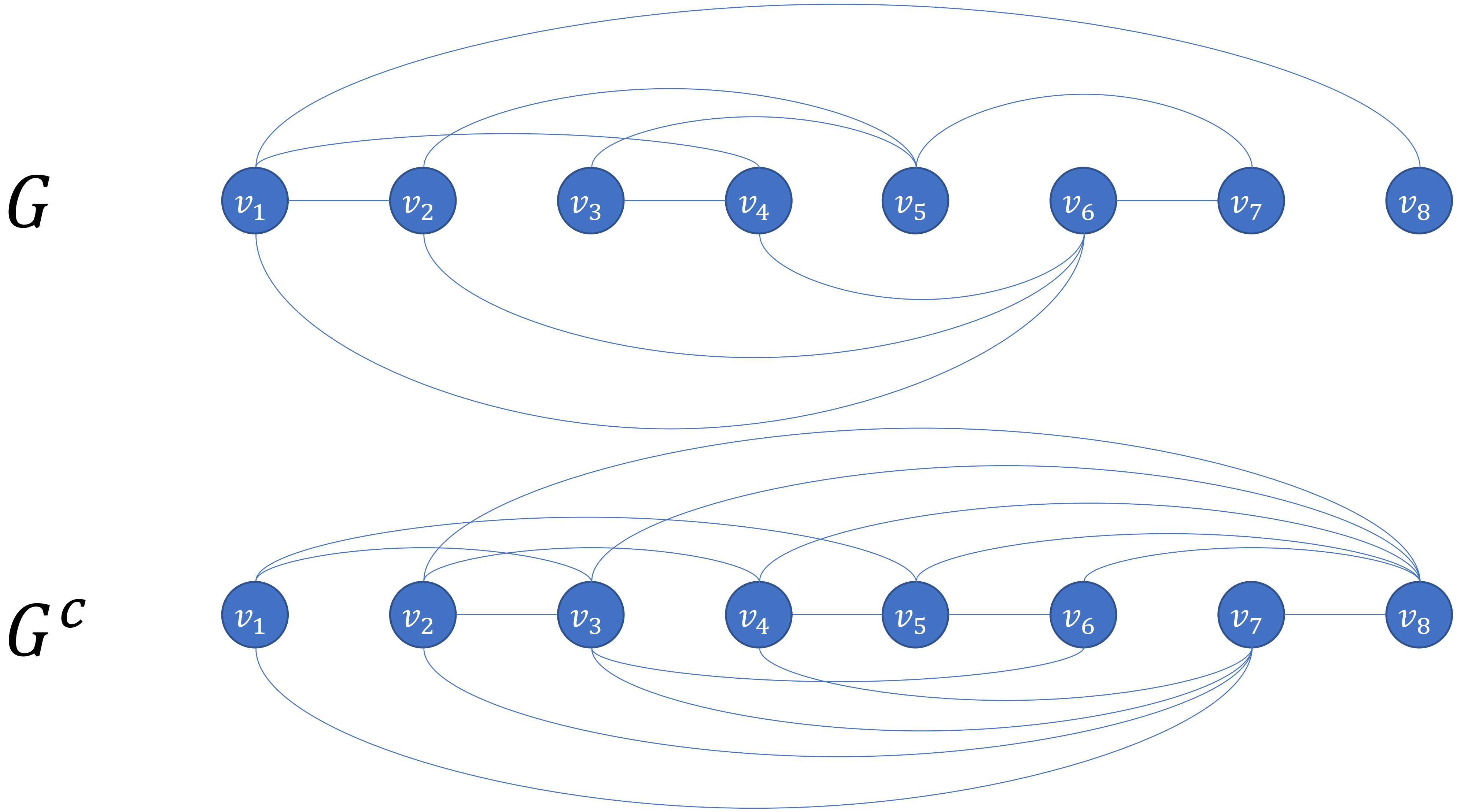
$MC(G)$ – maximum clique in G

In particular, we have:

$$MIS(G) = MC(G^{\mathrm{c}})$$

$$MC(G) = MIS(G^{c})$$

Constructing complement graph online



Constructing complement graph online

G: input items $(v_1, N_1), (v_2, N_2), \ldots, (v_i, N_i), \ldots, (v_n, N_n)$

G^c:

receive (v_i, N_i)

construct (v_i, N_i^c) where

$$N_i^c := \{v_1, \ldots, v_{i-1}\} \setminus N_i$$

Theorem

Maximum Independent Set $\leq$ Maximum Clique in the VAM-PH input model

Proof:

Let ALG be an online algorithm for Maximum Clique

Define ALG' for Maximum Independent Set on input G

(1) Construct online complement graph G^c

(2) Feed G^c to ALG, use same decisions as ALG

Theorem

Maximum Independent Set $\leq$ Maximum Clique in the VAM-PH input model

Proof:

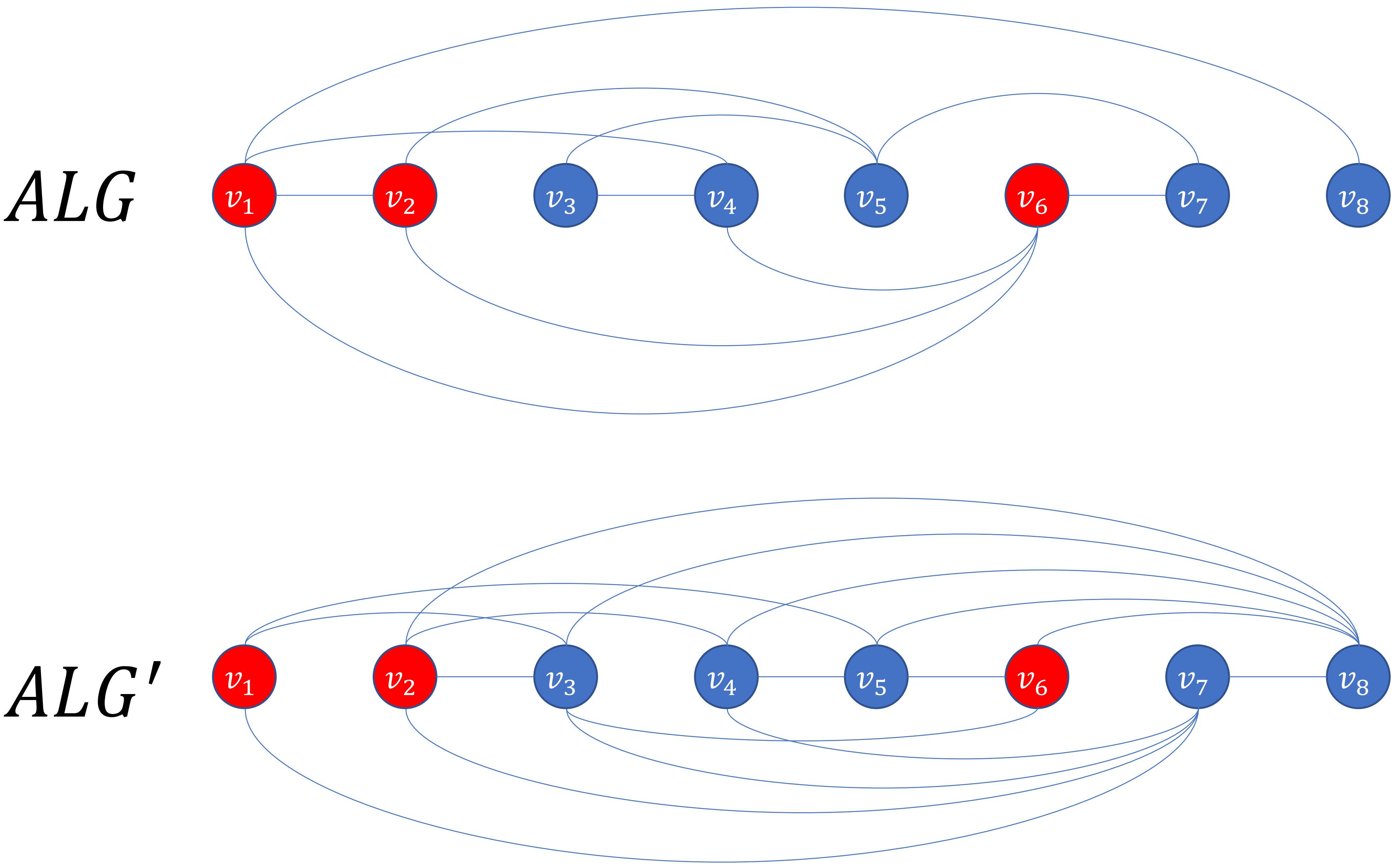
ALG finds S – a clique in G^c

ALG' finds same S – an independent set in G

$$\frac{|S|}{MC(G^c)} = \frac{|S|}{MIS(G)}$$

Hence competitive ratio is preserved for both randomized and deterministic algorithms.

QED



ALG
v1
v2
v3
v4
v5
v6
v7
v8
ALG′
v1
v2
v3
v4
v5
v6
v7
v8

Theorem

Let ALG be a deterministic algorithm for Maximum Independent Set problem in the VAM-PH input model.
$\rho(ALG) \geq n - 1$

Theorem

Let ALG be a **randomized** algorithm for Maximum Independent Set problem in the VAM-PH input model.
$\rho_{OBL}(ALG) = \Omega(n)$

Theorem

Maximum Independent Set $\leq$ Maximum Clique in the VAM-PH input model

Corollary 1

Let ALG be a deterministic algorithm for Maximum Clique in the VAM-PH input model. Then

$$\rho(ALG) \geq n - 1$$

Corollary 2

Let ALG be a randomized algorithm for Maximum Clique in the VAM-PH input model. Then

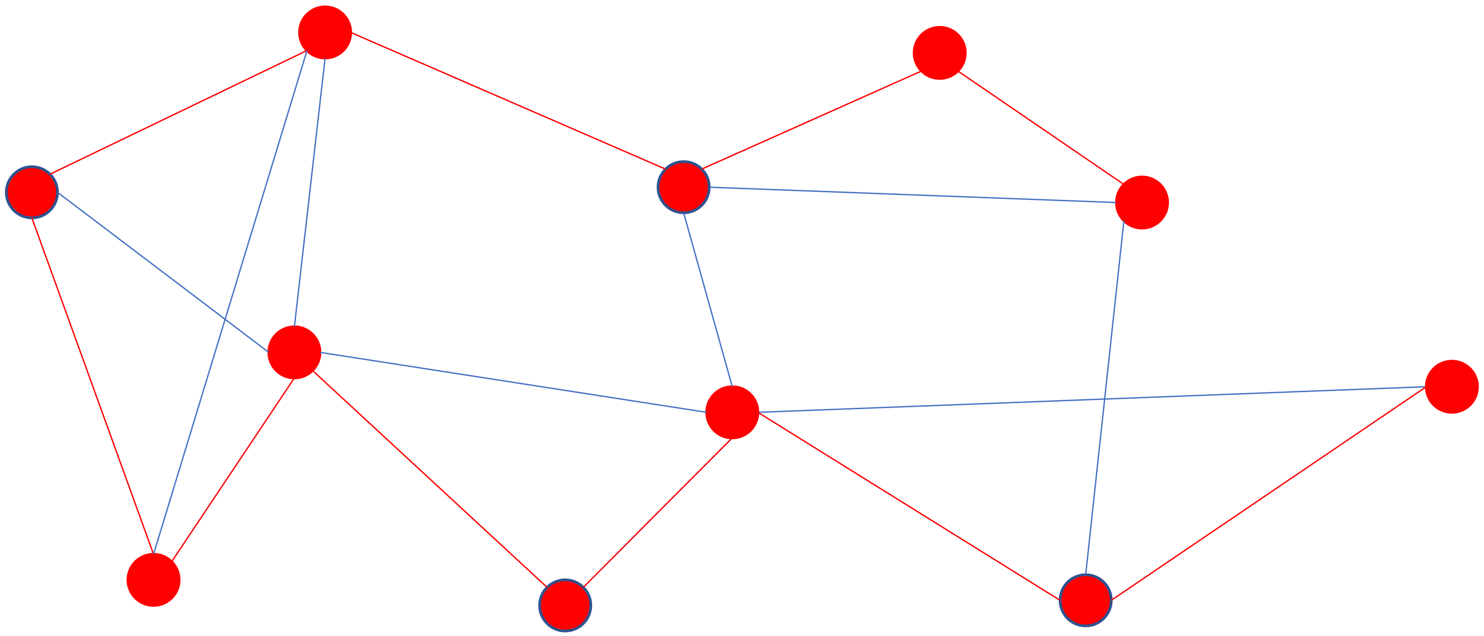
$$\rho_{OBL}(ALG) = \Omega(n)$$

Longest Path

Simple path

A path is called **simple** if it does not have repeated vertices

Longest Path problem: find a simple path that is as long as possible



Longest Path

Input: $G = (V, E, \prec)$ unweighted undirected graph

$\prec$ is a total order on E

$e_1 \prec e_2 \prec \cdots \prec e_m$ input sequence

Output: $d_1, \ldots, d_m$ where $d_i \in \{0,1\}$ indicates whether to include e_i in the path or not

Objective: maximize the length of the constructed path

$P = \{e_i : d_i = 1\}$ subject to P being a valid path

Theorem

Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have

$$\rho(ALG) \geq n - 3$$

Proof

Adversary fixes $V = \{v_1, \dots, v_n\}$

Adversary keeps presenting

$$\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_i, v_{i+1}\}$$

Until ALG accepts $\{v_i, v_{i+1}\}$ or until $i + 1 = n$

Theorem Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have

$$\rho(ALG) \geq n - 3$$

Proof

Case 1:

ALG doesn't accept any edges

adversary presents $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$

$OPT = n - 1$

$ALG = 0$

infinite competitive ratio

Theorem

Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have

$$\rho(ALG) \geq n - 3$$

Proof

Case 1:

ALG doesn't accept anything

Theorem	Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have $$\rho(ALG) \geq n - 3$$

Proof

Case 2:

ALG accepts $\{v_1, v_2\}$

adversary presents $\{v_3, v_4\}, \{v_4, v_5\} \dots, \{v_{n-1}, v_n\}$

$OPT = n - 3$

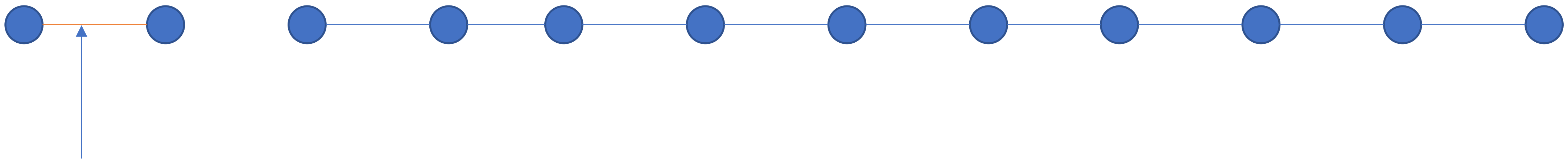
$ALG = 1$

competitive ratio $\geq n - 3$

Theorem	Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have $$\rho(ALG) \geq n - 3$$

Proof

Case 2:



ALG accepts

Theorem

Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have

$$\rho(ALG) \geq n - 3$$

Proof

Case 3:

ALG accepts $\{v_i, v_{i+1}\}$ for $i > 1$

adversary presents $\{v_{i-1}, v_{i+2}\}, \{v_{i+2}, v_{i+3}\} \dots, \{v_{n-1}, v_n\}$

$OPT = n - 3$

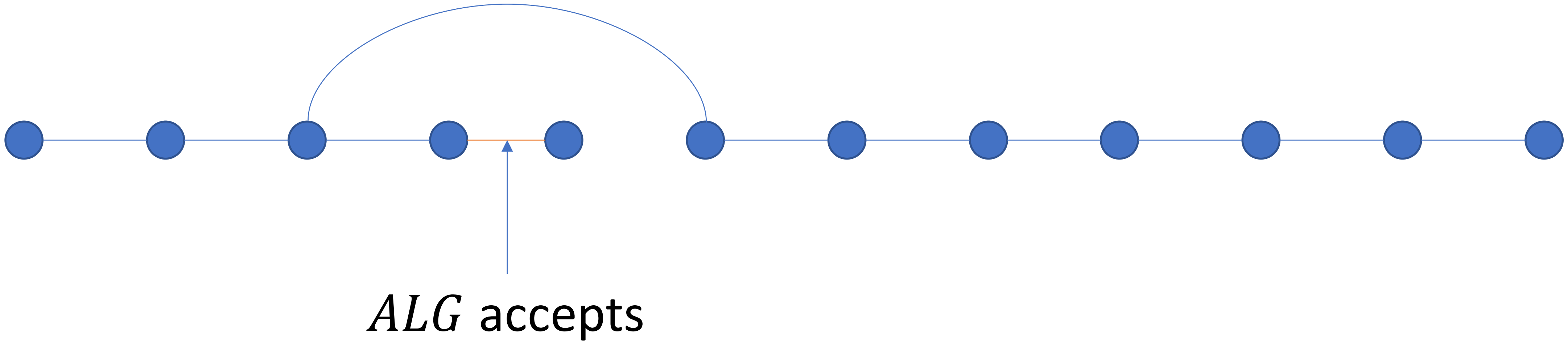
$ALG = 1$

competitive ratio $\geq n - 3$

Theorem	Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have $$\rho(ALG) \geq n - 3$$

Proof

Case 3:



Theorem	Let ALG be a deterministic algorithm for the Longest Path problem in the EM model. Then we have $\rho(ALG) \geq n - 3$

Proof

Overall get $\rho(ALG) \geq n - 3$

QED

Using a slightly more complicated argument and Yao's minimax principle, one can show the following:

Theorem	Let ALG be a randomized algorithm for the Longest Path problem in the EM model. Then we have $\rho_{OBL}(ALG) = \Omega(n)$

Minimum Spanning Tree (MST)

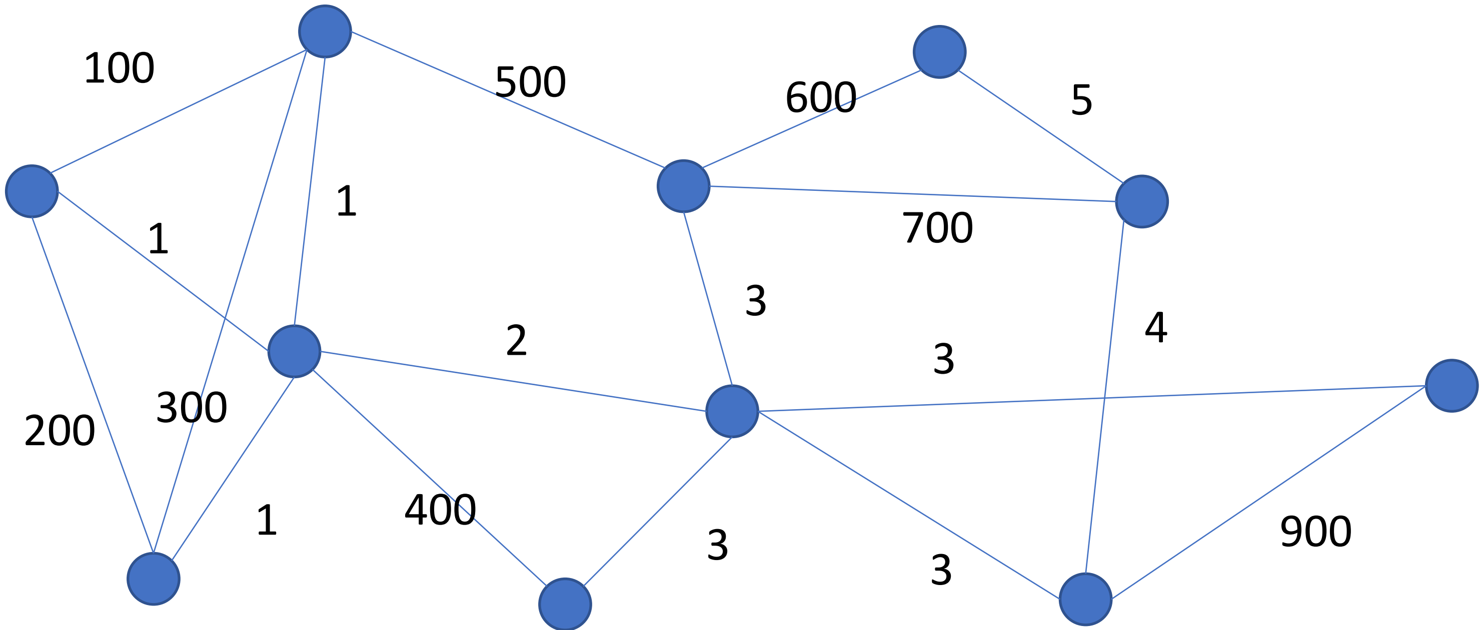
Minimum Spanning Tree

Graph $G = (V, E)$ with edge-weights $w : E \to \mathbb{R}$

Spanning tree $T = (V, E' \subseteq E)$ – connected acyclic subgraph of G

Weight of tree T is $w(T) = \sum_{e \in E'} w(e)$

Minimum spanning tree problem: find a tree of minimum weight



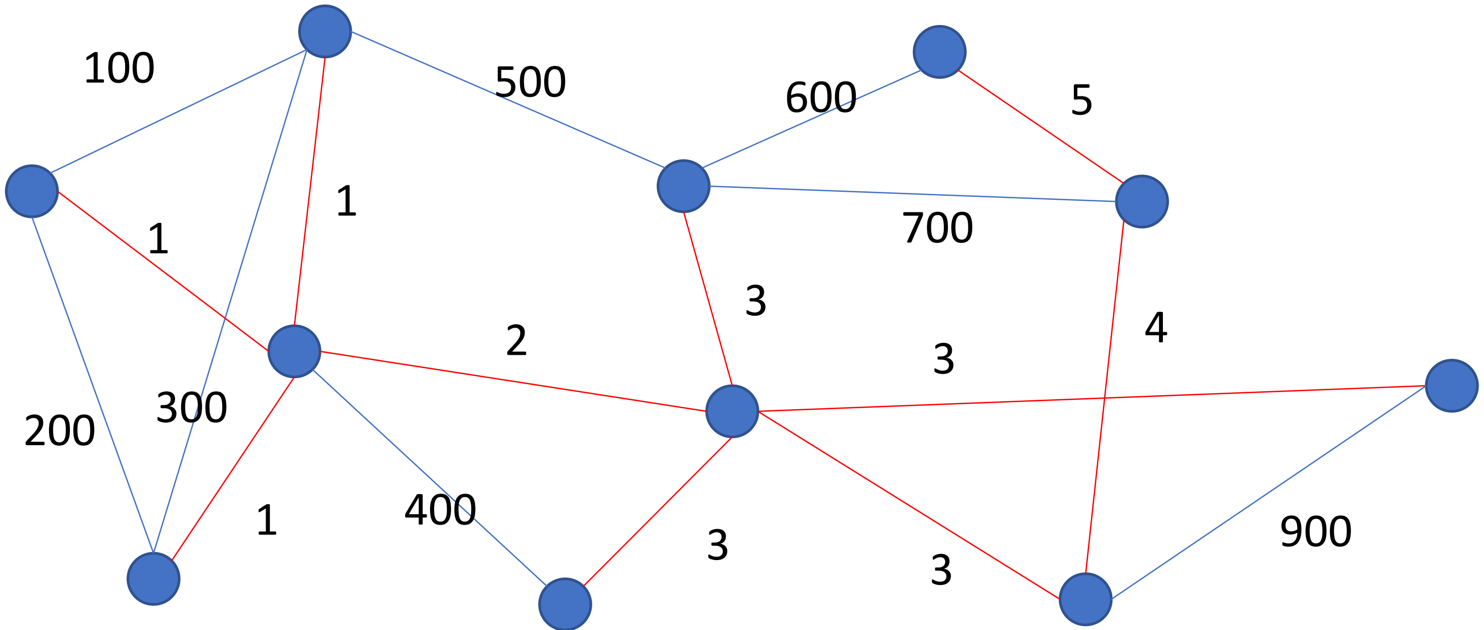
Minimum Spanning Tree

Graph $G = (V, E)$ with edge-weights $w : E \to \mathbb{R}$

Spanning tree $T = (V, E' \subseteq E)$ – connected acyclic subgraph of G

Weight of tree T is $w(T) = \sum_{e \in E'} w(e)$

Minimum spanning tree problem: find a tree of minimum weight



Minimum Spanning Tree

Input: $G = (V, E, w, \prec)$ edge-weighted undirected graph

$\prec$ is a total order on E

$w : E \to \mathbb{R}$ weights of edges

$(e_1, w(e_1)), (e_2, w(e_2)), \dots, (e_m, w(e_m))$ input sequence

Output: $d_1, \dots, d_m$ where $d_i \in \{0,1\}$ indicates whether to include e_i in the tree or not

Objective: minimize the weight $w(T)$ of the constructed spanning tree $T = (V, \{e_i : d_i = 1\})$ subject to T being a spanning tree

Theorem

Let ALG be a randomized algorithm for the Minimum Spanning Tree problem in the EM model. Then we have $\rho_{OBL}(ALG) = \Omega(n)$

Proof

We will use Yao's minimax principle

- present a distribution on inputs and show that
- every deterministic online algorithm has competitive ratio $\Omega(n)$ on average with respect to that distribution

Theorem

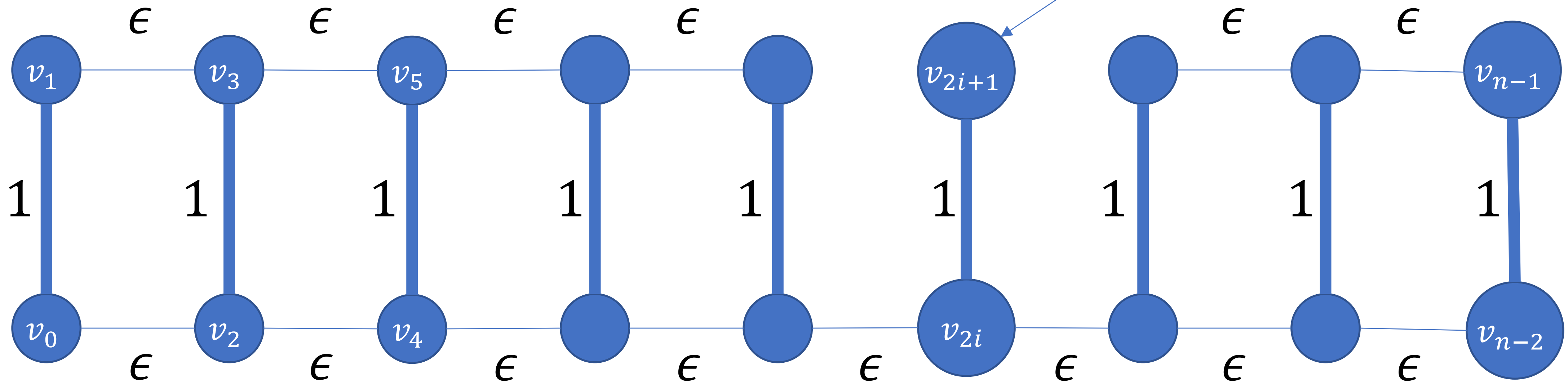
Let ALG be a randomized algorithm for the Minimum Spanning Tree problem in the EM model. Then we have $\rho_{OBL}(ALG) = \Omega(n)$

Proof

choose **random** $i \in \left\{0, 1, \dots, \frac{n}{2} - 1\right\}$

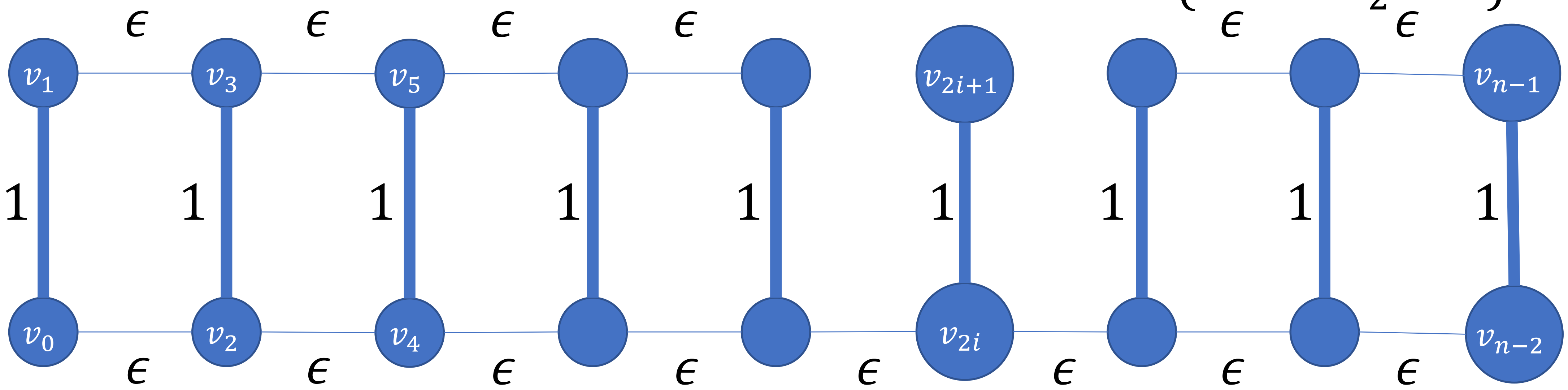
Adversary fixes $V = \{v_0, v_1, v_2, \dots, v_{n-1}\}$, n is even

Fix $\epsilon > 0$ arbitrary



Proof

choose **random** $i \in \left\{0, 1, \dots, \frac{n}{2} - 1\right\}$



Adversary presents:

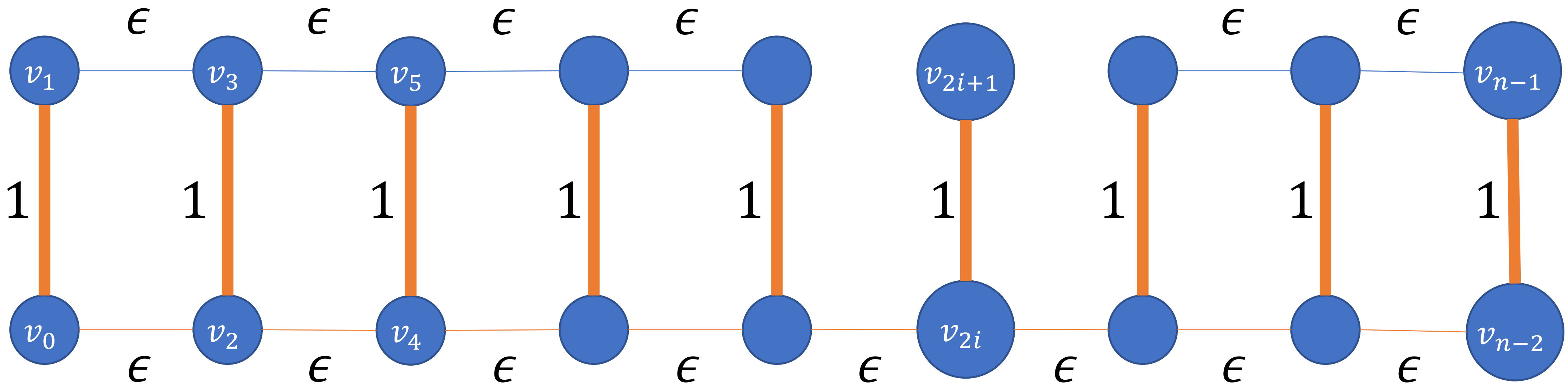
$\{v_0, v_1\}, \{v_0, v_2\}, \quad \{v_2, v_3\}, \{v_2, v_4\}, \quad \{v_4, v_5\}, \{v_4, v_6\}, \dots, \{v_{n-2}, v_{n-1}\}$

Followed by:

$\{v_1, v_3\}, \{v_3, v_5\}, \{v_5, v_7\}, \dots, \{v_{2i-3}, v_{2i-1}\}, \{v_{2i+3}, v_{2i+5}\}, \dots, \{v_{n-3}, v_{n-1}\}$

Proof

Solution of the algorithm:

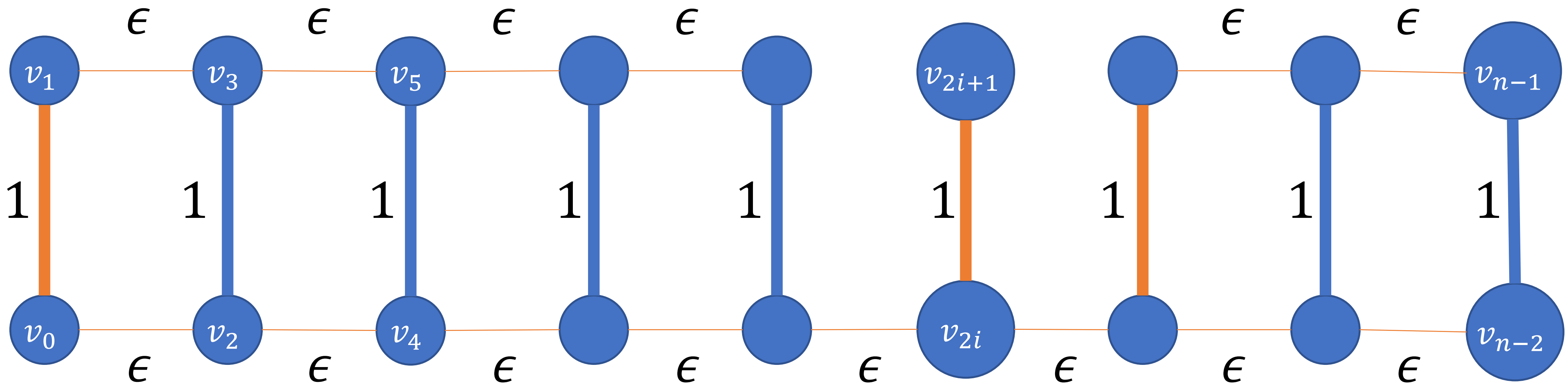


Why?

Therefore $\boldsymbol{ALG \geq \frac{n}{2}}$

Proof

Optimal Solution:



Therefore $OPT \leq 3 + \epsilon\left(\frac{n}{2} - 3\right)$

Theorem	Let ALG be a randomized algorithm for the Minimum Spanning Tree problem in the EM model. Then we have $\rho_{OBL}(ALG) = \Omega(n)$

Proof

$$ALG \geq \frac{n}{2}$$

$$OPT \leq 3 + \epsilon\left(\frac{n}{2} - 3\right)$$

For sufficiently small ϵ we have $OPT \leq 4$

Therefore the competitive ratio is $\Omega(n)$

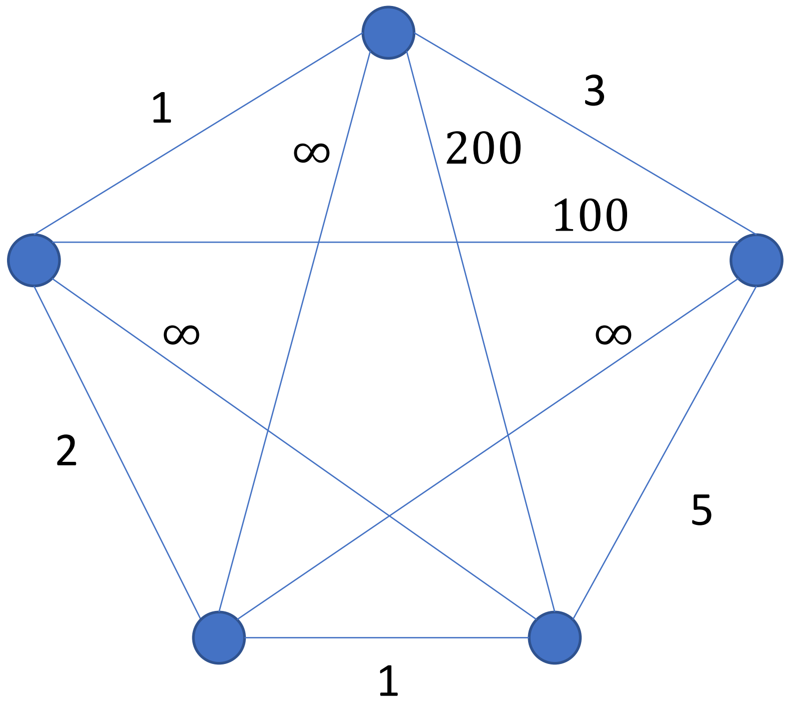
QED

Travelling Salesperson Problem (TSP)

Travelling Salesperson Problem

Complete graph $G = \left(V, \binom{V}{2}\right)$ with edge-weights $w : \binom{V}{2} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$

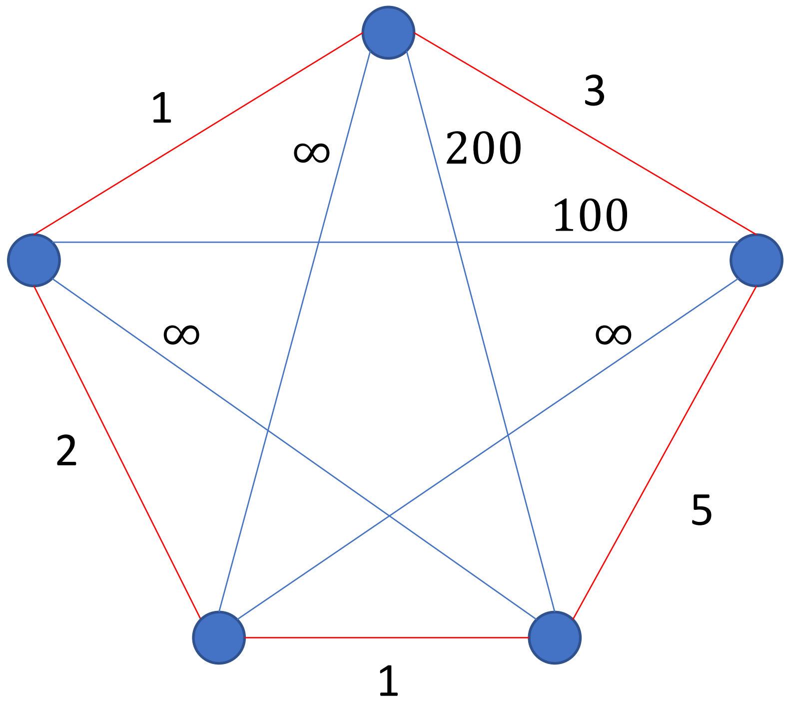
TSP: find a minimum weight route that visits every vertex exactly once and returns back to the original vertex



Travelling Salesperson Problem

Complete graph $G = \left(V, \binom{V}{2}\right)$ with edge-weights $w : \binom{V}{2} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$

TSP: find a minimum weight route that visits every vertex exactly once and returns back to the original vertex



Travelling Salesperson Problem

Input: $G = (V, E, w, \prec)$ edge-weighted undirected complete graph

$\prec$ is a total order on E

$w : E \to \mathbb{R}$ weights of edges

$(e_1, w(e_1)), (e_2, w(e_2)), \ldots, (e_m, w(e_m))$ input sequence

Output: $d_1, \ldots, d_m$ where $d_i \in \{0,1\}$ indicates whether to include v_i in the route or not

Objective: minimize the total weight $w(C)$ of the constructed route $C = \{e_i : d_i = 1\}$ subject to C being a route that visits every vertex exactly once

Theorem

Let ALG be a deterministic algorithm for TSP in the EM model. Then we have

$$\rho(ALG) = \infty$$

Proof

Adversary fixes $V = \{v_1, \dots, v_n\}$

First, adversary presents items

$$(\{v_1, v_2\}, 1), (\{v_1, v_3\}, 1), (\{v_1, v_4\}, 1), \dots$$

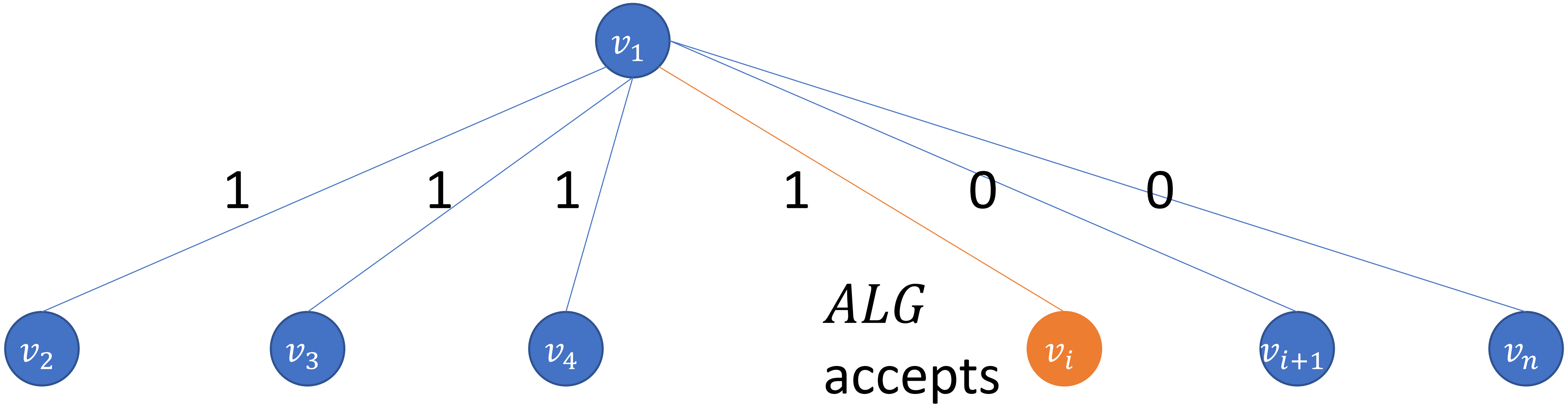
Consider i such that the adversary accepts $\{v_1, v_i\}$ for the first time

Theorem	Let ALG be a deterministic algorithm for TSP in the EM model. Then we have $\rho(ALG) = \infty$

Proof

Case ALG accepts $\{v_1, v_i\}$ for $i \leq n-2$ for the first time

Adversary presents $(\{v_1, v_{i+1}\}, 0), (\{v_1, v_{i+2}\}, 0), \ldots, (\{v_i, v_n\}, 0)$



Theorem

Let ALG be a deterministic algorithm for TSP in the EM model. Then we have

$$\rho(ALG) = \infty$$

Proof

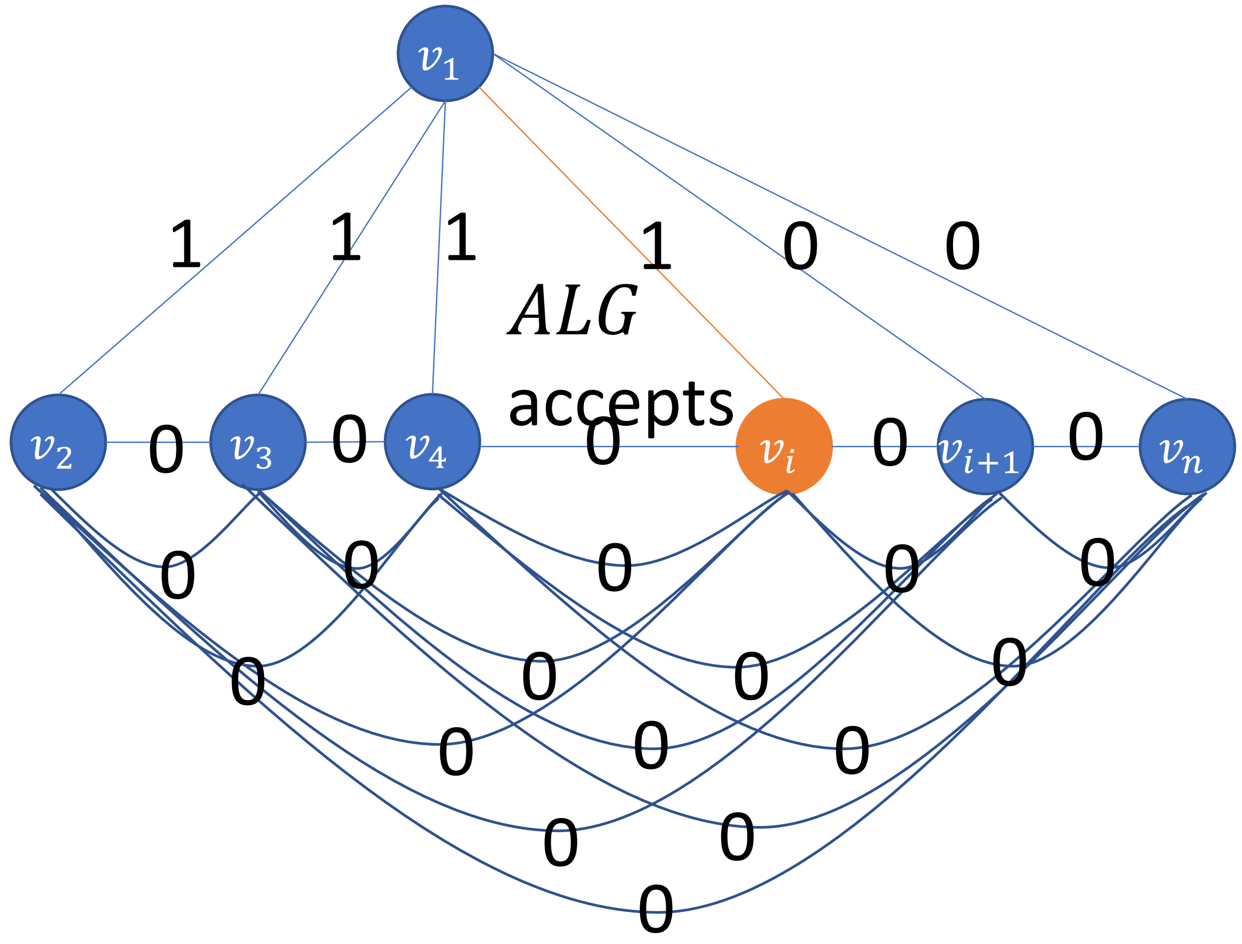
Case ALG accepts $\{v_1, v_i\}$ for $i \leq n - 2$ for the first time

Adversary presents $(\{v_1, v_{i+1}\}, 0), (\{v_1, v_{i+2}\}, 0), \dots, (\{v_i, v_n\}, 0)$

The rest of the edges are presented in arbitrary order with 0 weights

Then $ALG = 1$ and $OPT = 0$

Thus, competitive ratio is ∞



v_1
v_2
v_3
v_4
v_i
v_{i+1}
v_n
1
1
1
1
0
0
ALG
accepts
0
0
0
0
0

Theorem

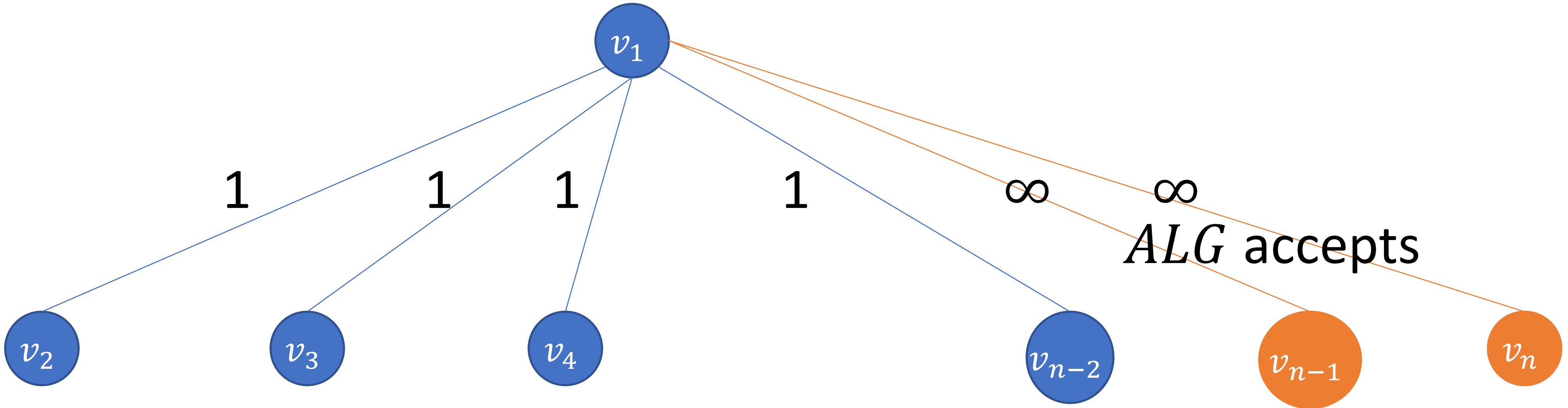
Let ALG be a deterministic algorithm for TSP in the EM model. Then we have

$$\rho(ALG) = \infty$$

Proof

Case ALG accepts $\{v_1, v_{n-1}\}$ and $\{v_1, v_n\}$

Adversary presents $(\{v_1, v_{n-1}\}, \infty), (\{v_1, v_n\}, \infty)$



Theorem

Let ALG be a deterministic algorithm for TSP in the EM model. Then we have

$$\rho(ALG) = \infty$$

Proof

Case ALG accepts $\{v_1, v_{n-1}\}$ and $\{v_1, v_n\}$

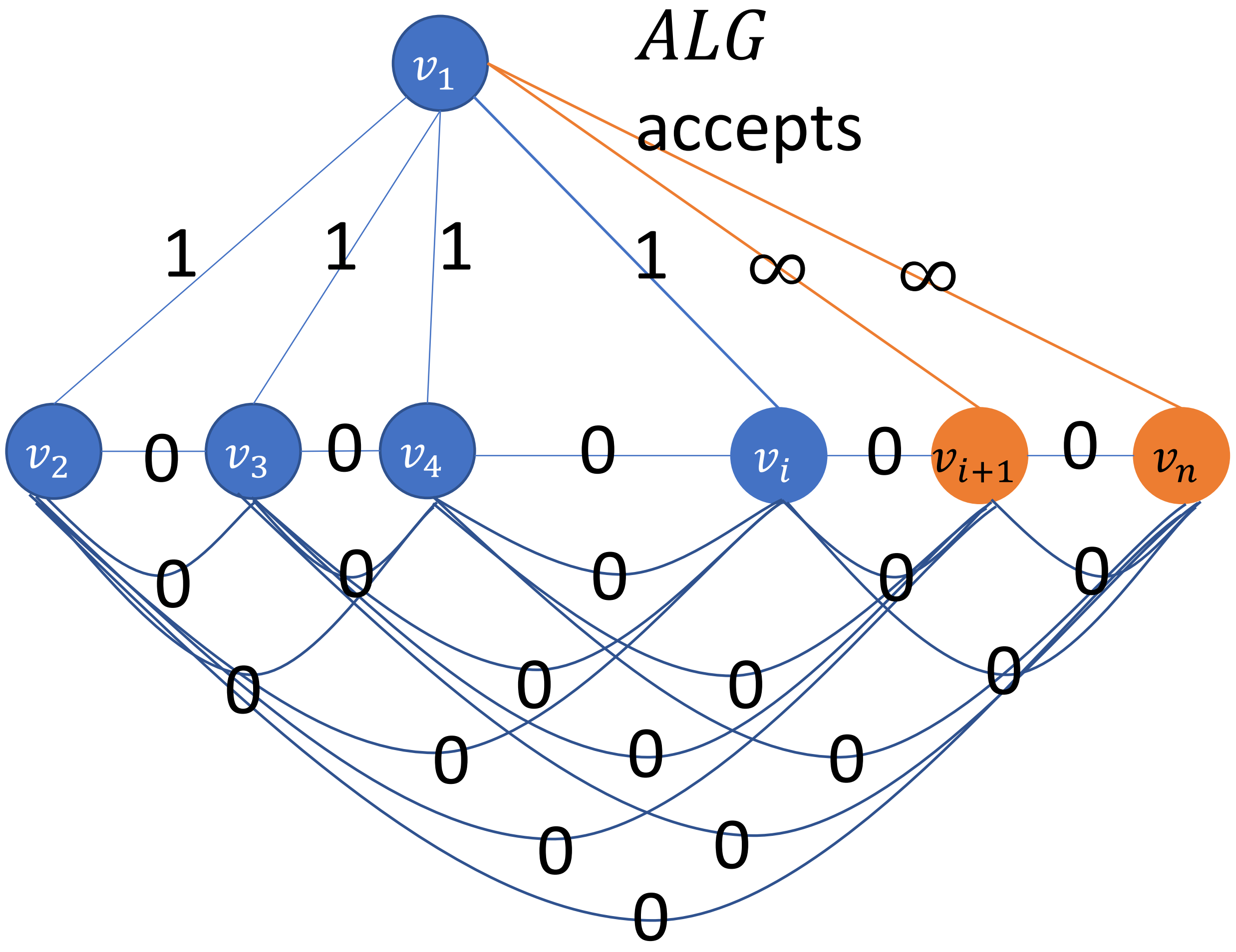
Adversary presents $(\{v_1, v_{n-1}\}, \infty), (\{v_1, v_n\}, \infty)$

The rest of the edges are presented in arbitrary order with weight 0

$$ALG = \infty$$
$$OPT = 2$$

Therefore competitive ratio is ∞

QED



ALG
accepts
v_1
v_2
v_3
v_4
v_i
v_{i+1}
v_n
1
1
1
1
∞
∞
0

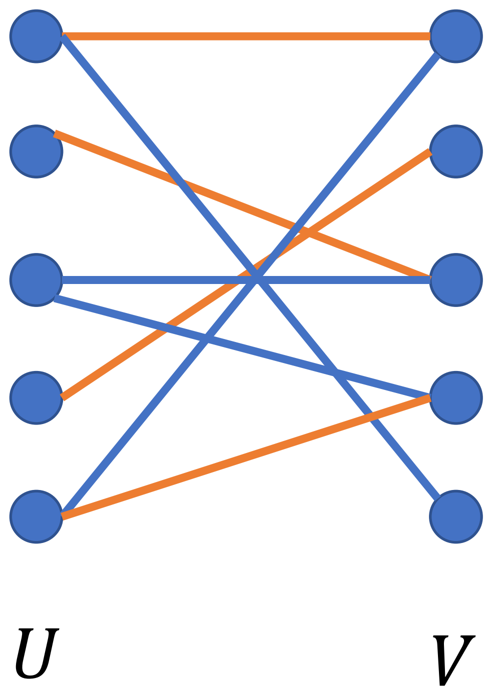
Maximum Bipartite Matching

Maximum Bipartite Matching

Bipartite graph $G = (U, V, E)$

Matching $M \subseteq E$ is a subset of vertex-disjoint edges

$\forall e_1, e_2 \in E$ we have $e_1 \cap e_2 = \emptyset$

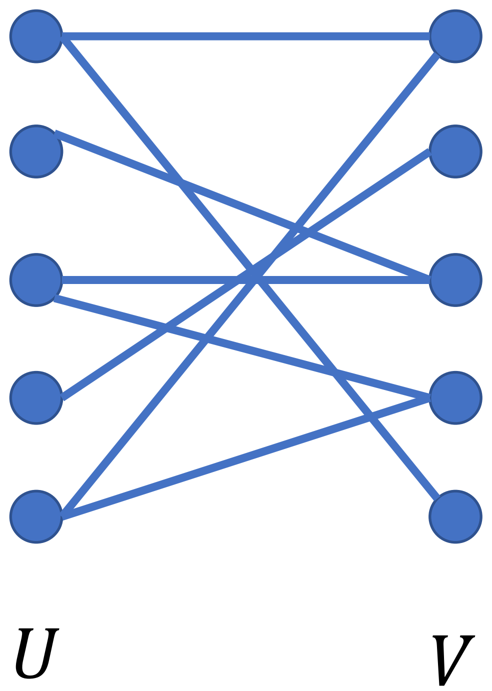


Vertex Arrival Model for bipartite graphs

One side V known in advance, called the offline side

Another side U arrives one node at a time, called the online side

When node $u \in U$ arrives also learn $N(u) \subseteq V$



Maximum Bipartite Matching

Input: $G = (U, V, E, \prec)$ unweighted undirected bipartite graph

$\prec$ is a total order on U

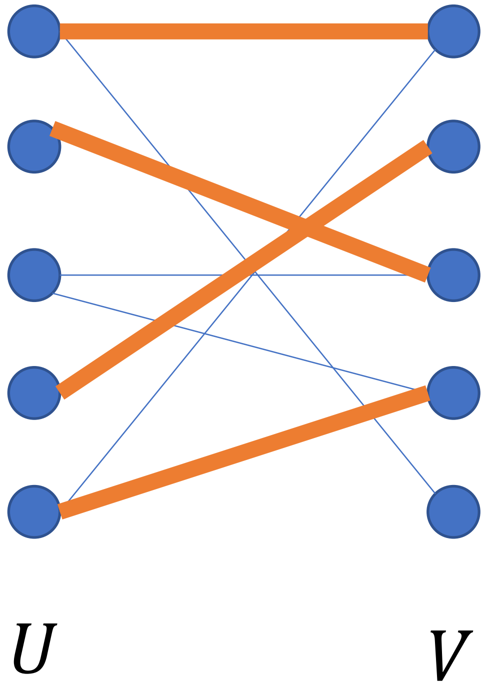
$(u_1, N(u_1)), (u_2, N(u_2)), \dots, (u_n, N(u_n))$ input sequence, where $u_1 \prec u_2 \prec \cdots \prec u_n$

Output: $d_1, \dots, d_n$ where $d_i \in N(u_i) \cup \{\perp\}$ indicates how to match u_i to its neighbor ($\perp$ indicates u_i remains unmatched)

Objective: maximize the size of the constructed matching

$M = \{\{u_i, d_i\} : d_i \neq \perp\}$ subject to M being a well-defined matching

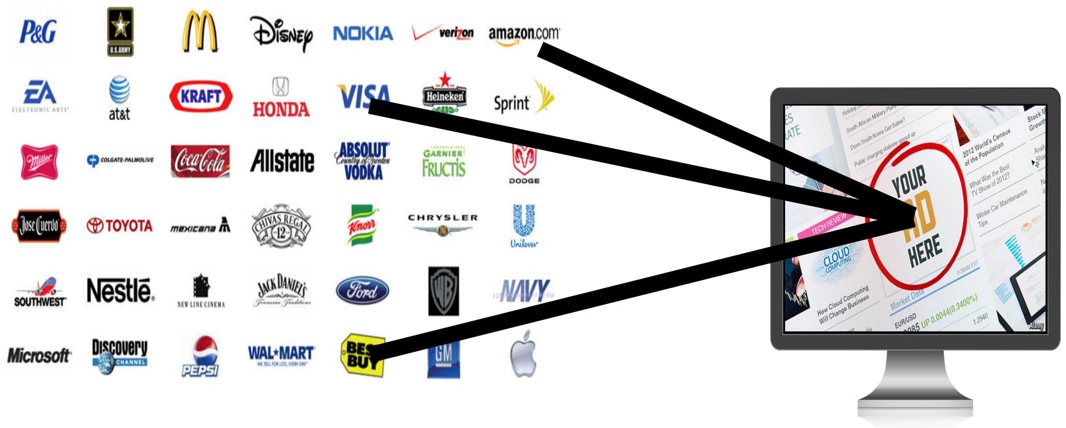
Maximum Bipartite Matching



Example Application: Online Advertising

Consider an online ad platform called G (for no particular reason)

- G has a database of advertisers
- Each advertiser has a limit on the number of times to be displayed
- User U clicks on a website
- G knows U better than U knows U
- G knows advertisers compatible with U



G wants to maximize the number of ads shown.

The underlying combinatorial problem: online bipartite matching.

Short History of Nearly Everything re Bipartite Matching

- Offline
  - rich history dating as far back as 1931 (König and Egerváry), maybe earlier
  - solvable optimally in polytime: Hopcrof-Karp [1974], matrix-mult. approach of Mucha-Sankowski [2004], electrical flow approach in sparse regime of Madry [2013], even more efficient approx. approaches.
- Online
  - introduced by Karp, Vazirani, Vazirani [1990]: adversarial input model
  - not solvable optimally: tight approx. ratio of 1/2 by det. algos, tight approx. ratio of 1 – 1/e = 0.632… by rand. algos
  - resurgence of interest in 2009 – Feldman et al. beat 1-1/e in known i.i.d., applications to online advertising
  - Bahmani, Kapralov [2010], Manshadi et al. [2011], Jaillet, Lu [2014], …

Simple Greedy Algorithm

Pick an ordering σ of offline vertices V

When a new item $(u, N(u))$ arrives

if there are no unmatched neighbors then u is unmatched

otherwise, pick the first neighbor v according to σ, match u to v

Some notation: $\sigma : V \to [n]$ – ordering or ranking

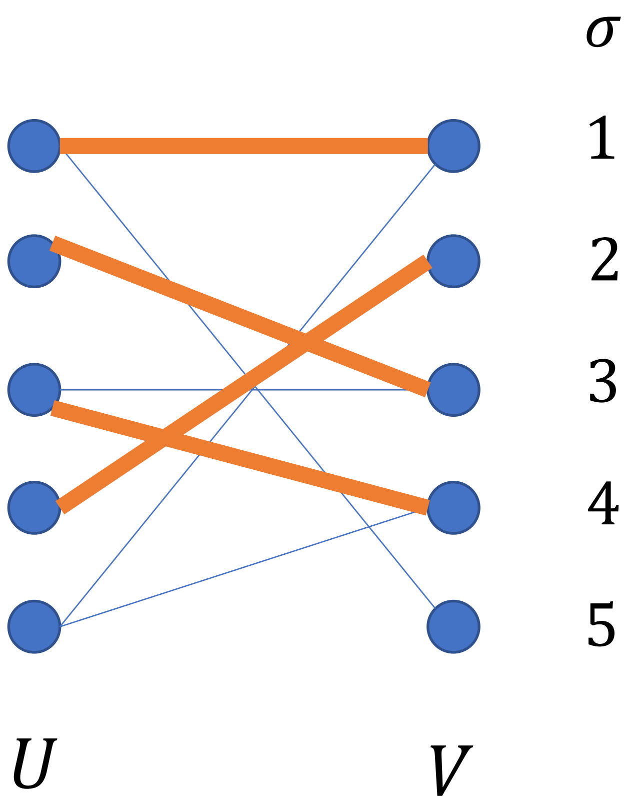
$\sigma(v)$ – rank of vertex v

$\sigma^{-1}(t)$ – vertex of rank t

v_1 has better rank than v_2 if $\sigma(v_1) < \sigma(v_2)$

$N_c(v)$ – currently available (unmatched) neighbors

Greedy example



Algorithm 13 Simple greedy algorithm for BMM.

procedure SIMPLEGREEDY

  V – set of offline vertices

  Fix a ranking σ on vertices V

  $M \leftarrow \emptyset$

  $i \leftarrow 1$

  while $i \leq n$ **do**

    New online vertex u_i arrives according to $\prec$ together with $N(u_i)$

    if $N_c(u) \neq \emptyset$ **then** ▷ if there is an unmatched vertex in $N(u_i)$

      ▷ select the vertex of best rank in $N_c(u_i)$

      $v \leftarrow \arg\min\{\sigma(v) : v \in N_c(u)\}$

      $M \leftarrow M \cup \{(u_i, v)\}$ ▷ match u_i with v

    $i \leftarrow i + 1$

Theorem	$\rho(SimpleGreedy) = 2$

Proof:

Part 1: $\rho(SimpleGreedy) \geq 2$

We present a **gadget-based** proof

We use the following gadget:

Adversary declares two offline vertices v_1, v_2

Without loss of generality assume that $\sigma(v_1) < \sigma(v_2)$

Adversary presents $(u_1, \{v_1, v_2\}), (u_2, \{v_1\})$

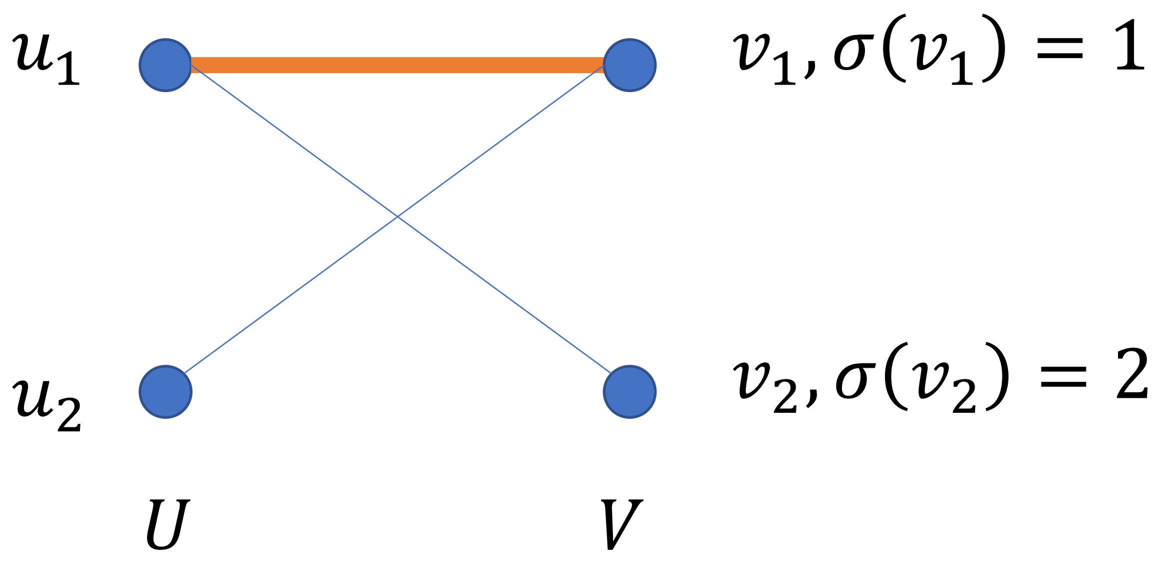
Theorem	$\rho(SimpleGreedy) = 2$

Proof:

Part 1: $\rho(SimpleGreedy) \geq 2$

$ALG = 1$

$OPT = 2$



Competitive ratio = 2

Theorem	$\rho(SimpleGreedy) = 2$

Proof:

Part 1: $\rho(SimpleGreedy) \geq 2$

This proves that strict competitive ratio is ≥ 2

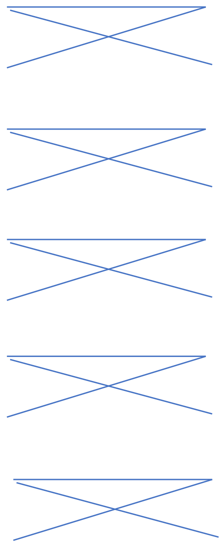
What about asymptotic?

Repeat the gadget!

Gives asymptotic competitive ratio

Can be used to show $\rho(ALG) \geq 2$ for any

deterministic algorithm ALG



Theorem	$\rho(SimpleGreedy) = 2$

Proof:

Part 2: $\rho(SimpleGreedy) \leq 2$

Let M be the matching constructed by $SimpleGreedy$

Claim: M is maximal, i.e., there is no edge $e = \{u, v\}$ that can be added to M

Pf by contradiction: suppose that $e = \{u, v\}$ is such that $M \cup \{e\}$ is also a valid matching

Therefore, u and v both were not matched by $SimpleGreedy$

Why wasn't u matched? It had at least one available neighbor v!

Contradiction! QED.

Theorem $\rho(SimpleGreedy) = 2$

Proof:

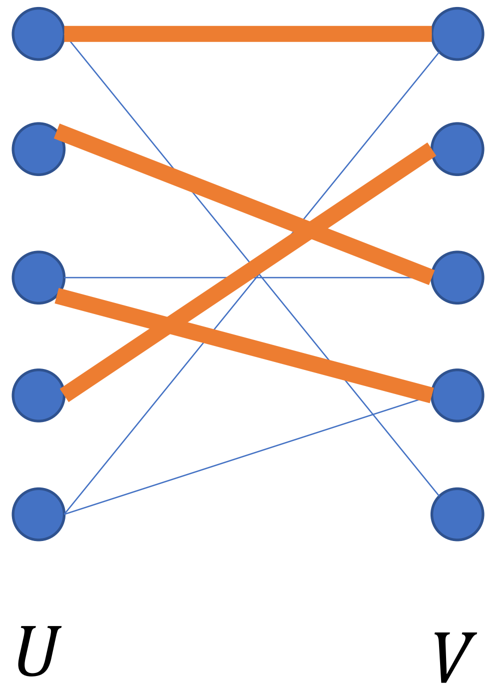
Part 2: $\rho(SimpleGreedy) \leq 2$

Let M be the matching constructed by $SimpleGreedy$

Claim: M is maximal

In other words, every edge is incident on some vertex in M

Therefore, if we take all vertices in M, we get a vertex cover!



Theorem $\rho(SimpleGreedy) = 2$

Proof:

Part 2: $\rho(SimpleGreedy) \leq 2$

Let M be the matching constructed by $SimpleGreedy$

Let OPT denote the size of maximum matching

Let $VC(G)$ denote the size of minimum vertex cover

Since M is maximal, we get $VC(G) \leq 2|M| = 2ALG$

Trivially, we have $VC(G) \geq OPT$ (why?)

Therefore $OPT \leq 2\,ALG$

QED

Deterministic Bipartite Matching

To sum up:

deterministic algorithms have $\rho(ALG) \geq 2$

$SimpleGreedy$ achieves this bound

What about randomized algorithms?

Simple Randomized Algorithm

Attempt 1: $SimpleRandom$

When an item $\left(u, N(u)\right)$ arrives

form $N_c(u)$ – unmatched neighbors of u

match u to a uniformly random element of $N_c(u)$

Exercise:

$$\rho_{OBL}(SimpleRandom) = 2$$

Thus, $SimpleRandom$ does not improve upon $SimpleGreedy$

Ranking Algorithm

Attempt 2: $Ranking$ algorithm, or KVV (Karp, Vazirani, Vazirani)

Pick an ordering/ranking σ on V uniformly at random

Run $SimpleGreedy$ with respect to σ

We will prove

Theorem $\rho_{OBL}(Ranking) = \dfrac{e}{e-1} \approx 1.582$

Algorithm 14 The Ranking algorithm for BMM.

procedure RANKING

V – set of offline vertices

Pick a ranking σ on vertices V *uniformly at random*

$M \leftarrow \emptyset$

$i \leftarrow 1$

while $i \leq n$ **do**

New online vertex u_i arrives according to $\prec$ together with $N(u_i)$

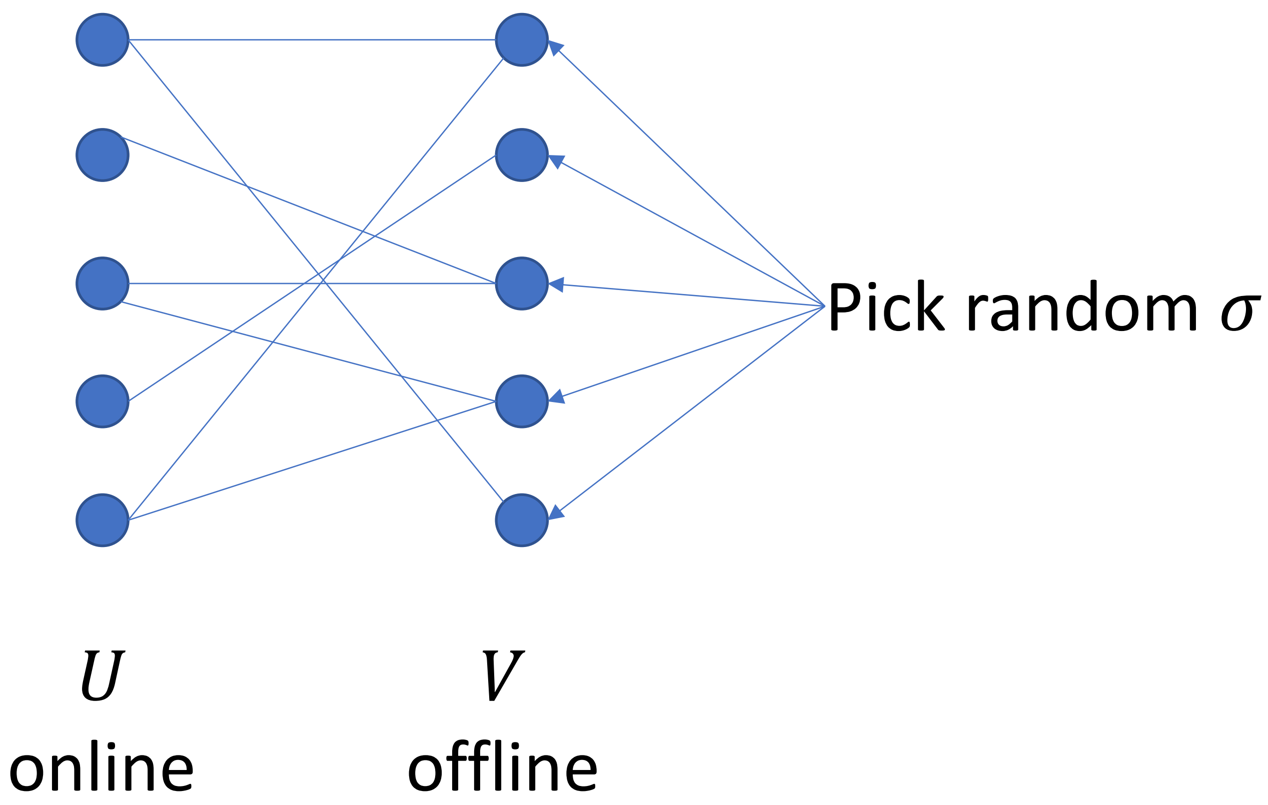
if $N_c(u) \neq \emptyset$ **then** ▷ if there is an unmatched vertex in $N(u_i)$

▷ select the vertex of best rank in $N_c(u_i)$

$v \leftarrow \arg\min\{\sigma(v) : v \in N_c(u)\}$

$M \leftarrow M \cup \{(u_i, v)\}$ ▷ match u_i with v

$i \leftarrow i + 1$

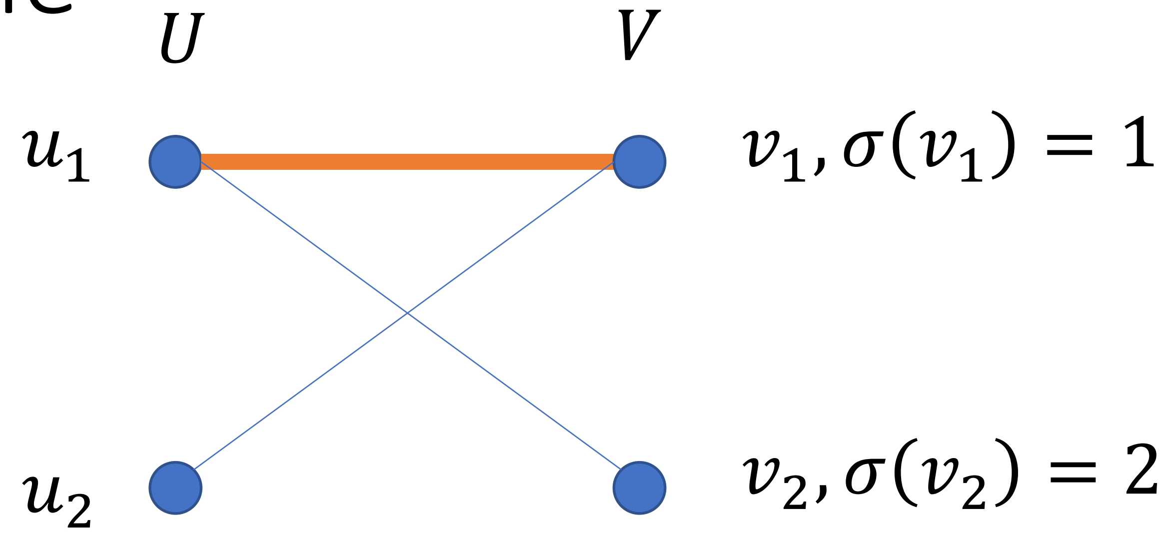


Pick random σ
U
online
V
offline

Ranking example

$ALG = 1$

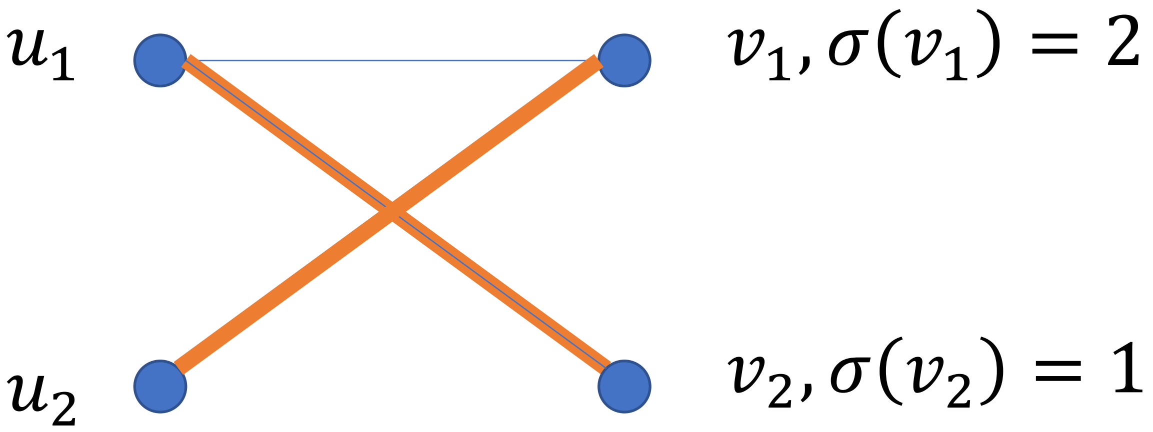
$OPT = 2$



Probability 1/2

$ALG = 2$

$OPT = 2$



Probability 1/2

Ranking example

$ALG = 1$ $OPT = 2$ Probability 1/2

$ALG = 2$ $OPT = 2$ Probability 1/2

$$\mathbb{E}(ALG) = \frac{3}{2} \qquad \frac{OPT}{\mathbb{E}(ALG)} = \frac{4}{3} \approx 1.333$$

Theorem $\rho_{OBL}(Ranking) \leq \frac{e}{e-1} \approx 1.582$

$G = (U, V, E)$ – input graph

Simplifying assumptions:

$|U| = |V| = n$ online and offline sides are of same size

let M^* denote some maximum matching, i.e., $|M^*| = OPT$

$|M^*| = |U|$, i.e., M^* is a perfect matching – it matches all nodes

Exercise: we can assume the above without loss of generality

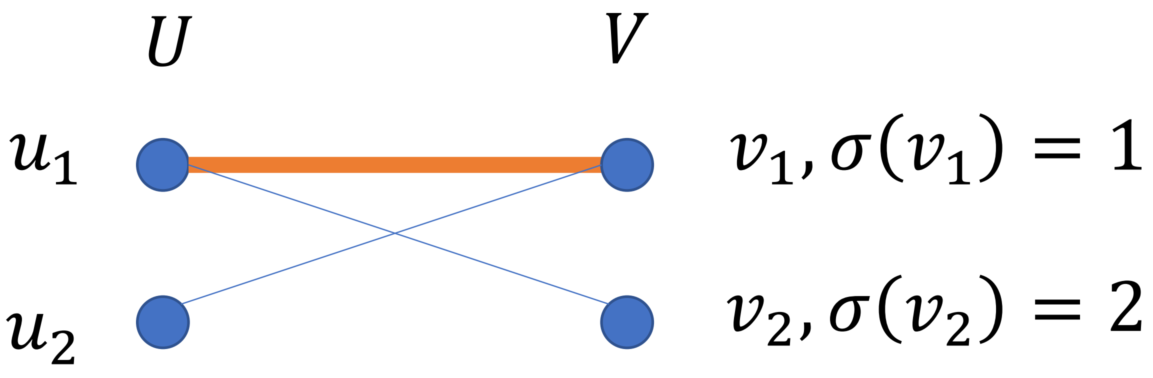
We are interested in how close $|M|$ is to n

Theorem $\rho_{OBL}(Ranking) \leq \dfrac{e}{e-1} \approx 1.582$

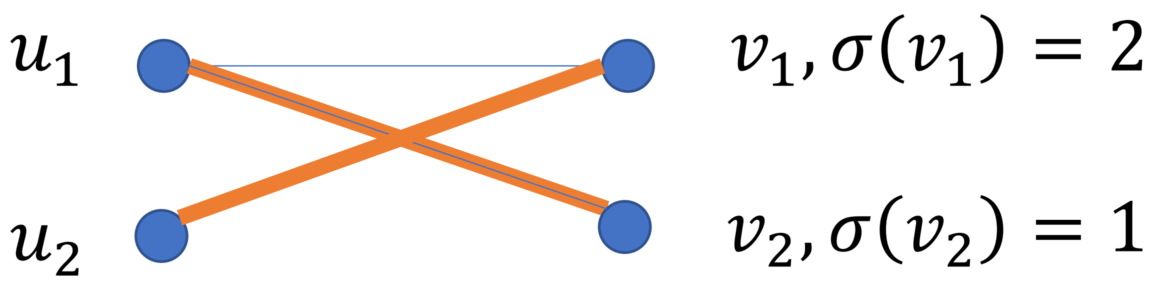
p_t - probability that vertex of rank t is matched by $Ranking$

Example:

Probability 1/2

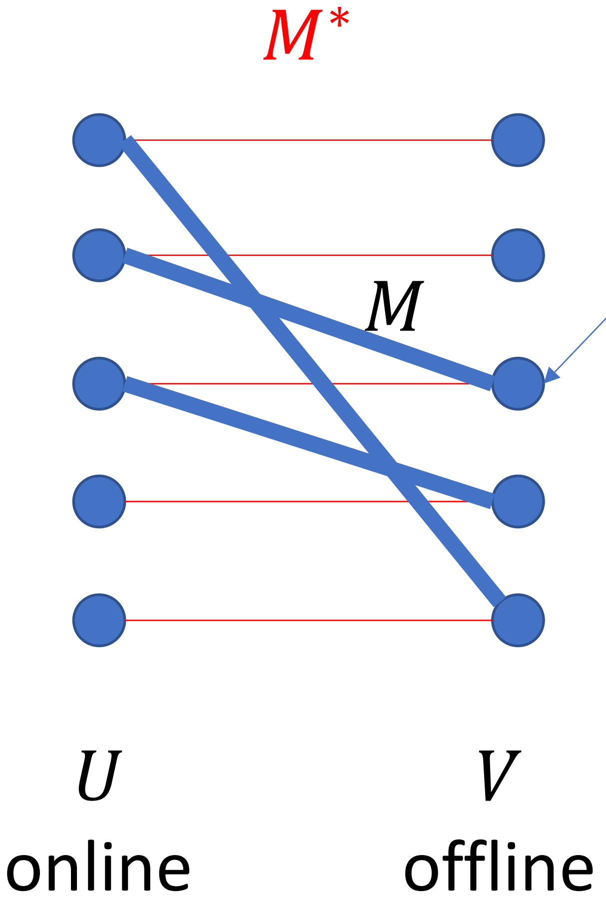


Probability 1/2



$p_1 = ?$
$p_2 = ?$

random σ



Vertex of rank t
p_t - probability it is matched

J_t - indicator that vertex of rank t matched by $Ranking$

$J_t = 1$ if vertex of rank t is matched
$J_t = 0$ otherwise

$$|M| = \sum_t J_t$$

$$\mathbb{E}(|M|) = \mathbb{E}\left(\sum_t J_t\right)$$

$$= \sum_t \mathbb{E}(J_t) = \sum_t 1 \cdot p_t + 0 \cdot (1 - p_t) = \sum_t p_t$$

Theorem	$\rho_{OBL}(Ranking) \leq \frac{e}{e-1} \approx 1.582$

We want to show that $\mathbb{E}(|M|)$ is large, i.e.,

$$\mathbb{E}(|M|) \geq \frac{e-1}{e} n = \left(1 - \frac{1}{e}\right) n$$

Alternatively

$$\sum_t p_t \geq \left(1 - \frac{1}{e}\right) n$$

KEY LEMMA:

For all $t \in [n]$ we have $1 - p_t \leq \frac{1}{n} \sum_{1 \leq s \leq t} p_s$

Theorem $\rho_{OBL}(Ranking) \le \frac{e}{e-1} \approx 1.582$

KEY LEMMA:

For all $t \in [n]$ we have $1 - p_t \le \frac{1}{n} \sum_{1 \le s \le t} p_s$

Assume KEY LEMMA and see how it implies the theorem

$p_1 = 1$ since there exists a perfect matching

$p_t \ge \left(1 - \frac{1}{n}\right)\left(\frac{n}{n+1}\right)^{t-1}$ follows by KEY LEMMA and simple induction

Theorem $\rho_{OBL}(Ranking) \leq \frac{e}{e-1} \approx 1.582$

$p_1 = 1$

$p_t \geq \left(1 - \frac{1}{n}\right)\left(\frac{n}{n+1}\right)^{t-1}$

$$\sum_t p_t = \frac{1}{n} + \left(1 - \frac{1}{n}\right) \sum_{1 \leq t \leq n} \left(\frac{n}{n+1}\right)^{t-1} \geq \frac{1}{n} + \left(1 - \frac{1}{n}\right) \frac{1 - \left(\frac{n}{n+1}\right)^n}{1 - \frac{n}{n+1}}$$

$$= \frac{1}{n} + \frac{n^2 - 1}{n} \left(1 - \left(\frac{n}{n+1}\right)^n\right) \geq n\left(1 - \left(1 - \frac{1}{n+1}\right)^n\right) \to n\left(1 - \frac{1}{e}\right)$$

Theorem $\rho_{OBL}(Ranking) \leq \frac{e}{e-1} \approx 1.582$

To finish the proof we need to prove

KEY LEMMA:

For all $t \in [n]$ we have $1 - p_t \leq \frac{1}{n} \sum_{1 \leq s \leq t} p_s$

A_t - the set of permutations σ such that $Ranking$ **matches** a vertex of rank t

B_t - the set of permutation σ such that $Ranking$ **doesn't match** a vertex of rank t

Theorem $\rho_{OBL}(Ranking) \leq \frac{e}{e-1} \approx 1.582$

KEY LEMMA:

For all $t \in [n]$ we have $1 - p_t \leq \frac{1}{n} \sum_{1 \leq s \leq t} p_s$

We construct an injection

$$[n] \times B_t \to \bigcup_{1 \leq s \leq t} A_s$$

Then $n|B_t| \leq \sum_{1 \leq s \leq t} |A_s|$

So $\frac{|B_t|}{n!} \leq \frac{1}{n} \sum_{1 \leq s \leq t} \frac{|A_s|}{n!}$

Which is equivalent to $1 - p_t \leq \frac{1}{n} \sum_{1 \leq s \leq t} p_s$

Theorem	$\rho_{OBL}(Ranking) \leq \frac{e}{e-1} \approx 1.582$

Injection

$$[n] \times B_t \to \bigcup_{1 \leq s \leq t} A_s$$

Means that we map an index $i \in [n]$ and a permutation where a vertex of rank t is not matched to some permutation where a vertex of rank $\leq t$ is matched.

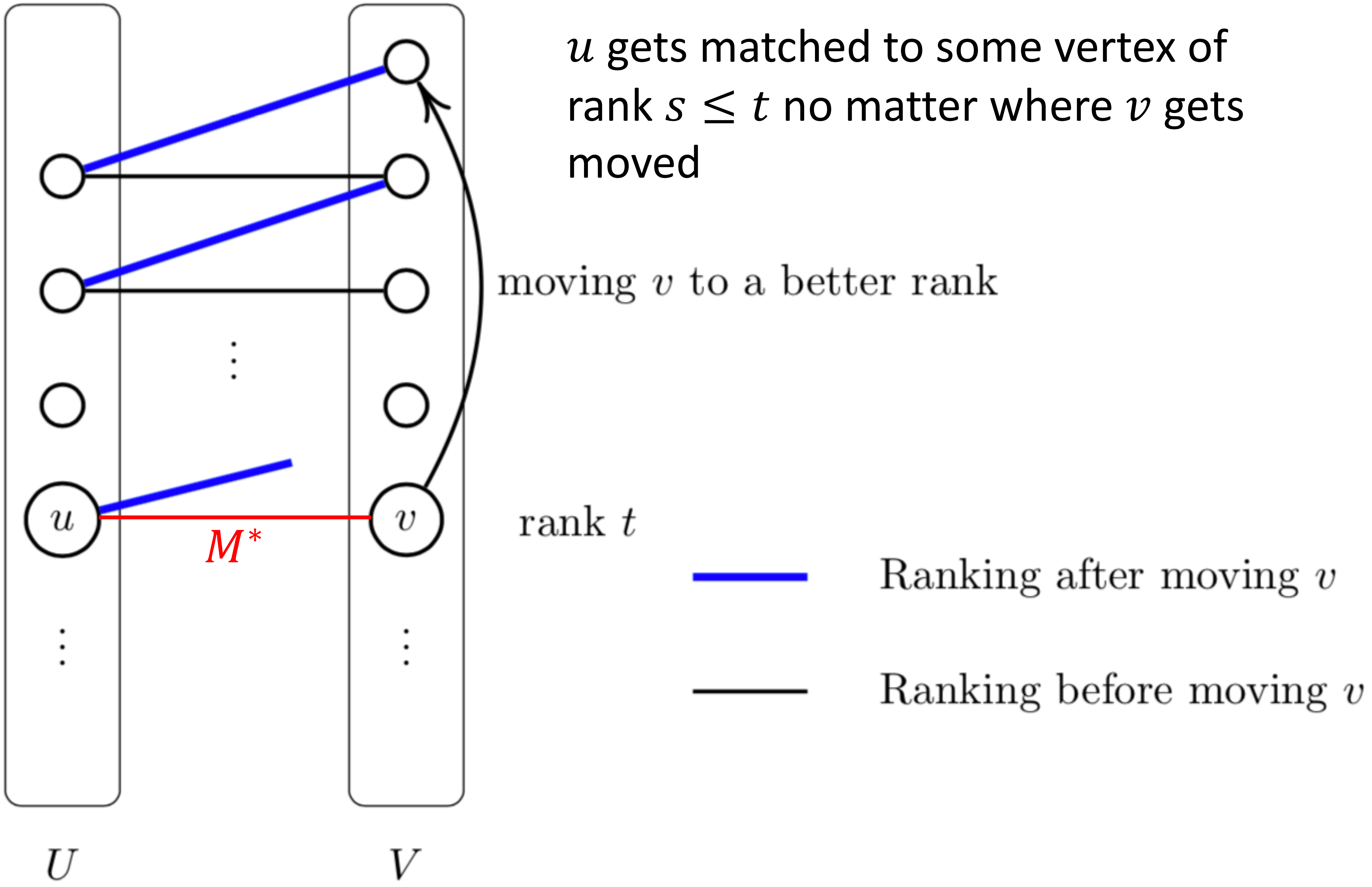
$\sigma \in B_t$ and $i \in [n]$

$v = \sigma^{-1}(t)$ – vertex of rank t

Obtain σ_i from σ by moving v into position i and shifting other elements accordingly

It is clearly injective, but is it well defined?

u gets matched to some vertex of rank $s \leq t$ no matter where v gets moved



This finishes the proof of the upper bound.

We also have a matching lower bound that holds for all randomized algorithms

Theorem

Let ALG be a randomized algorithm for Maximum Bipartite Matching. Then

$$\rho_{OBL}(ALG) \geq \frac{e}{e-1} \approx 1.582$$

Theorem

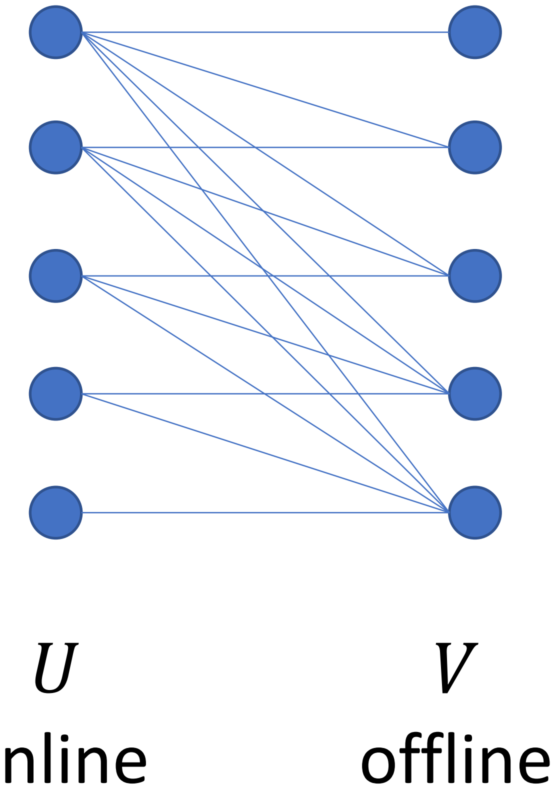
Let ALG be a randomized algorithm for Maximum Bipartite Matching. Then

$$\rho_{OBL}(ALG) \geq \frac{e}{e-1} \approx 1.582$$

Proof:

Use Yao's minimax argument

Consider a triangle graph



Theorem

Let ALG be a randomized algorithm for Maximum Bipartite Matching. Then

$$\rho_{OBL}(ALG) \geq \frac{e}{e-1} \approx 1.582$$

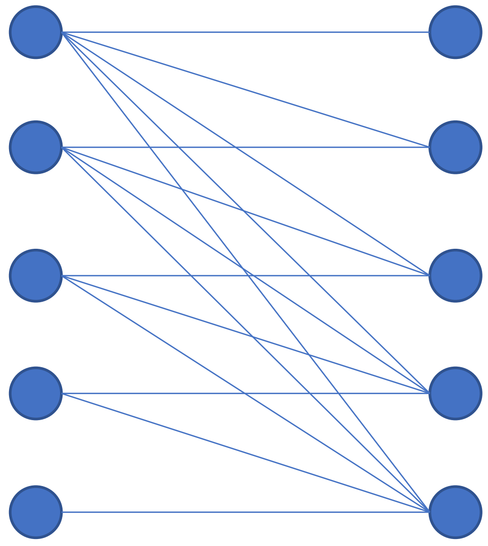
Proof:

Use Yao's minimax argument

Consider a triangle graph

Generate distribution on inputs by randomly reordering V

Can show that ALG will miss $\frac{1}{e}$ fraction on average. See text for details.

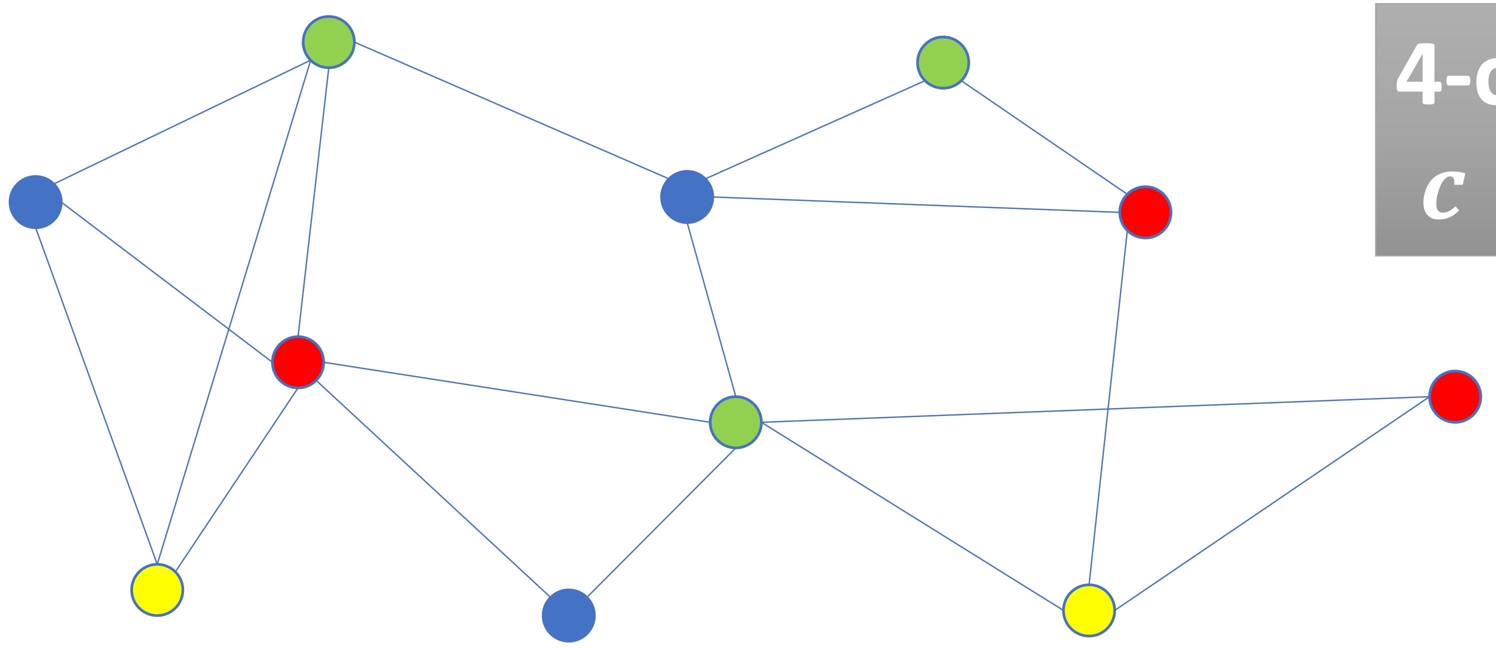


Graph Coloring

Graph Coloring

Graph k-coloring is a function $c : V \to \{1, 2, \ldots, k\}$ such that

for all $\{u, v\} \in E$ we have $c(u) \neq c(v)$



4-coloring

$c : V \to \{1, 2, 3, 4\}$

Graph Coloring

Input: $G = (V, E, \prec)$ unweighted undirected graph

$\prec$ is a total order on V

$(v_1, N_1), (v_2, N_2), \ldots, (v_n, N_n)$ input sequence, where

$v_1 \prec v_2 \prec \cdots \prec v_n$ and

$N_i = N(v_i) \cap \{v_j : j < i\}$

Output: $c : V \to [k]$ where $c(v_i)$ indicates the color assigned to vertex v_i

Objective: to find c so as to minimize k subject to c being a valid coloring, i.e., $\forall \{u, v\} \in E$ we have $c(u) \neq c(v)$

Graph Coloring

We are interested in graph coloring of bipartite graphs in VAM-PH

Thus, $OPT = 2$

Theorem

Let ALG be a deterministic online algorithm for Graph Coloring of bipartite graphs. Then

$$\rho(ALG) \geq \frac{\log n}{2}$$

Theorem

There is a deterministic online algorithm $CBIP$ for Graph Coloring of bipartite graphs such that

$$\rho(CBIP) \leq \log n$$

Theorem

Let ALG be a deterministic online algorithm for Graph Coloring of bipartite graphs. Then

$$\rho(ALG) \geq \frac{\log n}{2}$$

Proof:

Fix $k \in \mathbb{N}$. Define the following statement $S(k)$:

There is an adversarial strategy that presents a sequence of trees

$T_1, \ldots, T_k$ such that ALG uses k distinct colors

and the combined size of all trees is $\leq 2^k - 1$

We prove $S(k)$ by induction on k

$S(k)$: there is an adversarial strategy that presents a sequence of trees $T_1, \ldots, T_k$ such that ALG uses k distinct colors and the combined size of all trees is $\leq 2^k - 1$

Case $k = 1$:

T_1 - one tree

One color

Combined size = $1 = 2^1 - 1$

Case $k = 2$:

T_1 T_2

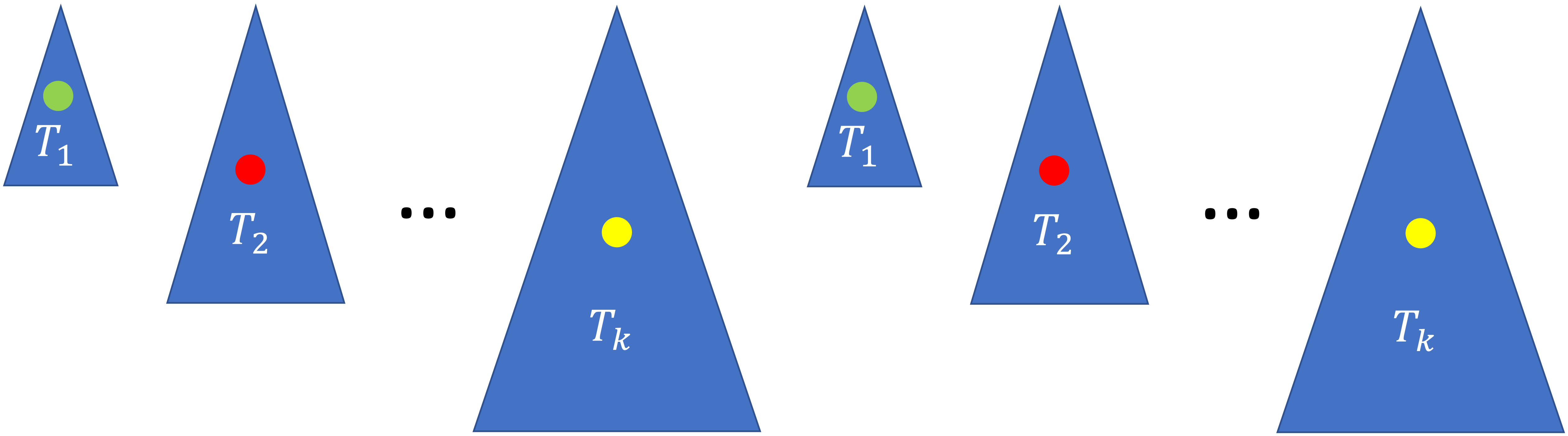
Two trees

Two colors

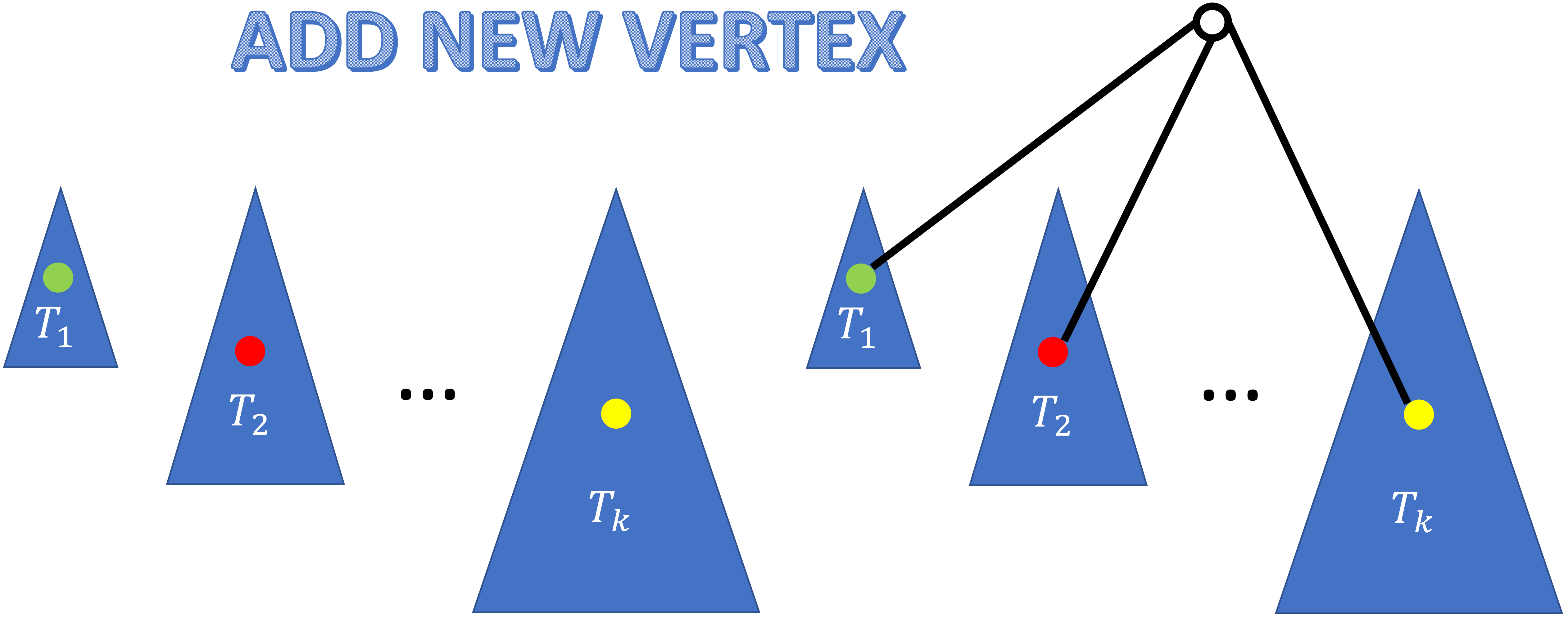
Combined size = $3 = 2^2 - 1$

By induction: $T_1, \ldots, T_k$ - k trees, k colors, combined size $\leq 2^k - 1$

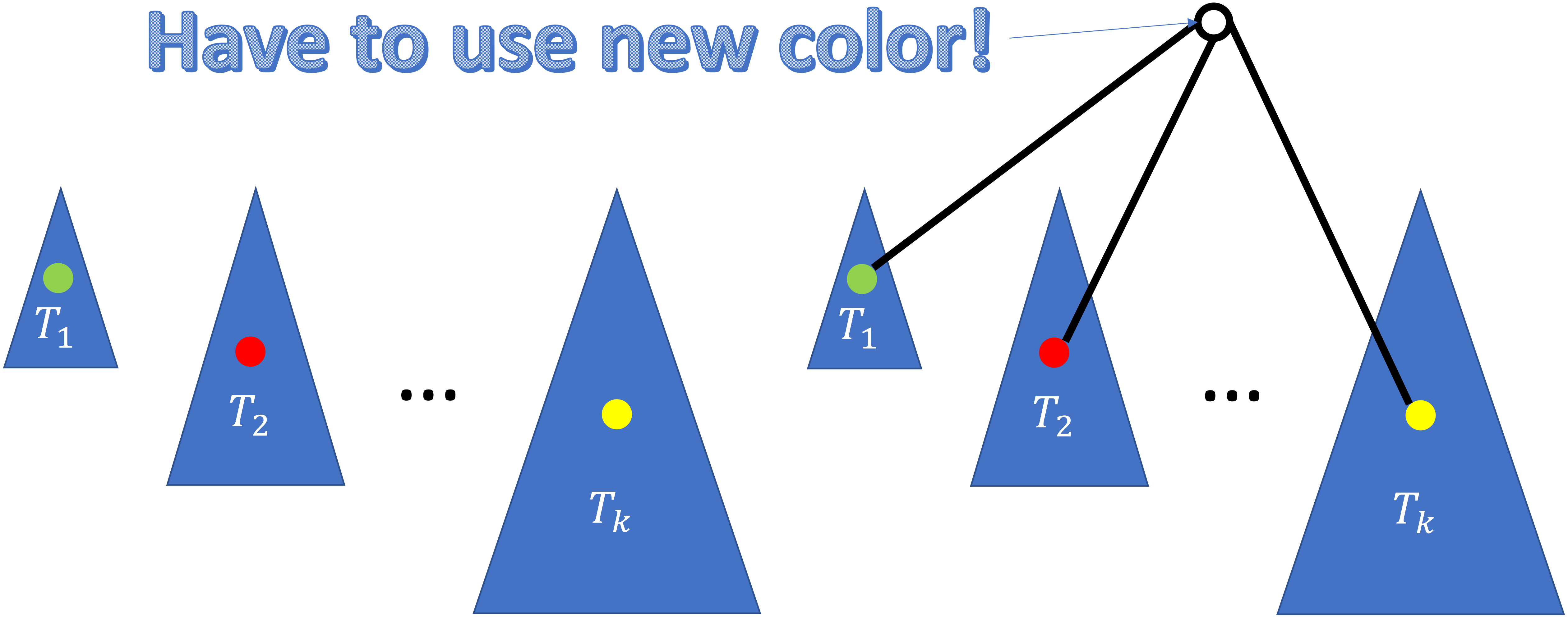
DUPLICATE!

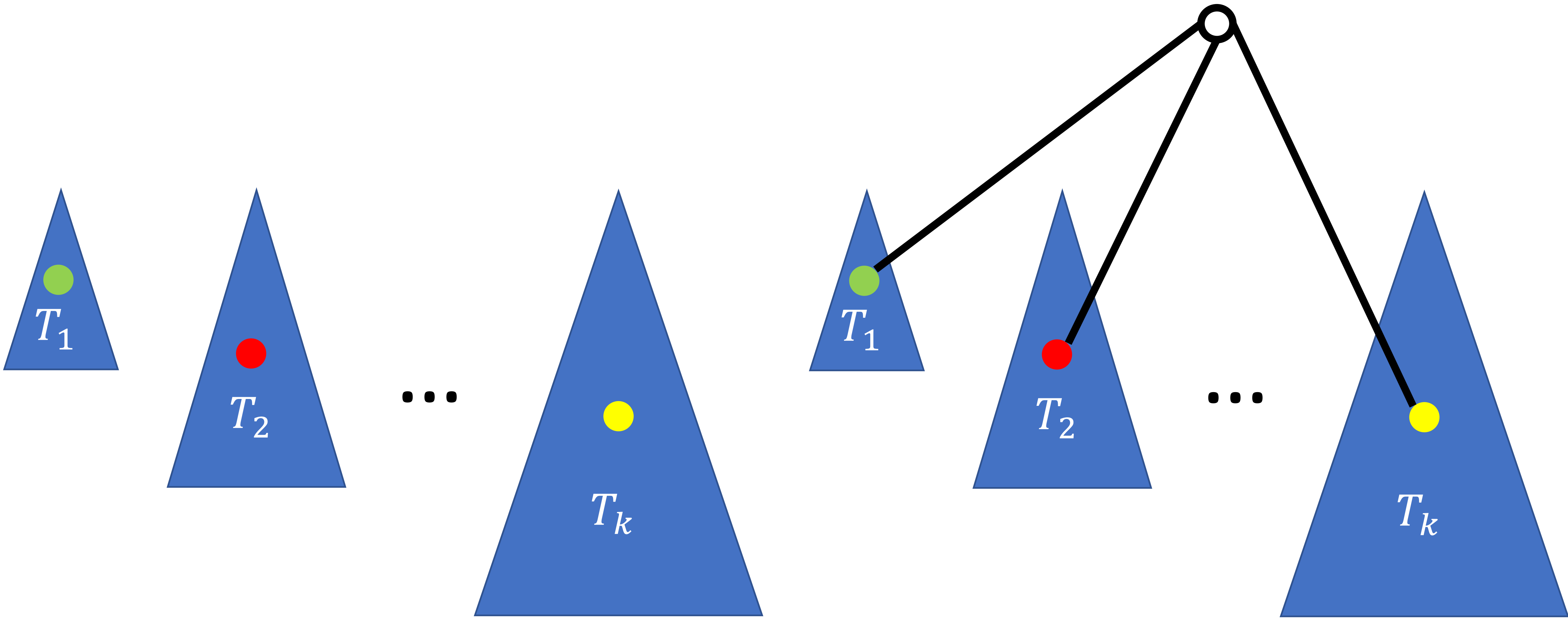


By induction: $T_1, \dots, T_k$ - k trees, k colors, combined size $\leq 2^k - 1$



By induction: $T_1, \ldots, T_k$ - k trees, k colors, combined size $\leq 2^k - 1$





Get $T_1, \ldots, T_k, T_{k+1}$ - $(k+1)$ trees, $k+1$ colors,

combined size $\leq (2^k - 1) + (2^k - 1) + 1 = 2^{k+1} - 1$

Theorem Let ALG be a deterministic online algorithm for Graph Coloring of bipartite graphs. Then

$$\rho(ALG) \geq \frac{\log n}{2}$$

Take $k = \log n$

Then adversary presents input $T_1, \ldots, T_k$ trees of combined size $2^k - 1 = n - 1$

Trees are bipartite so $OPT = 2$

ALG uses $\mathrm{k} = \log n$ colors

QED

$CBIP$ algorithm

When a vertex v arrives

compute C_v - connected component of v

compute bipartition of $C_v = S_v \cup \widetilde{S_v}$

$v \in S_v$ and $N(v) \subseteq \widetilde{S_v}$

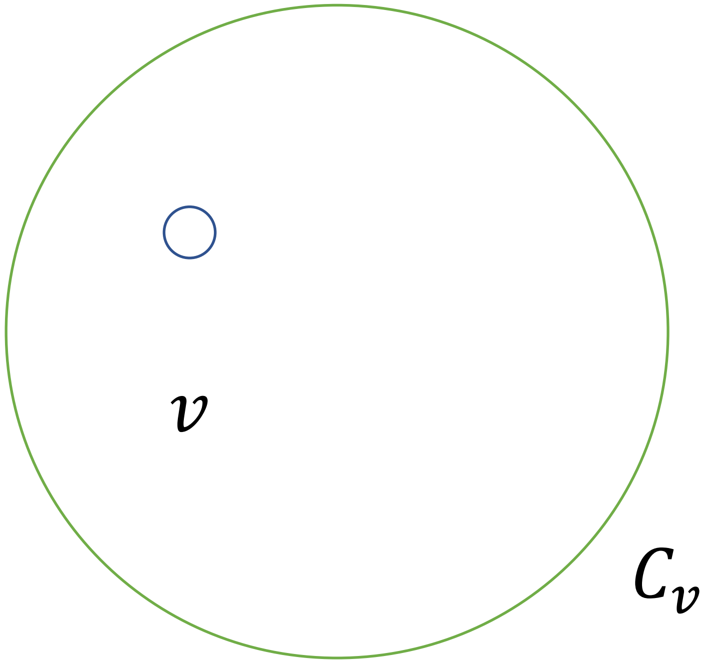
let i be the smallest color **not** in $\widetilde{S_v}$

color v with i

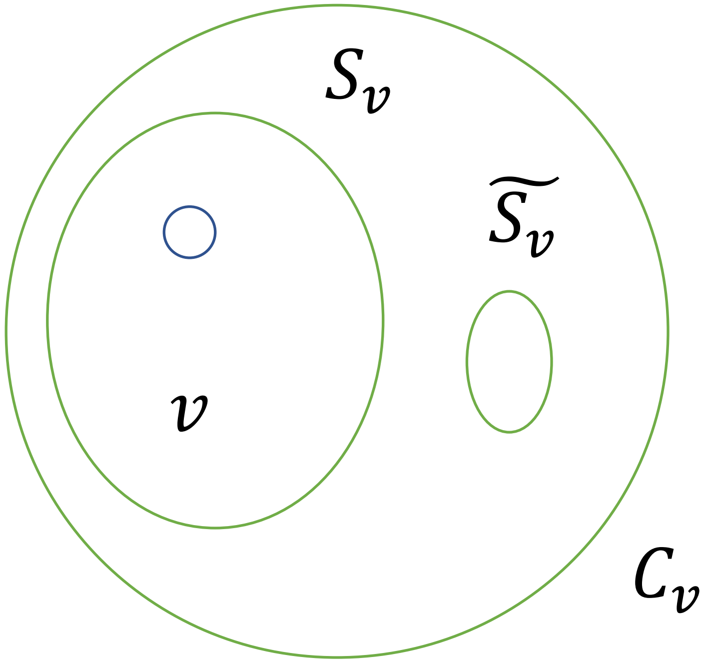
$CBIP$ example

v

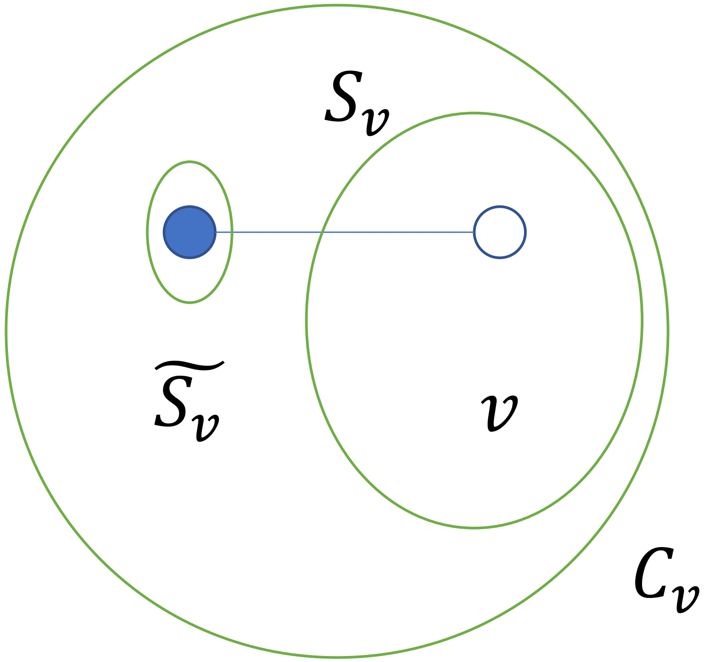
$CBIP$ example



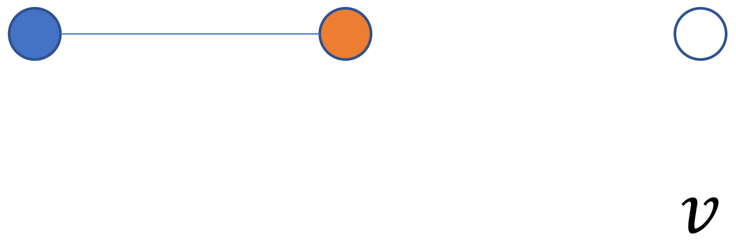
$CBIP$ example



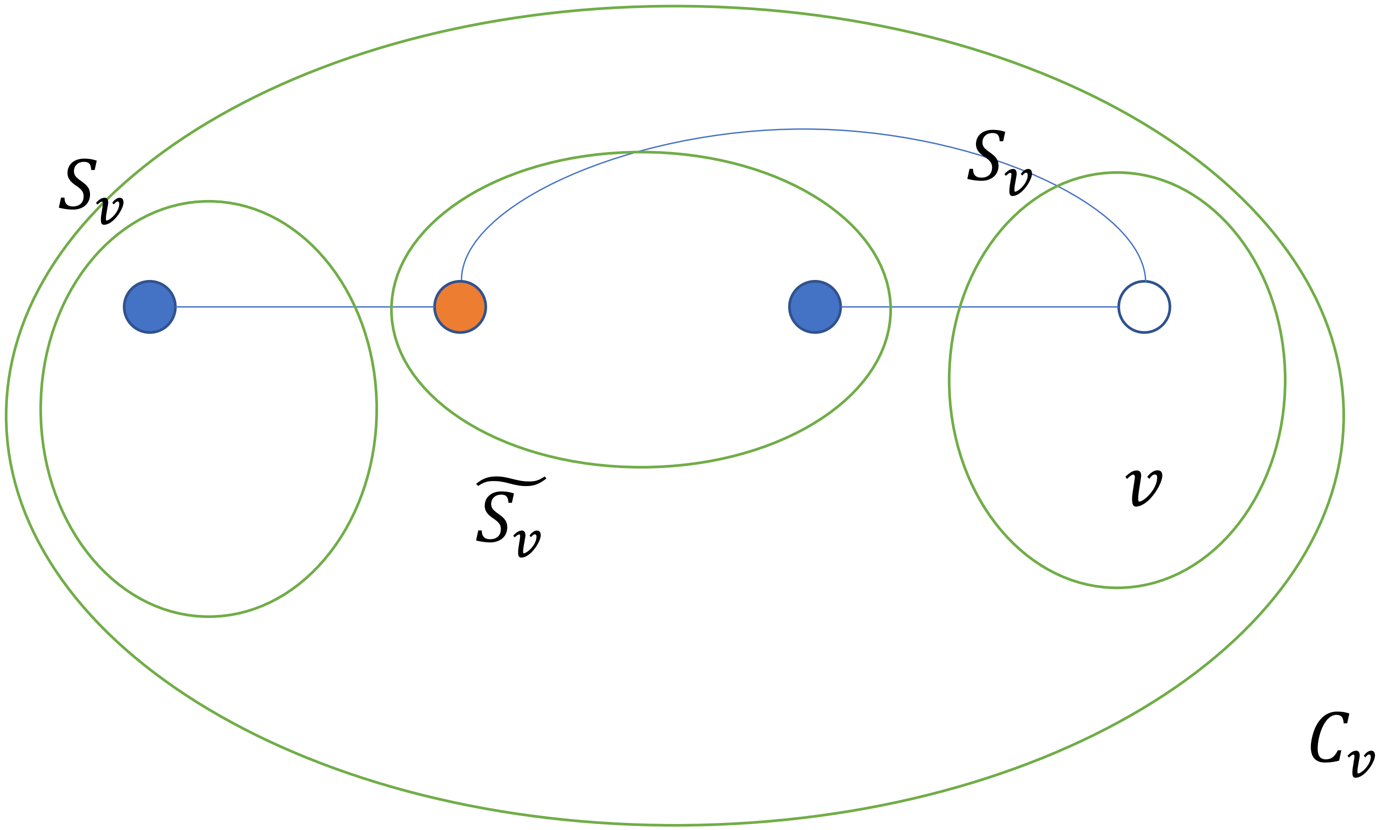
$CBIP$ example



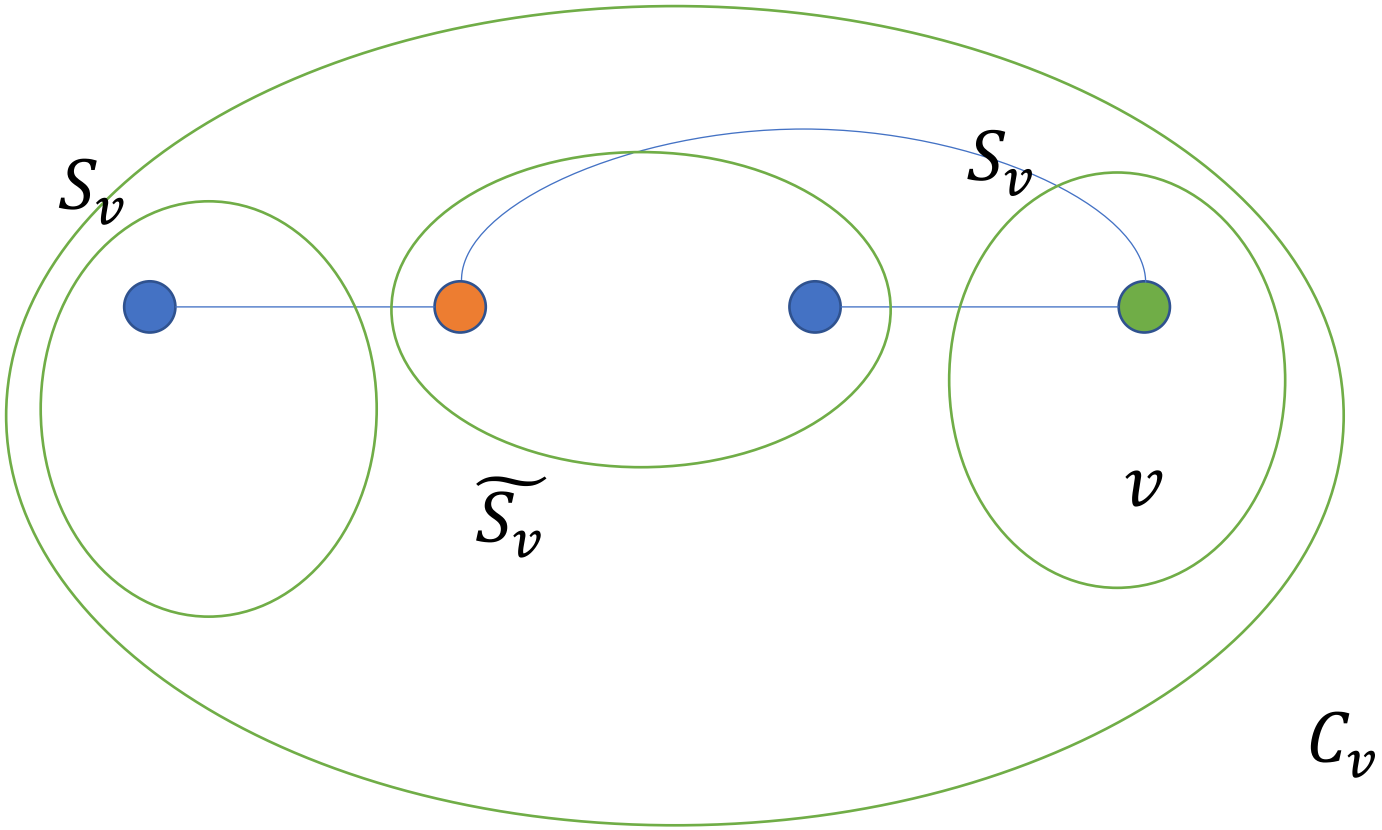
$CBIP$ example



$CBIP$ example



$CBIP$ example



Theorem $\rho(CBIP) \leq \log n$

Proof

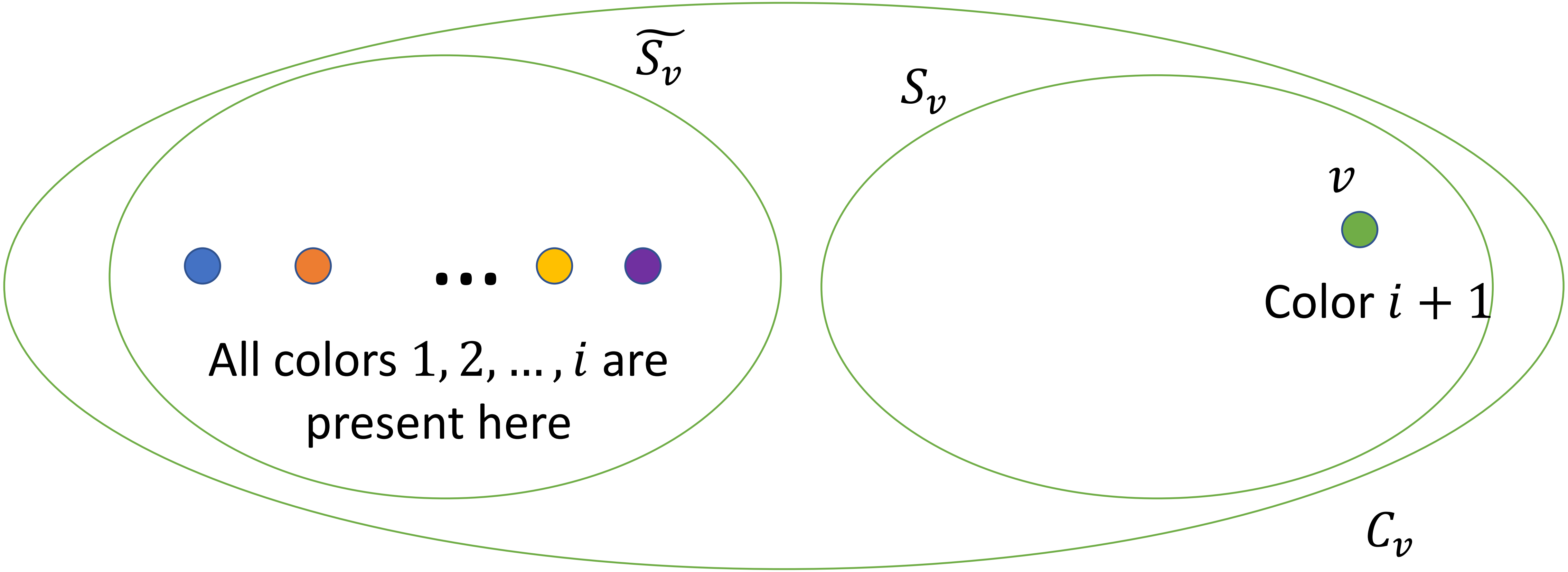
$n(i)$ – minimum number of nodes needed to force $CBIP$ use i distinct colors

We will prove $n(i) \geq \lceil 2^{i/2} \rceil$ by induction on i

Clearly $n(1) = 1$ and $n(2) = 2$

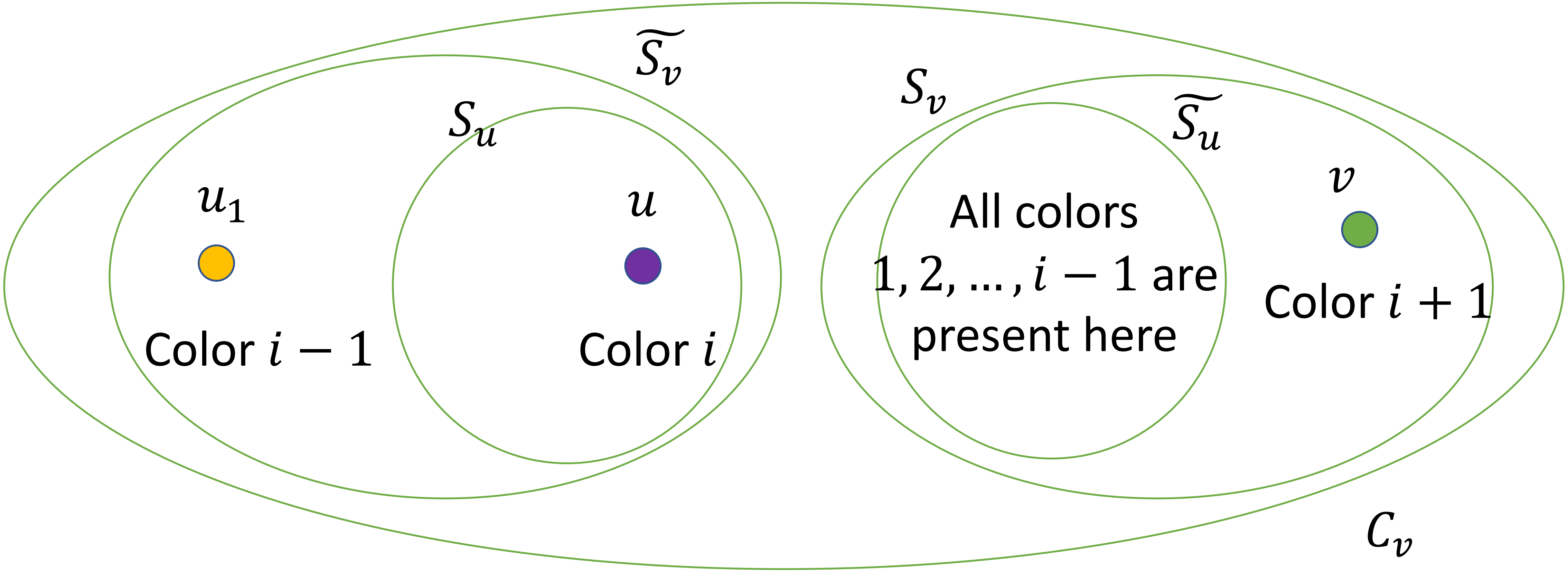
Theorem $\rho(CBIP) \leq \log n$

Inductive step: let v be the first vertex that is colored with color $i + 1$ by CBIP



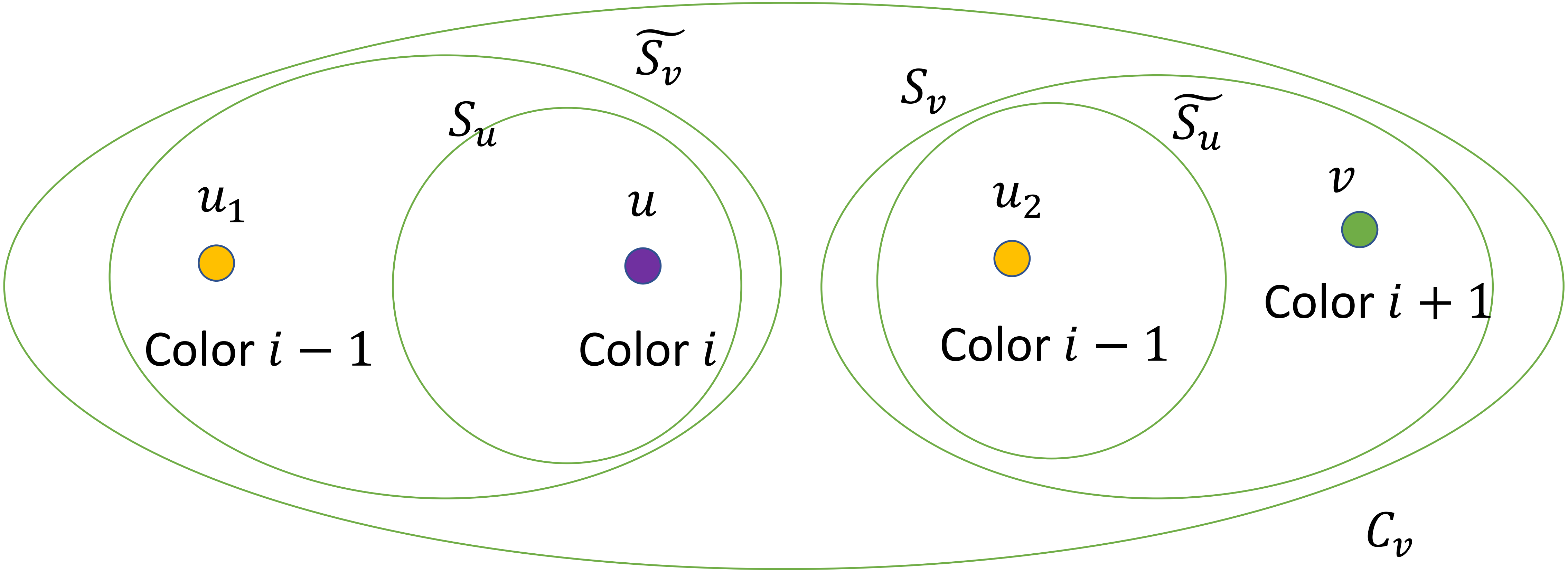
Theorem $\rho(CBIP) \leq \log n$

Inductive step: let v be the first vertex that is colored with color $i+1$ by CBIP



Theorem	$\rho(CBIP) \le \log n$

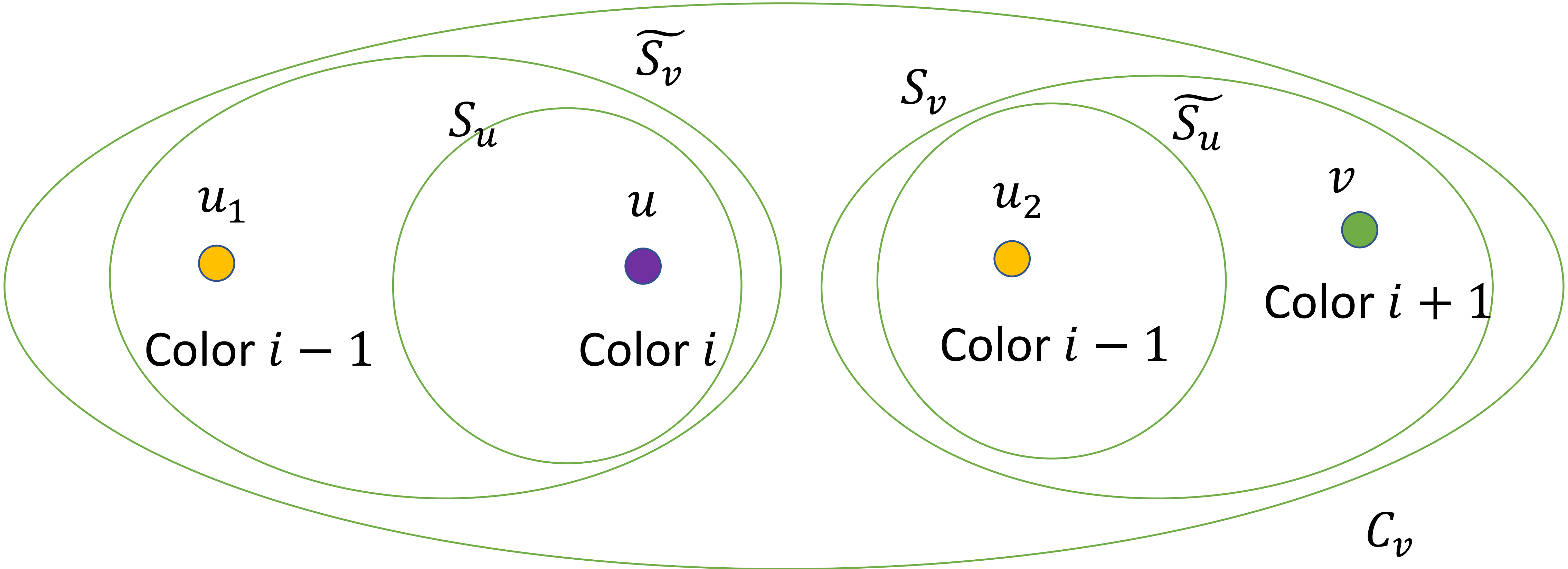
Inductive step: let v be the first vertex that is colored with color $i+1$ by CBIP



Theorem $\rho(CBIP) \le \log n$

Found two vertices u_1 and u_2 colored with $i-1$

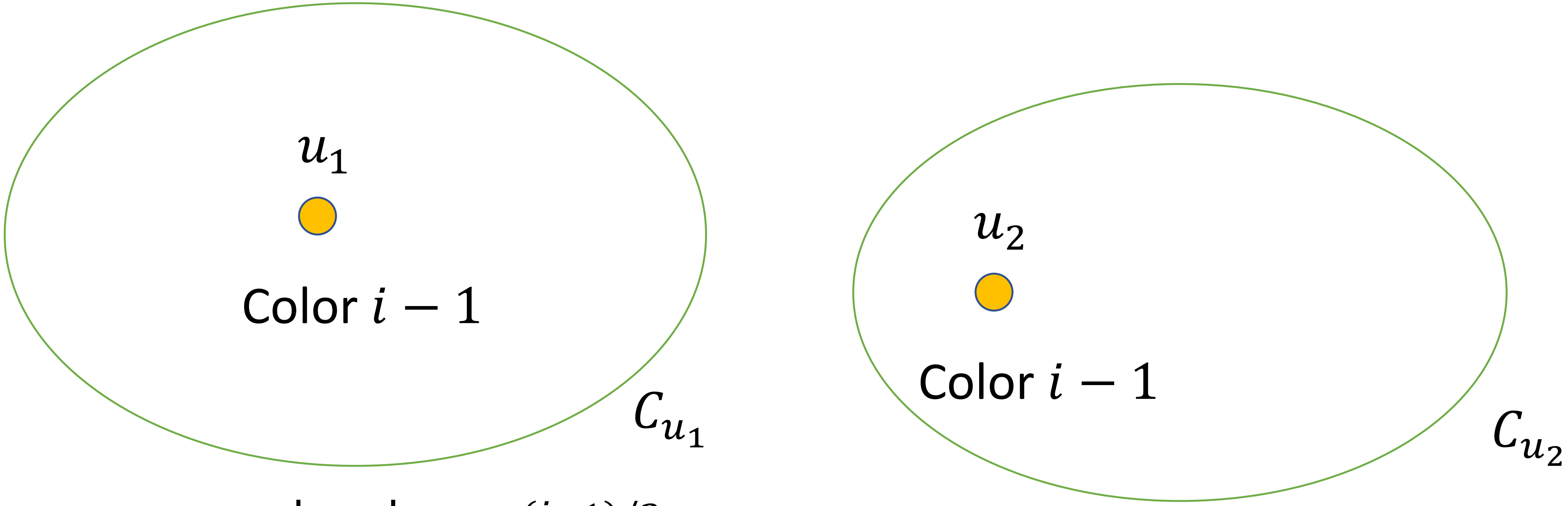
$$C_{u_1} \cap C_{u_2} = \emptyset$$



Theorem $\rho(CBIP) \leq \log n$

Found two vertices u_1 and u_2 colored with $i - 1$

$$C_{u_1} \cap C_{u_2} = \emptyset$$



By induction $|C_{u_1}| \geq \lceil 2^{(i-1)/2} \rceil$

By induction $|C_{u_2}| \geq \lceil 2^{(i-1)/2} \rceil$

Theorem	$\rho(CBIP) \leq \log n$

Therefore we have $\mathrm{n}(\mathrm{i}+1) \geq \left|C_{u_1}\right| + \left|C_{u_2}\right| \geq 2\lceil 2^{(i-1)/2} \rceil \geq \lceil 2^{(i+1)/2} \rceil$

Set $i = 2\log n$ then we get $n(i) = n$

Hence we get that on all bipartite graphs of size n $CBIP$ uses $\leq 2\log n$ colors, while $OPT = 2$

QED