Metric Dimension and Geodetic Set Parameterized by Vertex Cover

@STACS

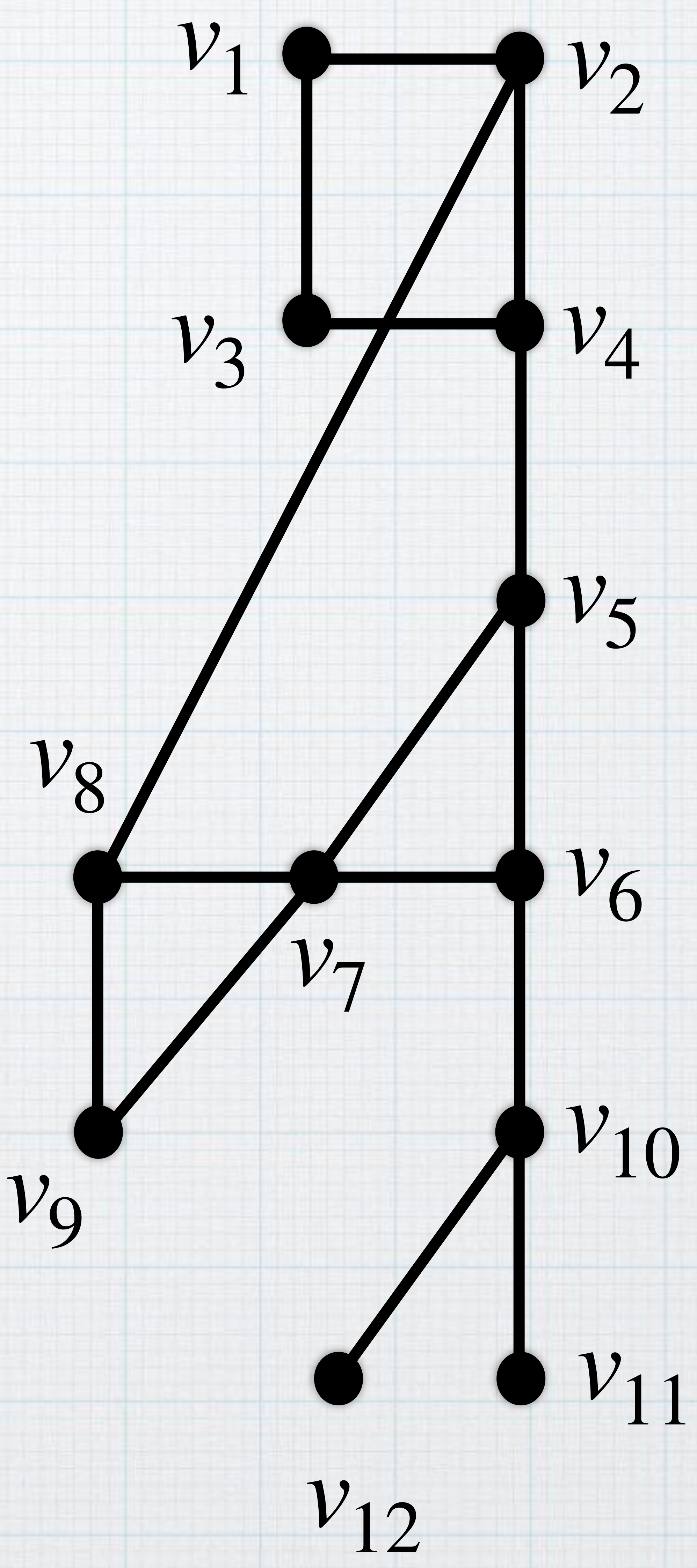
March 05, 2025

Prafullkumar Tale

Joint work with

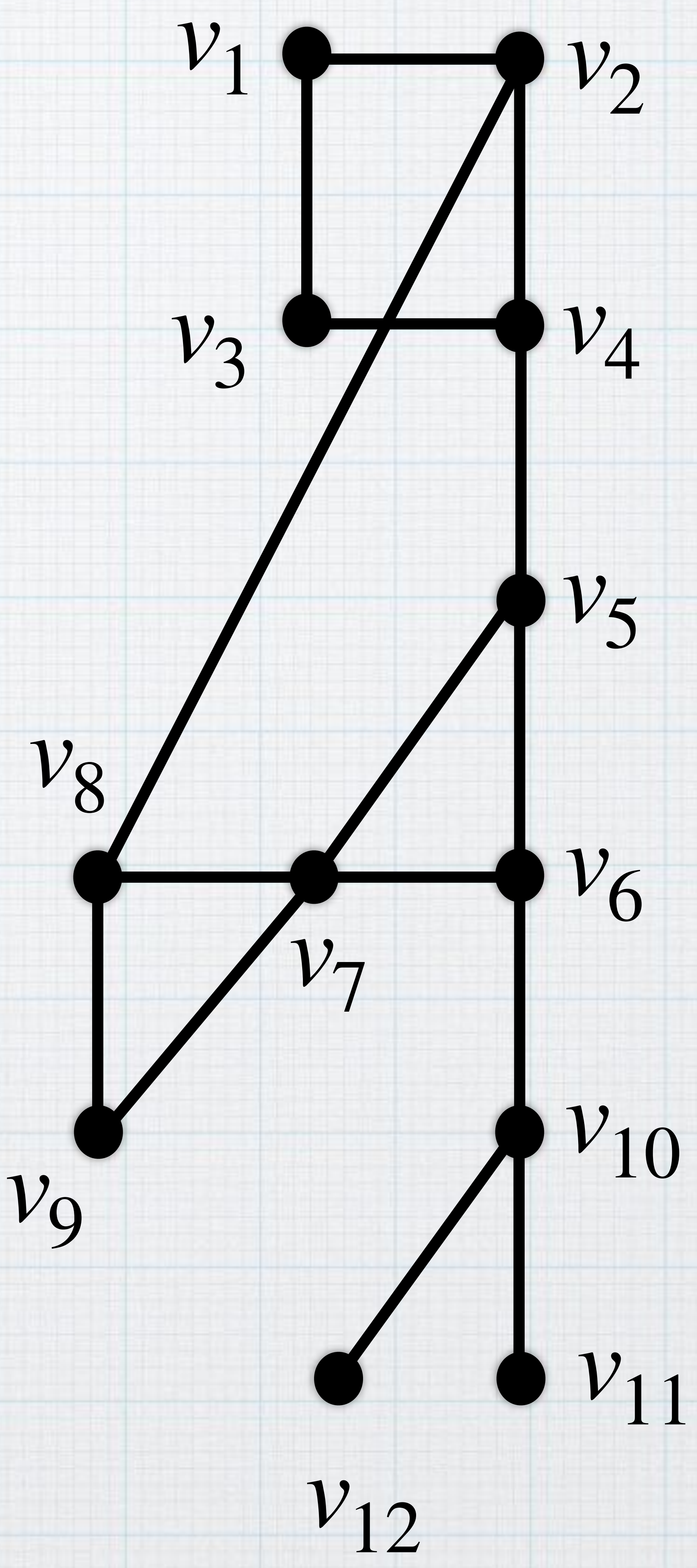
Florent Foucaud, Esther Galby, Liana Khazaliya, Shaohua Li, Fionn Mc Inerney, Roohani Sharma

Catch the Spy



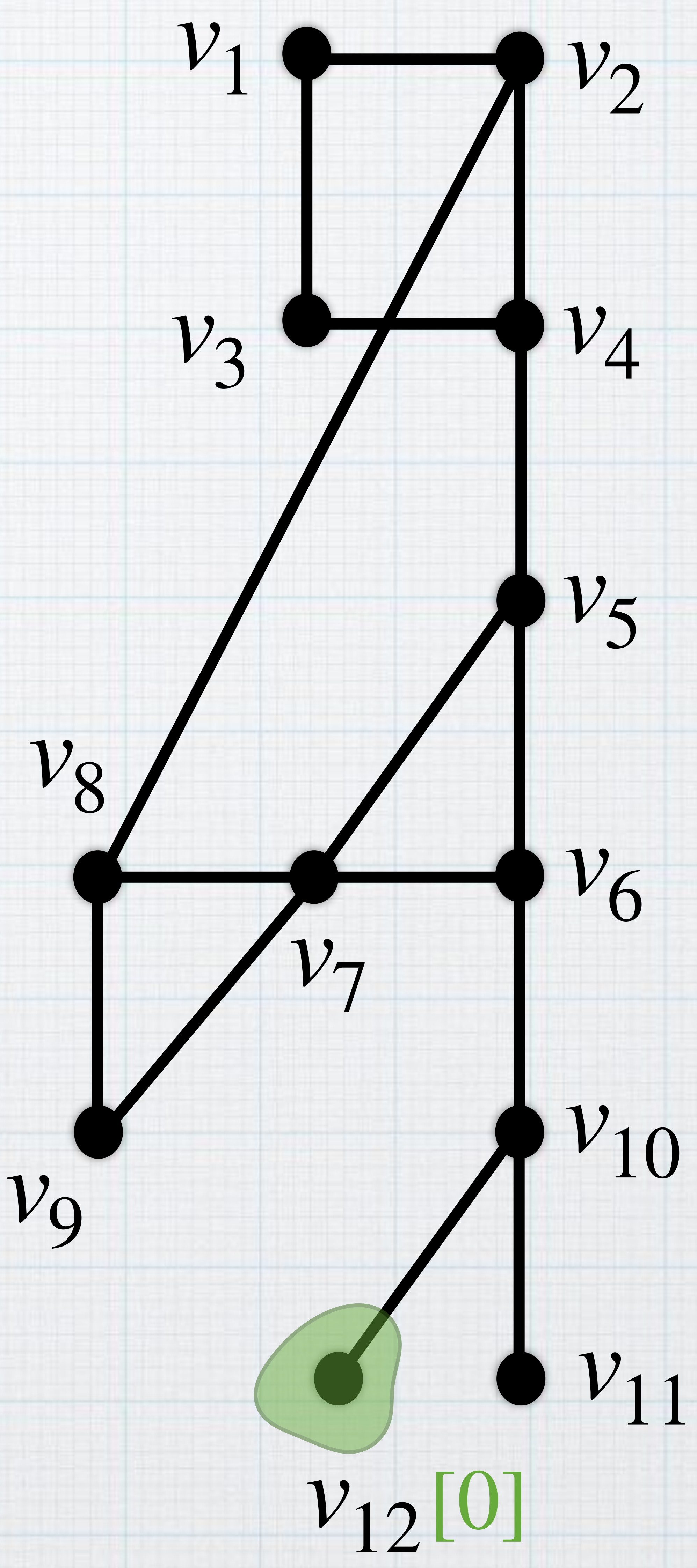
Catch the Spy

– Communication network of agents



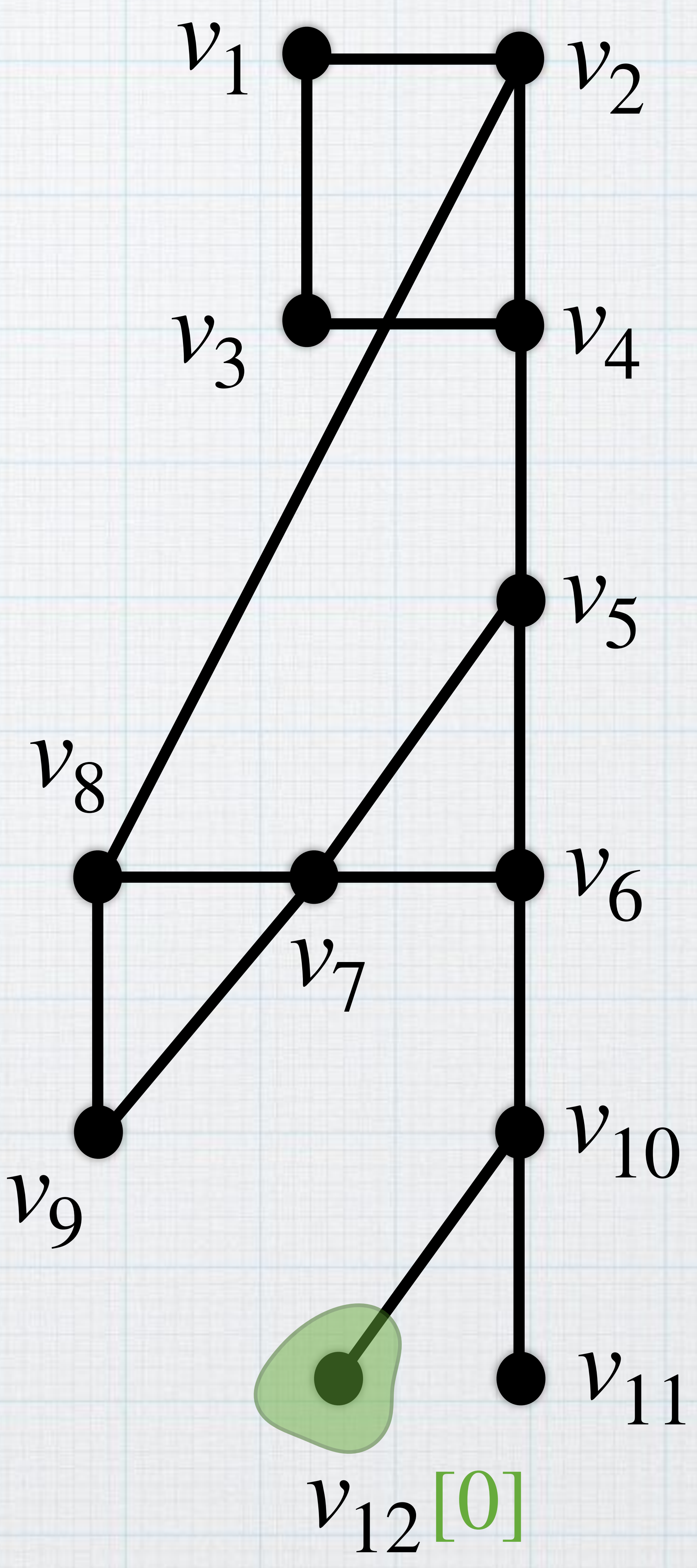
Catch the Spy

- Communication network of agents
- Drop a message at a node Ex. green at v_{12}



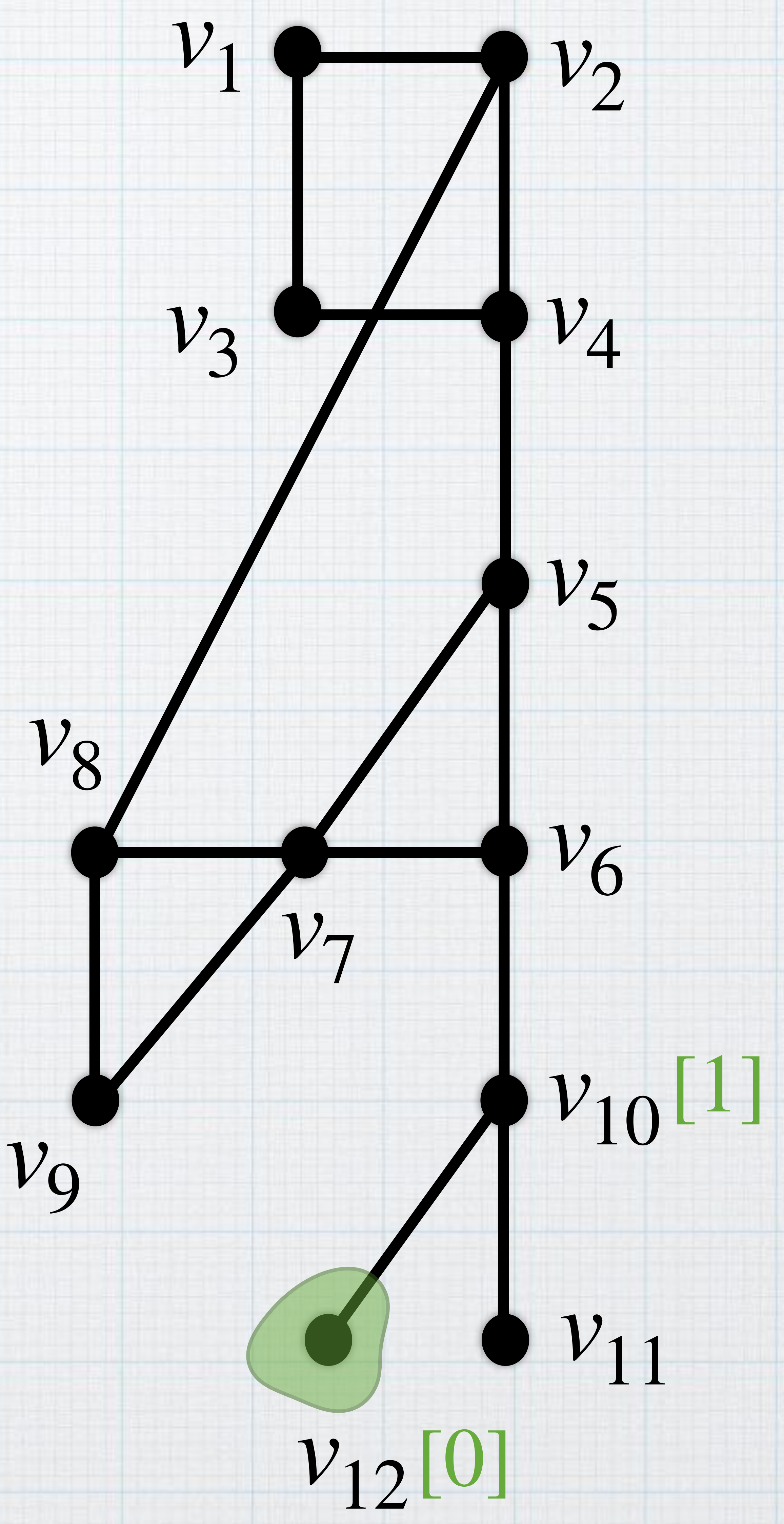
Catch the Spy

- Communication network of agents
- Drop a message at a node Ex. green at v_{12}
- Each agent conveys the message to all of its neighbours in the next time step



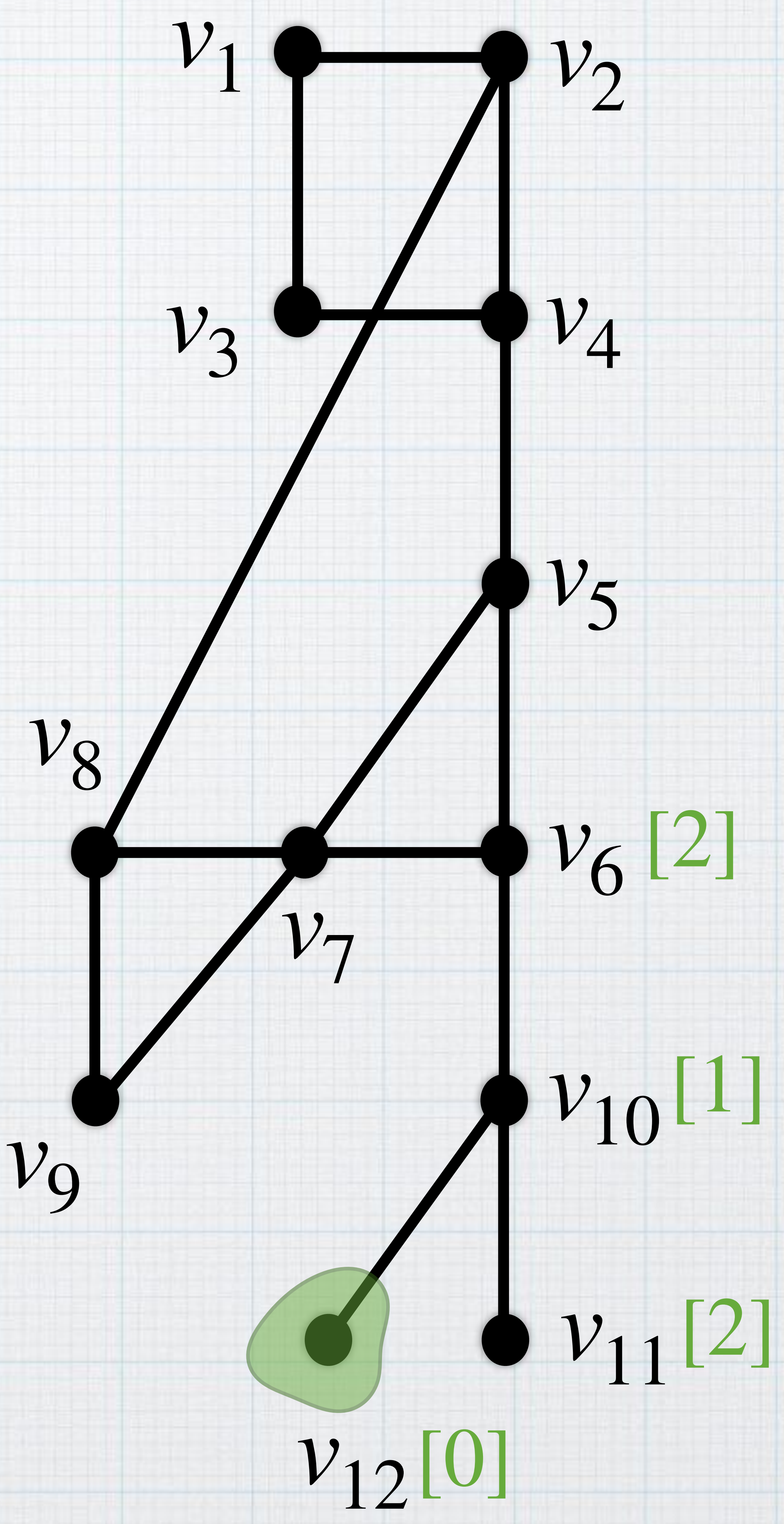
Catch the Spy

- Communication network of agents
- Drop a message at a node Ex. green at v_{12}
- Each agent conveys the message to all of its neighbours in the next time step



Catch the Spy

- Communication network of agents
- Drop a message at a node Ex. green at v_{12}
- Each agent conveys the message to all of its neighbours in the next time step

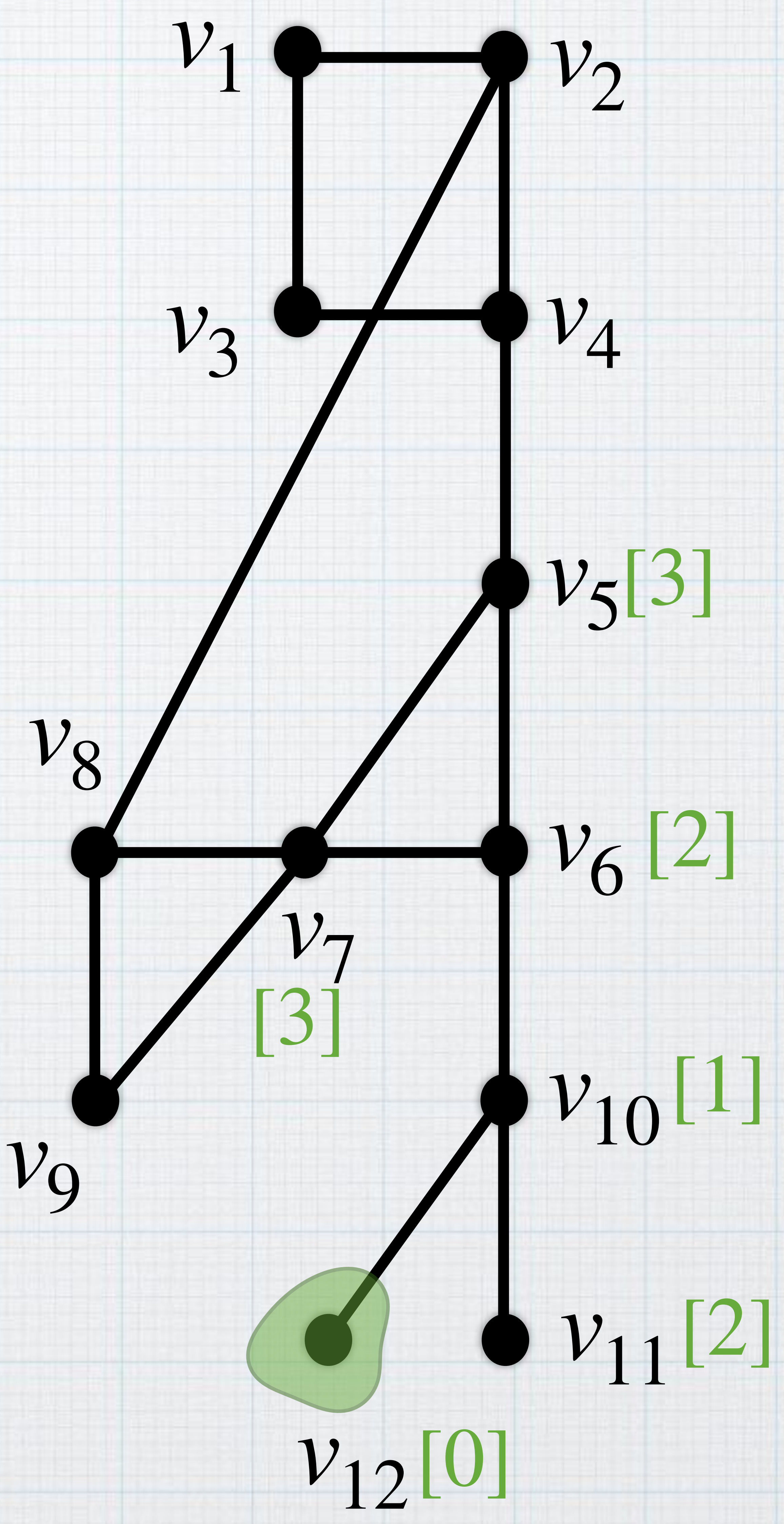


Catch the Spy

– Communication network of agents

– Drop a message at a node Ex. green at v_{12}

– Each agent conveys the message to all of its neighbours in the next time step

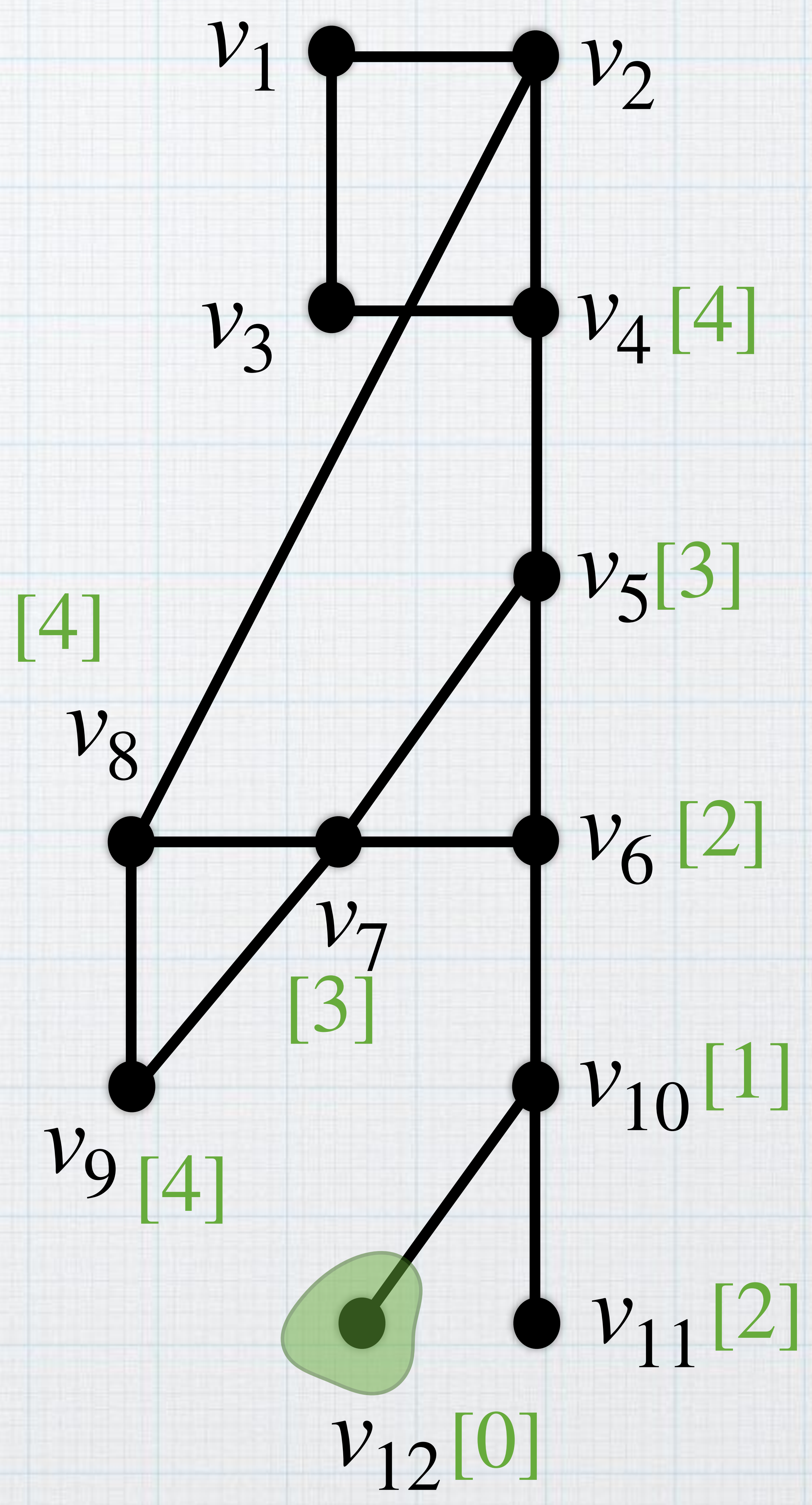


Catch the Spy

– Communication network of agents

– Drop a message at a node Ex. green at v_{12}

– Each agent conveys the message to all of its neighbours in the next time step

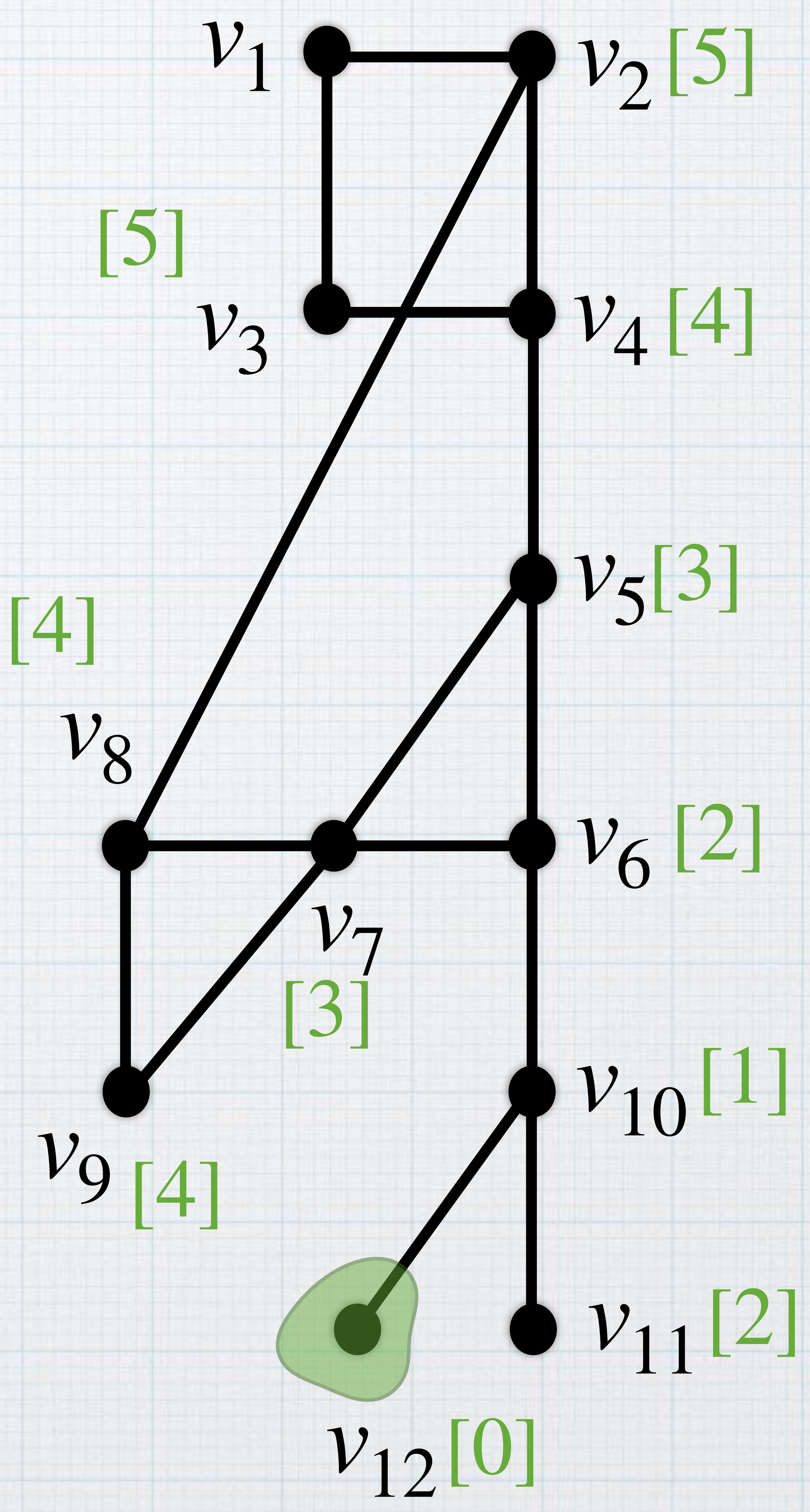


Catch the Spy

– Communication network of agents

– Drop a message at a node Ex. green at v_{12}

– Each agent conveys the message to all of its neighbours in the next time step

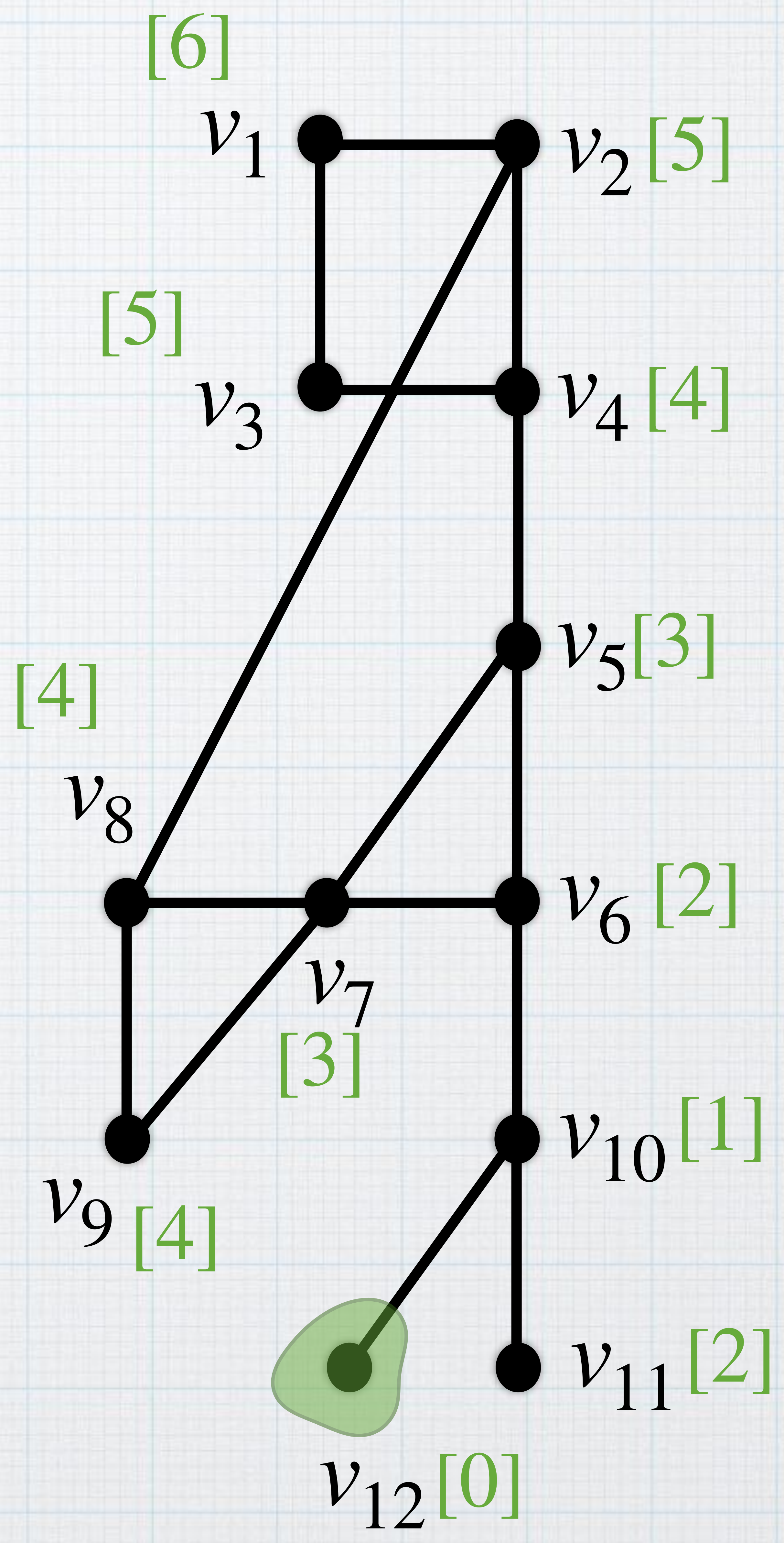


Catch the Spy

– Communication network of agents

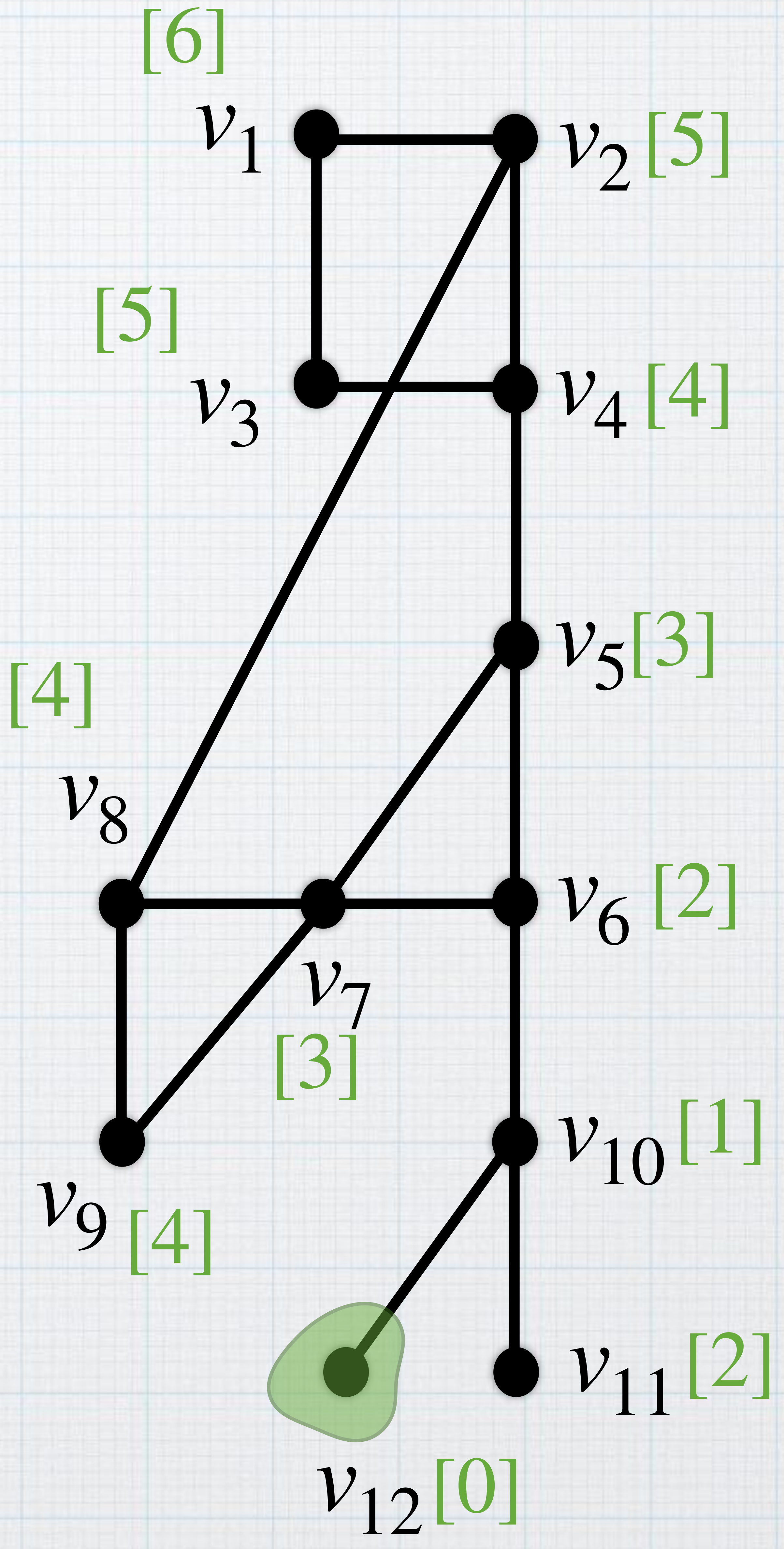
– Drop a message at a node Ex. green at v_{12}

– Each agent conveys the message to all of its neighbours in the next time step



Catch the Spy

- Communication network of agents
- Drop a message at a node Ex. green at v_{12}
- Each agent conveys the message to all of its neighbours in the next time step
- [Time at which the message is received for the first time]



Catch the Spy

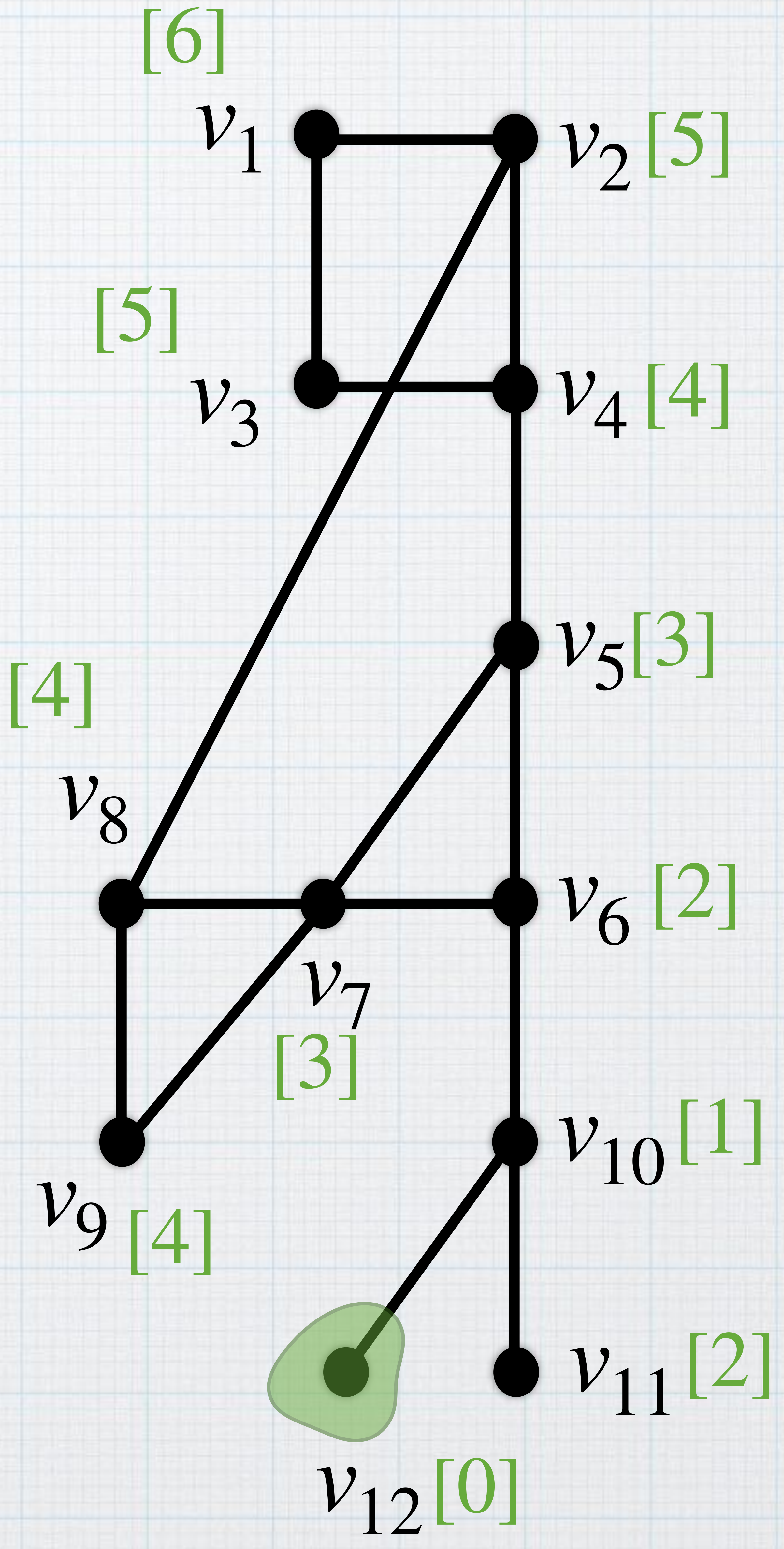
– Communication network of agents

– Drop a message at a node Ex. green at v_{12}

– Each agent conveys the message to all of its neighbours in the next time step

– [Time at which the message is received for the first time]

– Catch the message out of network



Catch the Spy

– Communication network of agents

– Drop a message at a node Ex. green at v_{12}

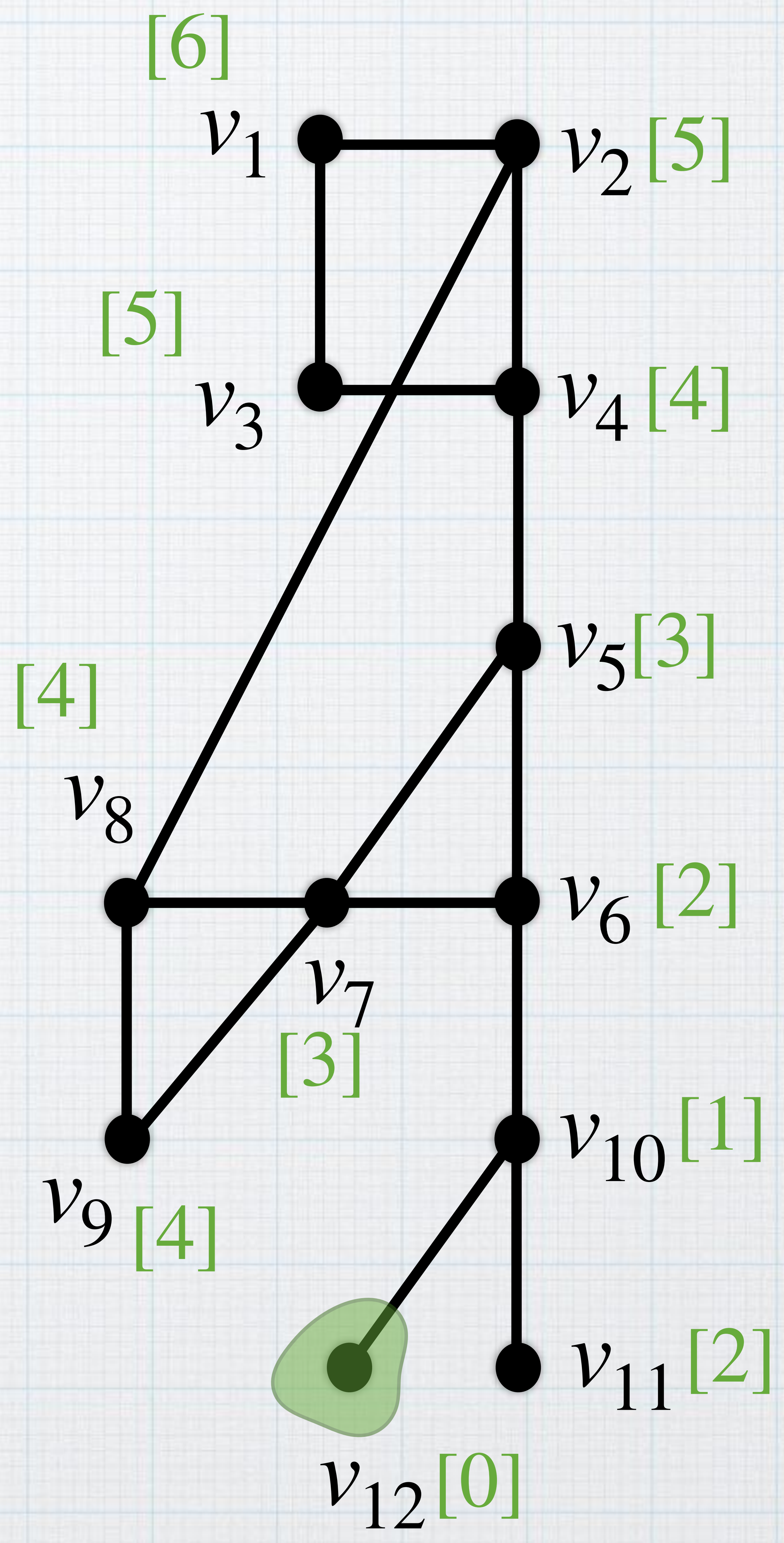
– Each agent conveys the message to all of its neighbours in the next time step

– [Time at which the message is received for the first time]

– Catch the message out of network

[0] $:\{v_{12}\}$ [1] $:\{v_{10}\}$ [2] $:\{v_6, v_{11}\}$

[3] $:\{v_5, v_7\}$ [4] $:\{v_4, v_8, v_9\}$ [5] $:\{v_6, v_{11}\}$ [6] $:\{v_1\}$



Catch the Spy

– Communication network of agents

– Drop a message at a node Ex. green at v_{12}

– Each agent conveys the message to all of its neighbours in the next time step

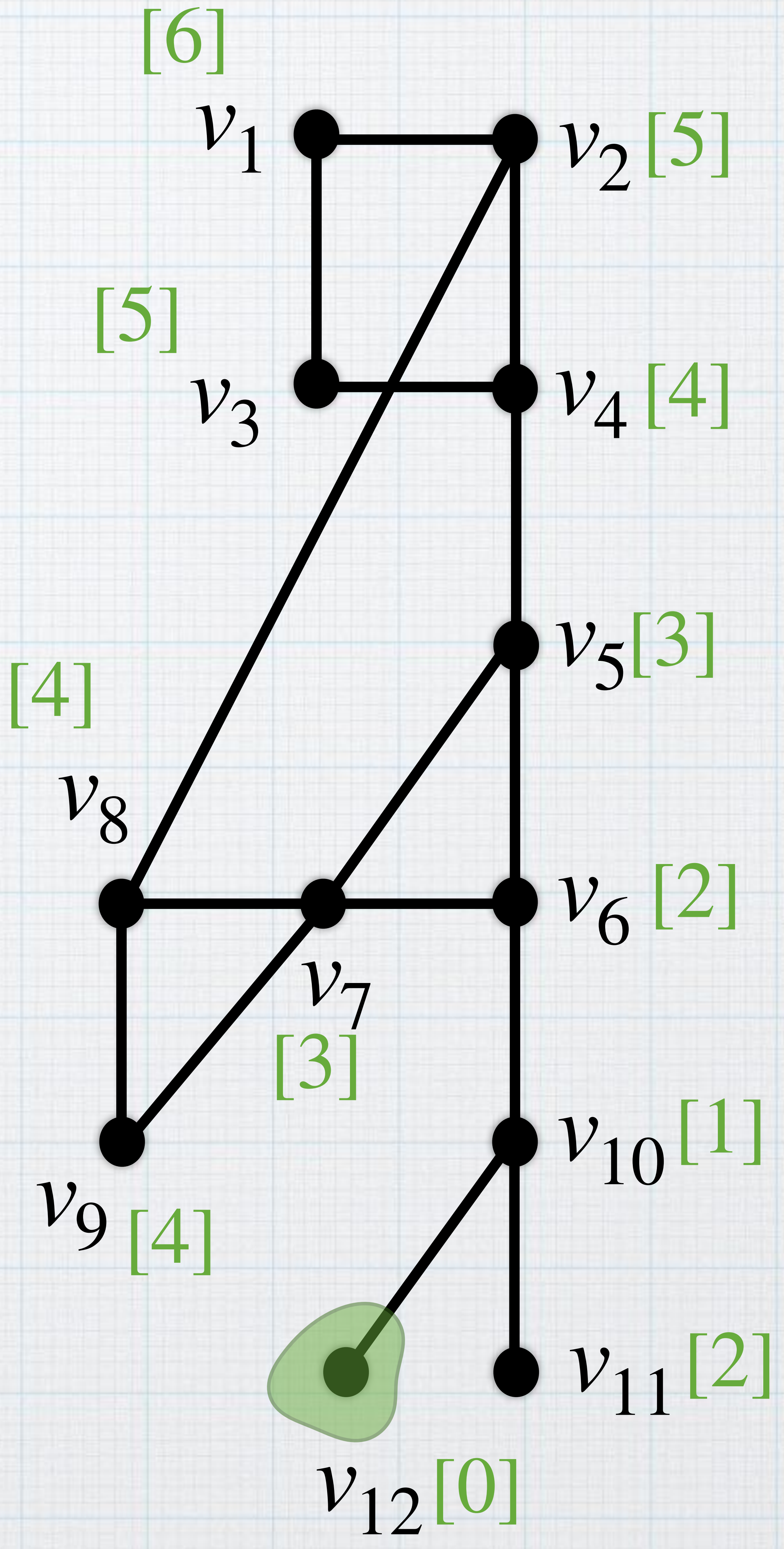
– [Time at which the message is received for the first time]

– Catch the message out of network

[0] $:\{v_{12}\}$ [1] $:\{v_{10}\}$ [2] $:\{v_6, v_{11}\}$

[3] $:\{v_5, v_7\}$ [4] $:\{v_4, v_8, v_9\}$ [5] $:\{v_6, v_{11}\}$ [6] $:\{v_1\}$

Q: Minimum nr of messages to catch the spy?

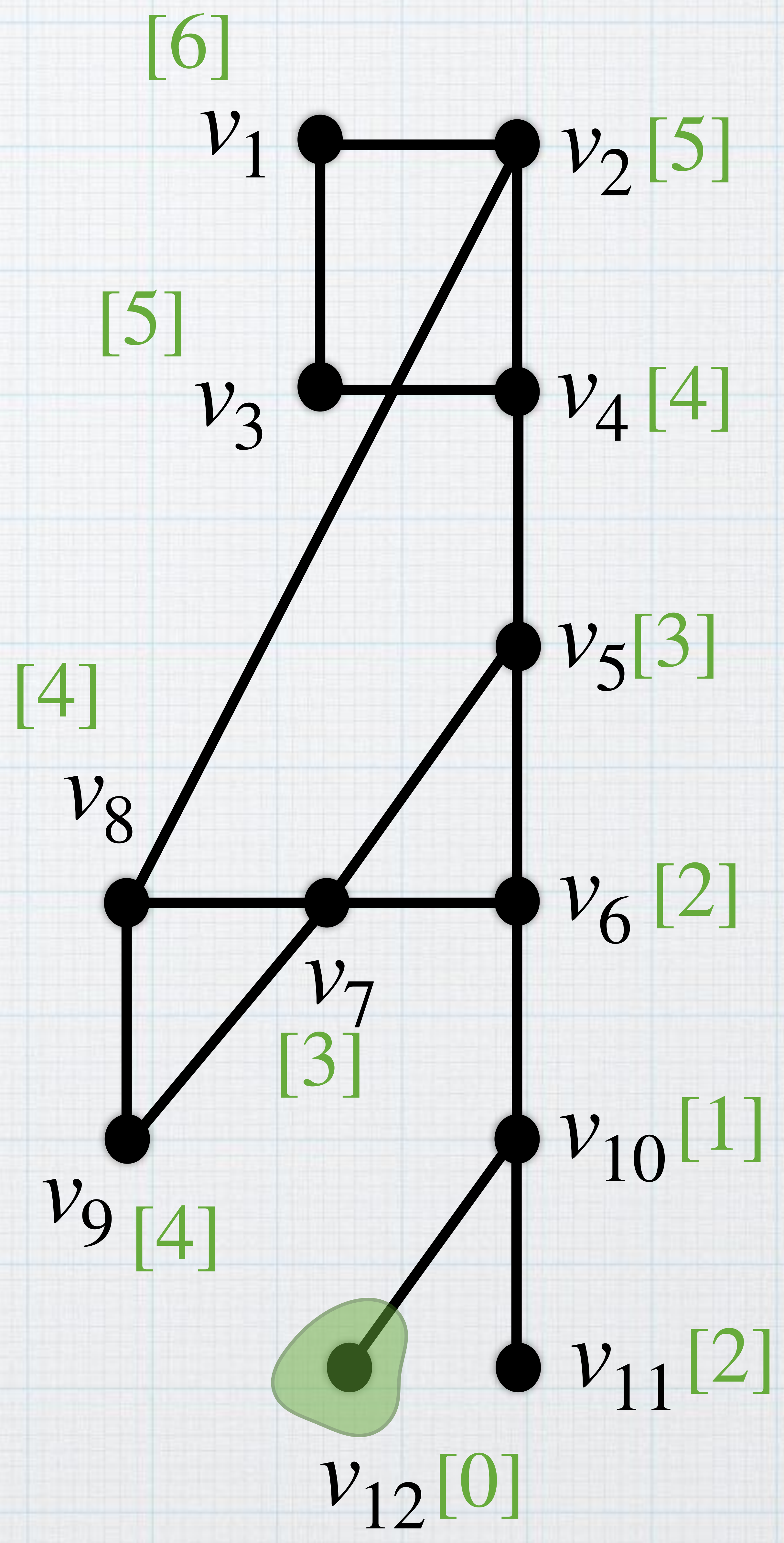


Catch the Spy

– Drop message green at v_{12}

[0] $:\{v_{12}\}$ [1] $:\{v_{10}\}$ [2] $:\{v_6, v_{11}\}$

[3] $:\{v_5, v_7\}$ [4] $:\{v_4, v_8, v_9\}$ [5] $:\{v_6, v_{11}\}$ [6] $:\{v_1\}$

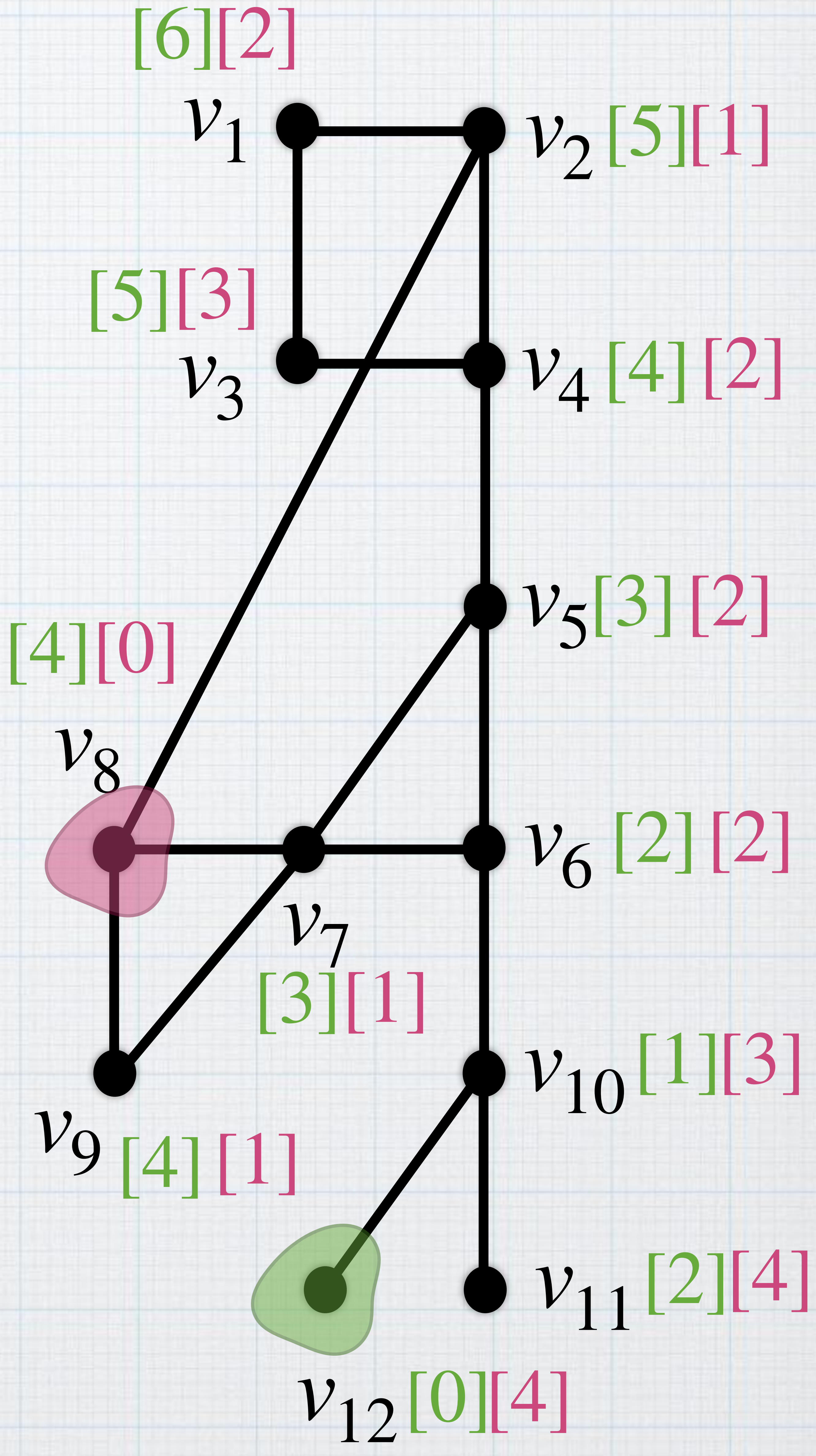


Catch the Spy

- Drop message green at v_{12}

[0] $:\{v_{12}\}$ [1] $:\{v_{10}\}$ [2] $:\{v_6, v_{11}\}$
[3] $:\{v_5, v_7\}$ [4] $:\{v_4, v_8, v_9\}$ [5] $:\{v_6, v_{11}\}$ [6] $:\{v_1\}$

- Drop message red at v_8



Catch the Spy

– Drop message green at v_{12}

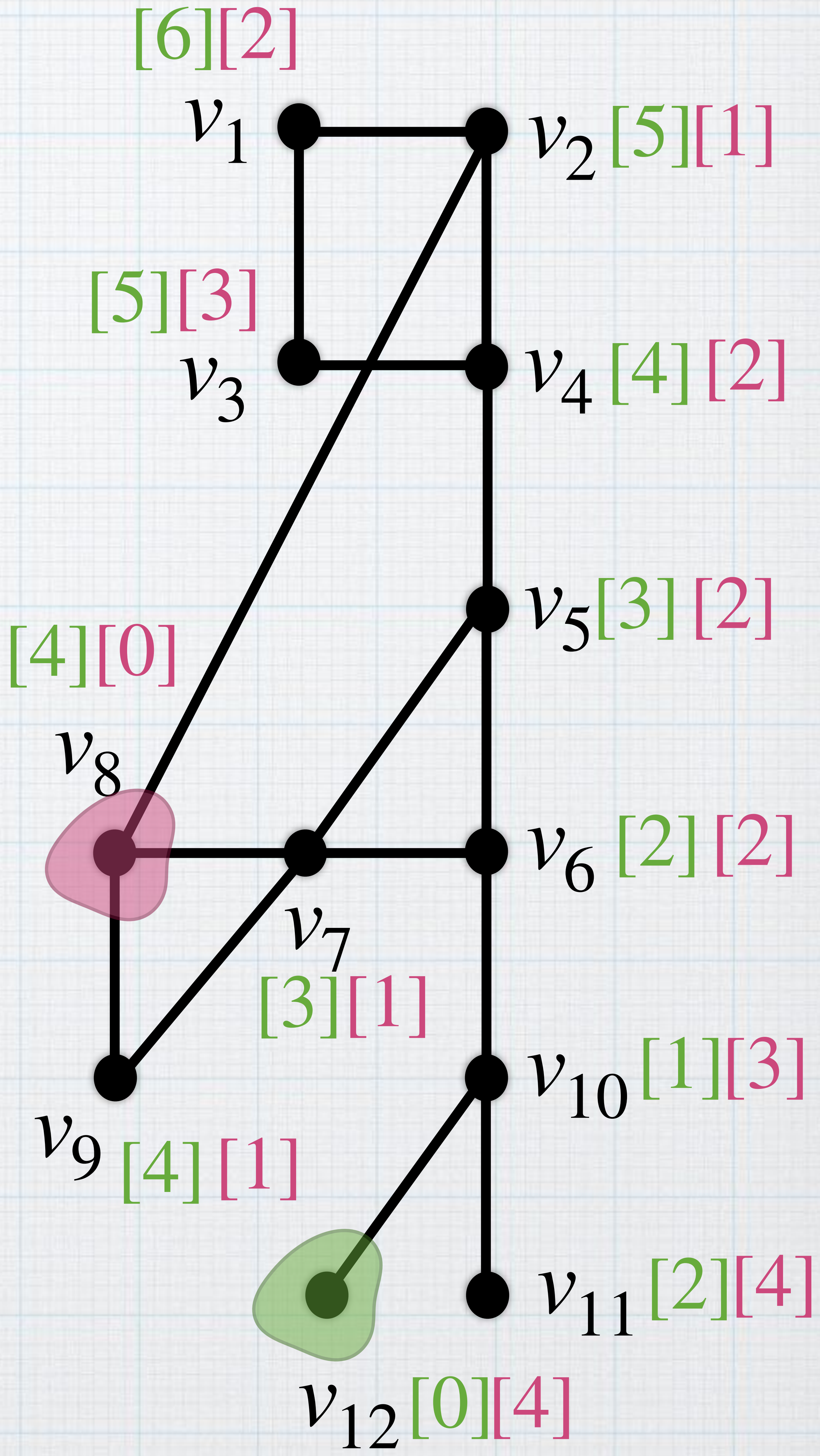
[0] $:\{v_{12}\}$ [1] $:\{v_{10}\}$ [2] $:\{v_6, v_{11}\}$

[3] $:\{v_5, v_7\}$ [4] $:\{v_4, v_8, v_9\}$ [5] $:\{v_6, v_{11}\}$ [6] $:\{v_1\}$

– Drop message red at v_8

[0] $:\{v_8\}$ [1] $:\{v_2, v_7, v_9\}$

[2] $:\{v_1, v_4, v_5, v_6\}$ [3] $:\{v_3, v_{10}\}$ [4] $:\{v_{11}, v_{12}\}$



Catch the Spy

- Drop message green at v_{12}

[0] $:\{v_{12}\}$ [1] $:\{v_{10}\}$ [2] $:\{v_6, v_{11}\}$

[3] $:\{v_5, v_7\}$ [4] $:\{v_4, v_8, v_9\}$ [5] $:\{v_6, v_{11}\}$ [6] $:\{v_1\}$

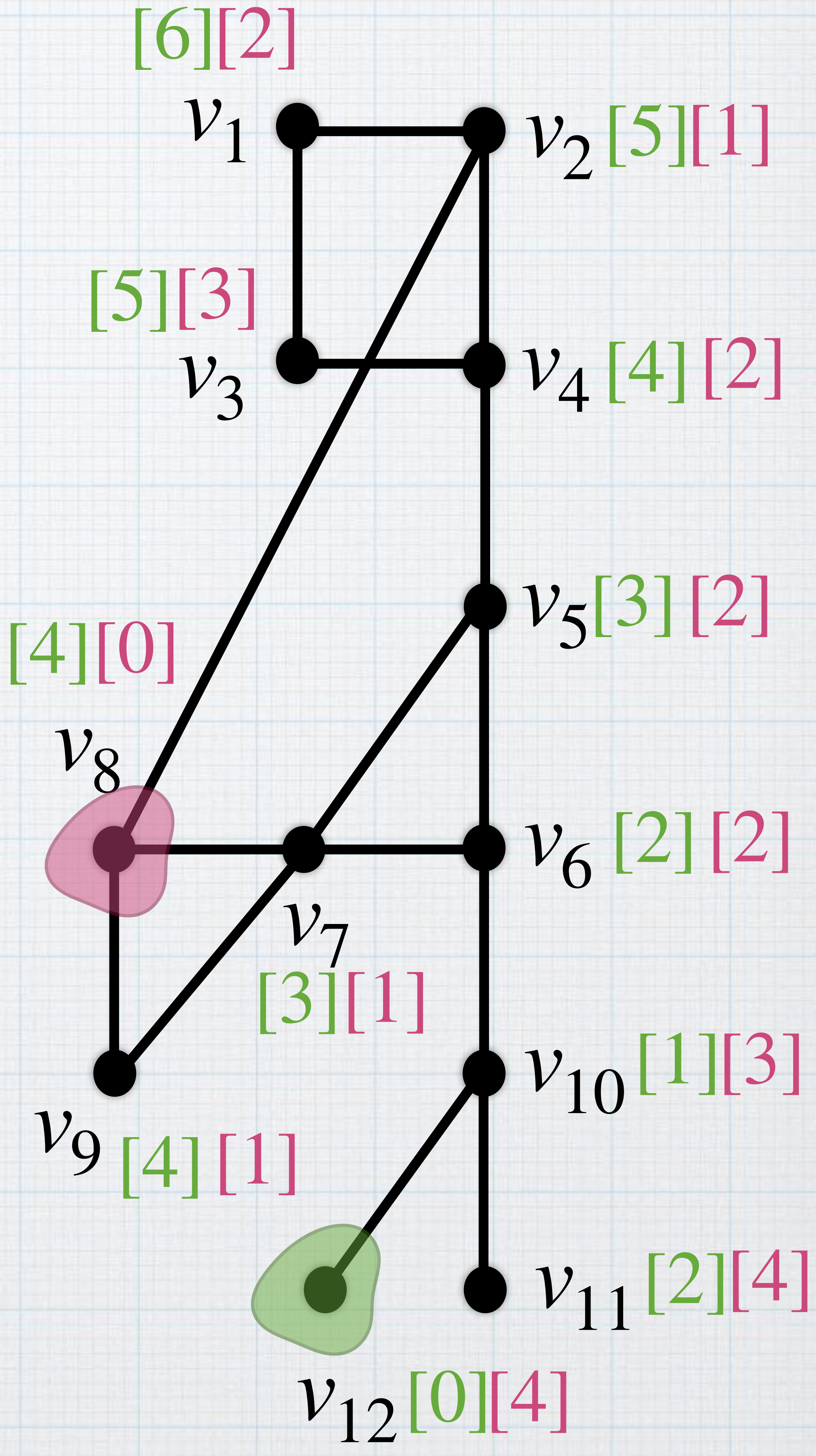
- Drop message red at v_8

[0] $:\{v_8\}$ [1] $:\{v_2, v_7, v_9\}$

[2] $:\{v_1, v_4, v_5, v_6\}$ [3] $:\{v_3, v_{10}\}$ [4] $:\{v_{11}, v_{12}\}$

- Any (valid) combination is unique.

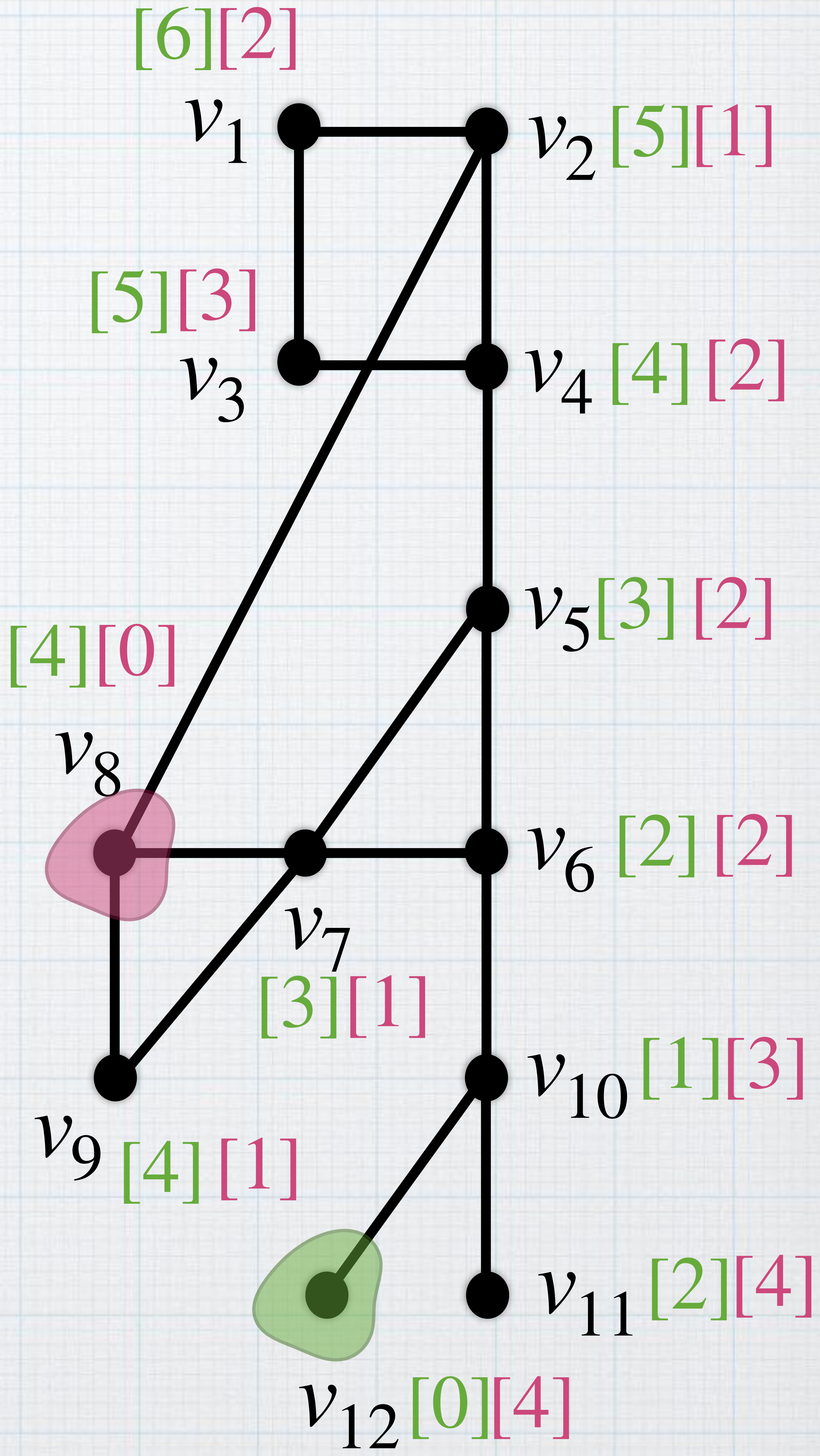
[2][2] — v_6, [4][2] — v_4, [4][1] — v_9



Catch the Spy

- Drop message green at v_{12}
- Drop message red at v_8
- Any (valid) combination is unique.

[2][2] — v_6, [4][2] — v_4, [4][1] — v_9

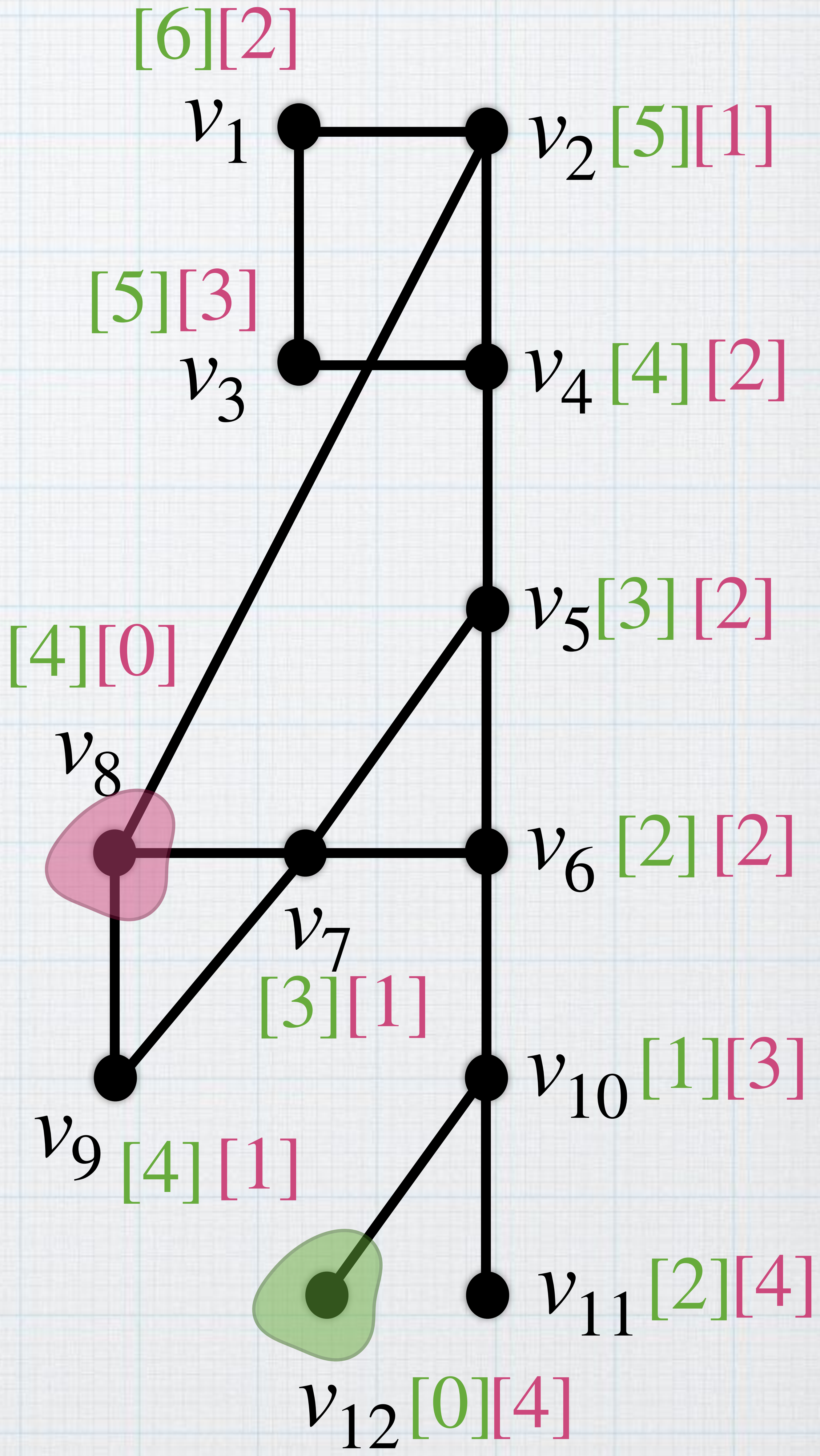


Catch the Spy

- Drop message green at v_{12}
- Drop message red at v_8
- Any (valid) combination is unique.

[2][2] — v_6, [4][2] — v_4, [4][1] — v_9

- $\{v_{12}, v_8\}$ is called a resolving set.



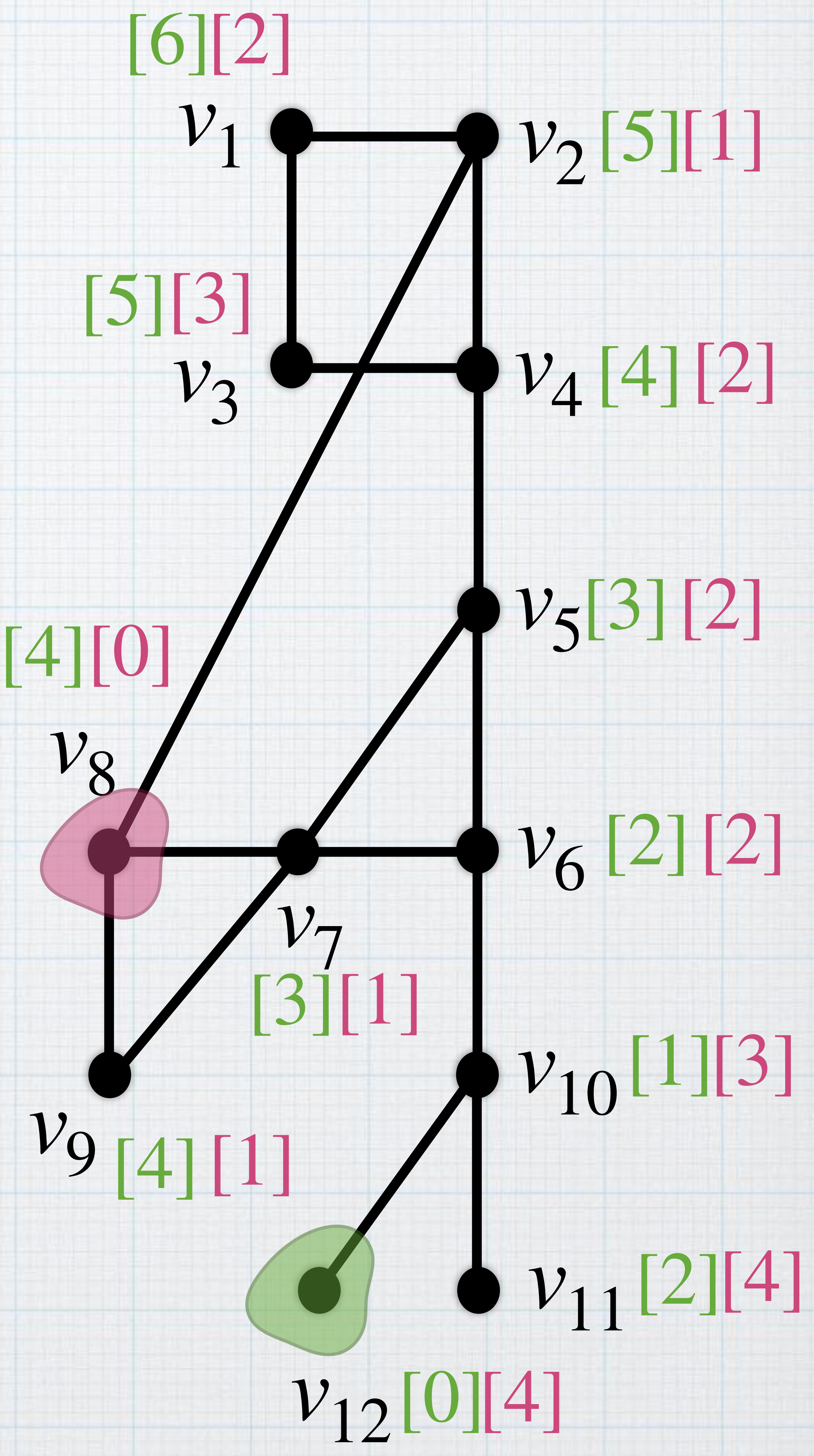
Catch the Spy

- Drop message green at v_{12}
- Drop message red at v_8
- Any (valid) combination is unique.

[2][2] — v_6, [4][2] — v_4, [4][1] — v_9

- $\{v_{12}, v_8\}$ is called a resolving set.

- A resolving set is an ordered set $S = \{s_1, s_2, ..\} \subseteq V(G)$ s.t. any $u \neq v \in V(G)$, $\langle d(u, s_1);\ d(u, s_2);\ ..\rangle \neq \langle d(v, s_1);\ d(v, s_2);\ ..\rangle$



Catch the Spy

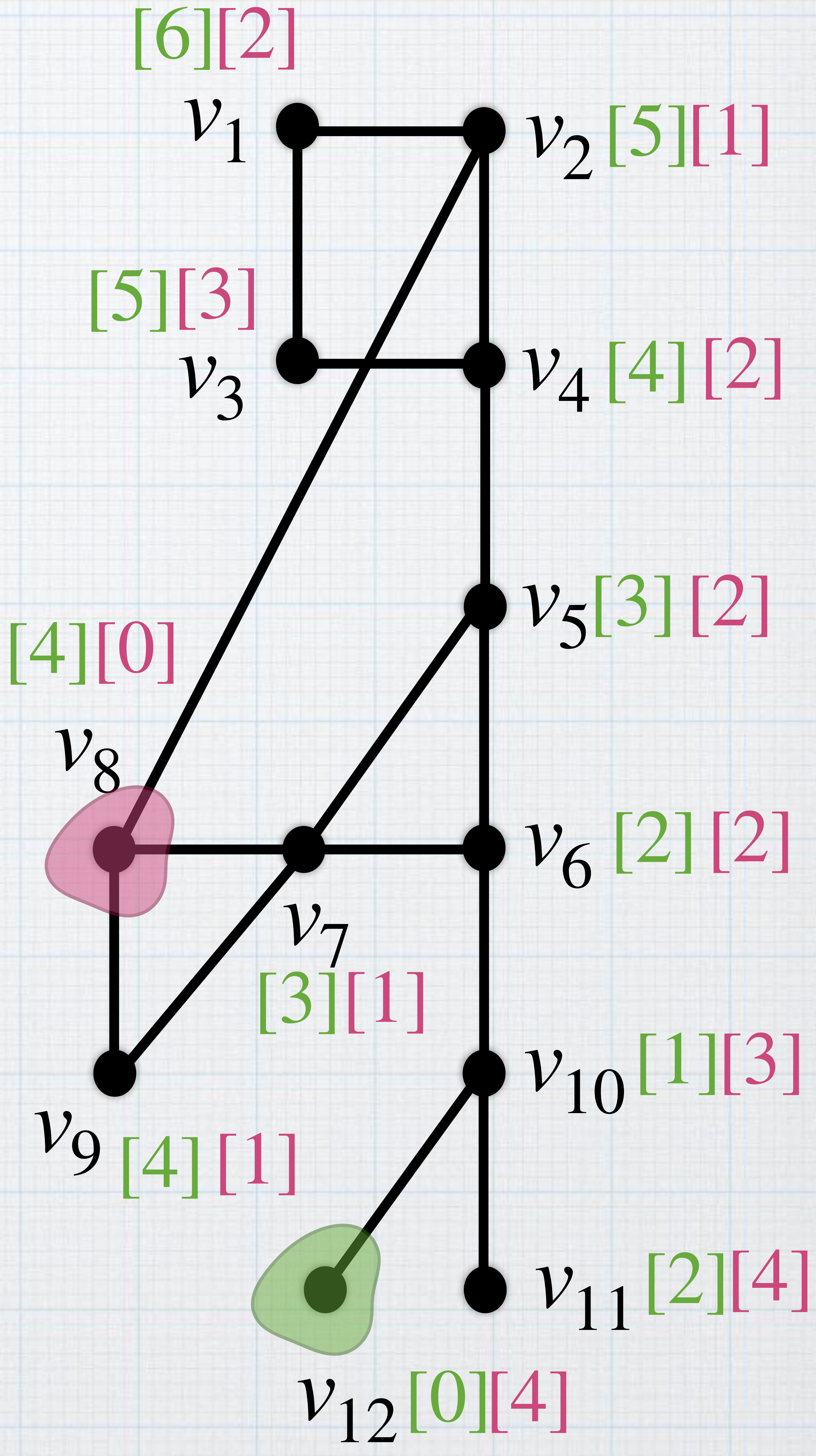
- Drop message green at v_{12}
- Drop message red at v_8
- Any (valid) combination is unique.

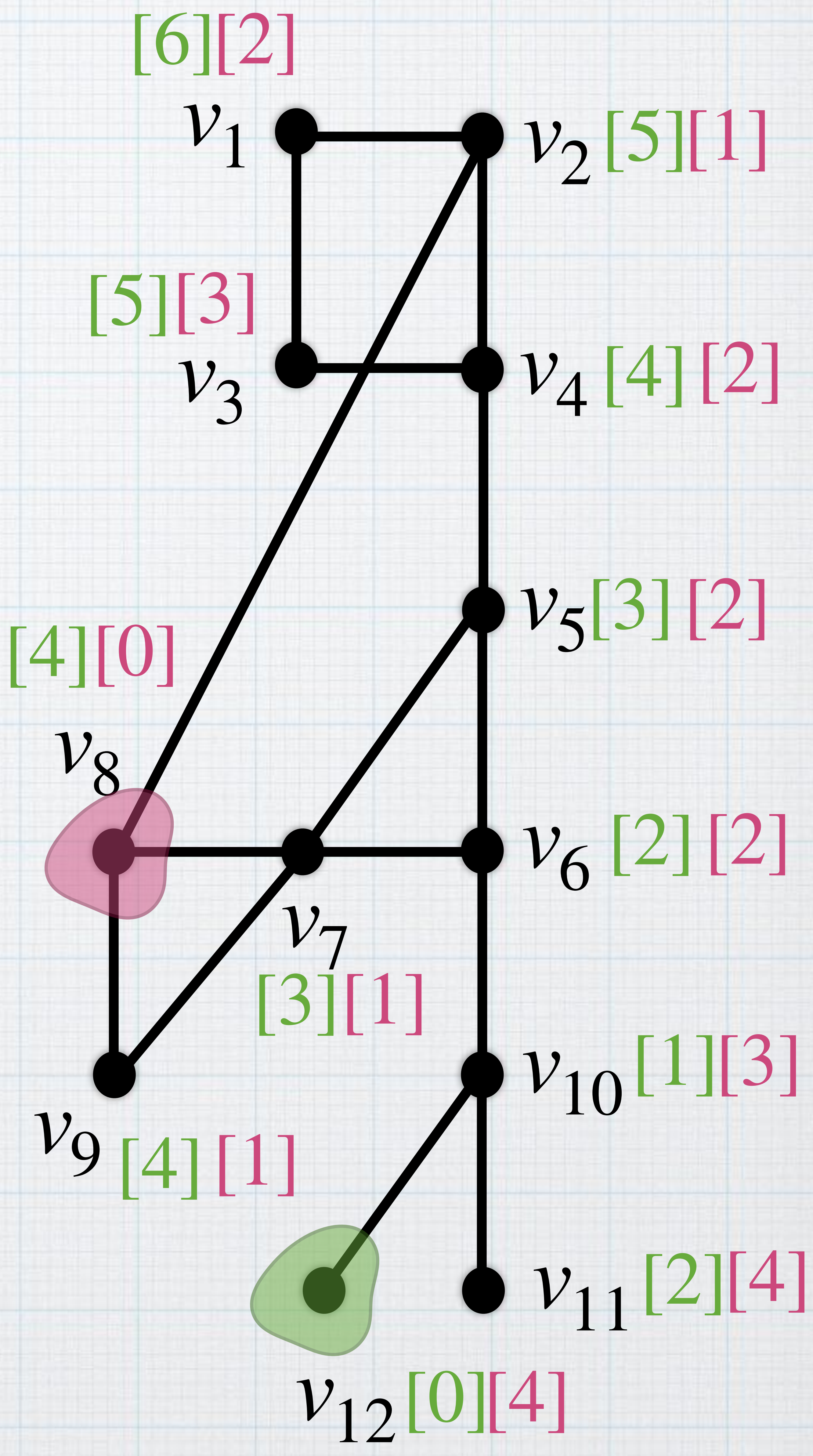
[2][2] — v_6, [4][2] — v_4, [4][1] — v_9

- $\{v_{12}, v_8\}$ is called a resolving set.

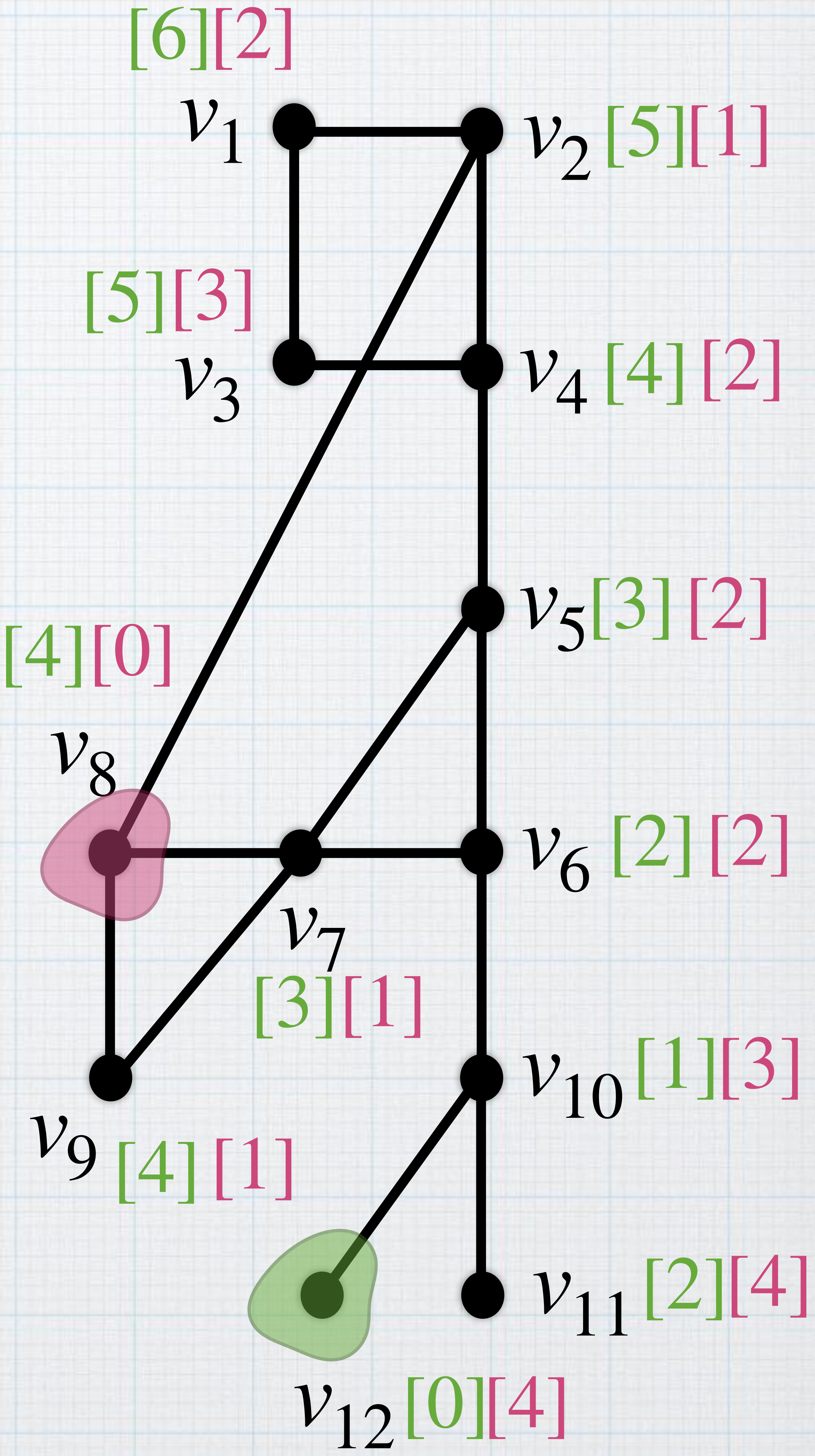
- A resolving set is an ordered set $S = \{s_1, s_2, ..\} \subseteq V(G)$ s.t. any $u \neq v \in V(G)$, $\langle d(u, s_1);\ d(u, s_2);\ ..\rangle \neq \langle d(v, s_1);\ d(v, s_2);\ ..\rangle$

- $d(u, s_i)$ = shortest distance from u to s_i




[6][2]
v_1
v_2 [5][1]
[5][3]
v_3
v_4 [4] [2]
v_5[3] [2]
[4][0]
v_8
v_6 [2] [2]
v_7
[3][1]
v_{10} [1][3]
v_9 [4] [1]
v_{11} [2][4]
v_{12} [0][4]

met–dim(G) = size of smallest resolving set

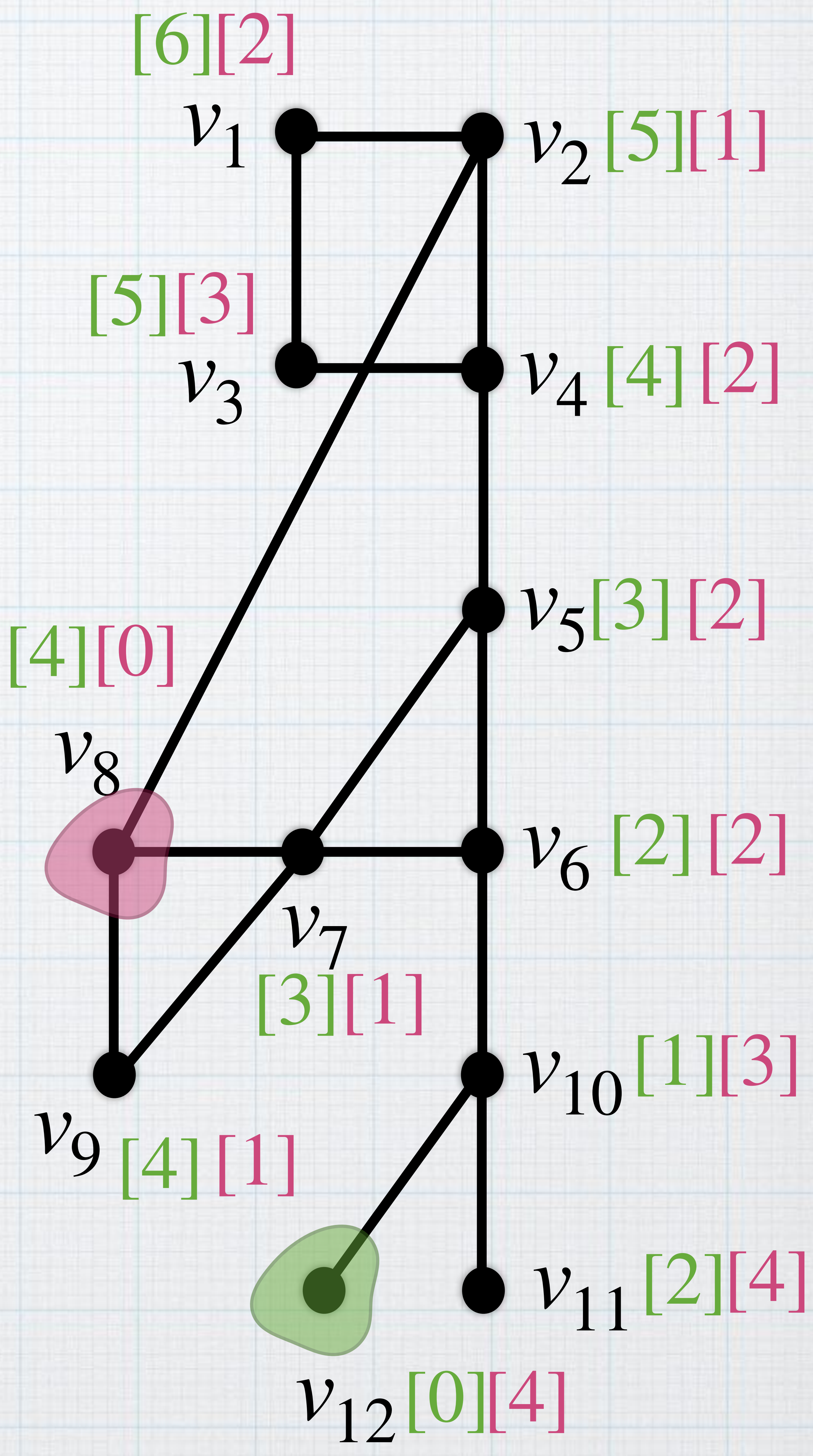


met–dim(G) = size of smallest resolving set

Metric Dimension

Input: Graph G, int k

Output: Does there exist a resolving set of size at most k?



met–dim(G) = size of smallest resolving set

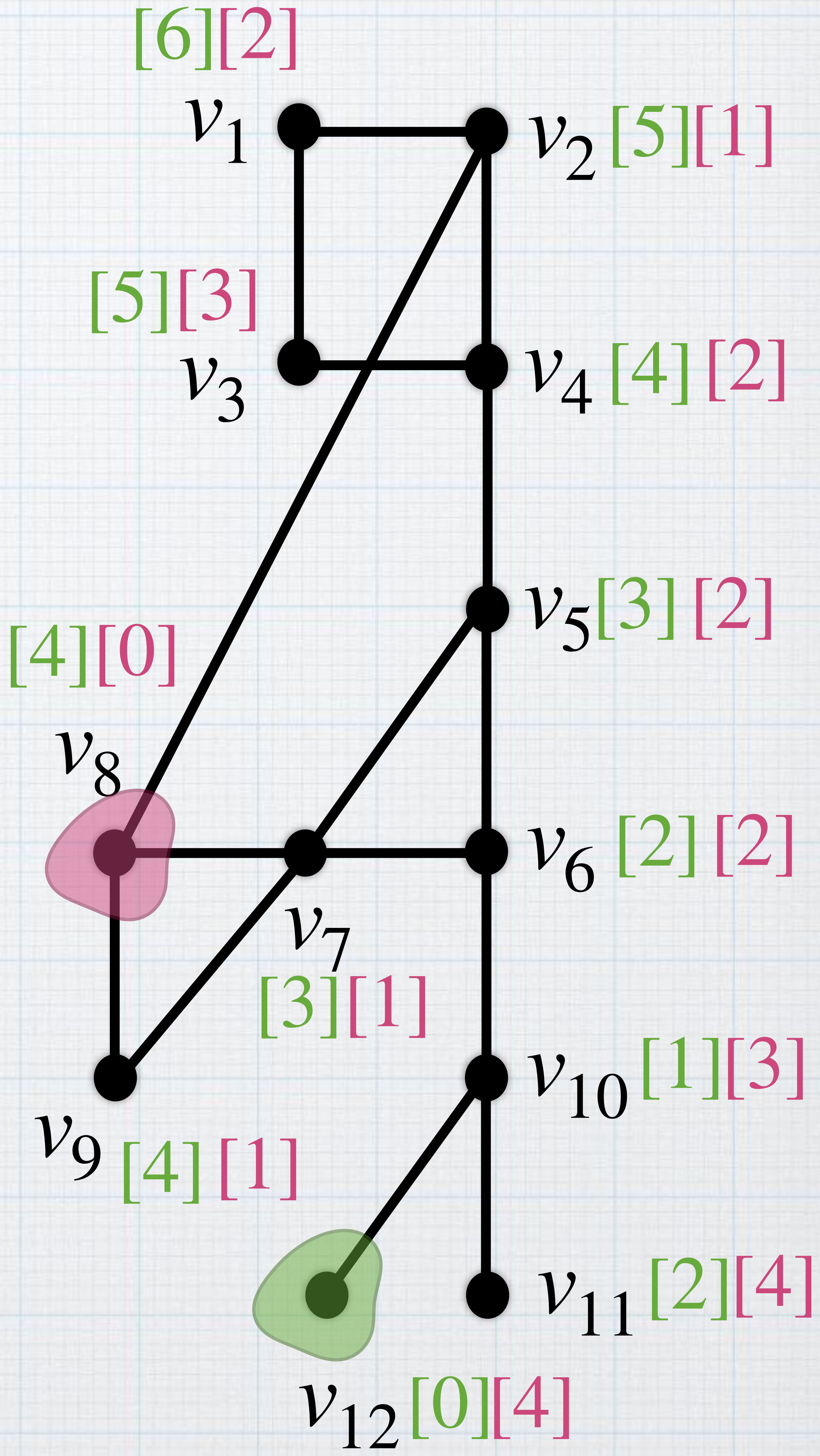
Metric Dimension

Input: Graph G, int k

Output: Does there exist a resolving set of size at most k?

– Solution at one part of the graph, plays important role in other part of the graph.

Ex. v_{12} resolves v_1, v_2.



met–dim(G) = size of smallest resolving set

Metric Dimension

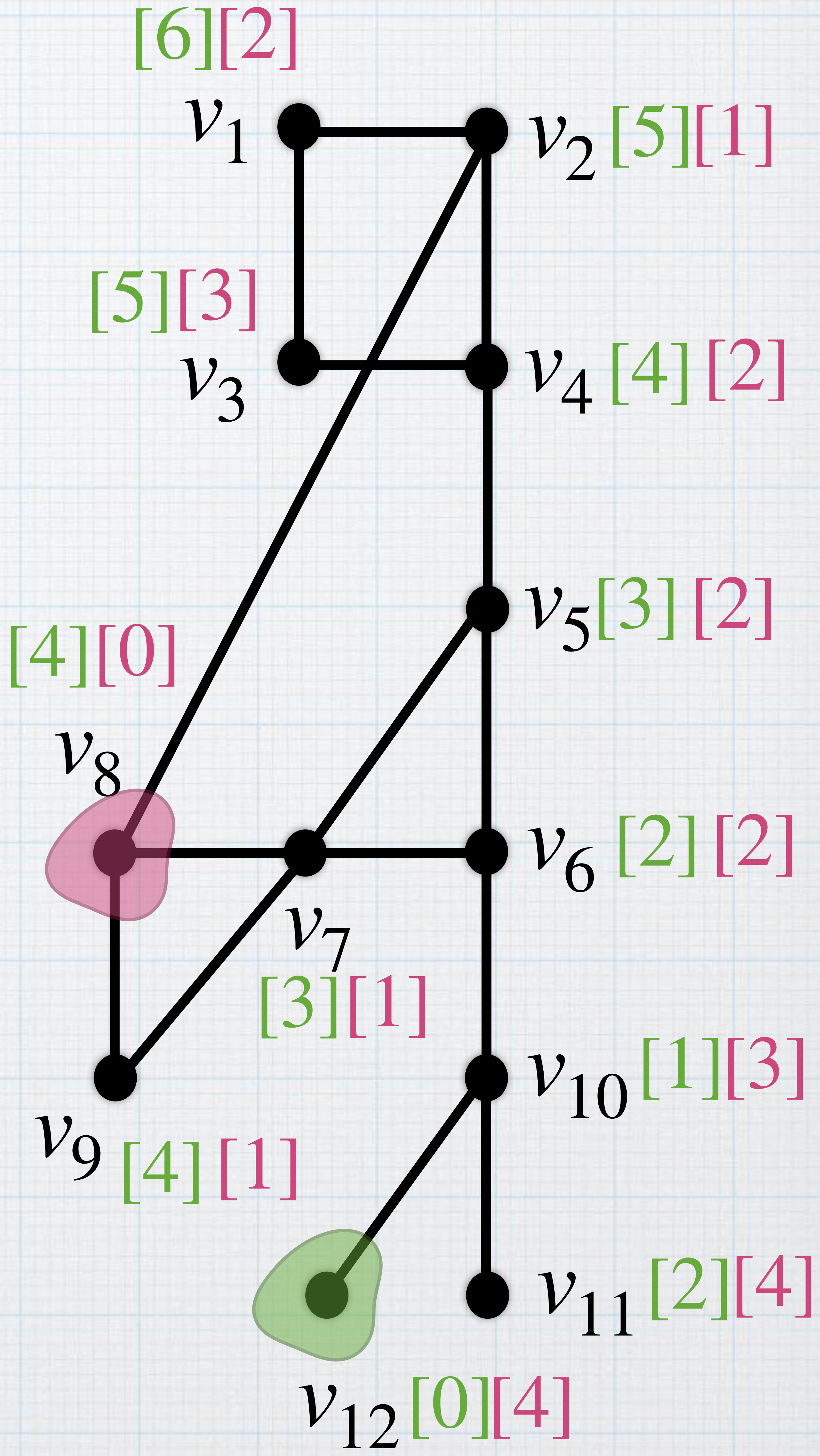
Input: Graph G, int k

Output: Does there exist a resolving set of size at most k?

– Solution at one part of the graph, plays important role in other part of the graph.

Ex. v_{12} resolves v_1, v_2.

– This phenomenon makes the problem hard,



met–dim(G) = size of smallest resolving set

Metric Dimension

Input: Graph G, int k

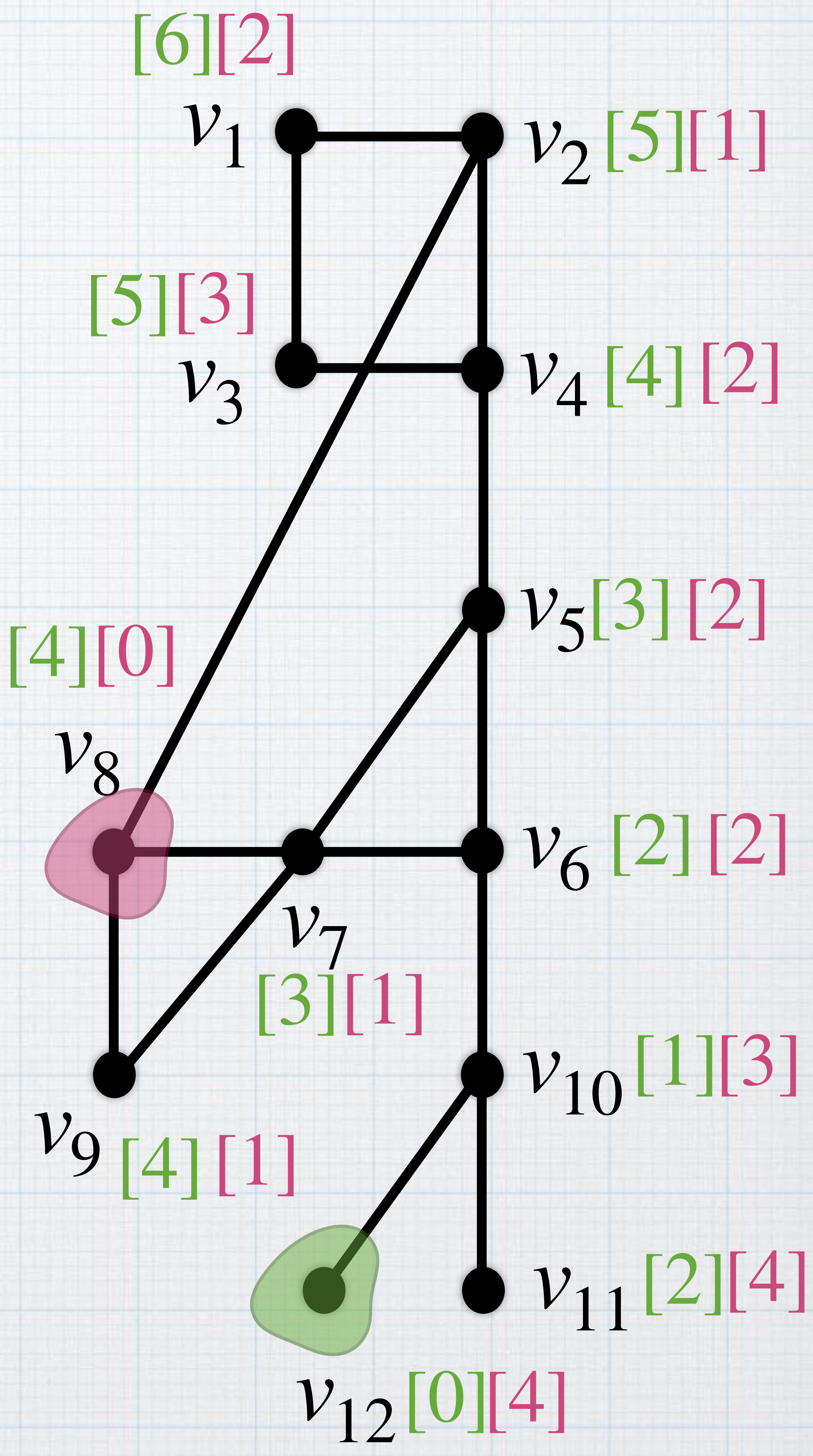
Output: Does there exist a resolving set of size at most k?

– Solution at one part of the graph, plays important role in other part of the graph.

Ex. v_{12} resolves v_1, v_2.

– This phenomenon makes the problem hard,

– but also makes it ideal to obtain exotic lower bounds.



Thm [FGKLIST (ICALP'24)]. The Metric Dimension problem on bounded diameter graphs

- admits $2^{2^{\mathcal{O}(\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$–time algo, but
- does not admit $2^{2^{o(\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Thm [FGKLIST (ICALP'24)]. The Metric Dimension problem on bounded diameter graphs

– admits $2^{2^{\mathcal{O}(\text{tw})}} \cdot n^{\mathcal{O}(1)}$–time algo, but

– does not admit $2^{2^{o(\text{tw})}} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

First NP-Complete problems to admit such lower bounds

Thm [FGKLIST (ICALP'24)]. The Metric Dimension problem on bounded diameter graphs
– admits $2^{2^{\mathcal{O}(\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$–time algo, but
– does not admit $2^{2^{o(\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

First NP–Complete problems to admit such lower bounds

Earlier known results were Σ_2^p–complete or Π_2^p–complete.

Thm [FGKLIST (ICALP'24)]. The Metric Dimension problem on bounded diameter graphs

– admits $2^{2^{\mathcal{O}(\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$–time algo, but

– does not admit $2^{2^{o(\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

First NP–Complete problems to admit such lower bounds

Earlier known results were Σ_2^p–complete or Π_2^p–complete.

How far can we push this result?

Thm [FGKLIST (ICALP'24)]. The Metric Dimension problem on bounded diameter graphs

– admits $2^{2^{\mathcal{O}(\text{tw})}} \cdot n^{\mathcal{O}(1)}$–time algo, but

– does not admit $2^{2^{o(\text{tw})}} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

First NP-Complete problems to admit such lower bounds

Earlier known results were Σ_2^p-complete or Π_2^p-complete.

How far can we push this result?

Holds for feedback vertex set, pathwidth, and treedepth.

Thm [FGKLIST (ICALP'24)]. The Metric Dimension problem on bounded diameter graphs

– admits $2^{2^{\mathcal{O}(\text{tw})}} \cdot n^{\mathcal{O}(1)}$–time algo, but

– does not admit $2^{2^{o(\text{tw})}} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

First NP–Complete problems to admit such lower bounds

Earlier known results were Σ_2^p–complete or Π_2^p–complete.

How far can we push this result?

Holds for feedback vertex set, pathwidth, and treedepth.

How about vertex cover?

Our results

Our results

Thm. The Metric Dimension and Geodetic Set problems

– admit $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo, but

– do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Our results

Thm. The Metric Dimension and Geodetic Set problems
– admit $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo, but
– do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Thm. The Metric Dimension and Geodetic Set problems
– admit kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices, but
– do not admit a kernel with $2^{o(\mathsf{vc})}$ vertices unless the ETH fails.

Our results

Thm. The Metric Dimension and Geodetic Set problems
- admit $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo, but
- do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

(Split Contraction [ALSZ, STACS 2017])

Thm. The Metric Dimension and Geodetic Set problems
- admit kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices, but
- do not admit a kernel with $2^{o(\mathsf{vc})}$ vertices unless the ETH fails.

Our results

Thm. The Metric Dimension and Geodetic Set problems
– admit $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo, but
– do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

(Split Contraction [ALSZ, STACS 2017])

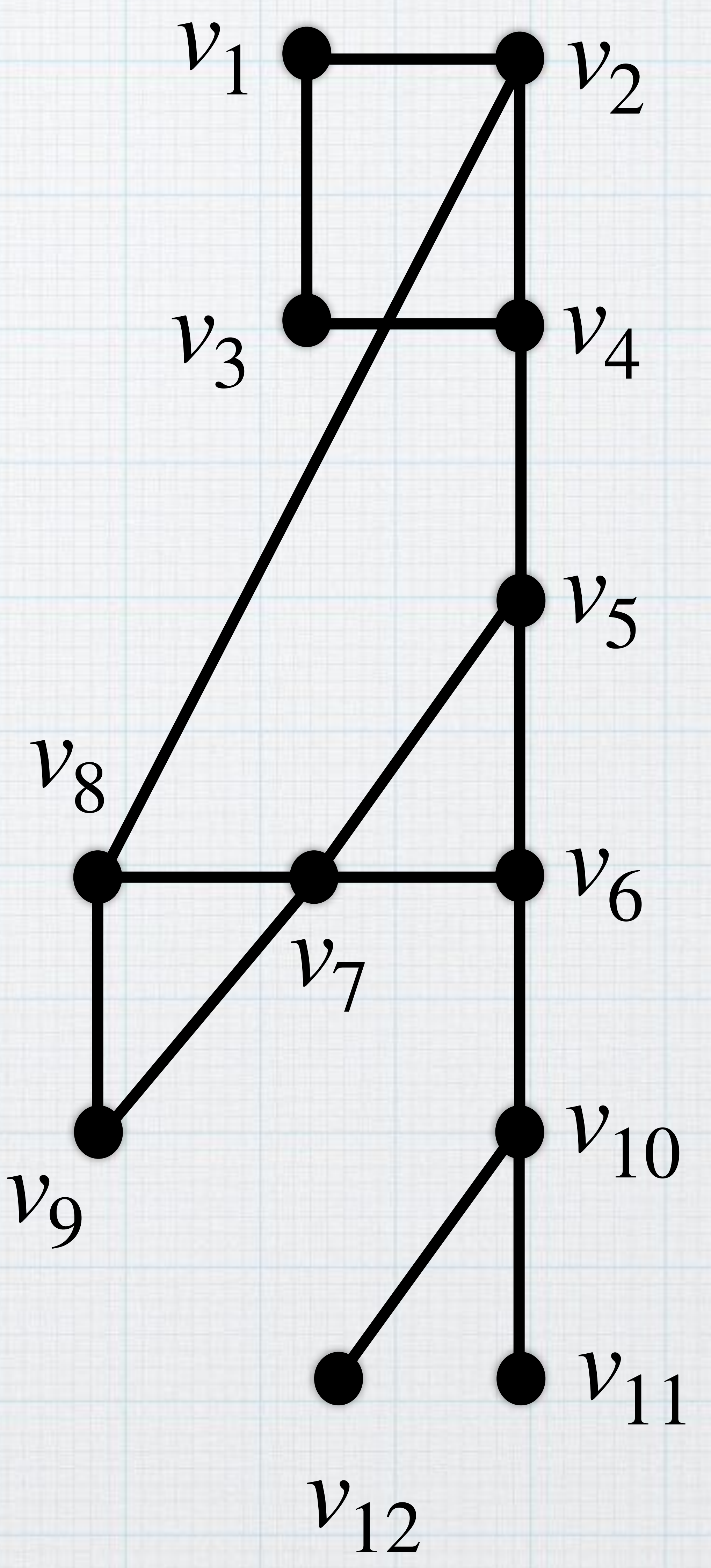
Thm. The Metric Dimension and Geodetic Set problems
– admit kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices, but
– do not admit a kernel with $2^{o(\mathsf{vc})}$ vertices unless the ETH fails.

(Edge Clique Cover [CPP, SODA'13], Biclique Cover [CIK, IPEC'16], Strong Met–Dim [FGKLIST, ICALP'24])

Geodetic Set

Input: Graph G, int k

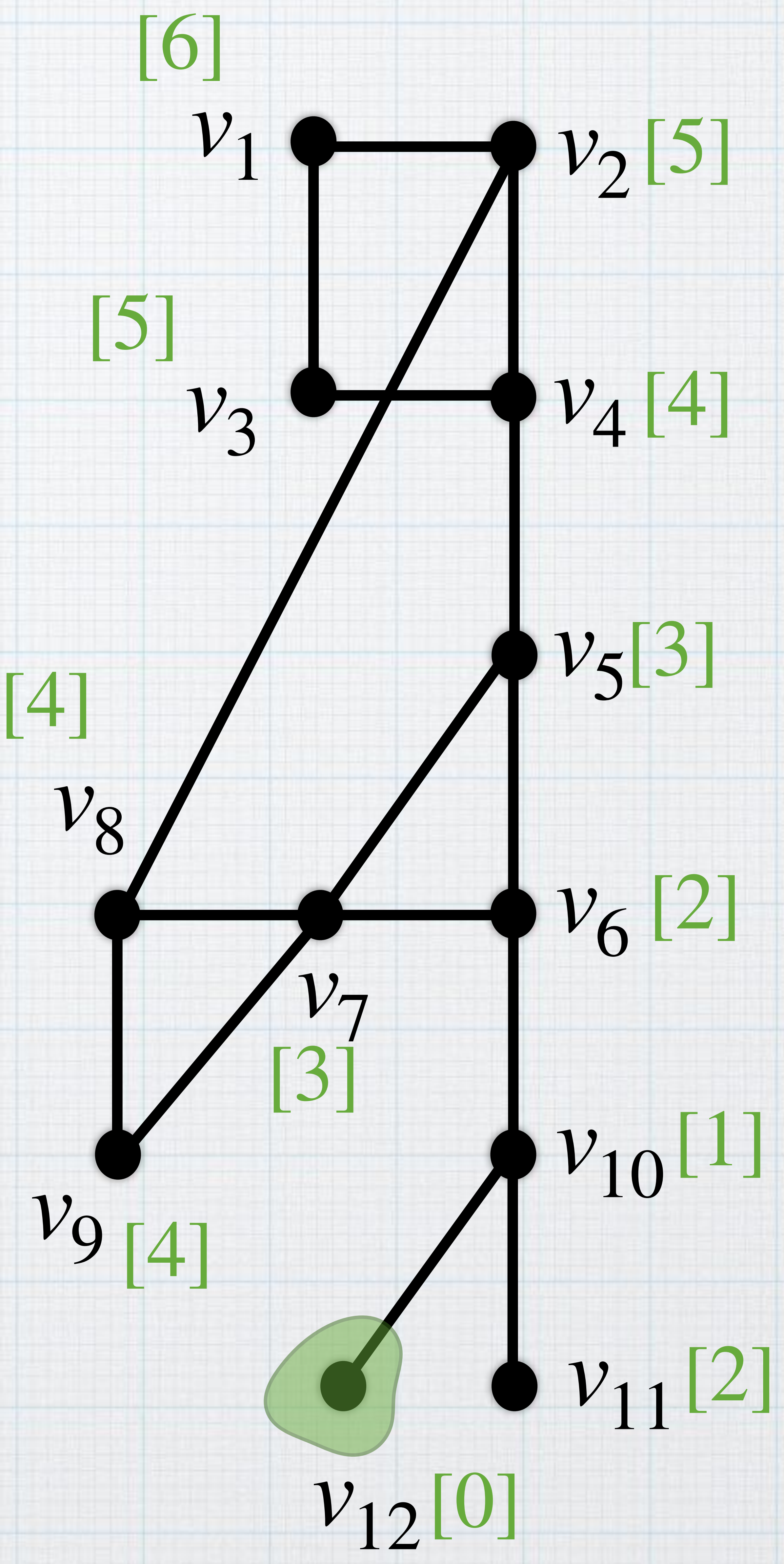
Output: $\exists ?$ a subset S of k vertices s.t. for any vertex u is covered by a shortest path between two vertices in S



Geodetic Set

Input: Graph G, int k

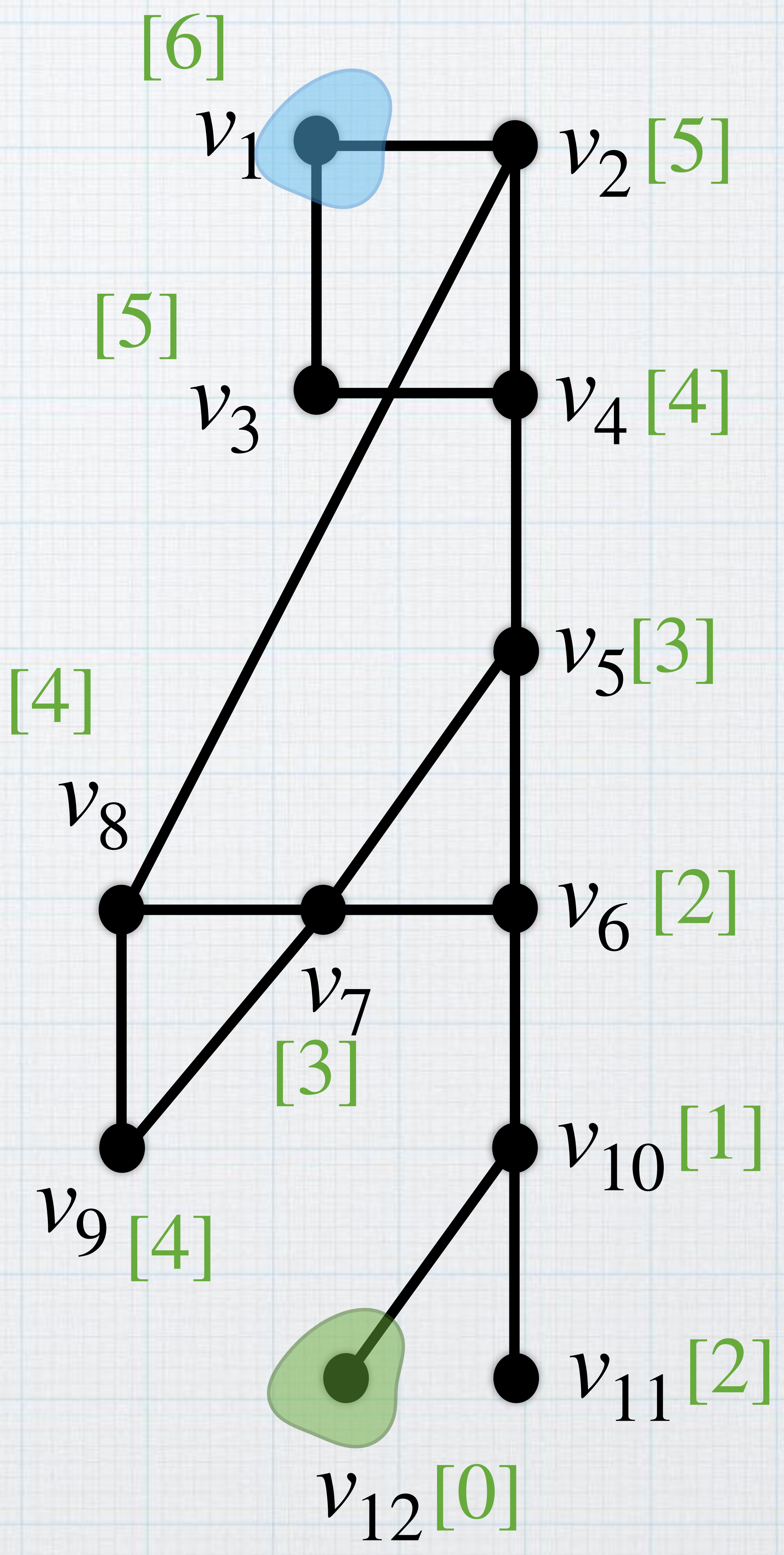
Output: $\exists ?$ a subset S of k vertices s.t. for any vertex u is covered by a shortest path between two vertices in S



Geodetic Set

Input: Graph G, int k

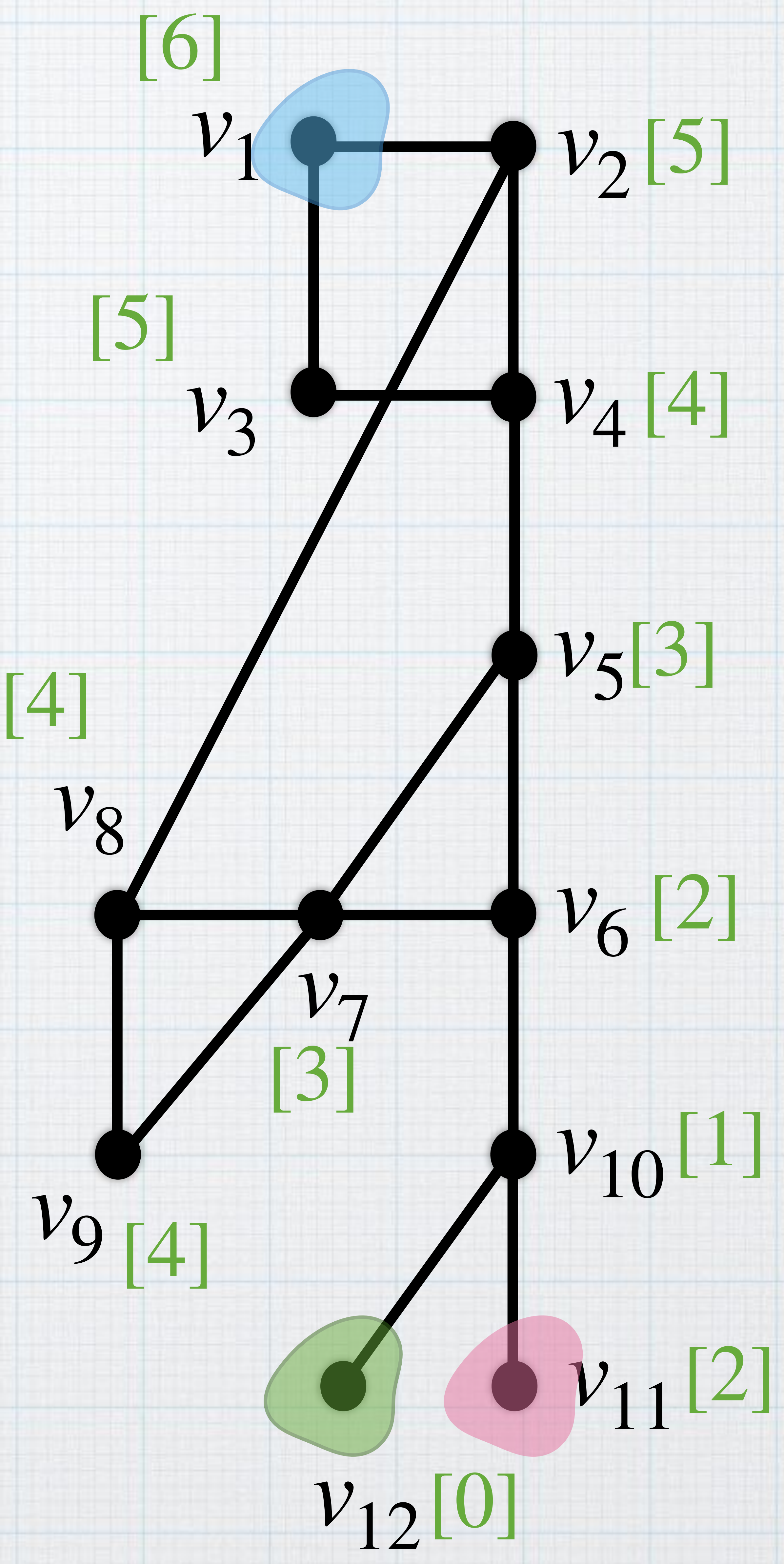
Output: $\exists?$ a subset S of k vertices s.t. for any vertex u is covered by a shortest path between two vertices in S



Geodetic Set

Input: Graph G, int k

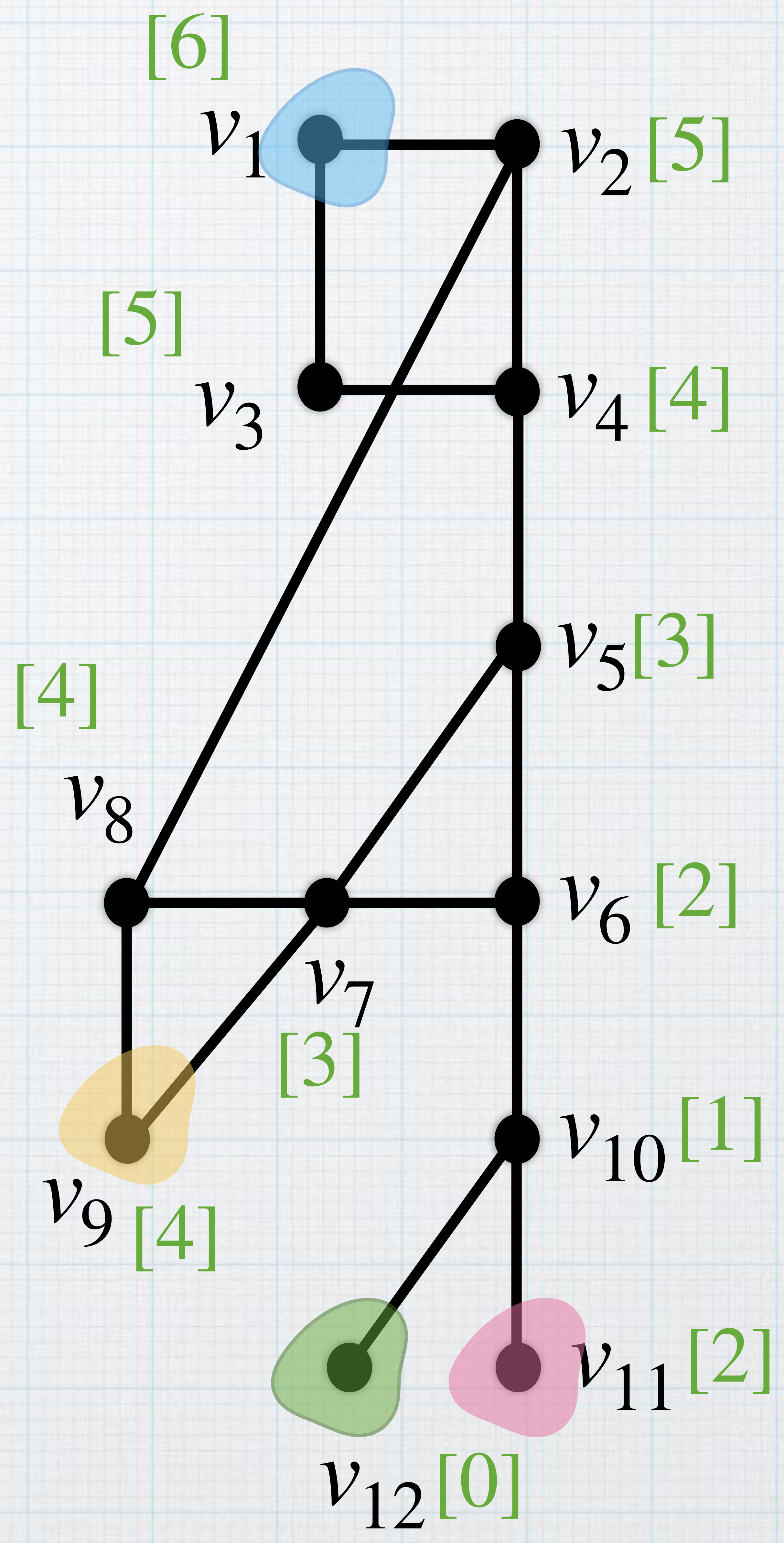
Output: $\exists ?$ a subset S of k vertices s.t. for any vertex u is covered by a shortest path between two vertices in S



Geodetic Set

Input: Graph G, int k

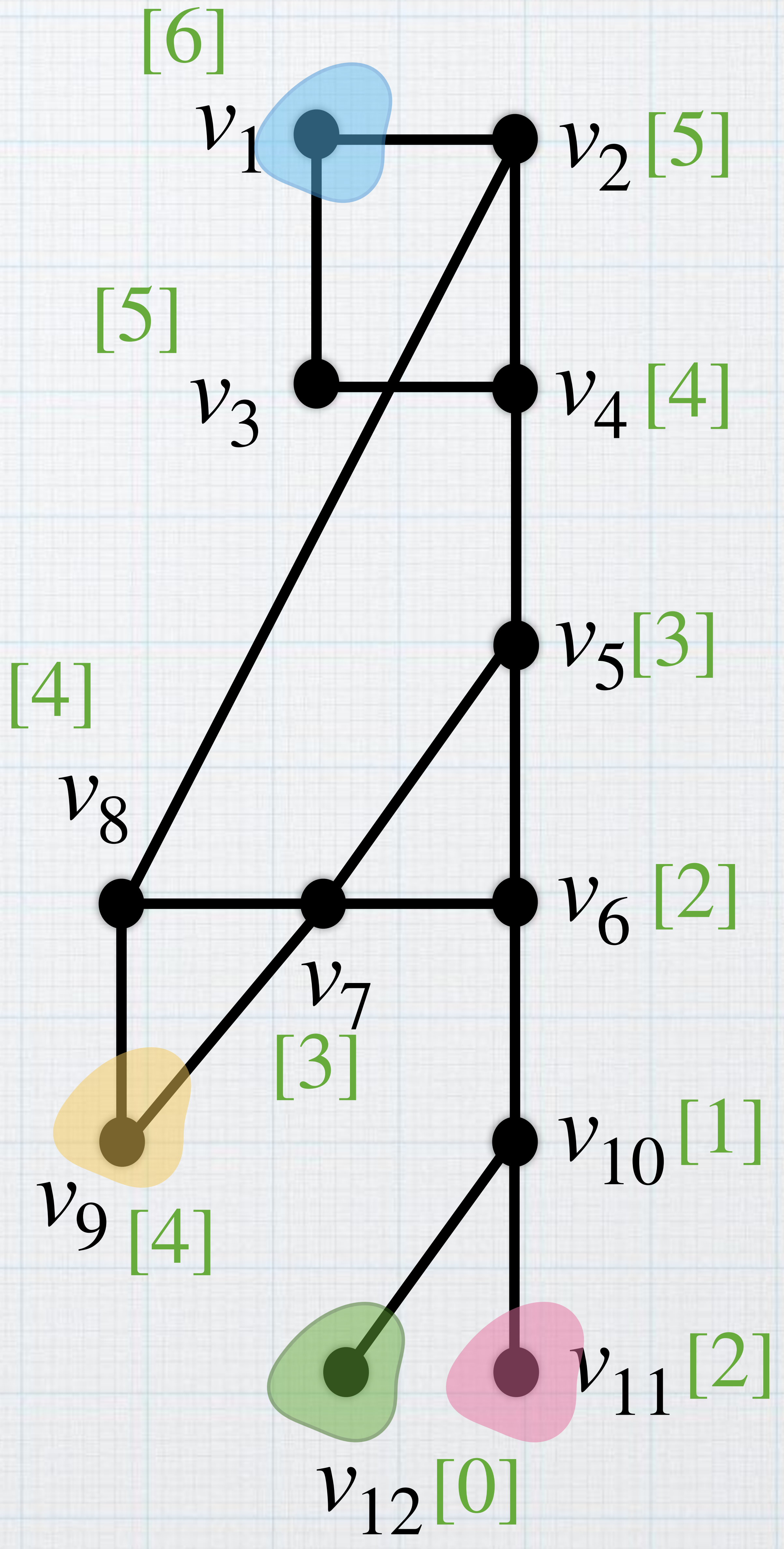
Output: $\exists ?$ a subset S of k vertices s.t. for any vertex u is covered by a shortest path between two vertices in S



Geodetic Set

Input: Graph G, int k

Output: $\exists ?$ a subset S of k vertices s.t. for any vertex u is covered by a shortest path between two vertices in S

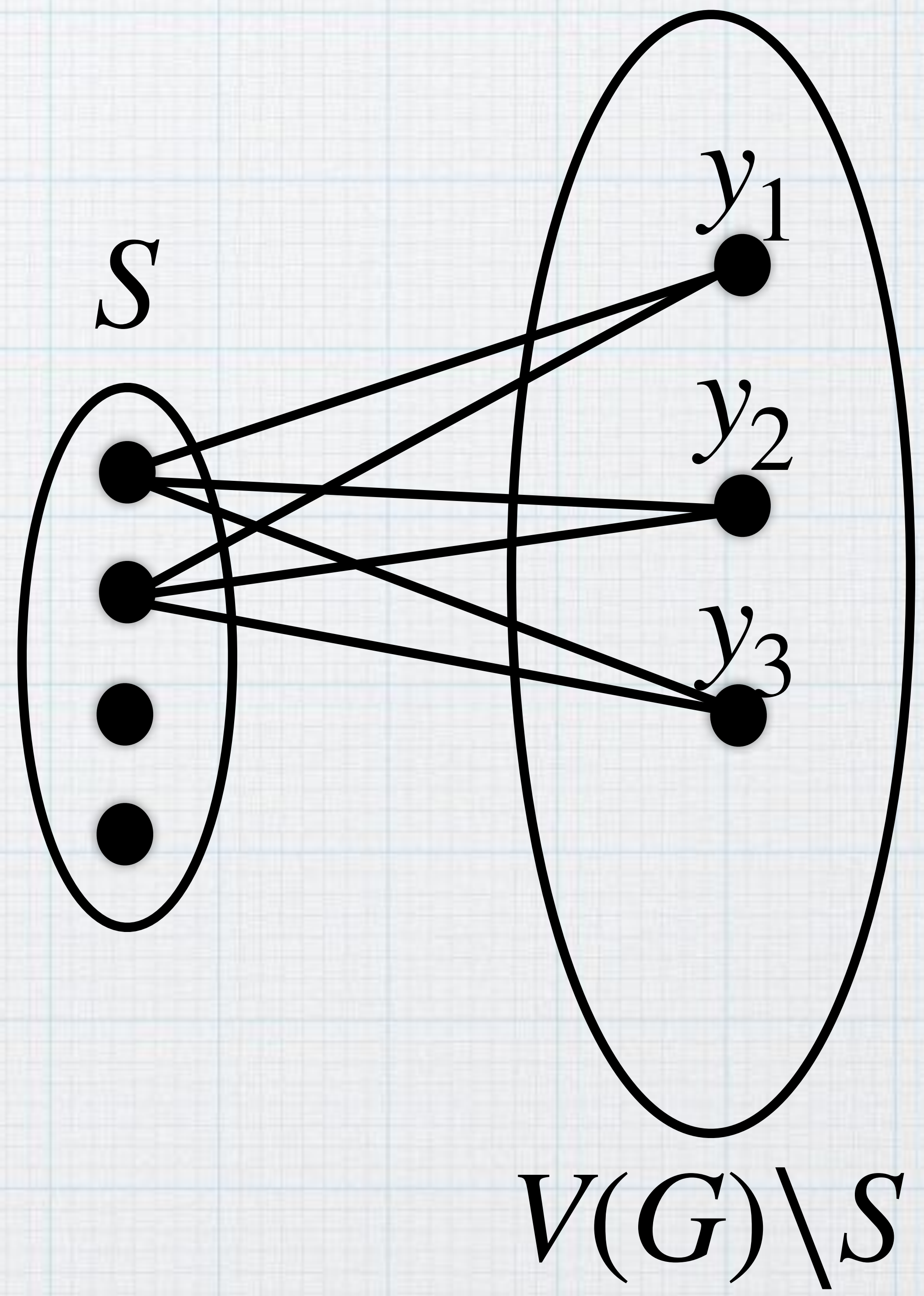


Solution at one part of the graph, plays important role in other part of the graph.

Ex. v_{12} and v_1 covers a lot of vertices.

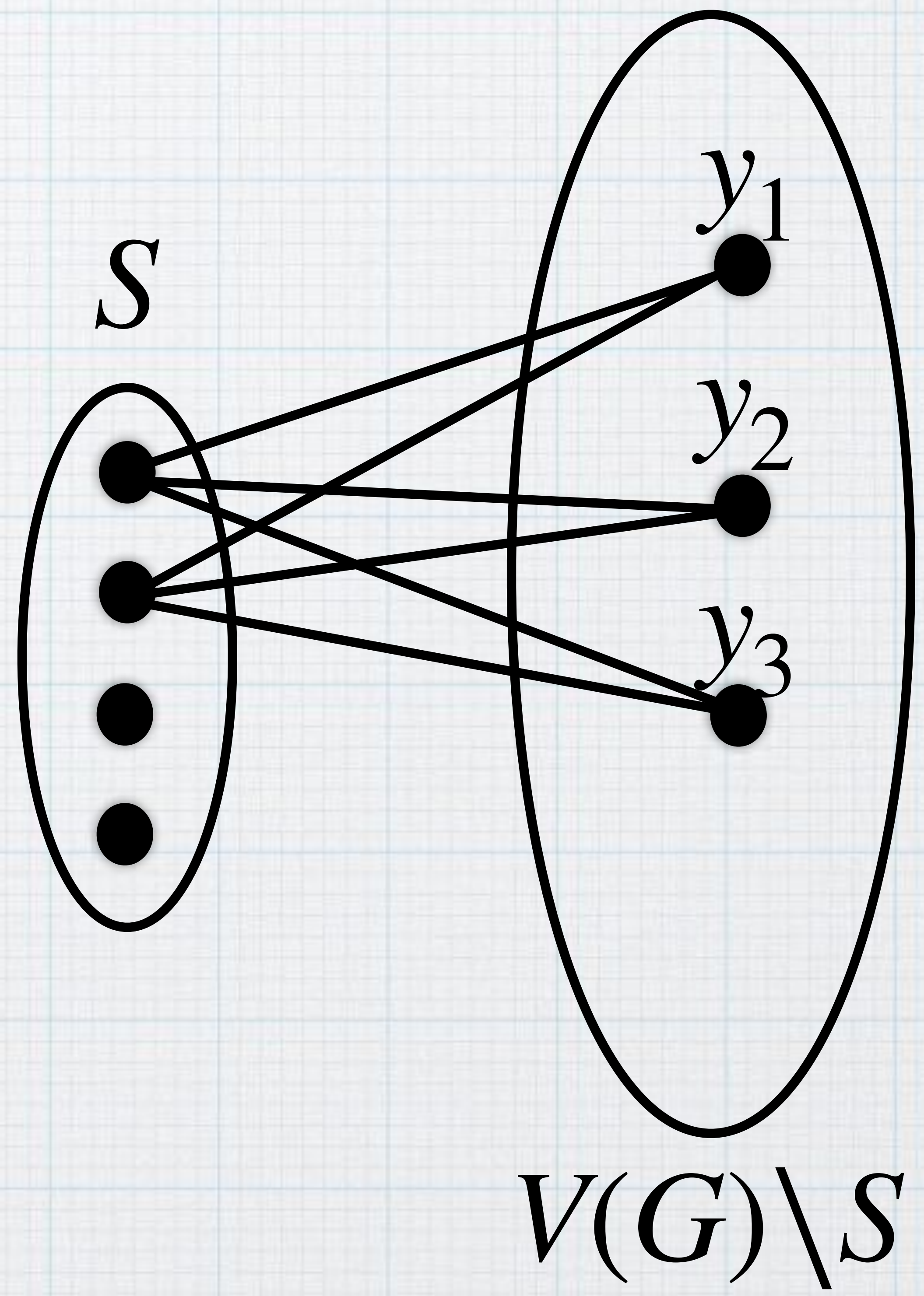
Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices and $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo.

Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices and $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo.



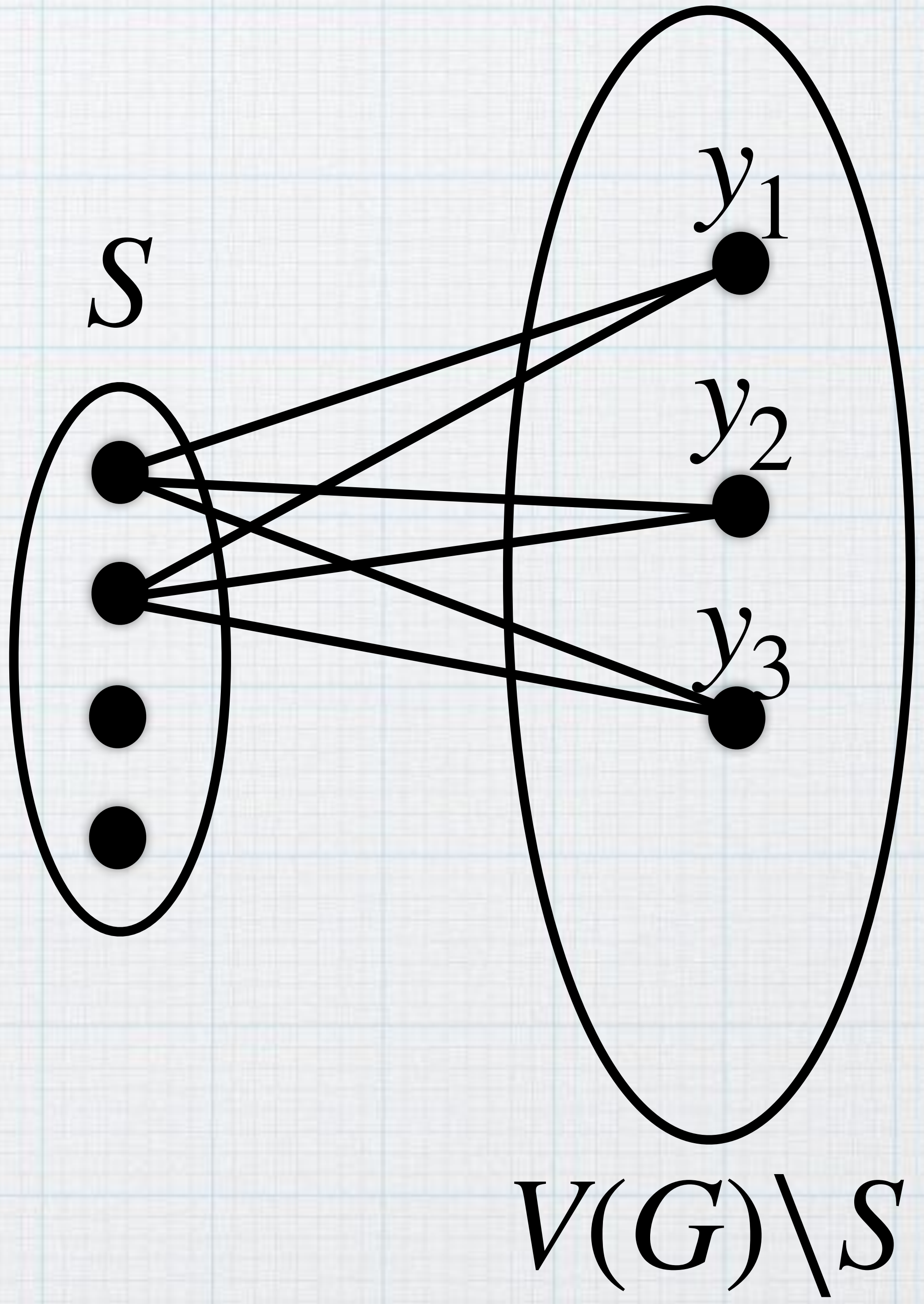
Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices and $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo.

Obs: If $N(y_1) = N(y_2) = N(y_3)$ for y_1, y_2, y_3 in ind–set, then any resolving set contains at least two of them.



Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\mathrm{vc})}$ vertices and $2^{(\mathrm{vc})^2} \cdot n^{\mathcal{O}(1)}$-time algo.

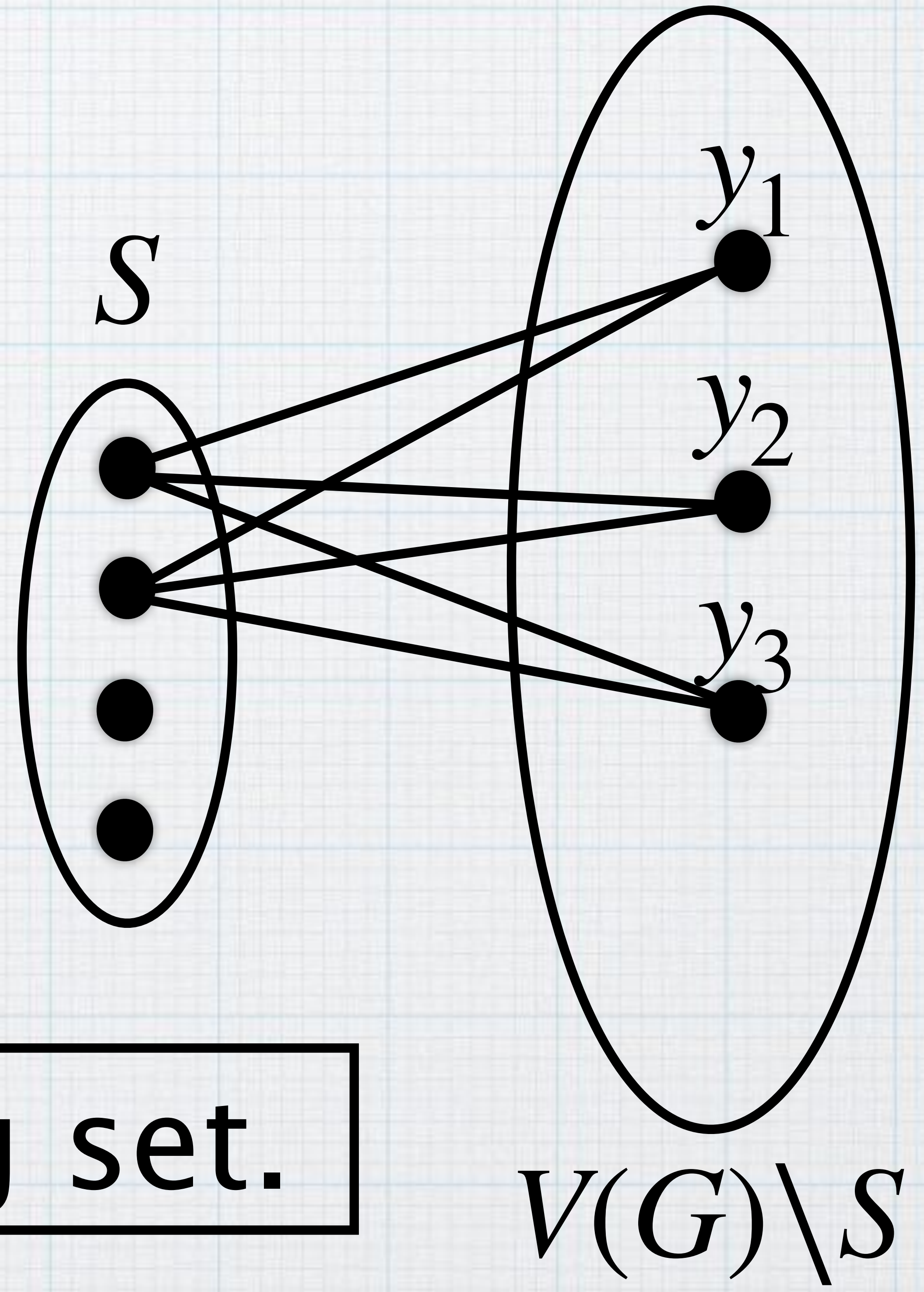
Obs: If $N(y_1) = N(y_2) = N(y_3)$ for y_1, y_2, y_3 in ind-set, then any resolving set contains at least two of them.



$\Rightarrow$ kernel with $2^{\mathcal{O}(\mathrm{vc})}$ vertices

Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices and $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo.

Obs: If $N(y_1) = N(y_2) = N(y_3)$ for y_1, y_2, y_3 in ind–set, then any resolving set contains at least two of them.

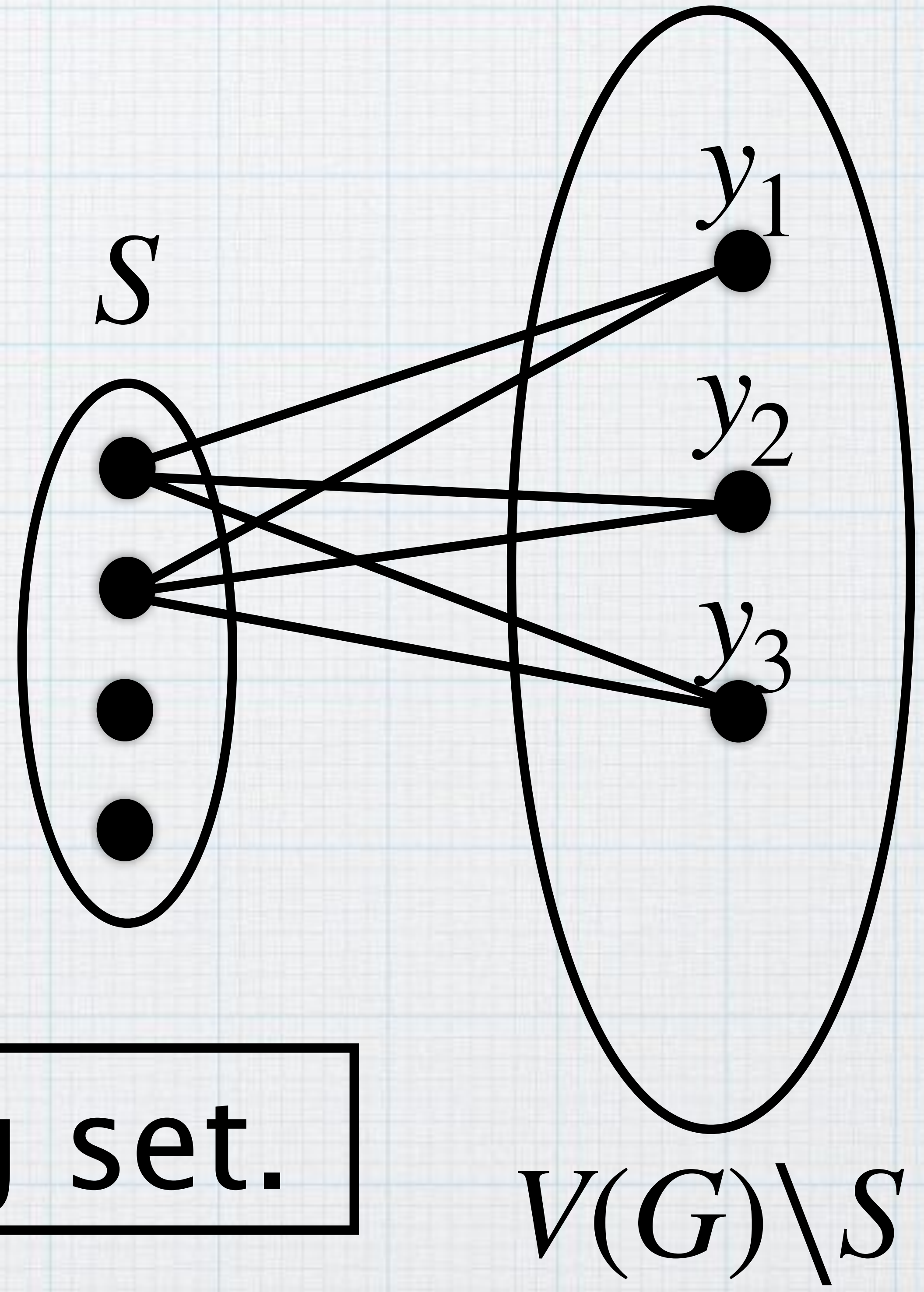


$\Rightarrow$ kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices

Obs: Forced vertices + vertex cover S is a resolving set.

Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices and $2^{(\mathsf{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo.

Obs: If $N(y_1) = N(y_2) = N(y_3)$ for y_1, y_2, y_3 in ind–set, then any resolving set contains at least two of them.



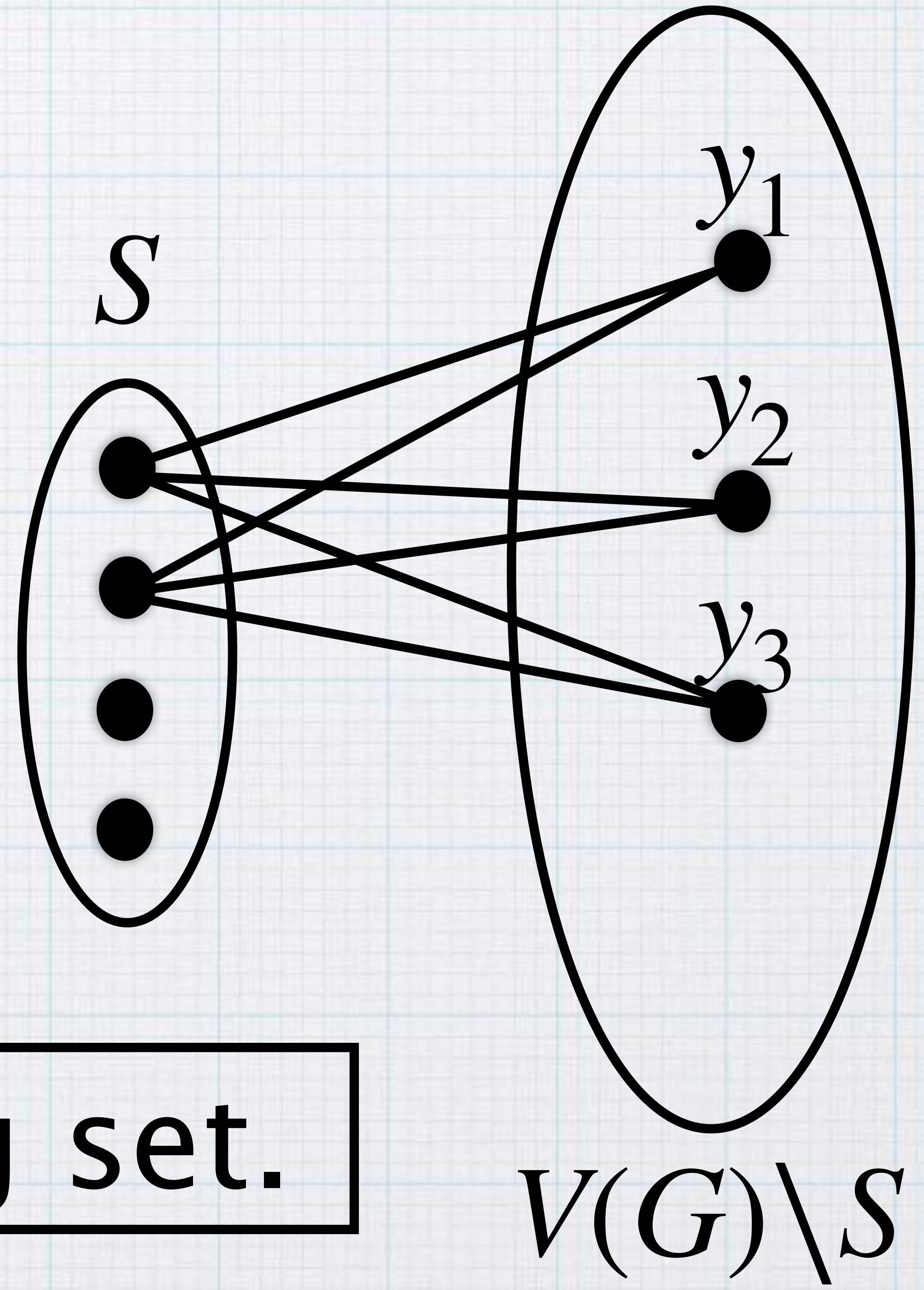
$\Rightarrow$ kernel with $2^{\mathcal{O}(\mathsf{vc})}$ vertices

Obs: Forced vertices + vertex cover S is a resolving set.

$\Rightarrow$ `free vertices' is a resolving set is at most vc

Thm. Metric Dimension admits kernel with $2^{\mathcal{O}(\text{vc})}$ vertices and $2^{(\text{vc})^2} \cdot n^{\mathcal{O}(1)}$–time algo.

Obs: If $N(y_1) = N(y_2) = N(y_3)$ for y_1, y_2, y_3 in ind–set, then any resolving set contains at least two of them.



$\Rightarrow$ kernel with $2^{\mathcal{O}(\text{vc})}$ vertices

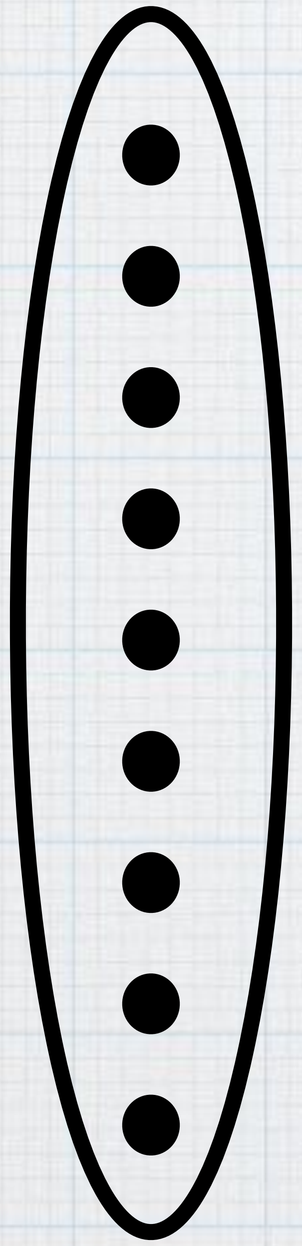
Obs: Forced vertices + vertex cover S is a resolving set.

$\Rightarrow$ `free vertices' is a resolving set is at most vc

$\Rightarrow$ algorithm with desired running time

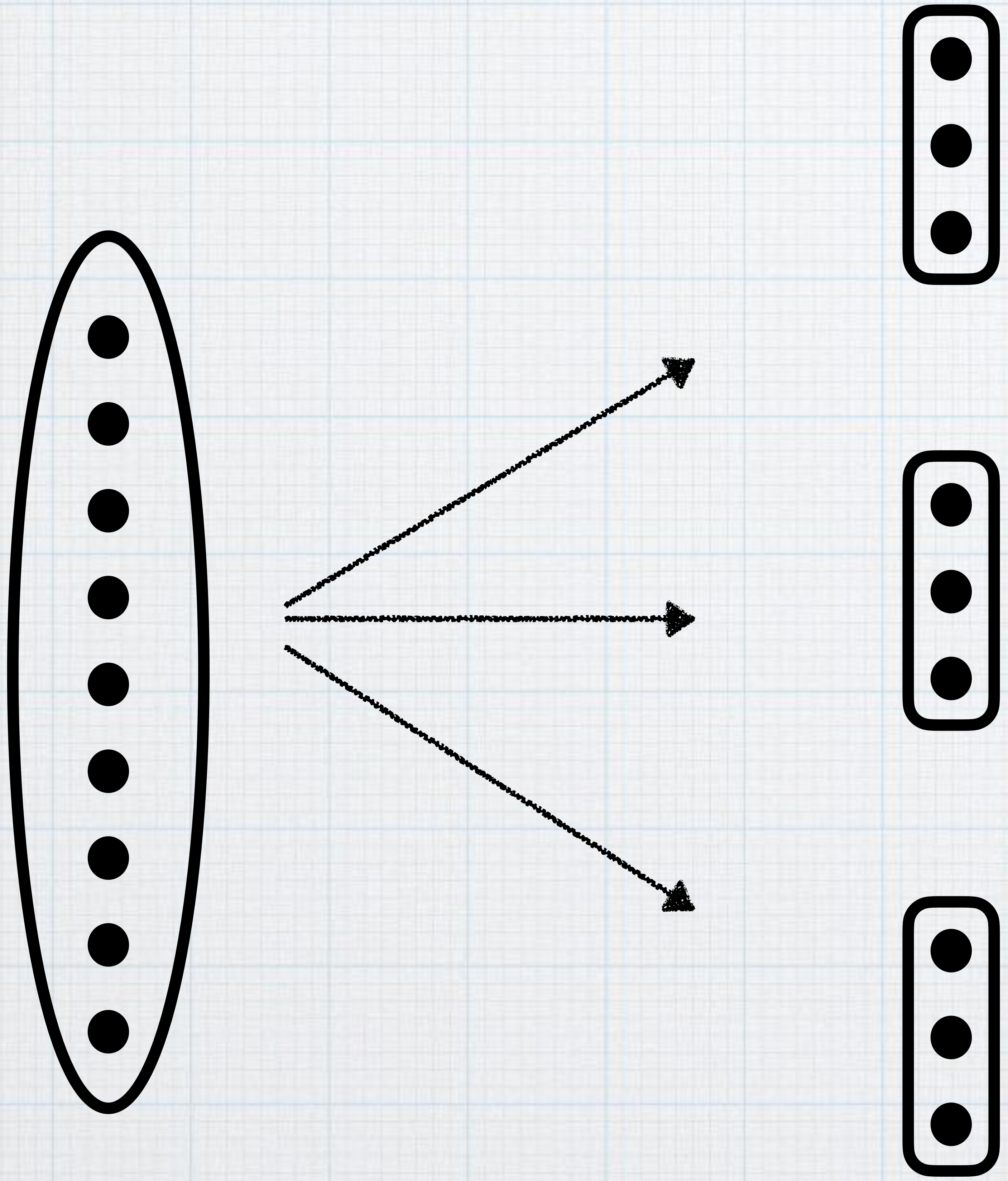
3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$



n var

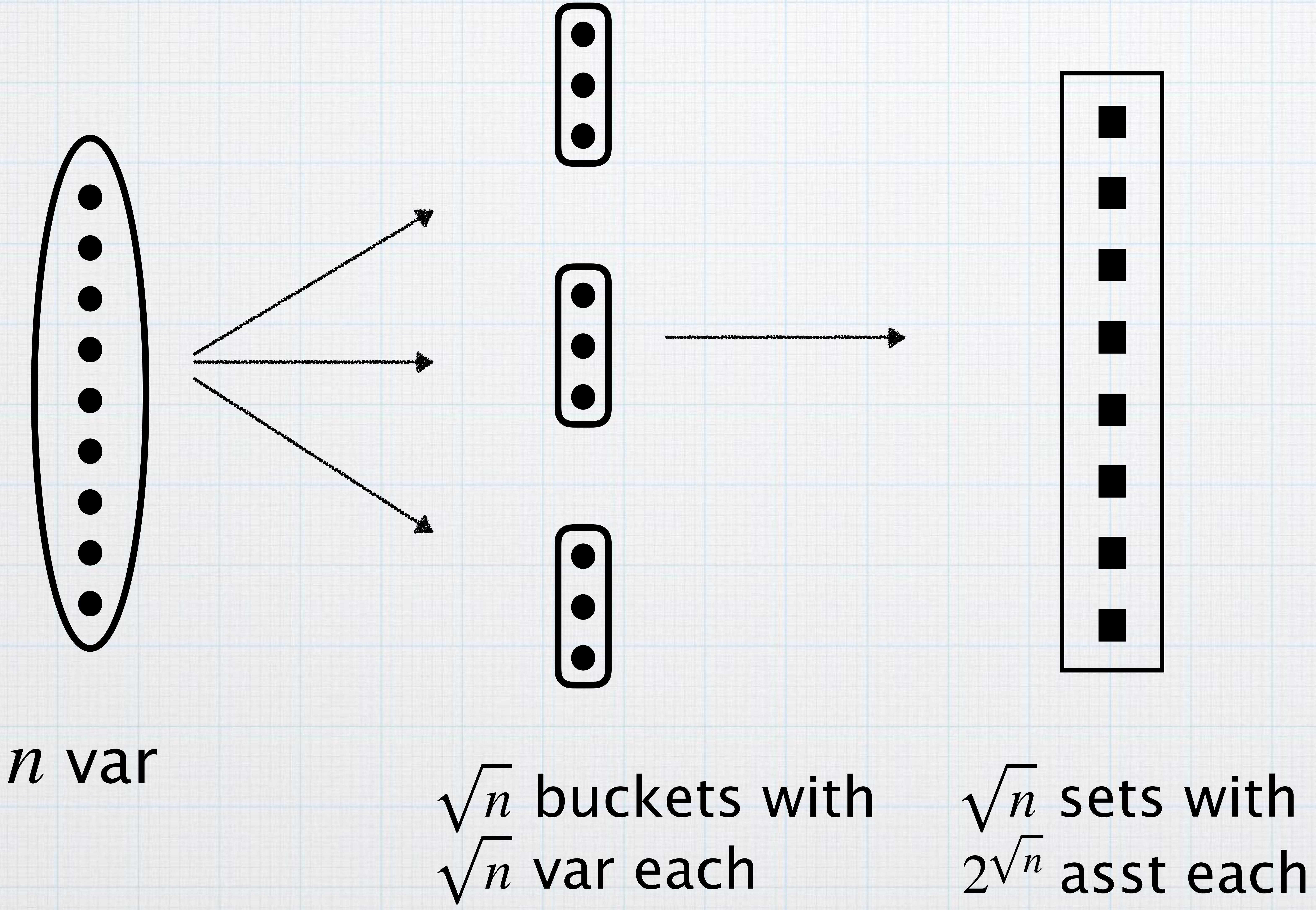
3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$



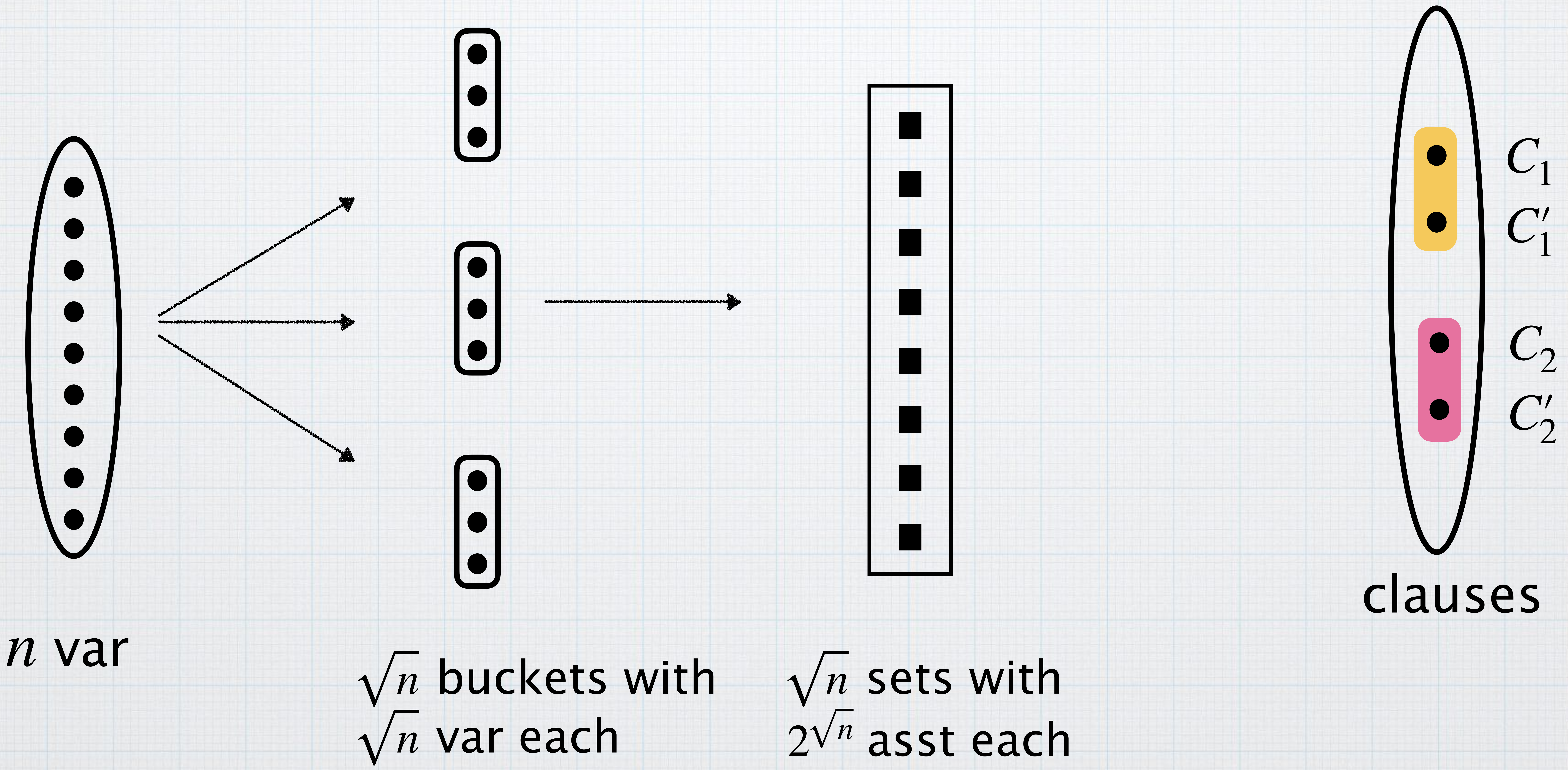
n var

$\sqrt{n}$ buckets with
$\sqrt{n}$ var each

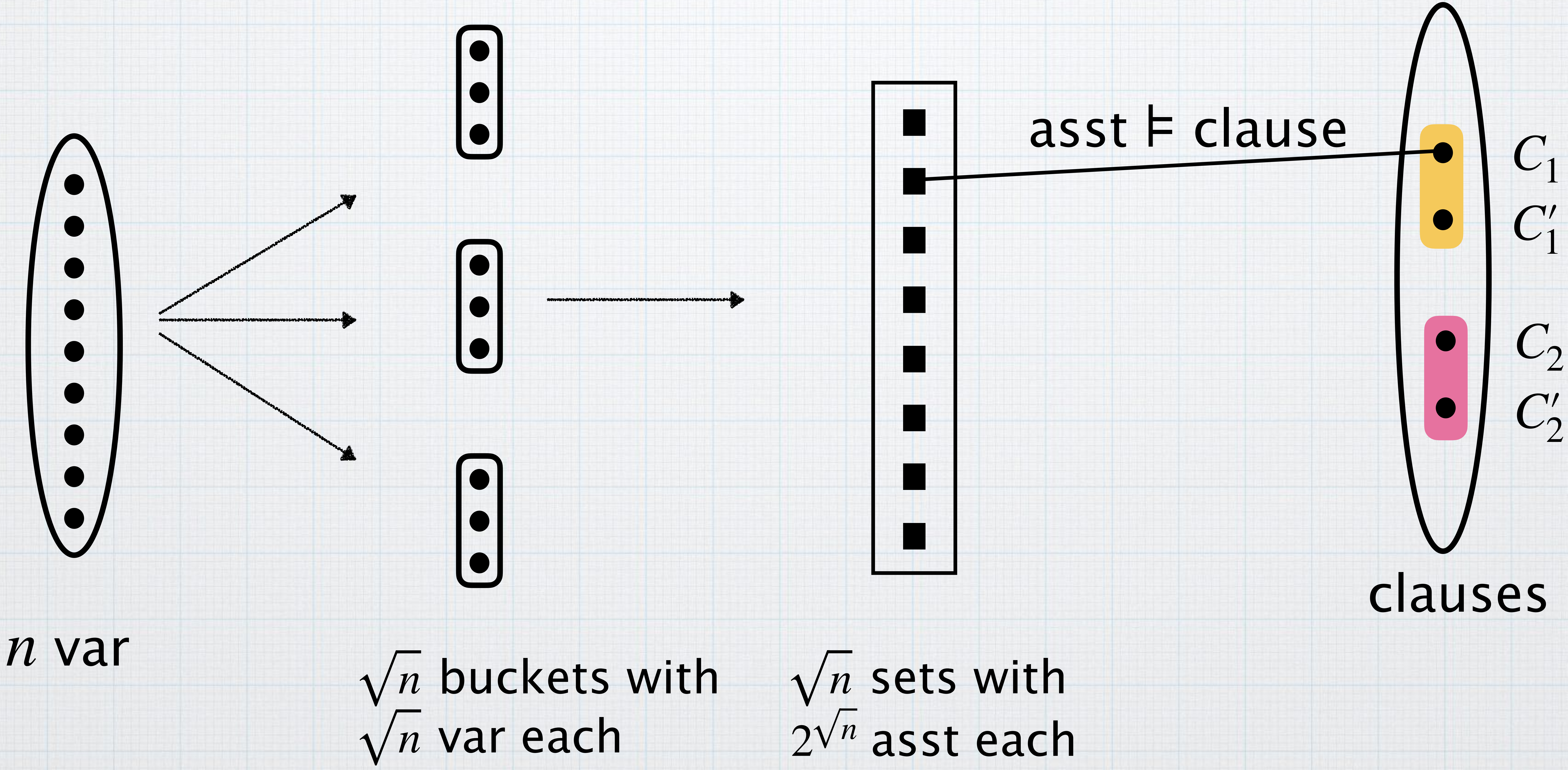
3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$



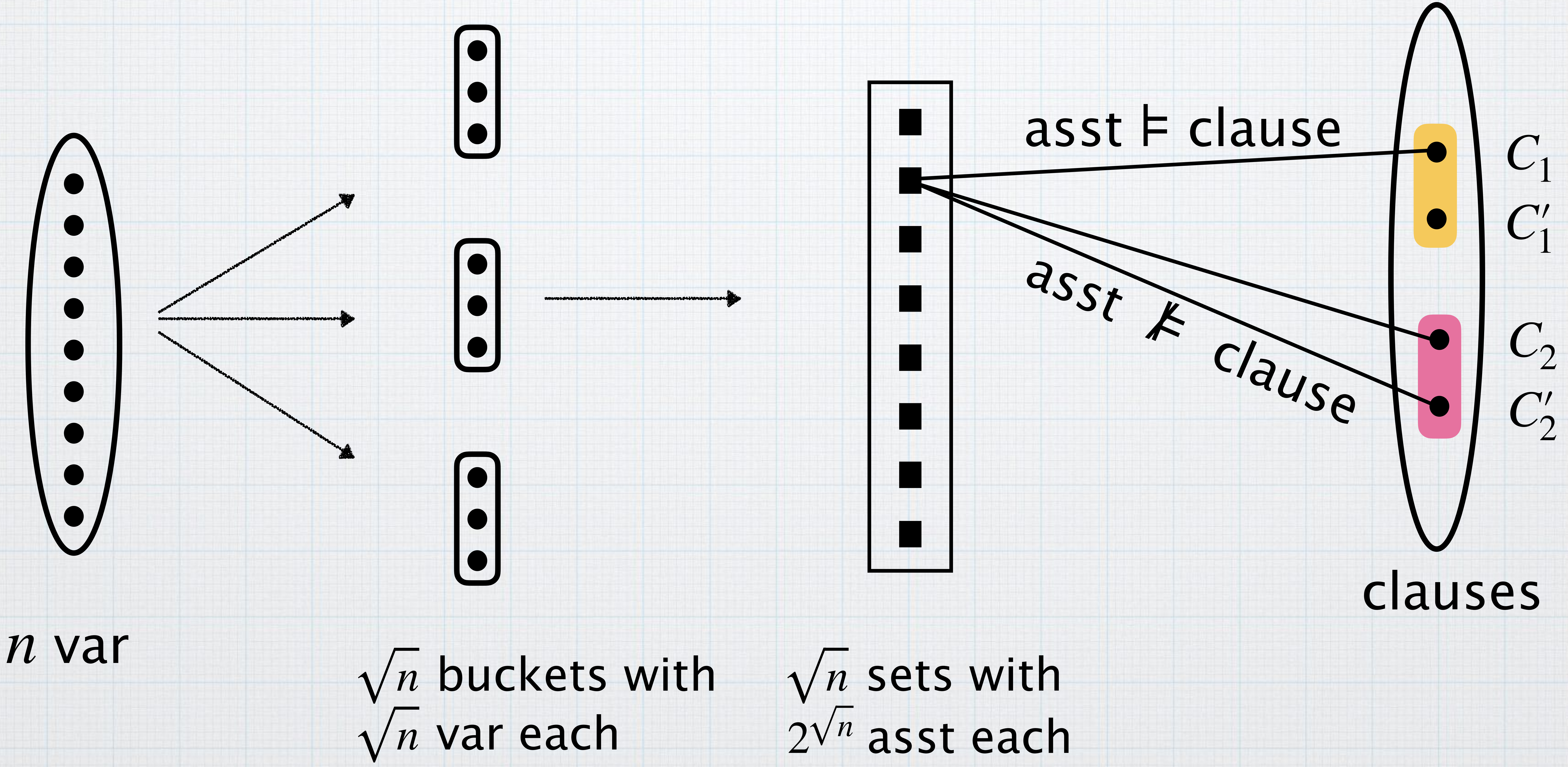
3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$



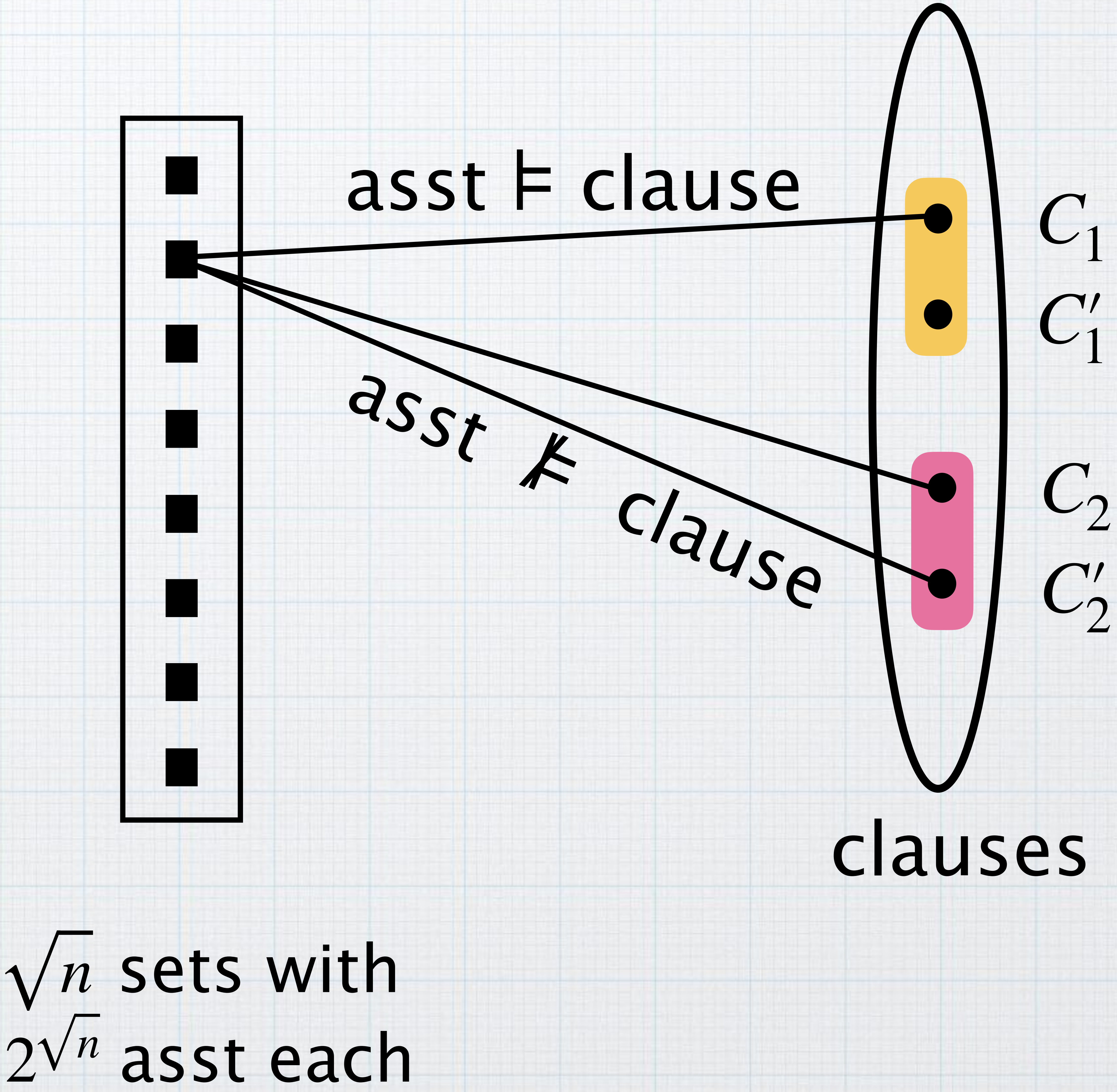
3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$



3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

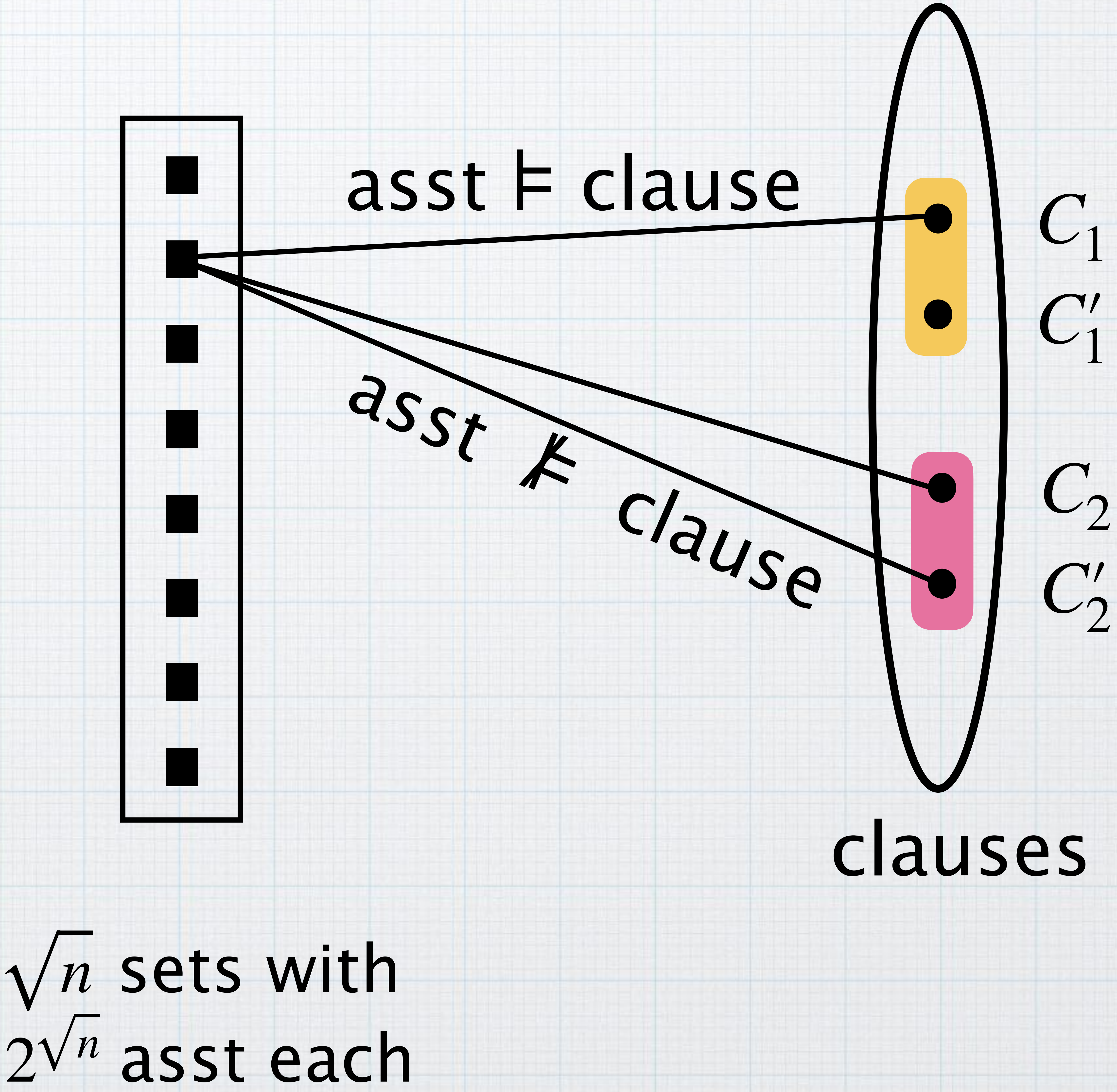


3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$



3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Q: How to identify vertices corresponding to assignments?

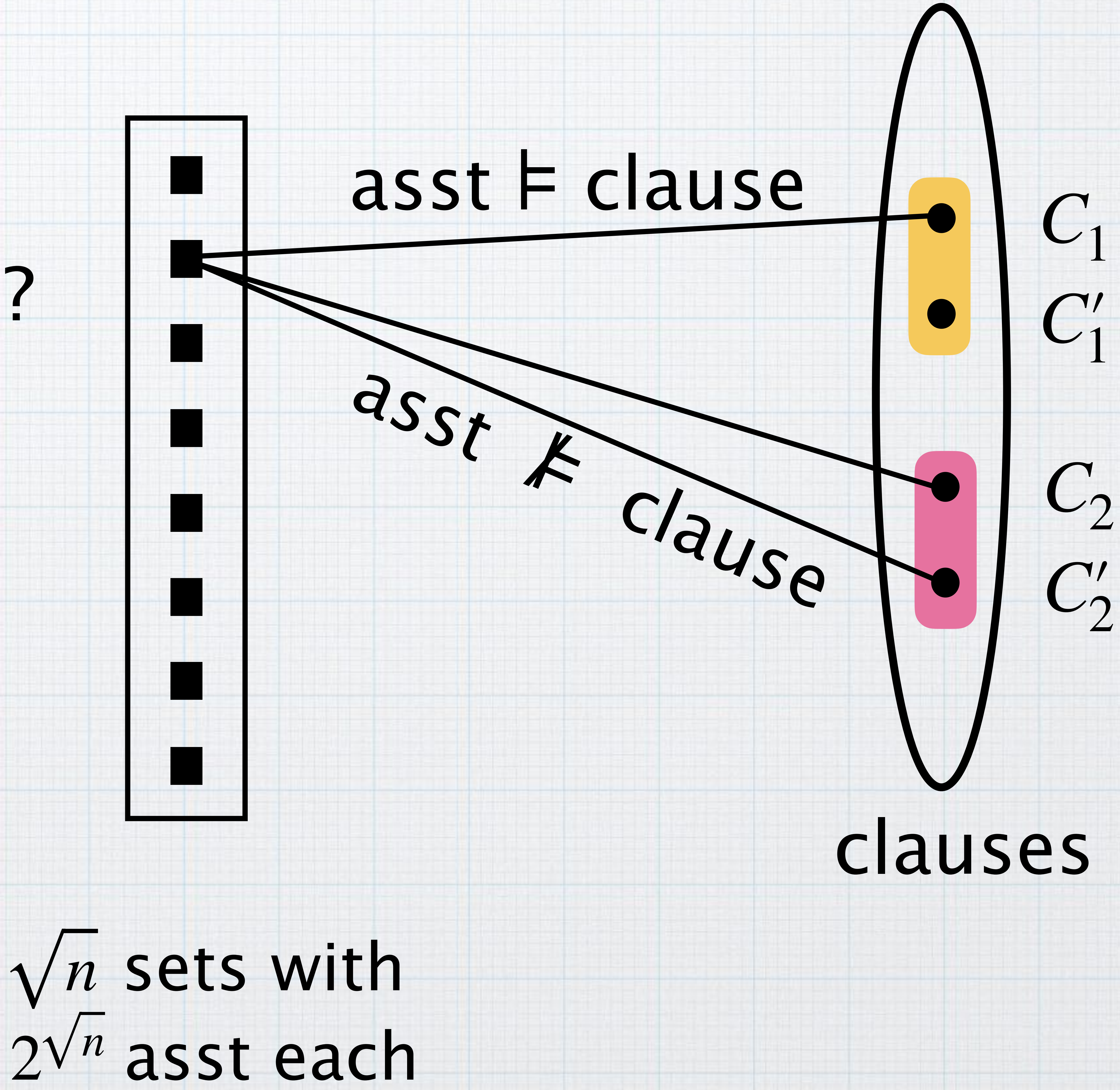


3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Q: How to identify vertices corresponding to assignments?

Q: How to resolve following pairs?

$\{C_1, C_2\}$ $\{C'_1, C_2\}$ ...

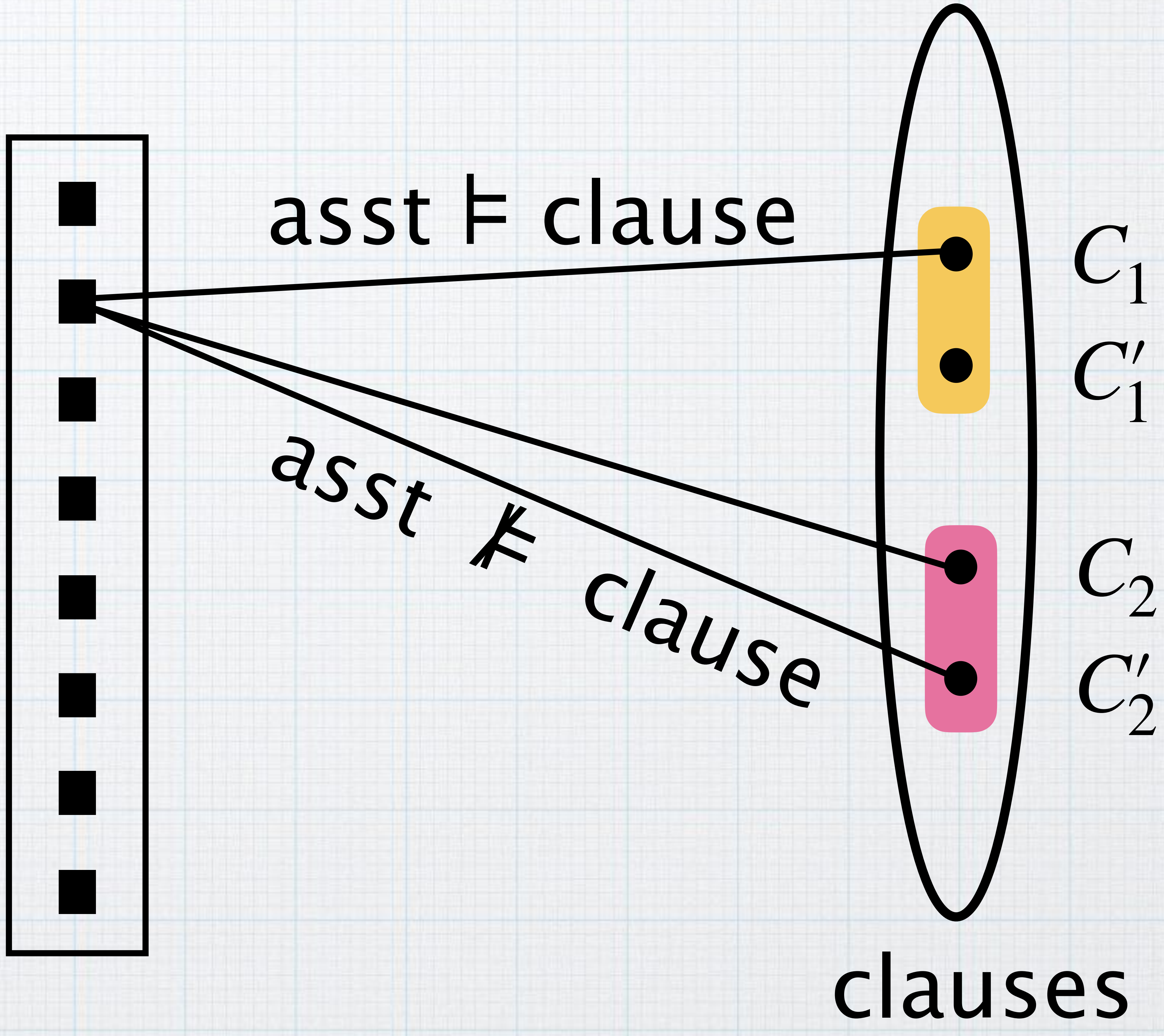


3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Q: How to identify vertices corresponding to assignments?

Q: How to resolve following pairs?
$\{C_1, C_2\}$ $\{C_1', C_2\}$ ...

Q: How to control the vertex cover? i.e. limit the interactions



$\sqrt{n}$ sets with
$2^{\sqrt{n}}$ asst each

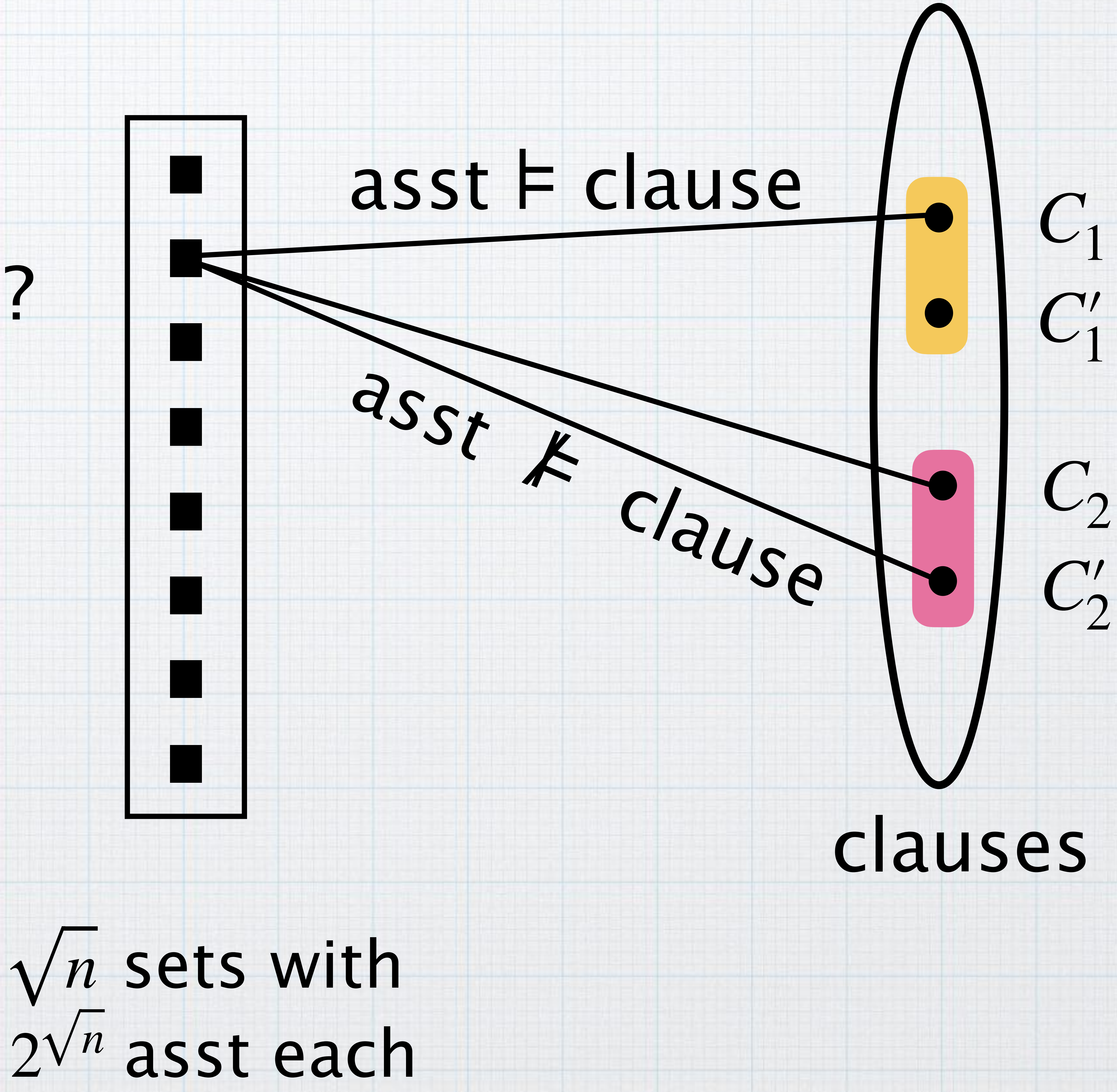
3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Q: How to identify vertices corresponding to assignments?

Q: How to resolve following pairs?
$\{C_1, C_2\}$ $\{C_1', C_2\}$ …

Using bit encoding which cost log(set-size) extra vertices

Q: How to control the vertex cover? i.e. limit the interactions



3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

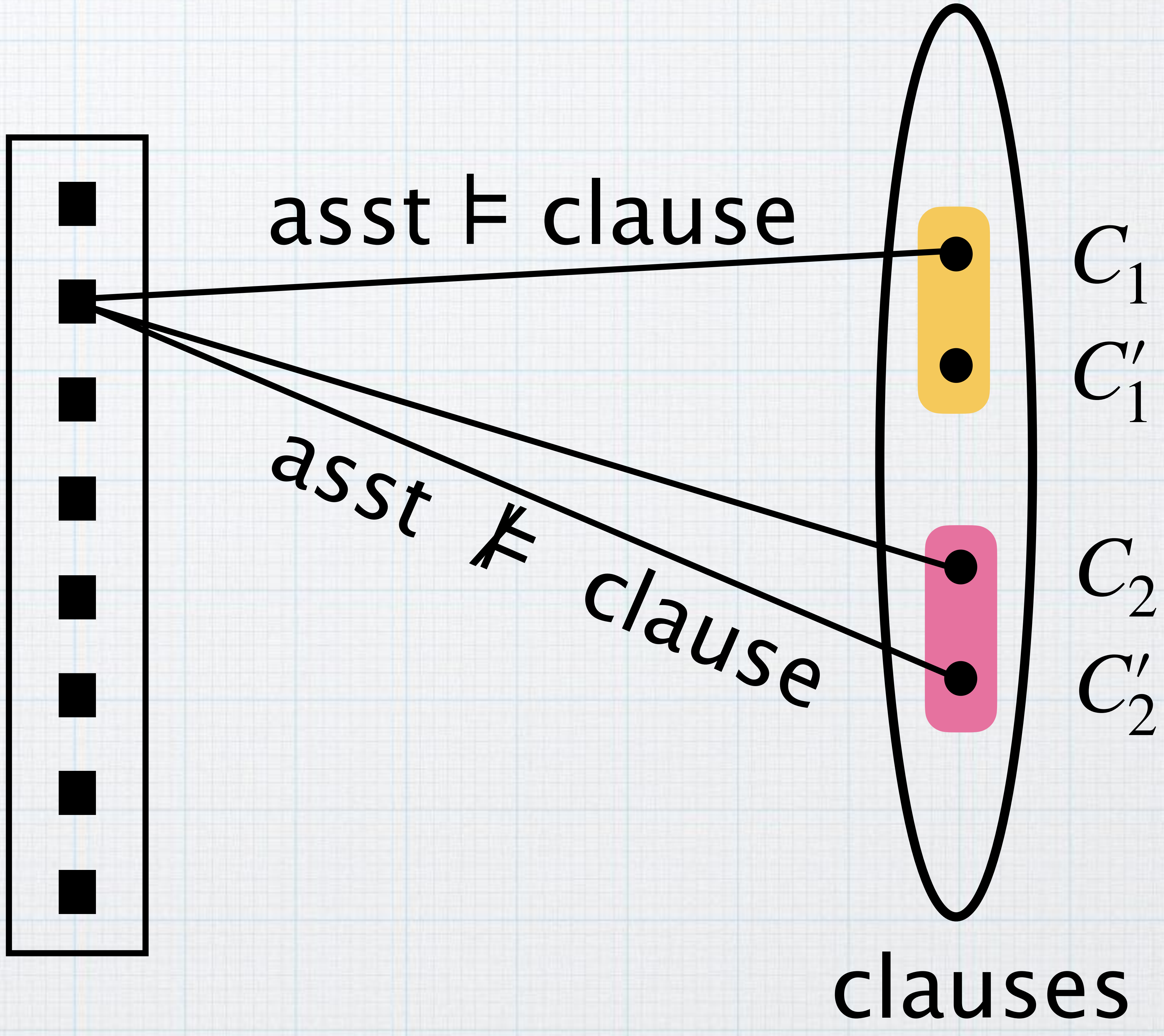
Q: How to identify vertices corresponding to assignments?

Q: How to resolve following pairs?
$\{C_1, C_2\}$ $\{C_1', C_2\}$ ...

Using bit encoding which cost log(set-size) extra vertices

Q: How to control the vertex cover? i.e. limit the interactions

Sperner Family



$\sqrt{n}$ sets with
$2^{\sqrt{n}}$ asst each

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Sperner Family: Collection $\mathcal{F}$ of subsets of a universe such that for any two sets A_1, A_2 in $\mathcal{F}$ neither $A_1 \subseteq A_2$ nor $A_2 \subseteq A_1$.

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Sperner Family: Collection $\mathcal{F}$ of subsets of a universe such that for any two sets A_1, A_2 in $\mathcal{F}$ neither $A_1 \subseteq A_2$ nor $A_2 \subseteq A_1$.

i.e. $A_1 \cap \bar{A}_2 \neq \emptyset$ for any A_2 in $\mathcal{F} \setminus \{A_1\}$

3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Sperner Family: Collection $\mathcal{F}$ of subsets of a universe such that for any two sets A_1, A_2 in $\mathcal{F}$ neither $A_1 \subseteq A_2$ nor $A_2 \subseteq A_1$.

i.e. $A_1 \cap \bar{A}_2 \neq \emptyset$ for any A_2 in $\mathcal{F} \setminus \{A_1\}$

Ex: Universe: $\{1,2,\ldots,2p\}$ for an integer p and
$\mathcal{F}$ — each set contains exactly p elements.

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Sperner Family: Collection $\mathcal{F}$ of subsets of a universe such that for any two sets A_1, A_2 in $\mathcal{F}$ neither $A_1 \subseteq A_2$ nor $A_2 \subseteq A_1$.

i.e. $A_1 \cap \bar{A}_2 \neq \emptyset$ for any A_2 in $\mathcal{F} \setminus \{A_1\}$

Ex: Universe: $\{1,2,\ldots,2p\}$ for an integer p and
$\mathcal{F}$ — each set contains exactly p elements.

For $p = 3$, $A_1 = \{1,3,5\}$, $A_2 = \{1,3,6\}$ are sets in $\mathcal{F}$

3–SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Sperner Family: Collection $\mathcal{F}$ of subsets of a universe such that for any two sets A_1, A_2 in $\mathcal{F}$ neither $A_1 \subseteq A_2$ nor $A_2 \subseteq A_1$.

i.e. $A_1 \cap \bar{A}_2 \neq \emptyset$ for any A_2 in $\mathcal{F} \setminus \{A_1\}$

Ex: Universe: $\{1,2,\ldots,2p\}$ for an integer p and
$\mathcal{F}$ — each set contains exactly p elements.

For $p = 3$, $A_1 = \{1,3,5\}$, $A_2 = \{1,3,6\}$ are sets in $\mathcal{F}$

For universe of size $\mathcal{O}(p)$, we get $\mathcal{F}$ of size $2^{\mathcal{O}(p)}$

3–SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Sperner Family: Collection $\mathcal{F}$ of subsets of a universe such that for any two sets A_1, A_2 in $\mathcal{F}$ neither $A_1 \subseteq A_2$ nor $A_2 \subseteq A_1$.

i.e. $A_1 \cap \bar{A}_2 \neq \emptyset$ for any A_2 in $\mathcal{F} \setminus \{A_1\}$

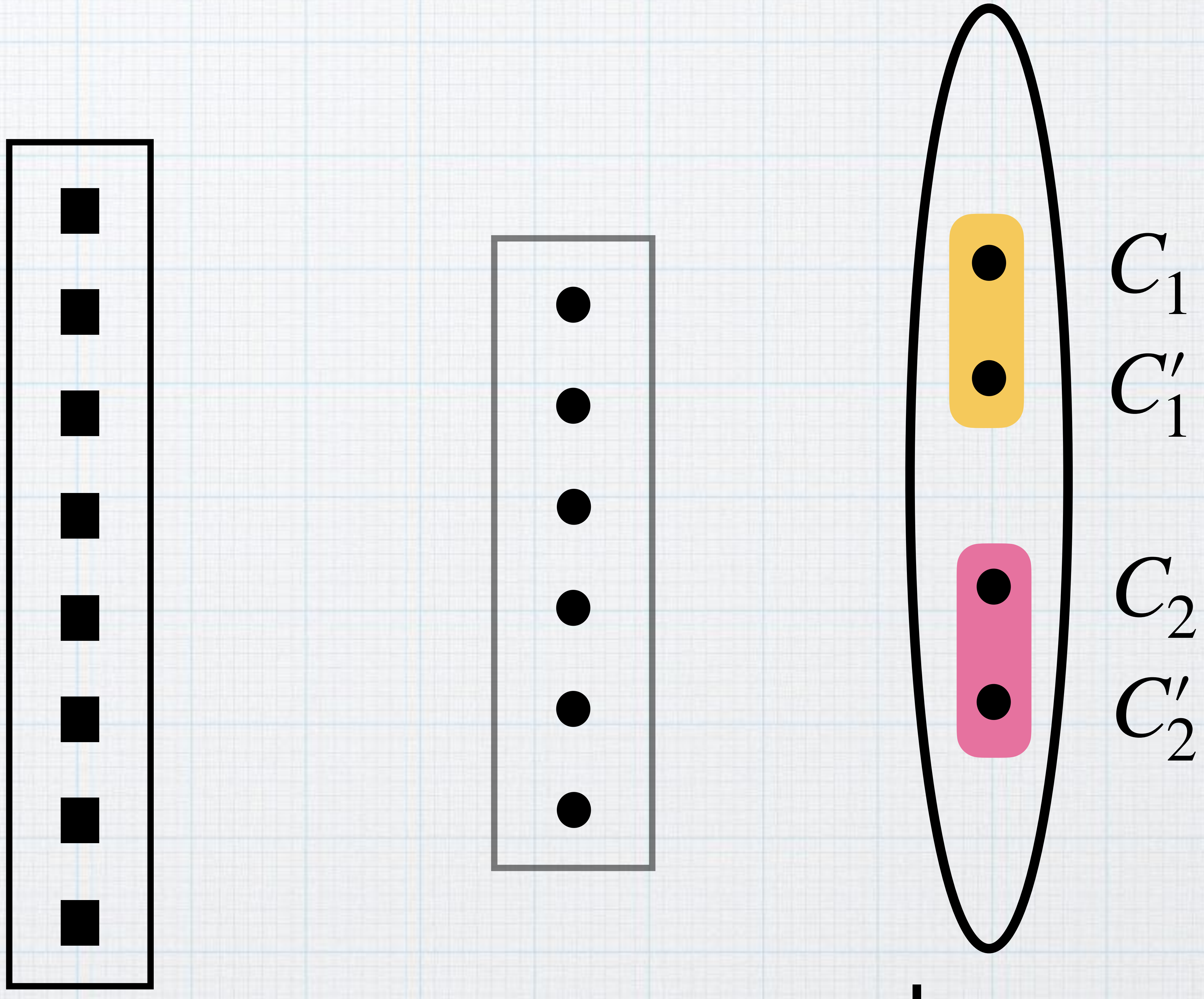
Ex: Universe: $\{1,2,\ldots,2p\}$ for an integer p and
$\mathcal{F}$ — each set contains exactly p elements.

For $p = 3$, $A_1 = \{1,3,5\}$, $A_2 = \{1,3,6\}$ are sets in $\mathcal{F}$

For universe of size $\mathcal{O}(p)$, we get $\mathcal{F}$ of size $2^{\mathcal{O}(p)}$

For $p = \mathcal{O}(\sqrt{n})$, $\mathcal{F}$ is of size $2^{\mathcal{O}(\sqrt{n})}$, i.e. unique set for each asst.

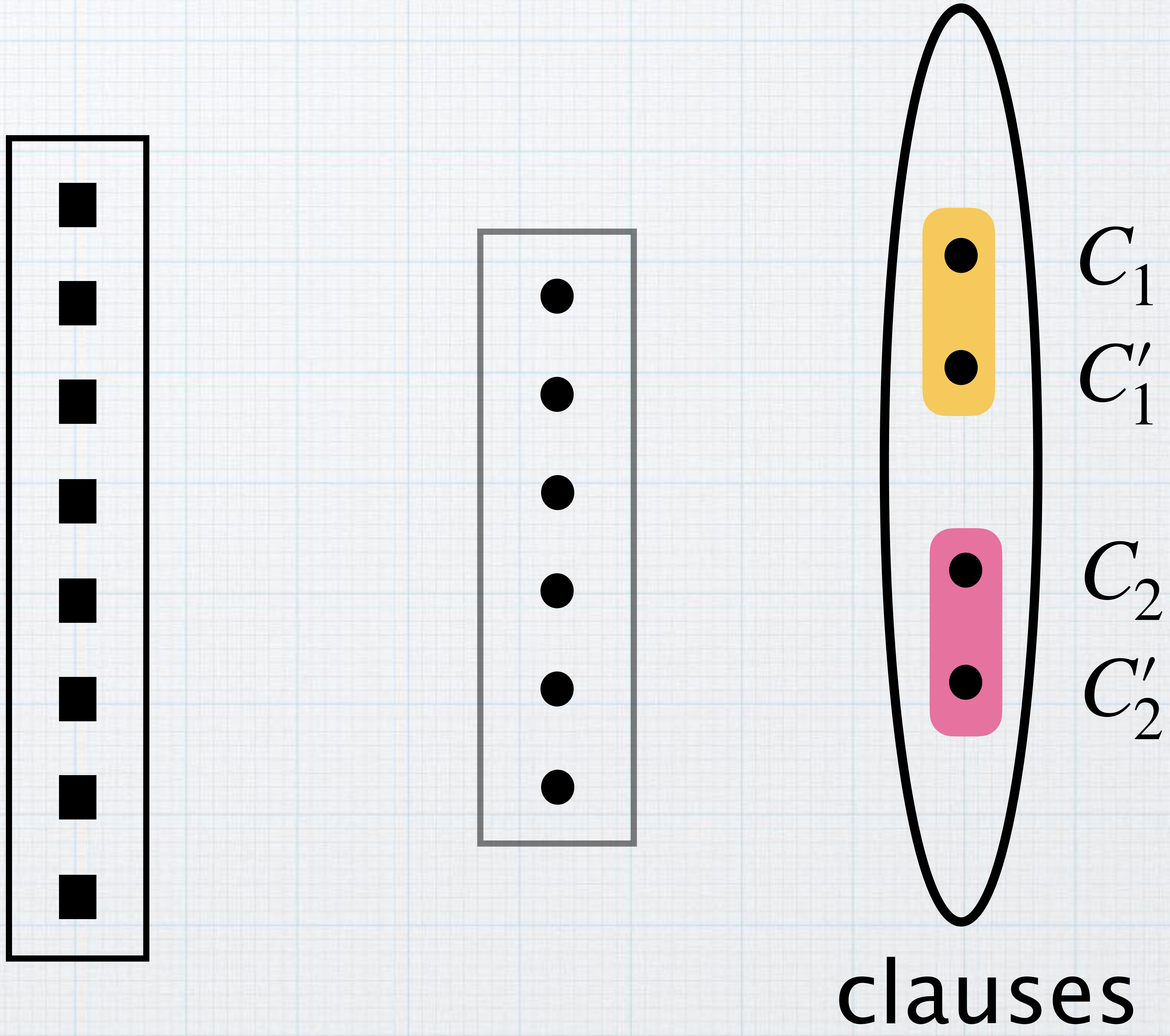
3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$



$\sqrt{n}$ sets with
$2^{\sqrt{n}}$ asst each

3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

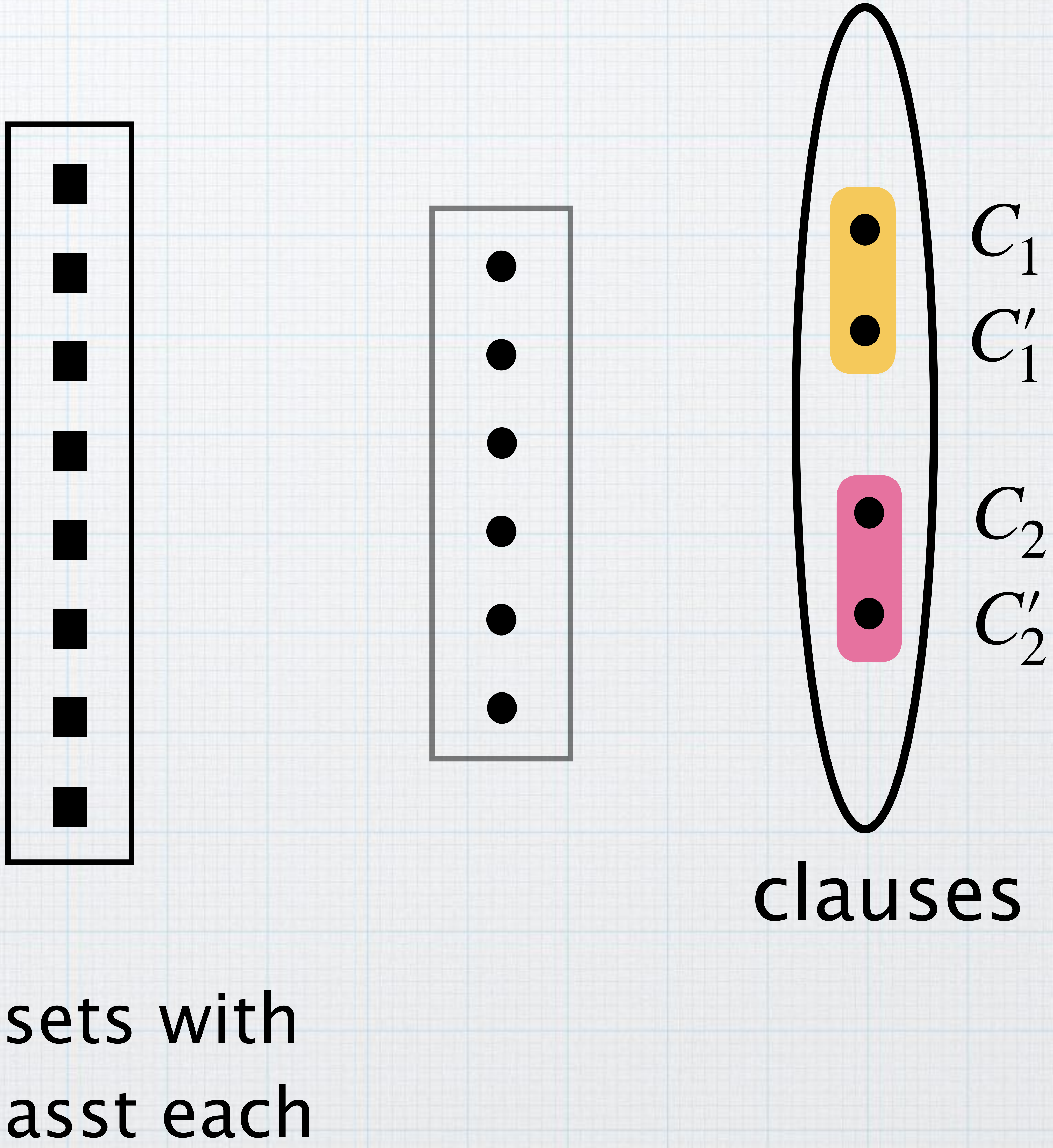


$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

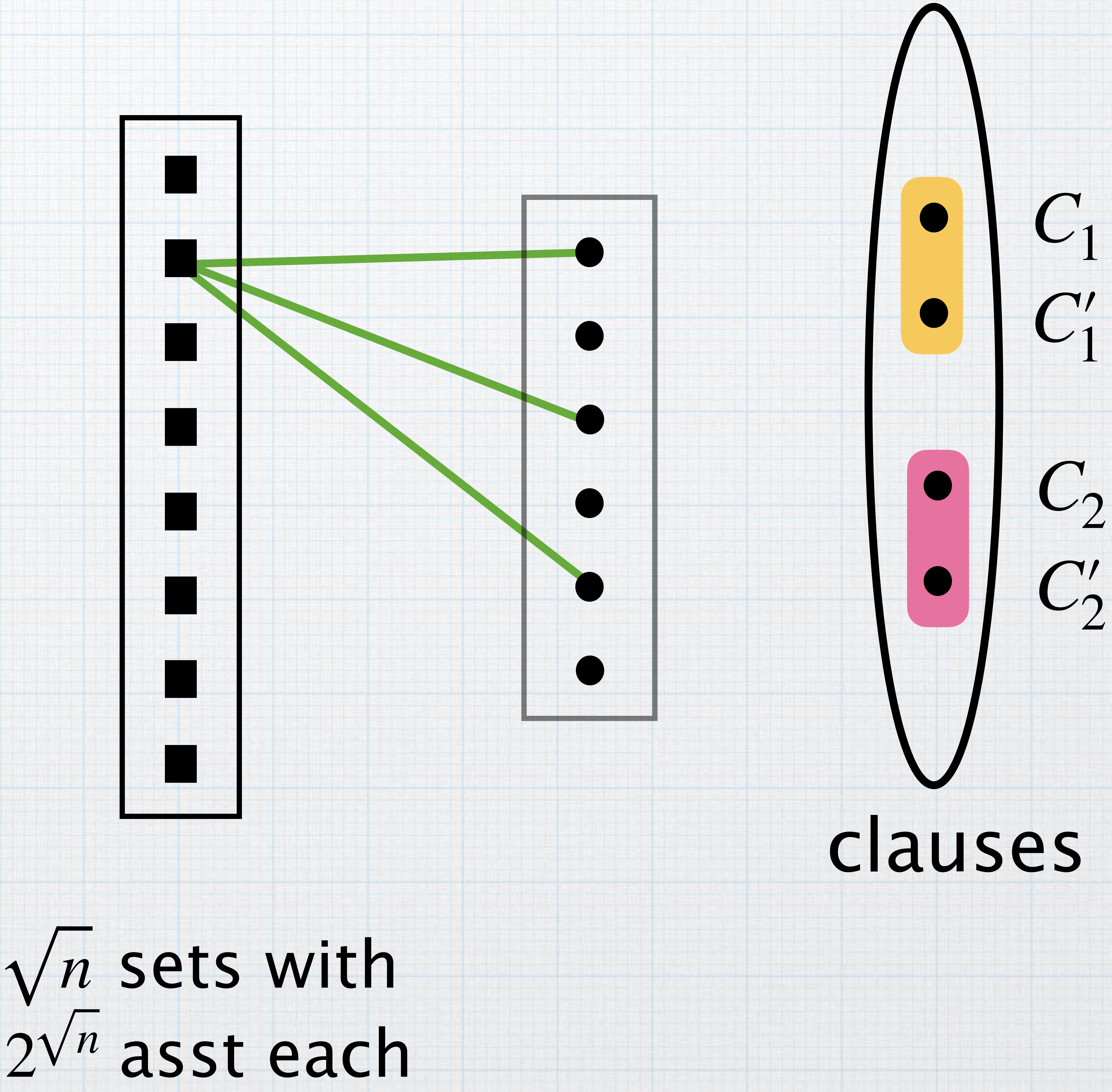
Asst a_1 connected to $A_1 = \{1,3,5\}$



3–SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

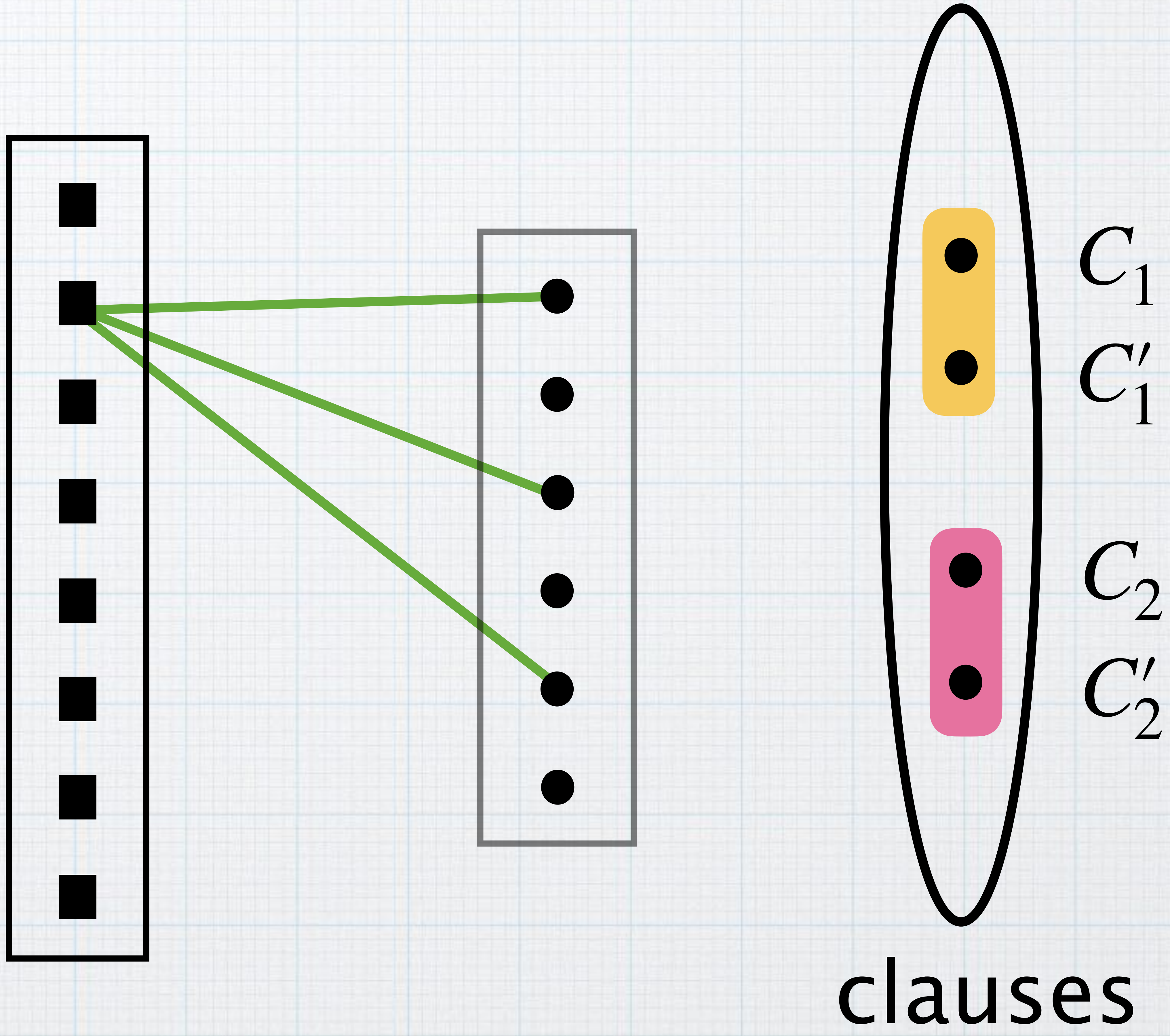


3–SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst



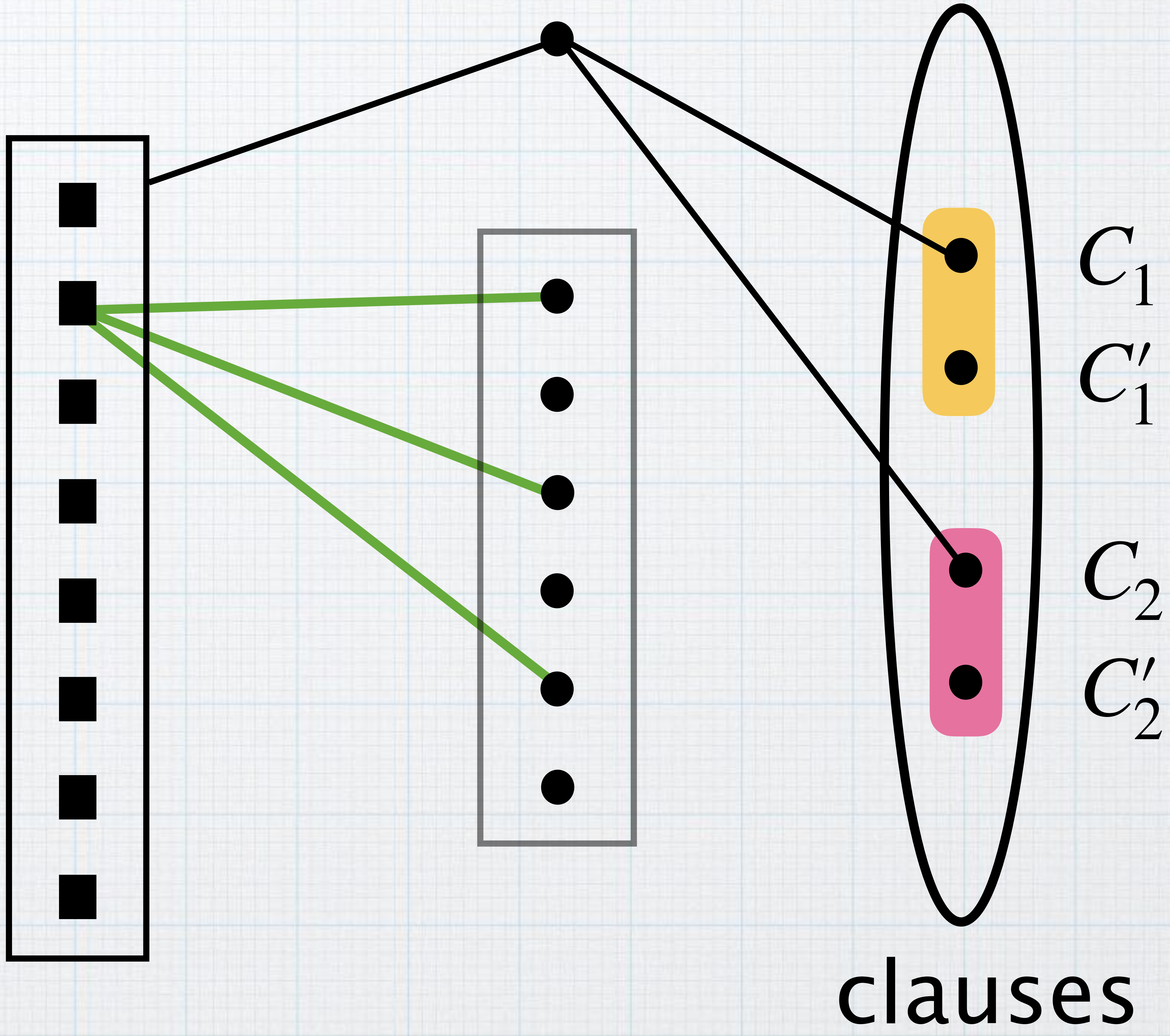
$\sqrt{n}$ sets with
$2^{\sqrt{n}}$ asst each

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst



$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

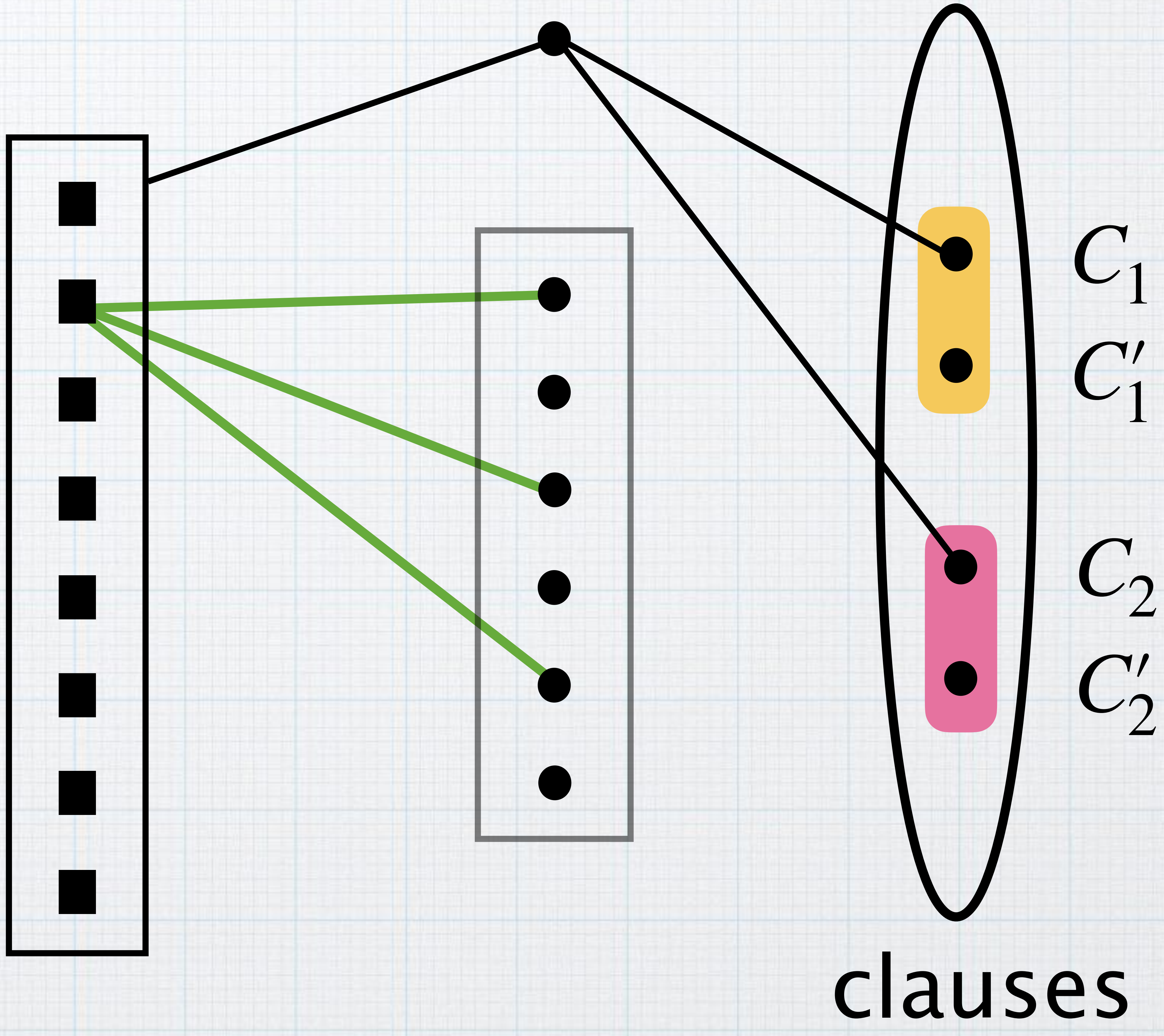
3–SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst

If $a_1 \nvDash C_1$, connect C_1' to $\bar{A}_1 = \{2,4,6\}$



$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

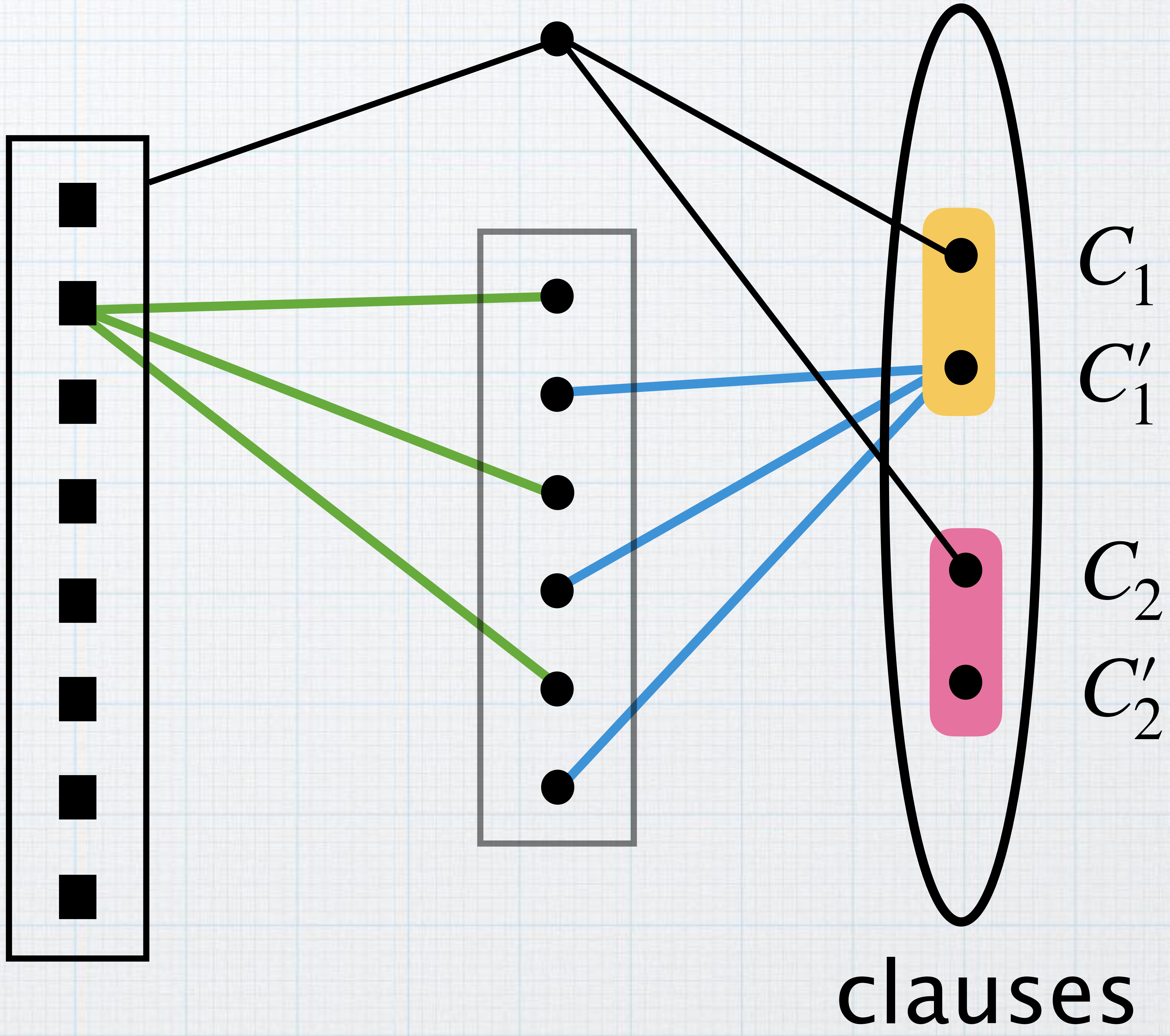
3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst

If $a_1 \nvDash C_1$, connect C_1' to $\bar{A}_1 = \{2,4,6\}$



$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

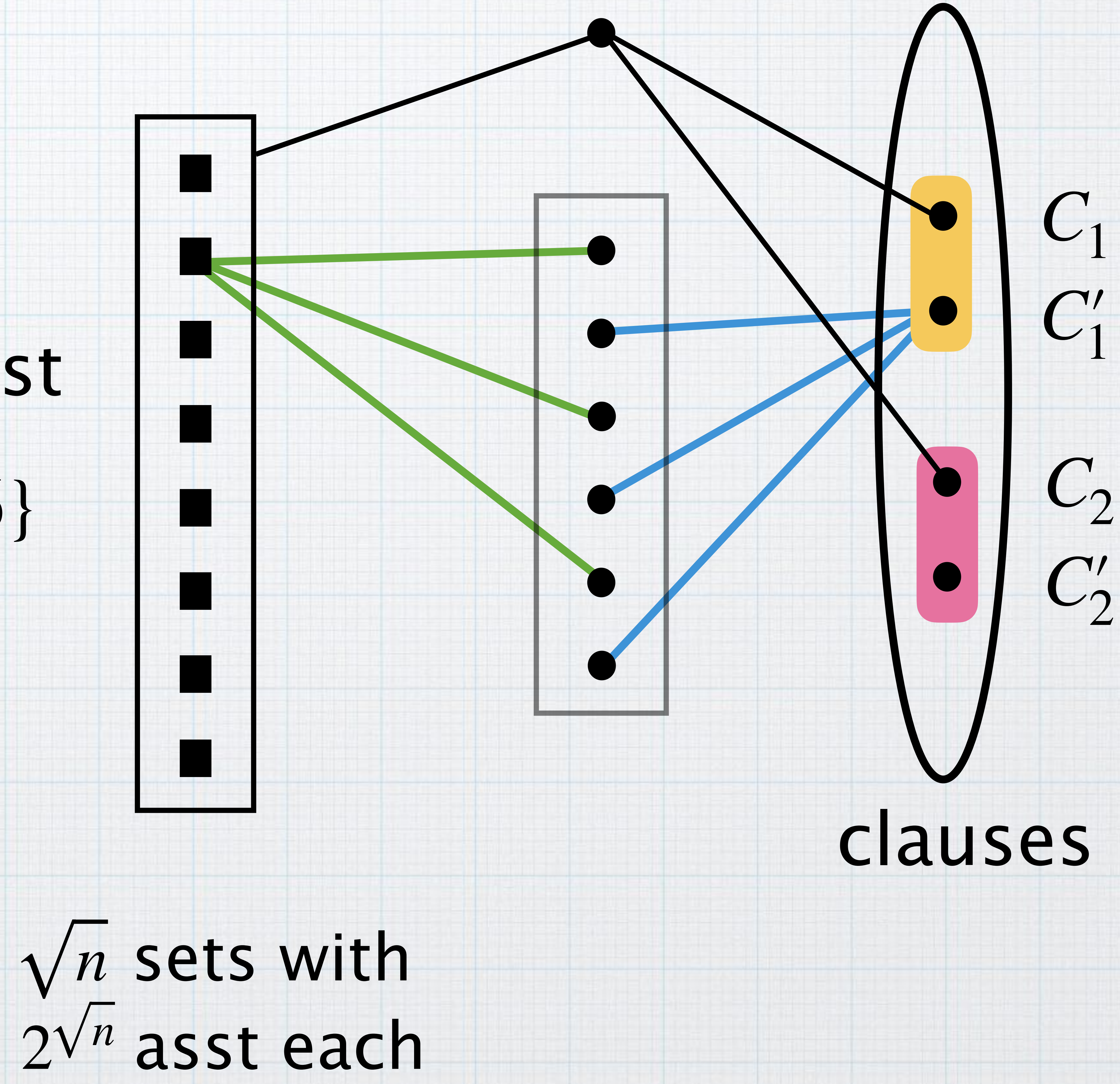
Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst

If $a_1 \vDash C_1$, connect C_1' to $\bar{A}_1 = \{2,4,6\}$

If $a_1 \nvDash C_2$ and $a_2 \vDash C_2$, connect C_2' to $\bar{A}_2$ for some $A_2 \neq A_1$



3-SAT (with n var) to Metric Dimension with vc = $\mathcal{O}(\sqrt{n})$

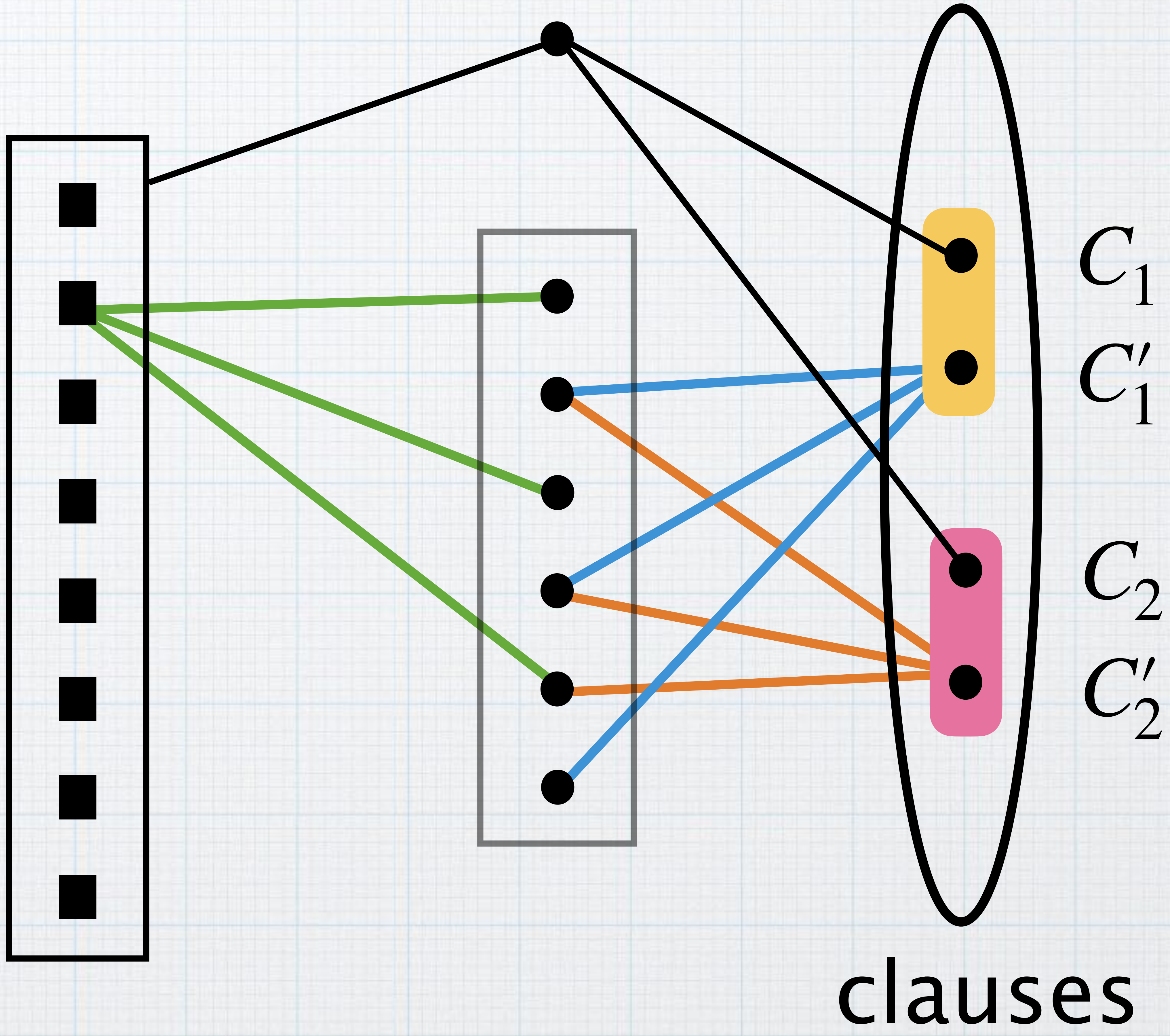
Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst

If $a_1 \vDash C_1$, connect C_1' to $\bar{A}_1 = \{2,4,6\}$

If $a_1 \nvDash C_2$ and $a_2 \vDash C_2$, connect C_2' to $\bar{A}_2$ for some $A_2 \neq A_1$



$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

3–SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

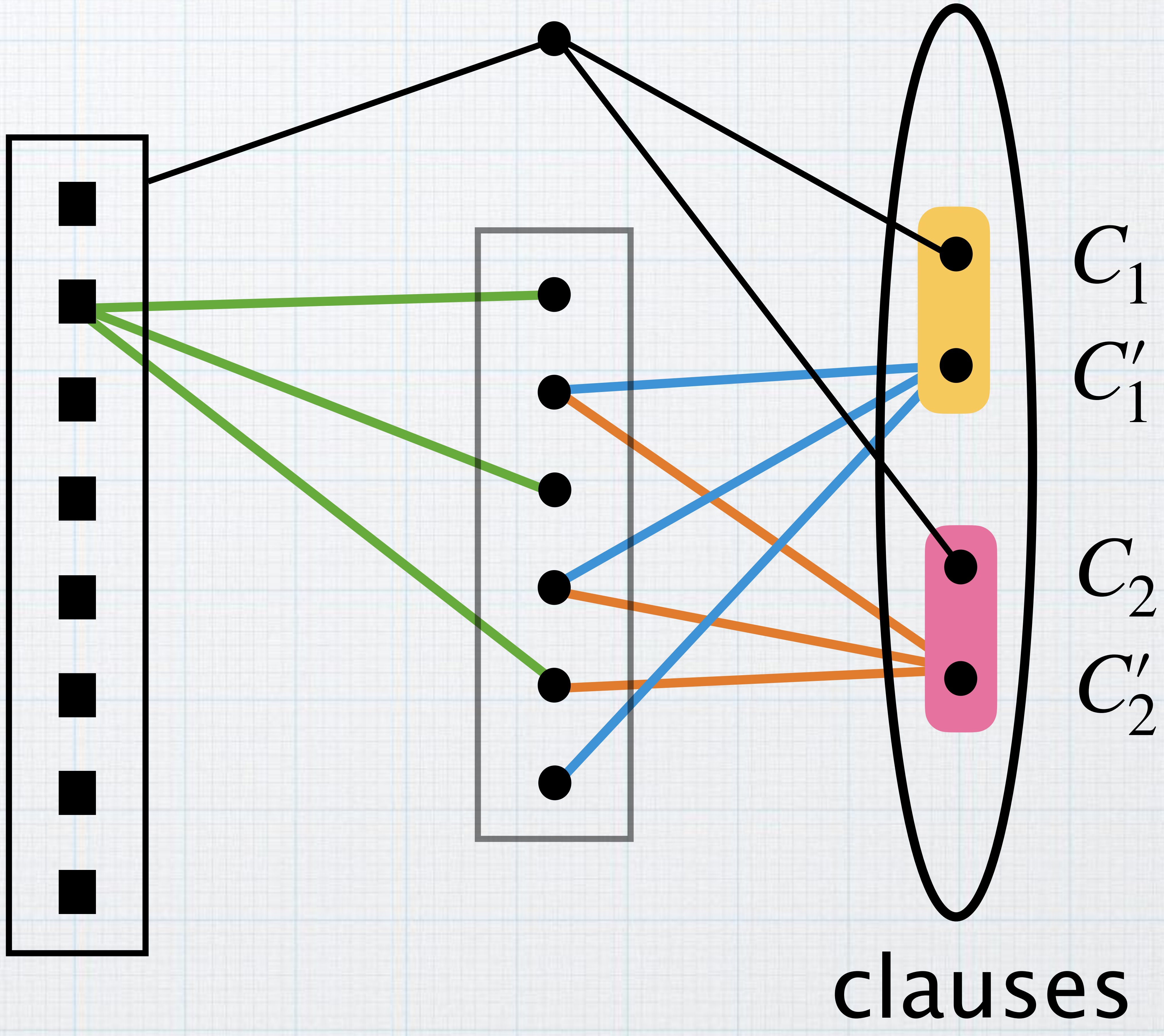
Asst a_1 connected to $A_1 = \{1,3,5\}$

C_1 C_2 are at dist two from any asst

If $a_1 \vDash C_1$, connect C_1' to $\bar{A}_1 = \{2,4,6\}$

If $a_1 \nvDash C_2$ and $a_2 \vDash C_2$, connect C_2' to $\bar{A}_2$ for some $A_2 \neq A_1$

$\Rightarrow dist(a_1, C') = 3$ iff $a_1 \vDash C$



$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

3-SAT (with n var) to Metric Dimension with vc $= \mathcal{O}(\sqrt{n})$

Replace direct connection via universe of Sperner Family

Asst a_1 connected to $A_1 = \{1,3,5\}$

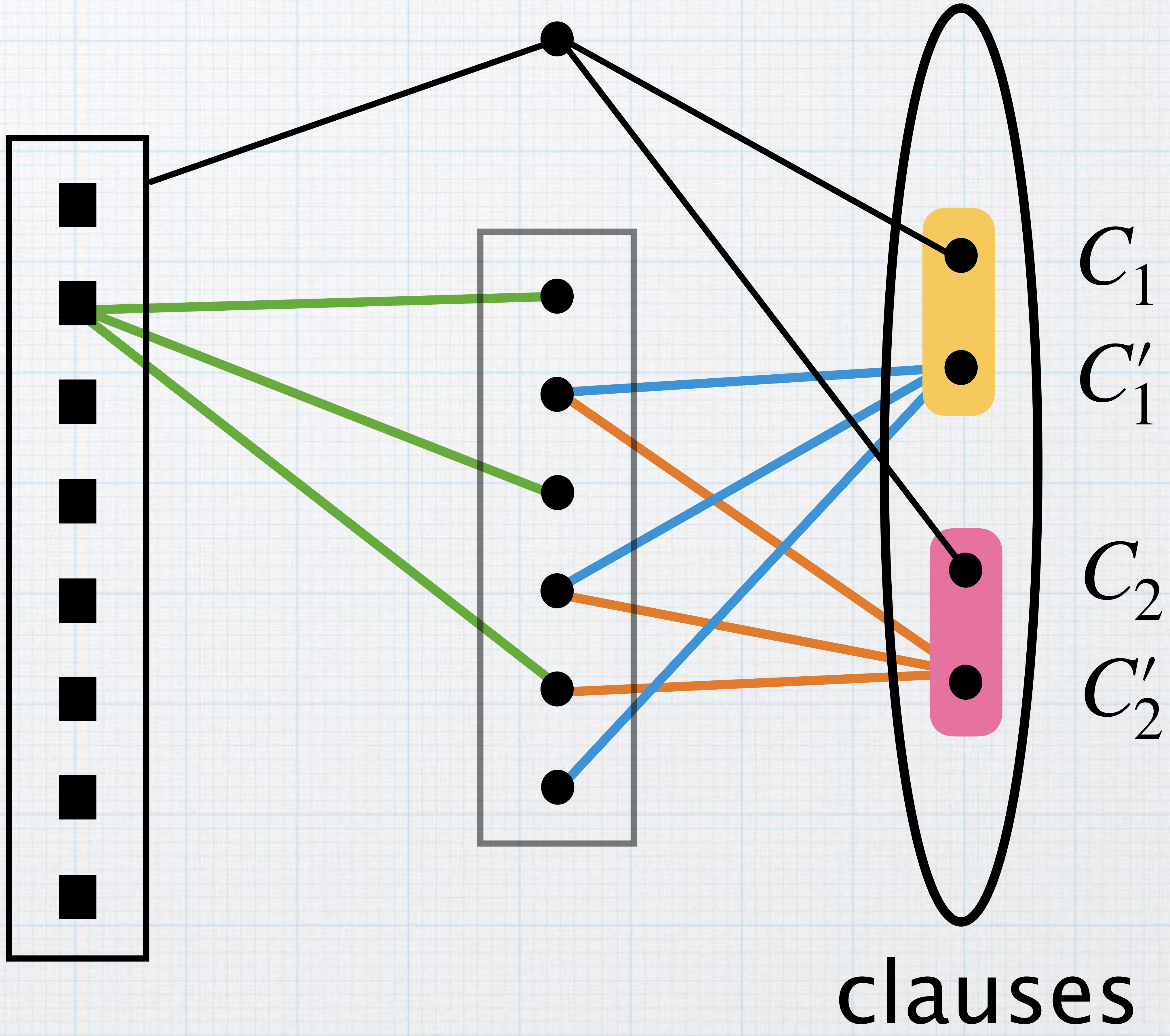
C_1 C_2 are at dist two from any asst

If $a_1 \vDash C_1$, connect C_1' to $\bar{A}_1 = \{2,4,6\}$

If $a_1 \nvDash C_2$ and $a_2 \vDash C_2$, connect C_2' to $\bar{A}_2$ for some $A_2 \neq A_1$

$\Rightarrow dist(a_1, C') = 3$ iff $a_1 \vDash C$

$\Rightarrow \mathrm{vc}(G) = \mathcal{O}(\sqrt{n})$



$\sqrt{n}$ sets with $2^{\sqrt{n}}$ asst each

Thm. Metric Dimension do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Thm. Metric Dimension do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Recall that `free vertices' is a resolving set is at most vc

Thm. Metric Dimension do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Recall that `free vertices' is a resolving set is at most vc

Hence, kernel with $2^{o(\mathsf{vc})}$ vertices will imply $2^{o(\mathsf{vc}^2)}$ algo.

Thm. Metric Dimension do not admit $2^{o(\mathsf{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Recall that `free vertices' is a resolving set is at most vc

Hence, kernel with $2^{o(\mathsf{vc})}$ vertices will imply $2^{o(\mathsf{vc}^2)}$ algo.

Thm. The Metric Dimension do not admit a kernel with $2^{o(\mathsf{vc})}$ vertices unless the ETH fails.

Thm. Metric Dimension do not admit $2^{o(\text{vc}^2)} \cdot n^{\mathcal{O}(1)}$ algo unless the ETH fails.

Recall that `free vertices' is a resolving set is at most vc

Hence, kernel with $2^{o(\text{vc})}$ vertices will imply $2^{o(\text{vc}^2)}$ algo.

Thm. The Metric Dimension do not admit a kernel with $2^{o(\text{vc})}$ vertices unless the ETH fails.

Any other (metric graph) NP-Complete problems that admit such lower bounds?

Thank you