Online Covering

Secretaries, Prophets, and Universal Maps

FOCS 2021 + Forthcoming Work

Roie Levin

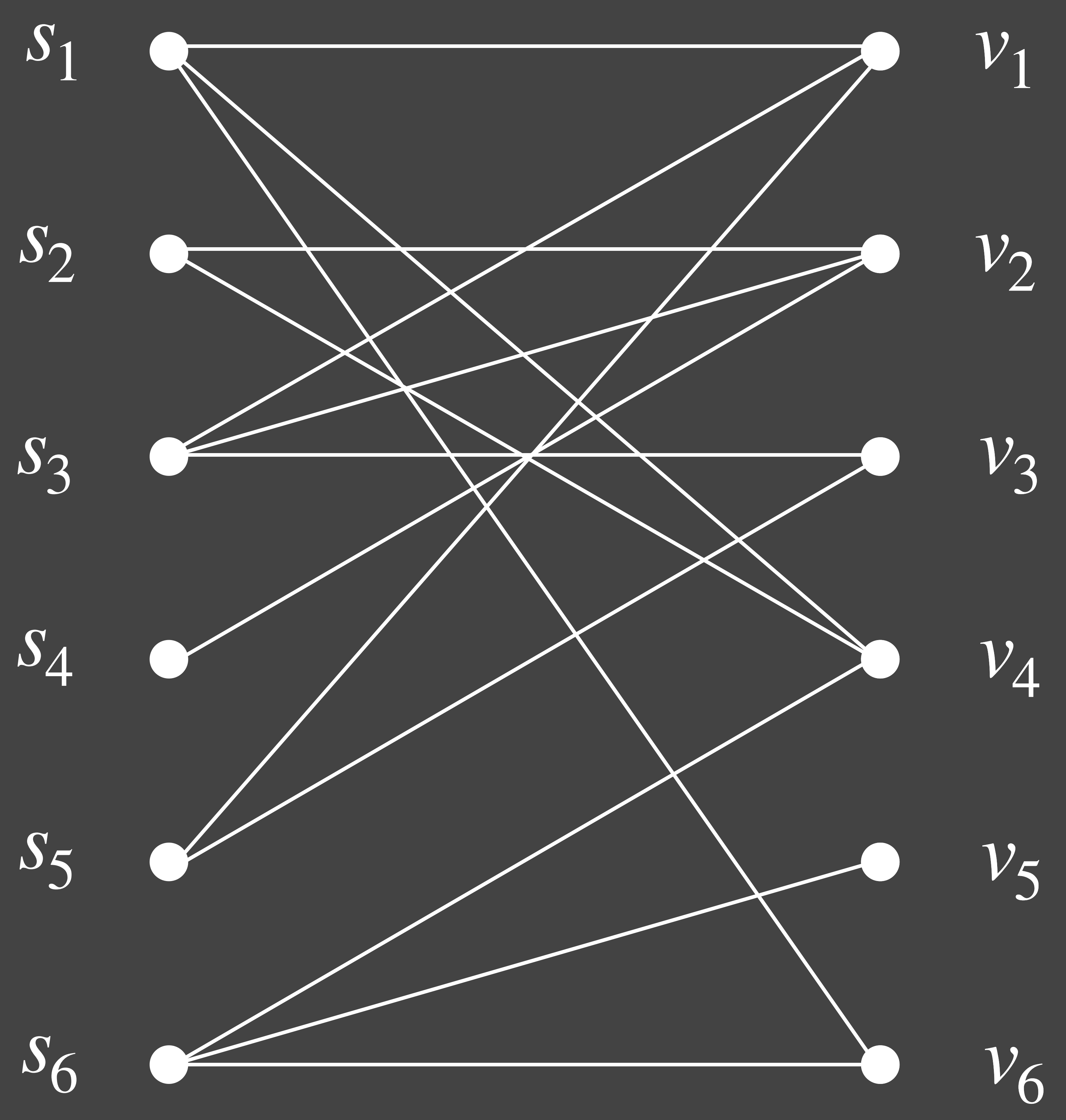
Anupam Gupta (CMU)

Gregory Kehne (Harvard)

Set Cover

$\mathcal{S}$

m sets



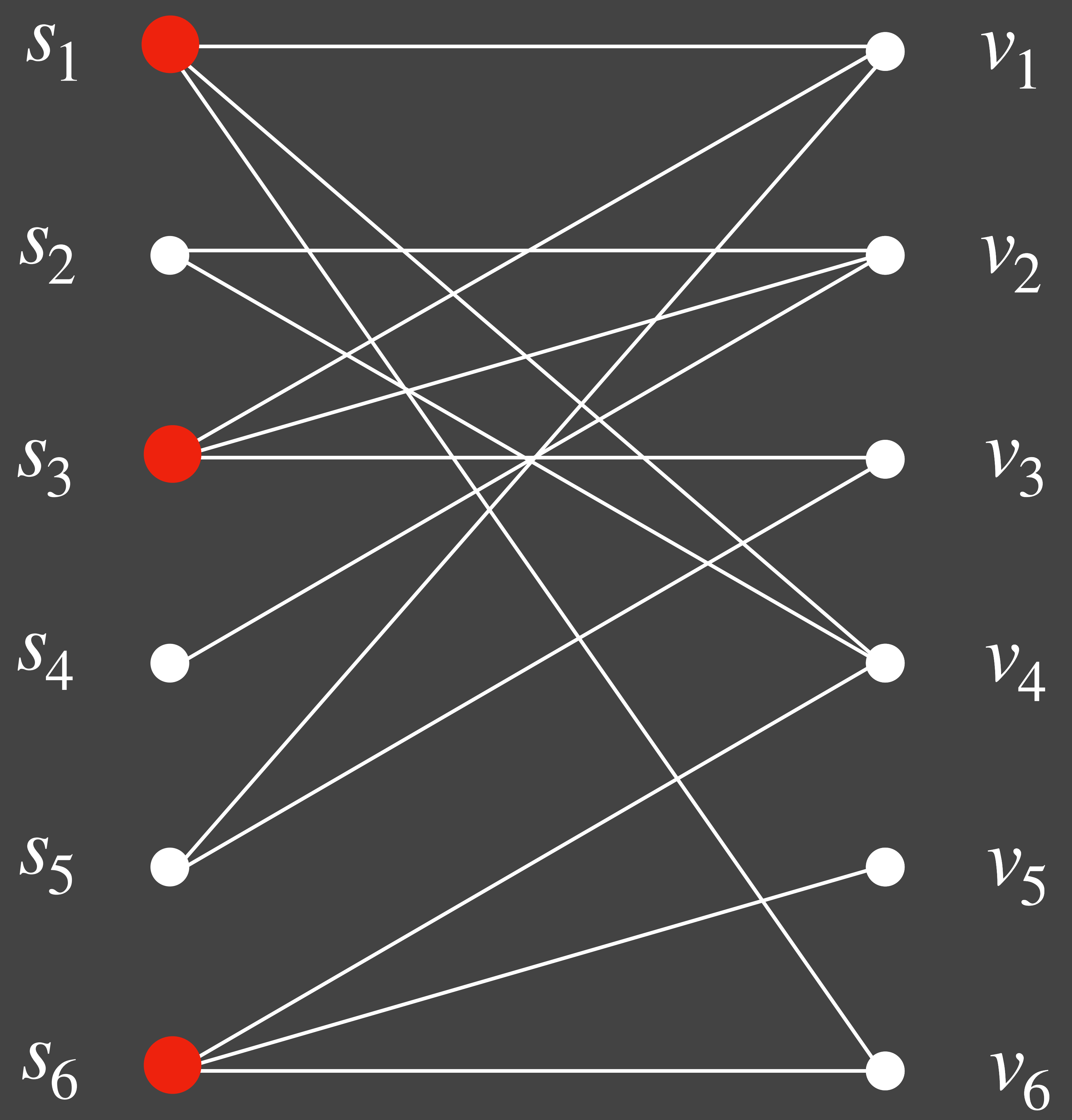
$\mathcal{U}$

n elements

Set Cover

$\mathcal{S}$

m sets



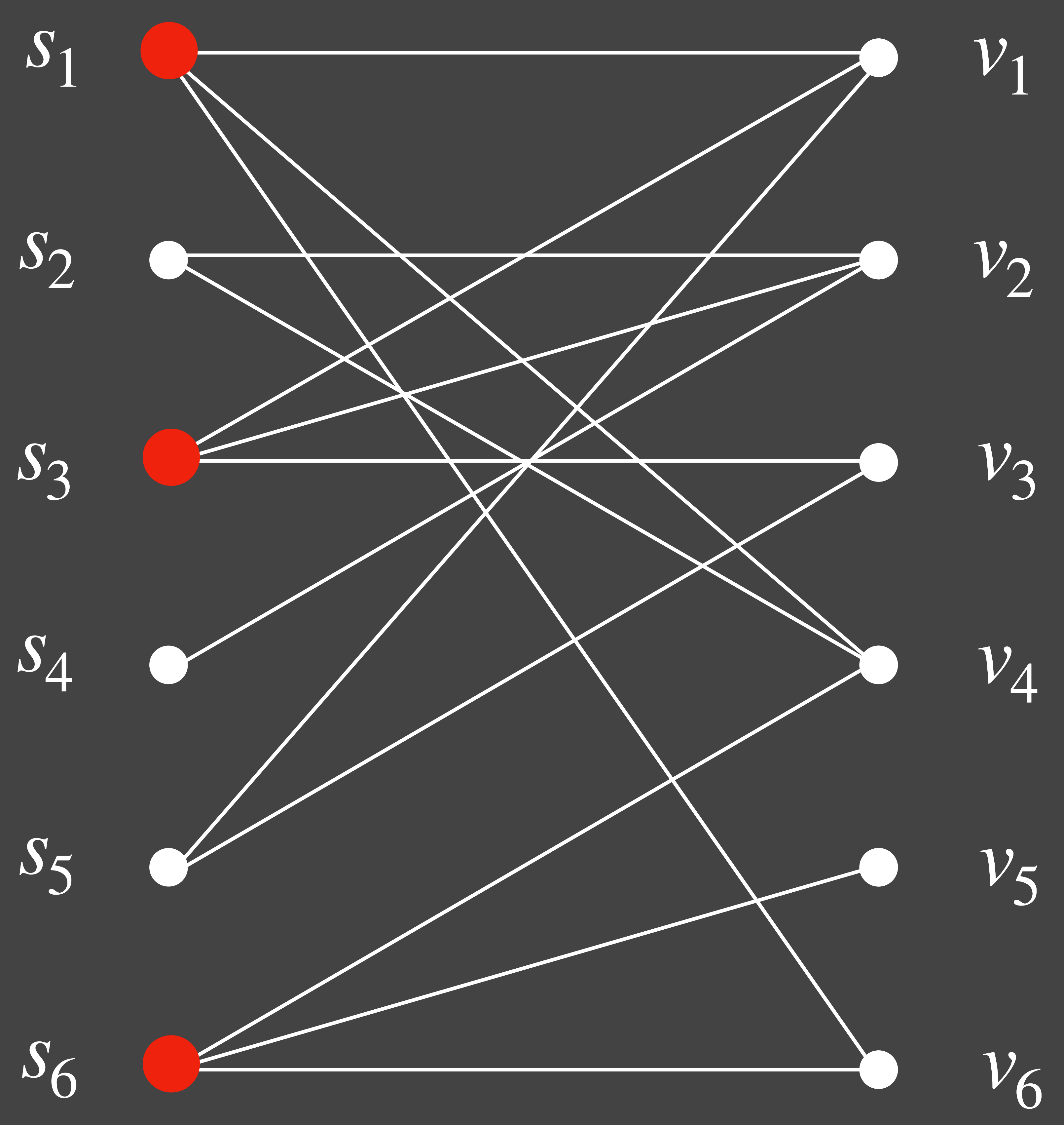
$\mathcal{U}$

n elements

Set Cover

Apx: $\log n + 1$
[Johnson 74],[Lovasz 75],
[Chvatal 79]

$\mathcal{S}$
m sets



$\mathcal{U}$
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets

s_1

s_2

s_3

s_4

s_5

s_6

$\mathcal{U}$

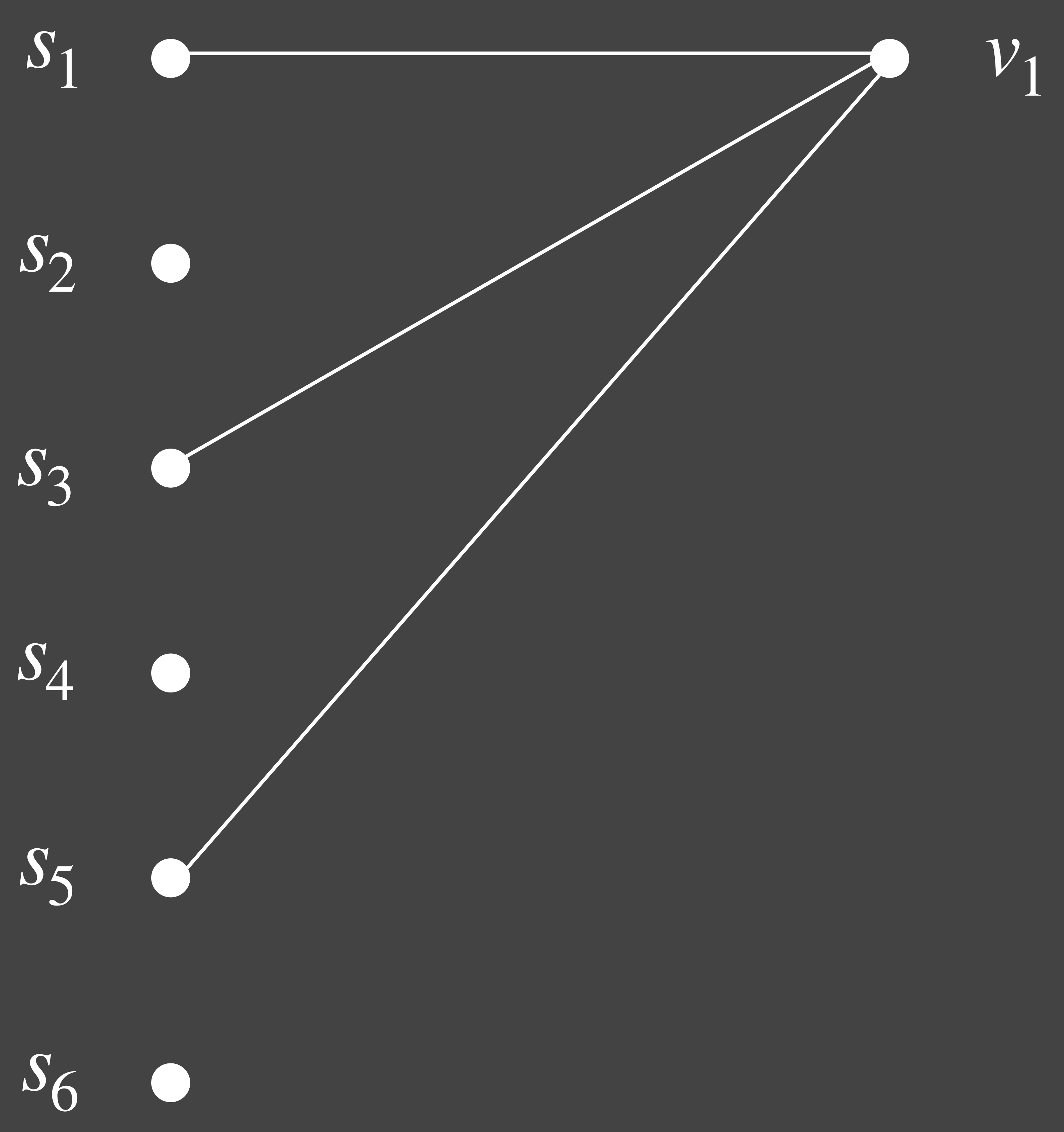
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



$\mathcal{U}$

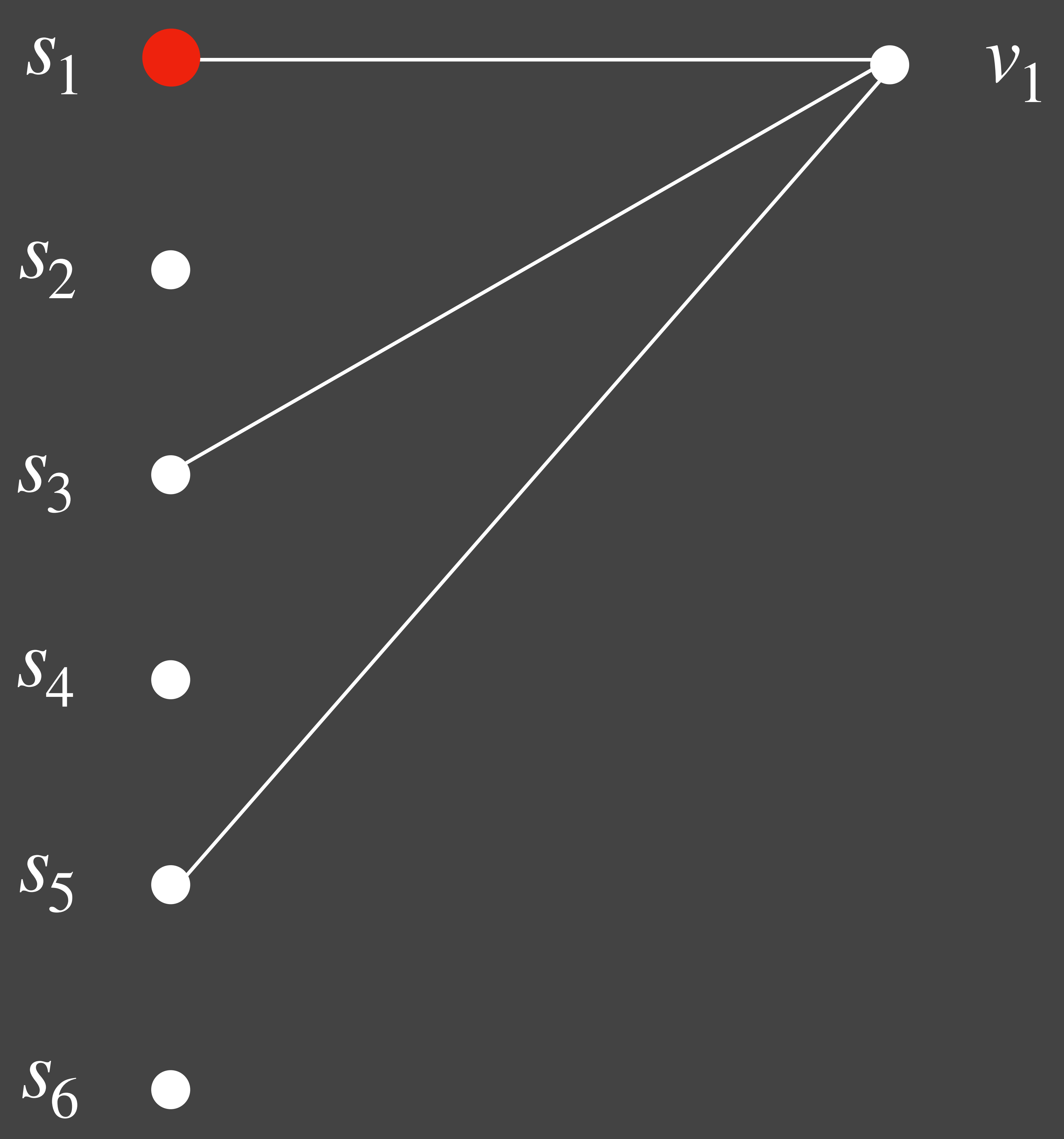
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



$\mathcal{U}$

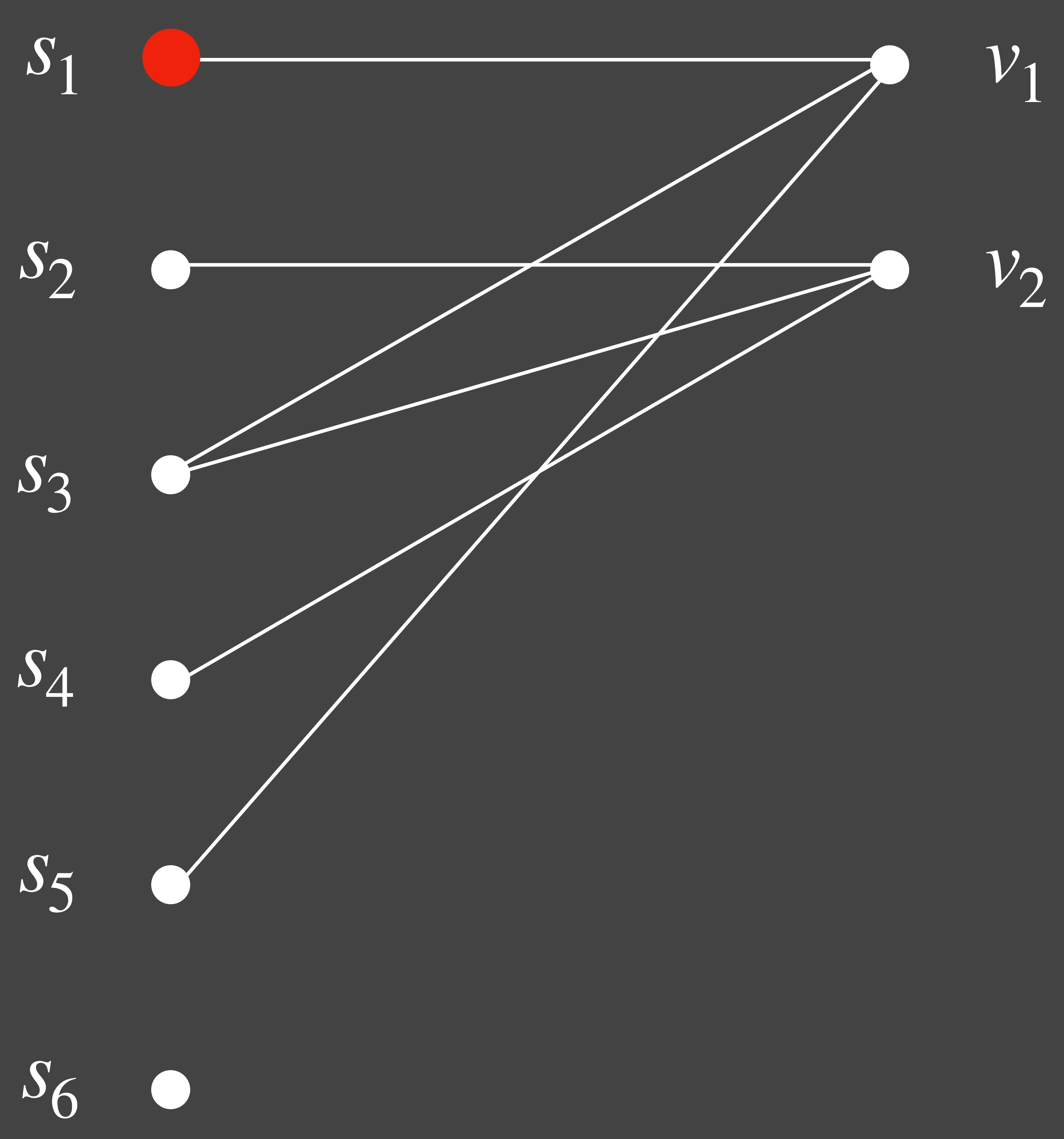
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



$\mathcal{U}$

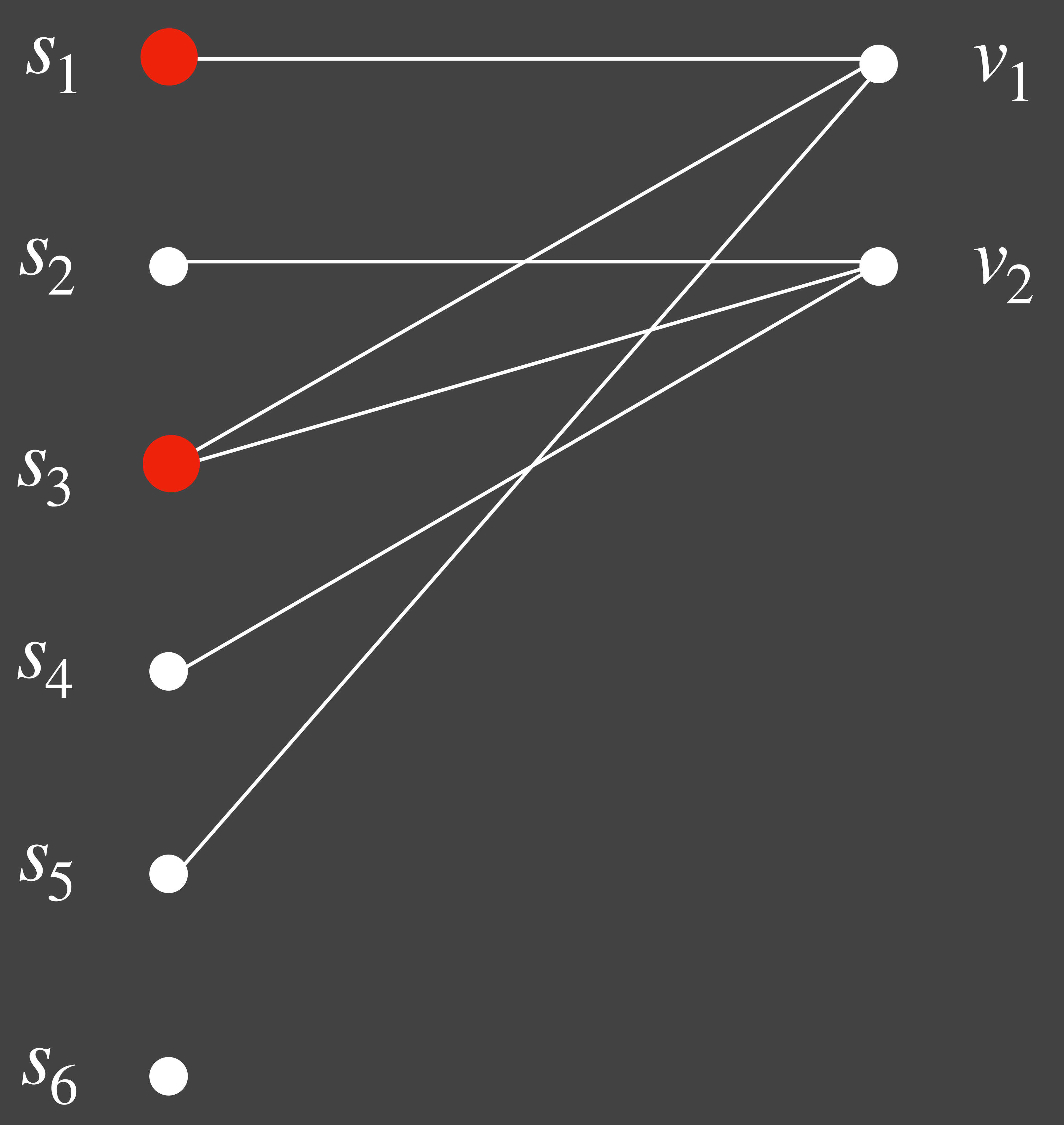
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



$\mathcal{U}$

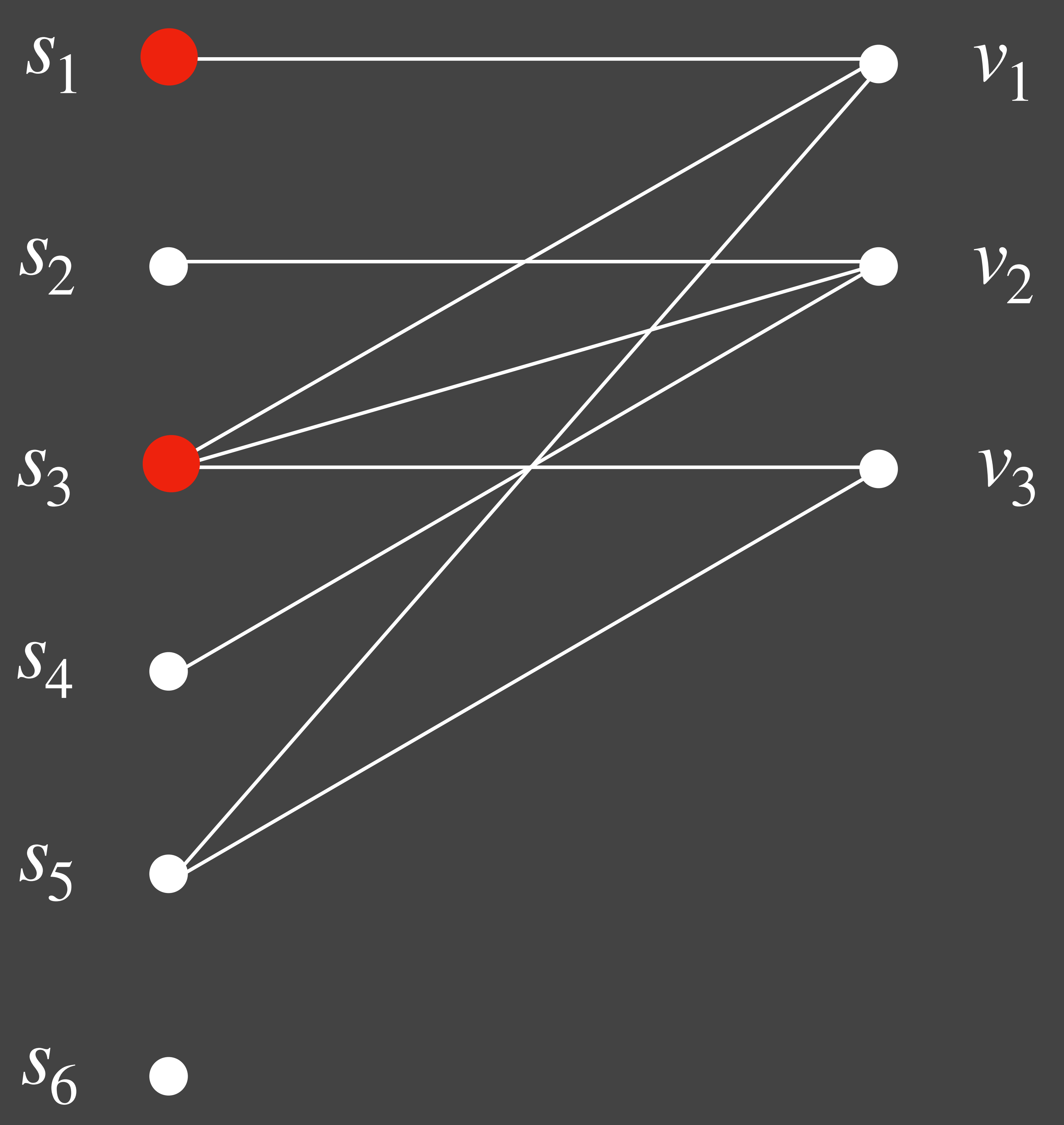
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



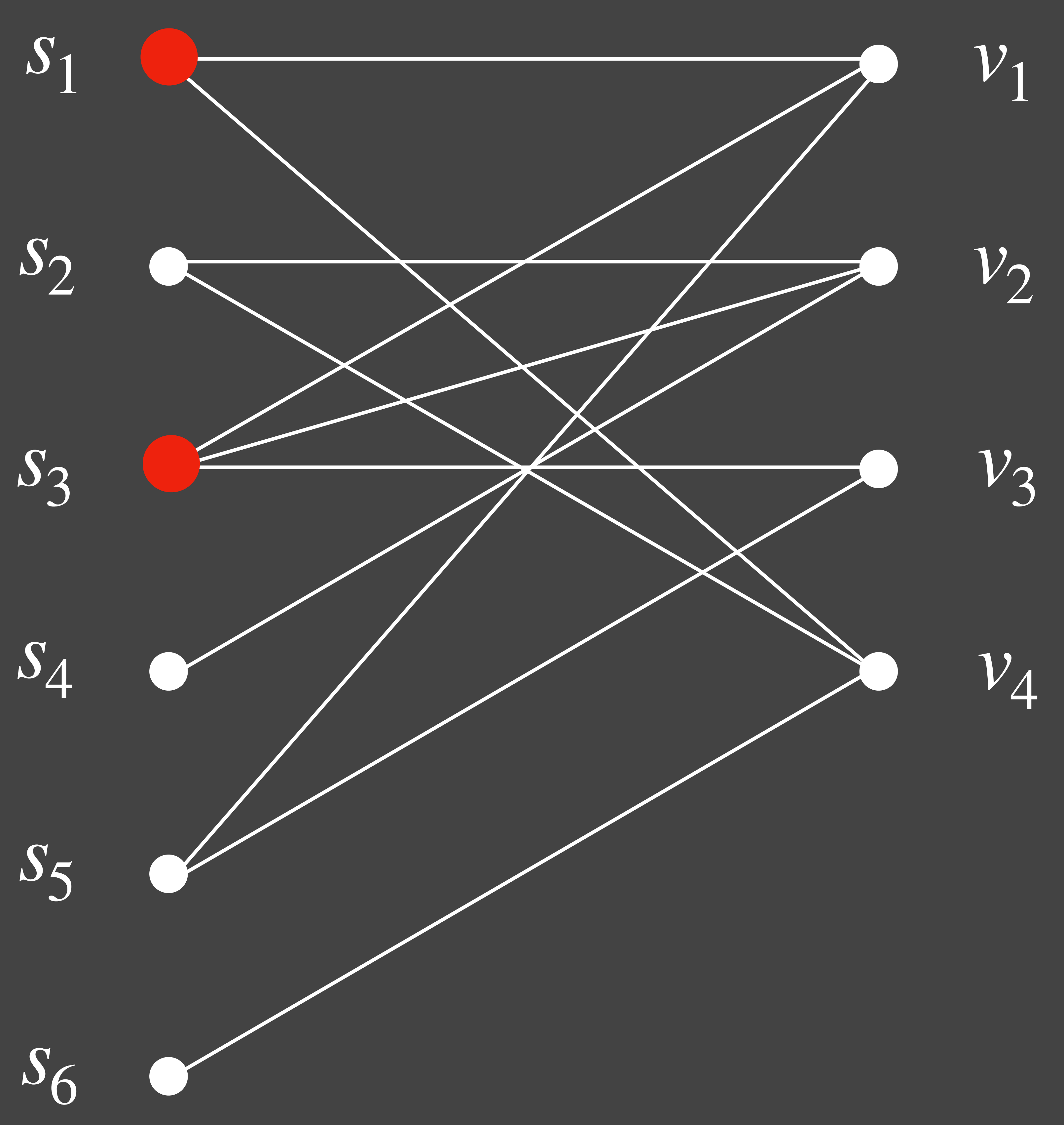
$\mathcal{U}$

n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$
m sets

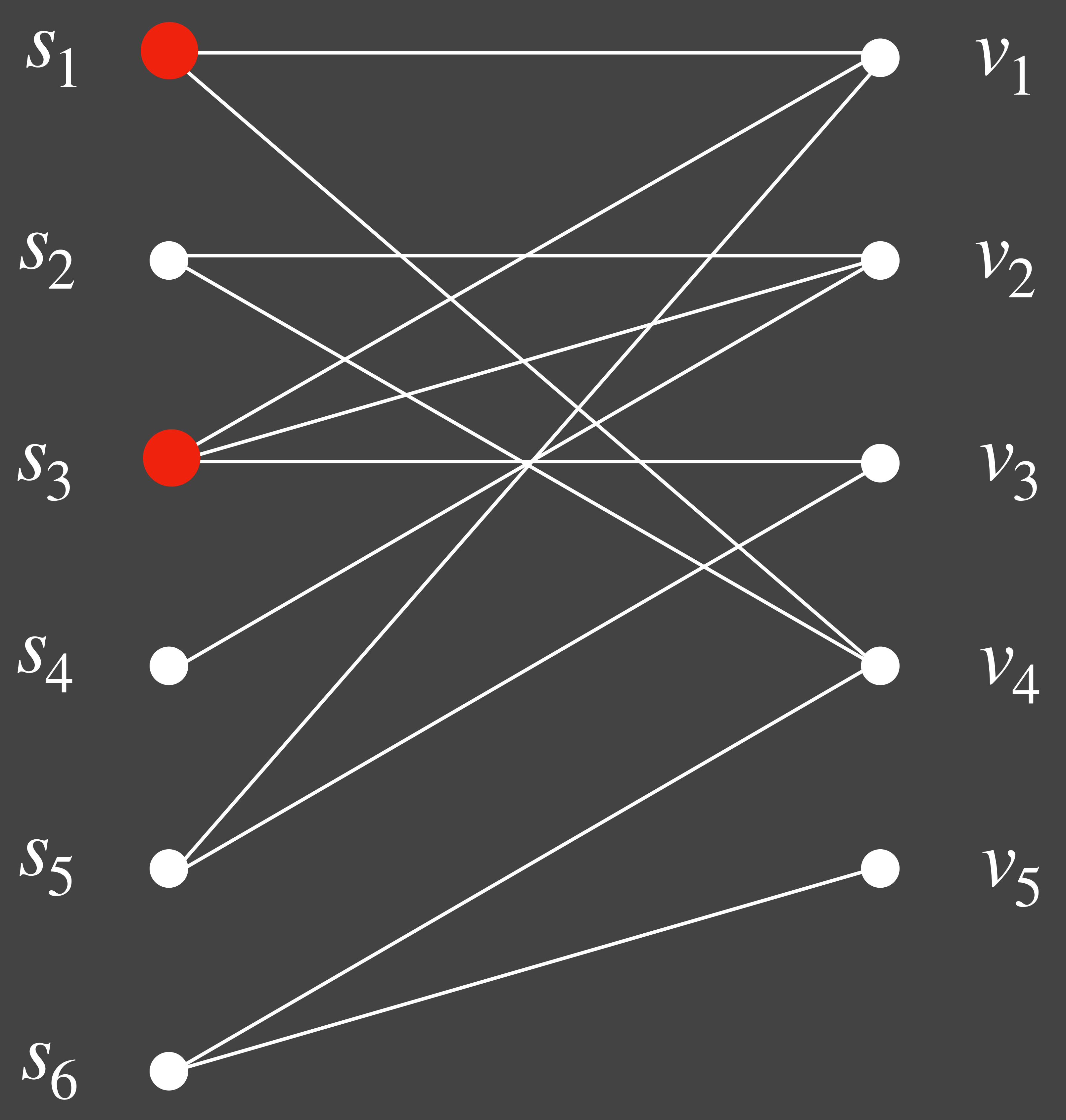


$\mathcal{U}$
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$
m sets



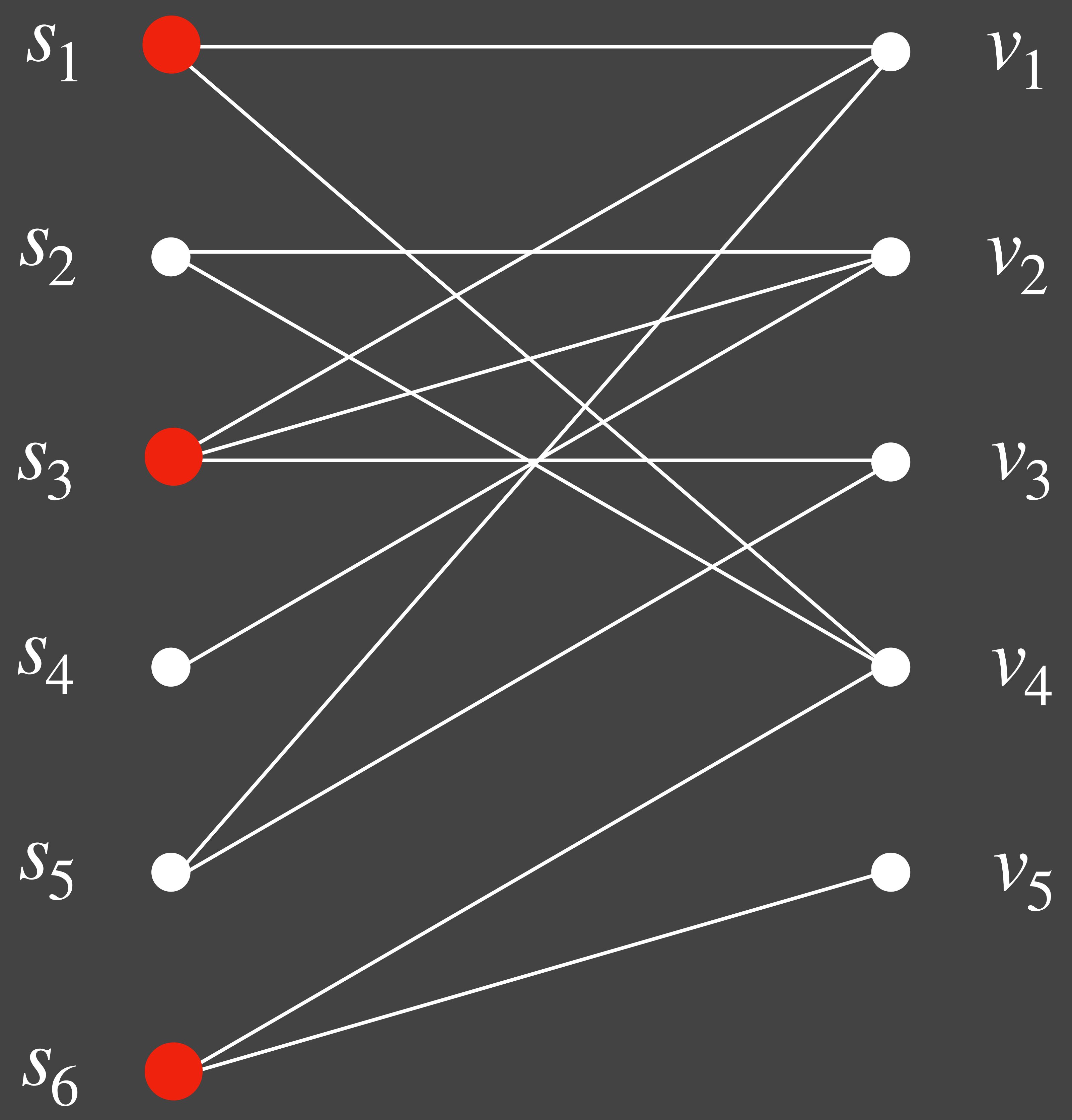
$\mathcal{U}$
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



$\mathcal{U}$

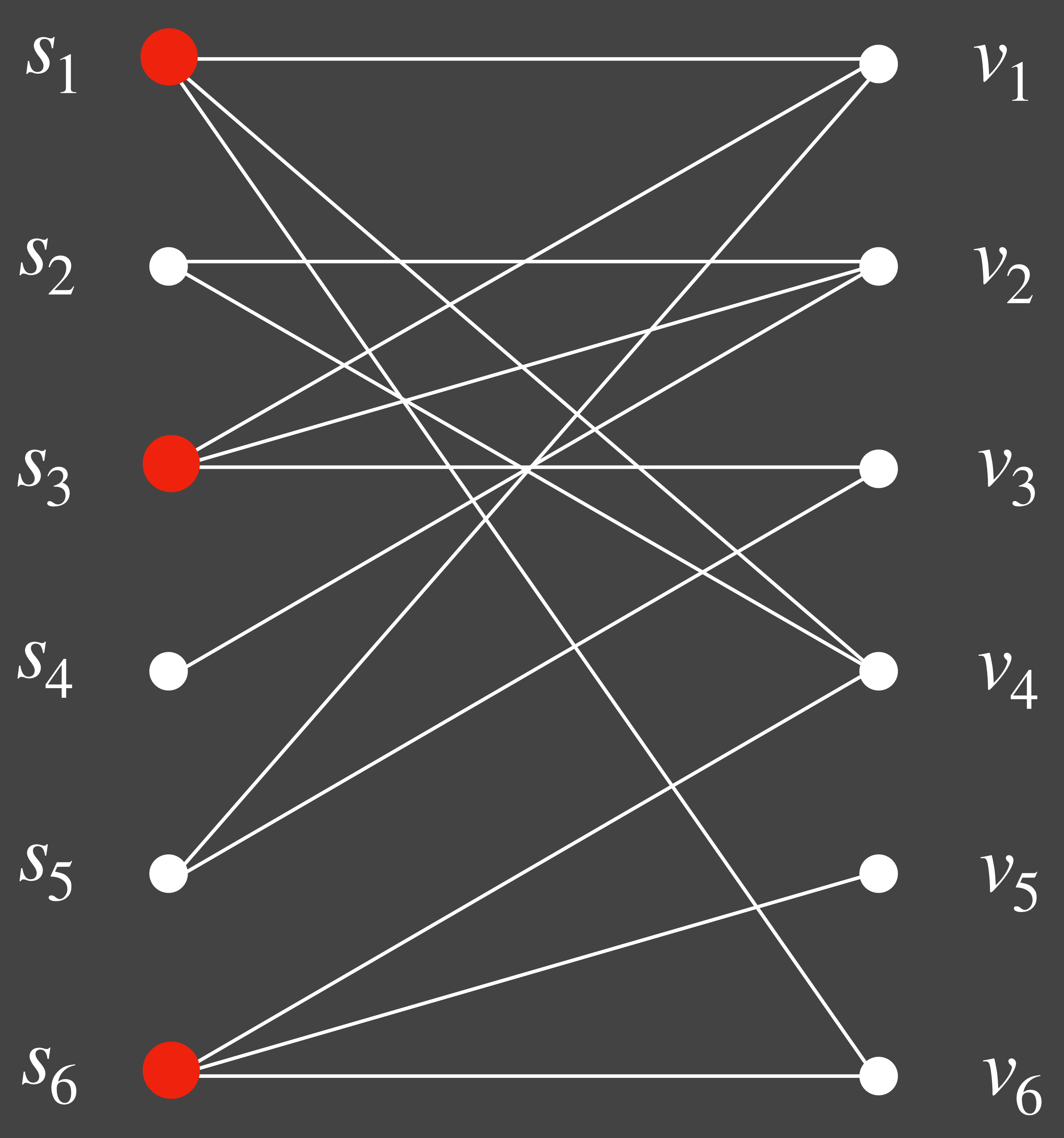
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

$\mathcal{S}$

m sets



$\mathcal{U}$

n elements

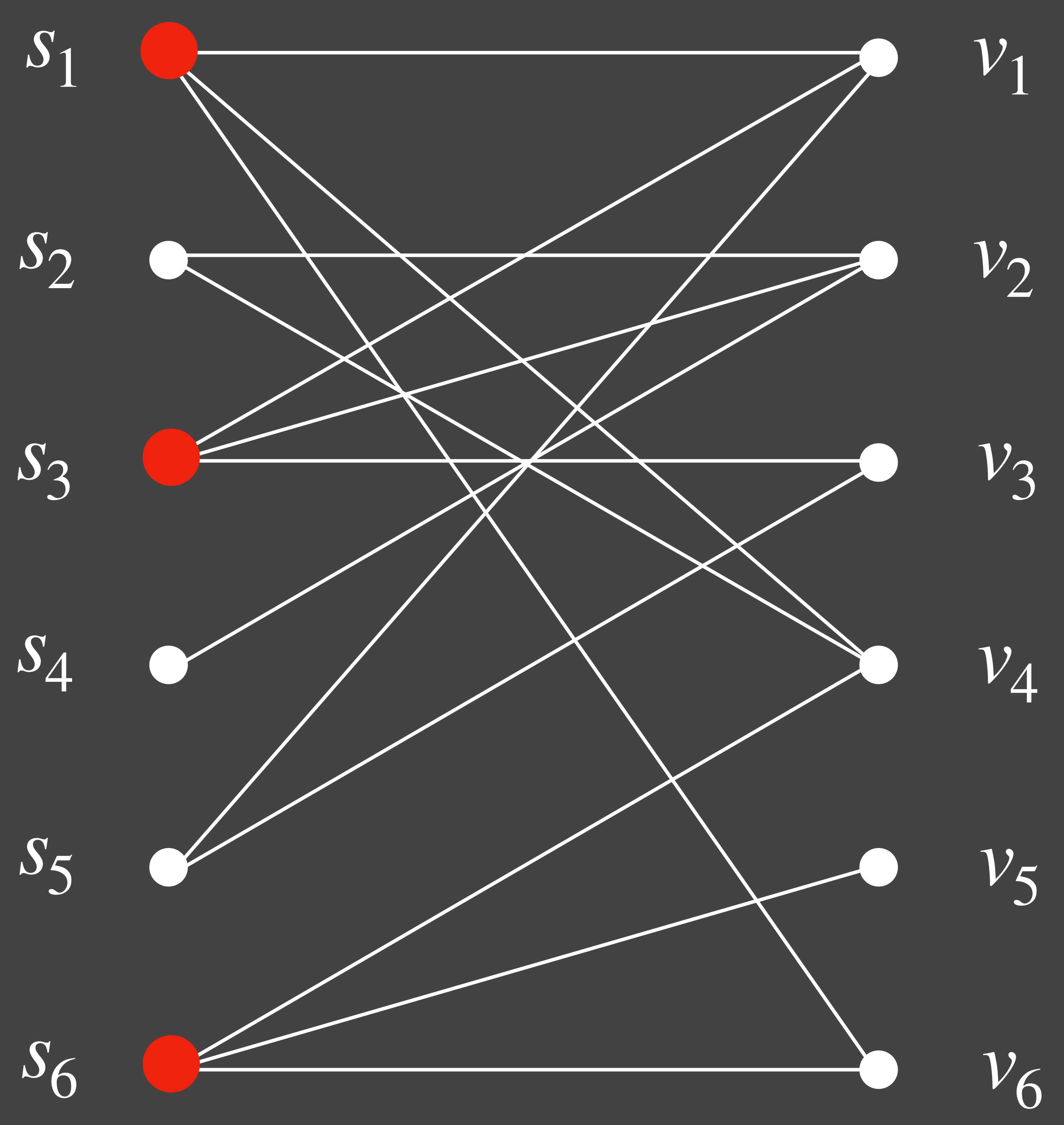
Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

CR: $O(\log n \log m)$

[Alon+ 03]
[Buchbinder Naor 09]

$\mathcal{S}$
m sets



$\mathcal{U}$
n elements

Online Set Cover

[Alon Awerbuch Azar Buchbinder Naor 03]

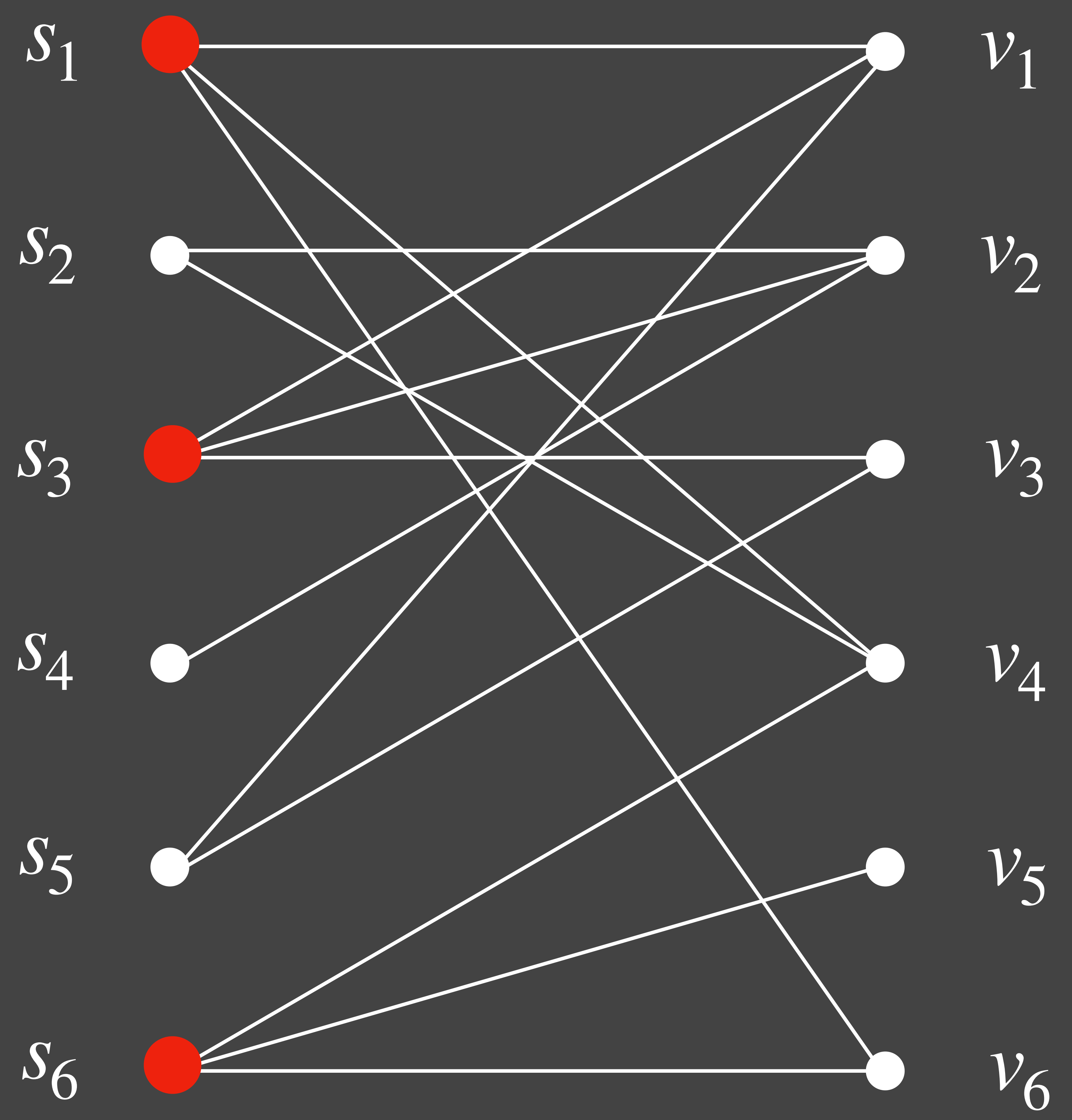
CR: $O(\log n \log m)$

[Alon+ 03]

[Buchbinder Naor 09]

$\mathcal{S}$

m sets



$\mathcal{U}$

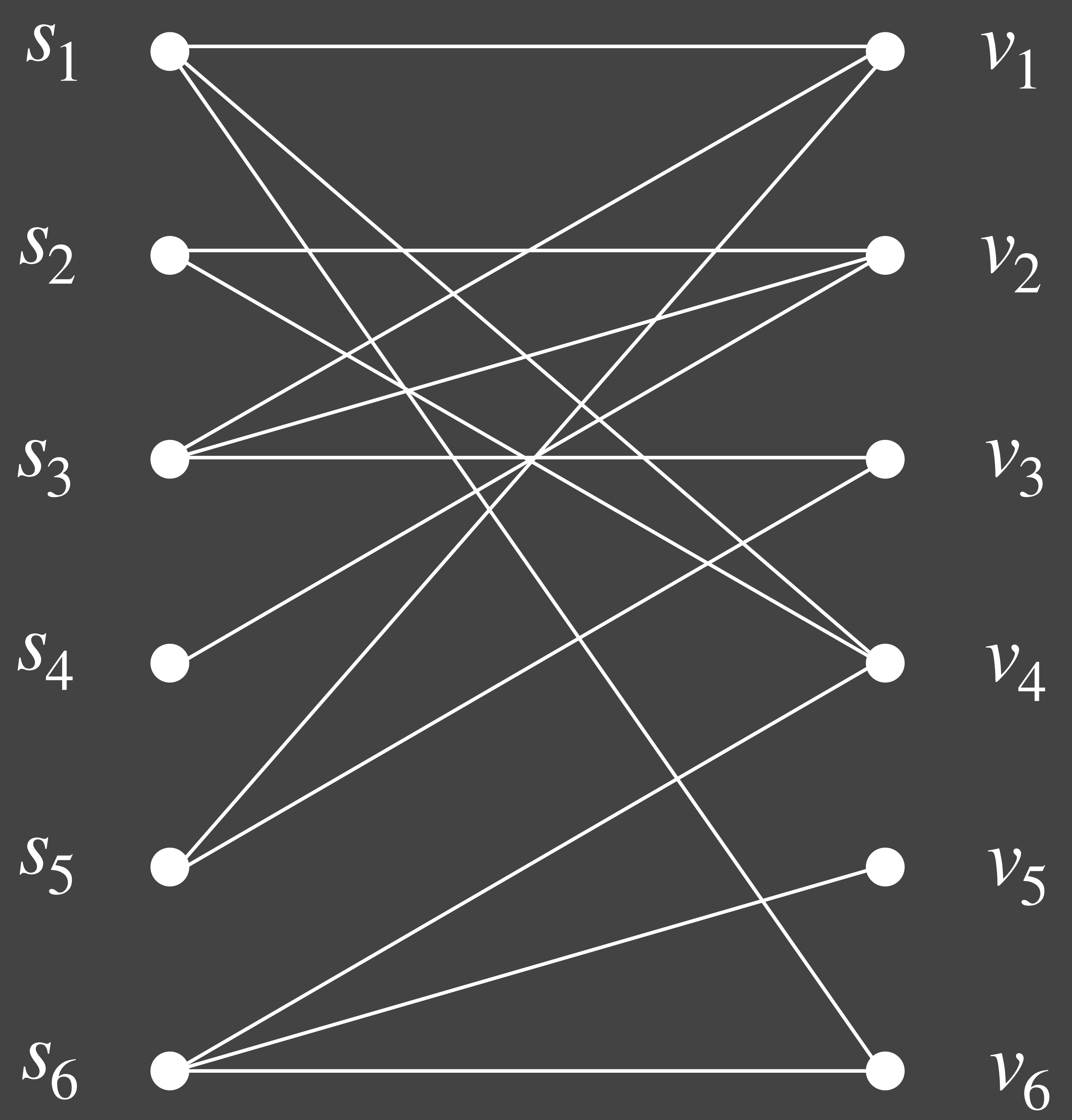
n elements

Q: What happens beyond the worst case?

Relaxation 1: Random Order (RO)

$\mathcal{S}$

m sets



$\mathcal{U}$

n elements

Relaxation 1: Random Order (RO)

$\mathcal{S}$

m sets

s_1

s_2

s_3

s_4

s_5

s_6

$\mathcal{U}$

n elements

Relaxation 1: Random Order (RO)

$\mathcal{S}$

m sets

s_1

s_2

s_3

s_4

s_5

s_6

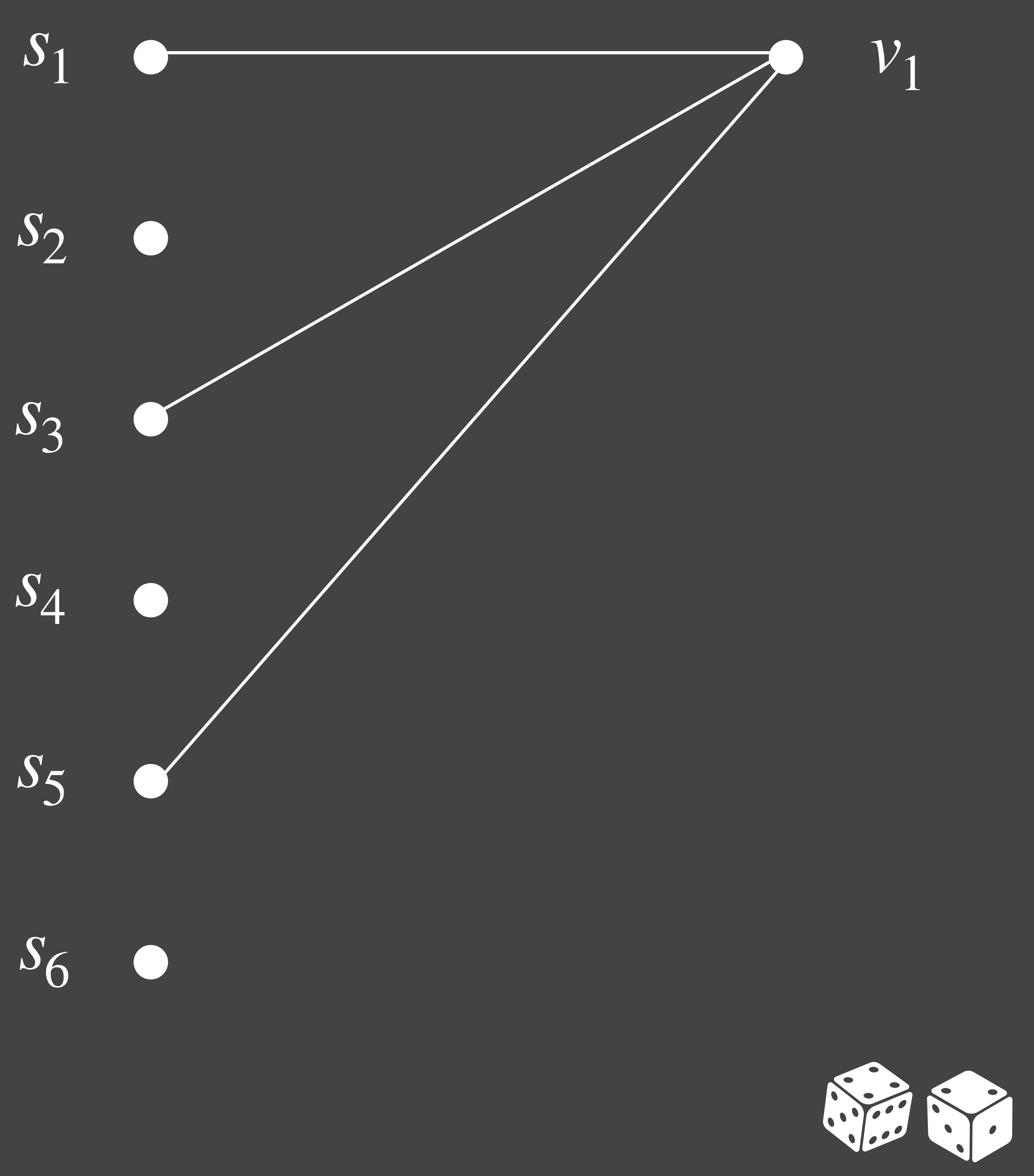
$\mathcal{U}$

n elements

Relaxation 1: Random Order (RO)

$\mathcal{S}$

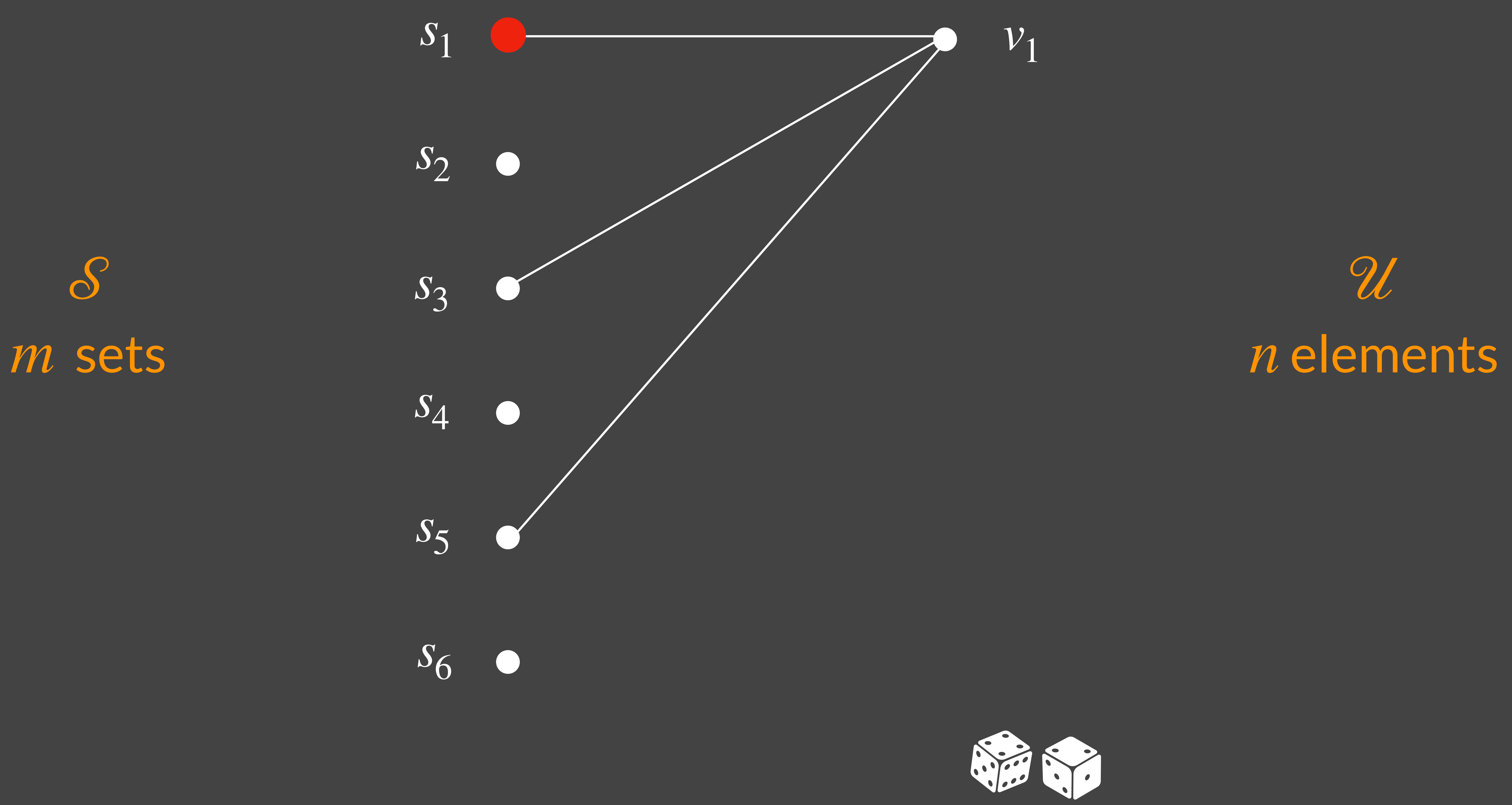
m sets



$\mathcal{U}$

n elements

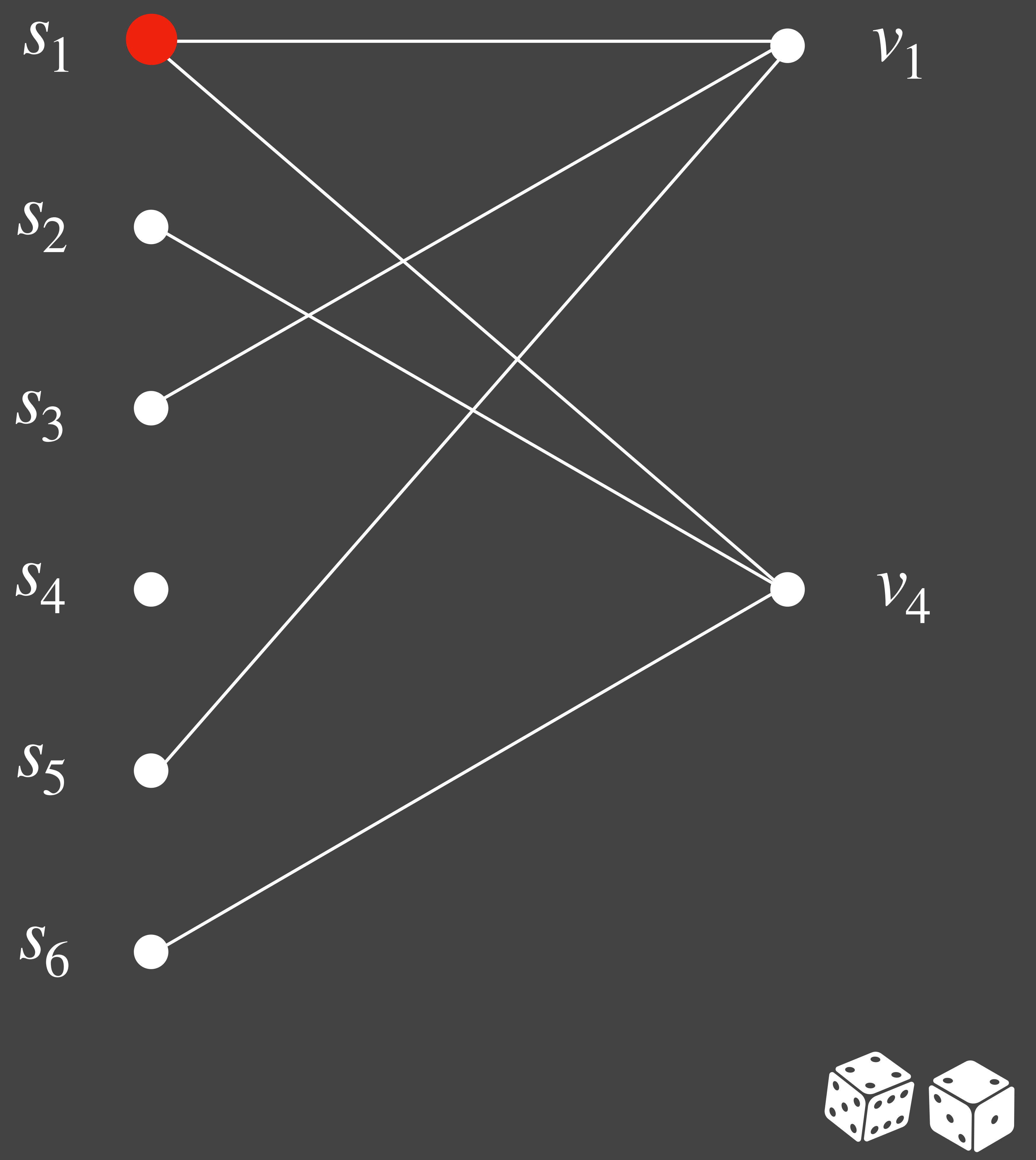
Relaxation 1: Random Order (RO)



Relaxation 1: Random Order (RO)

$\mathcal{S}$

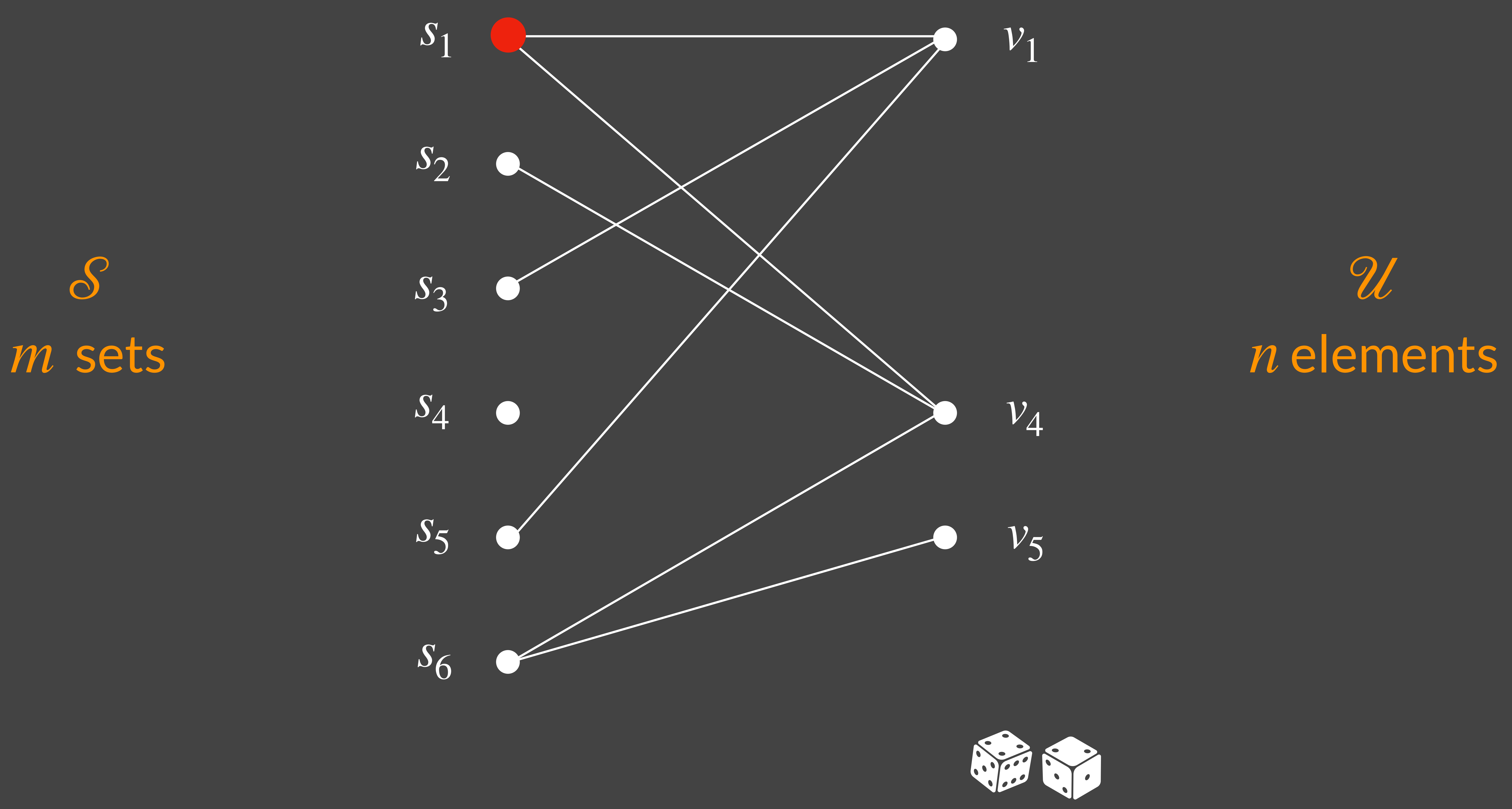
m sets



$\mathcal{U}$

n elements

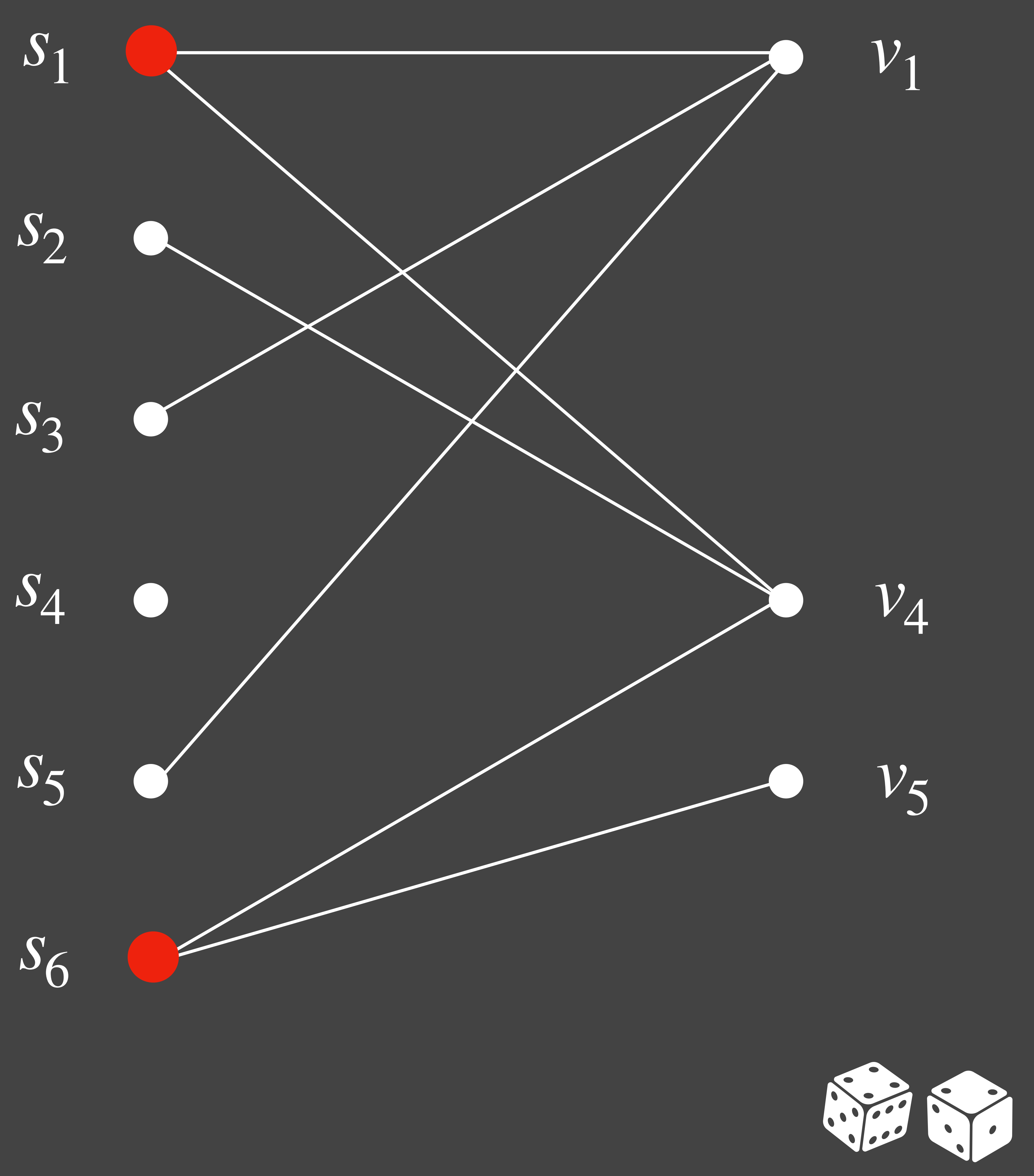
Relaxation 1: Random Order (RO)



Relaxation 1: Random Order (RO)

$\mathcal{S}$

m sets

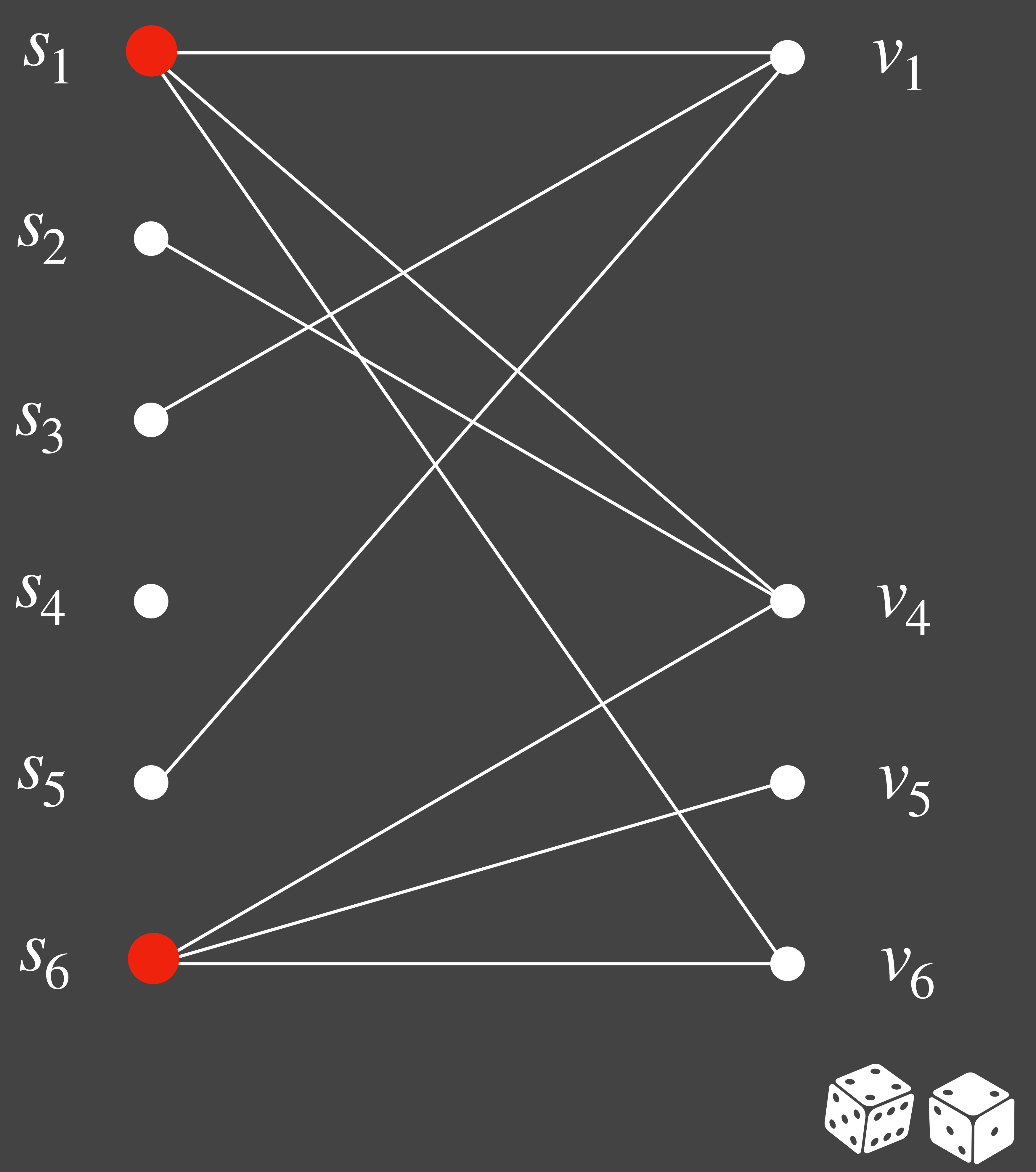


$\mathcal{U}$

n elements

Relaxation 1: Random Order (RO)

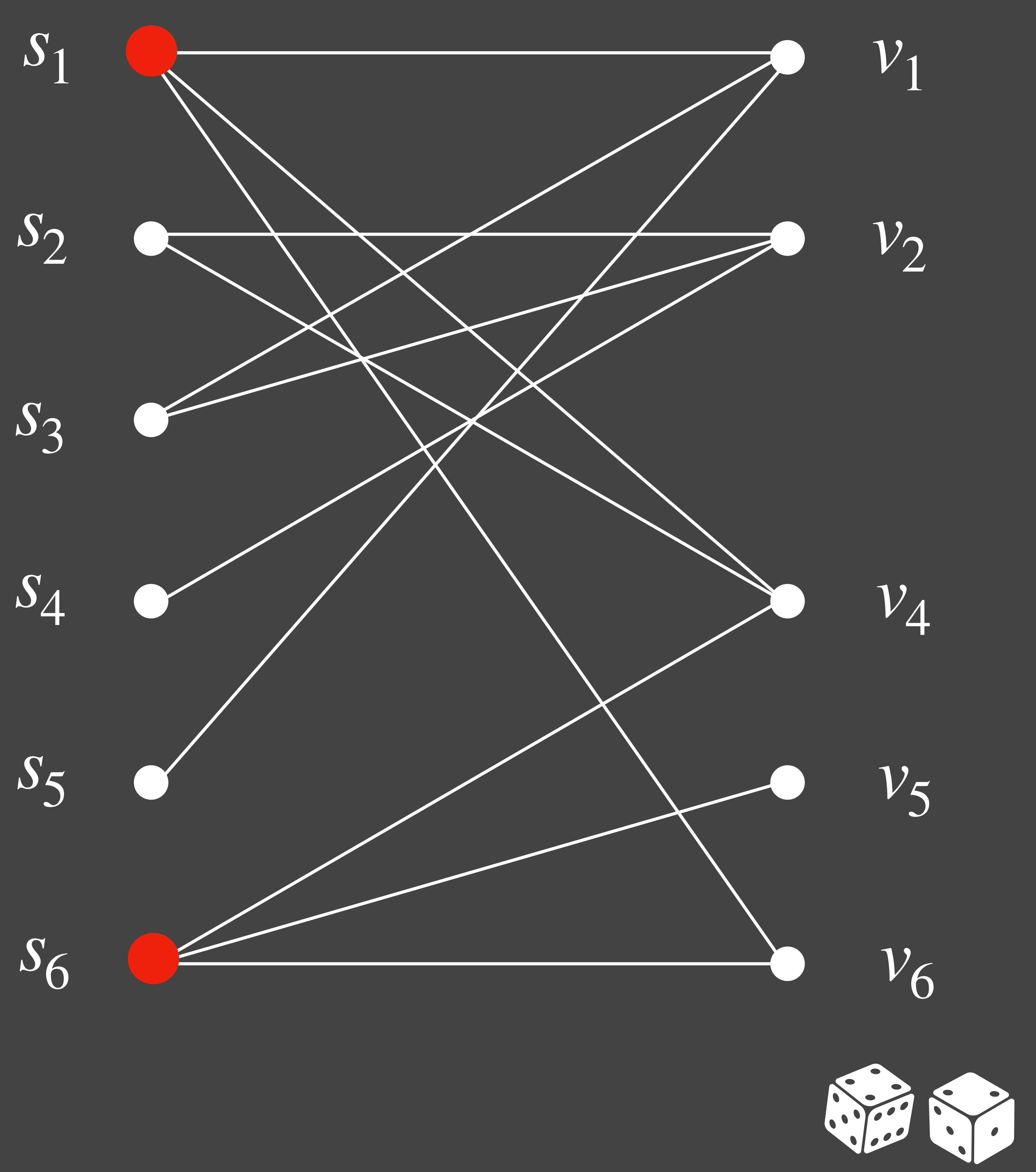
$\mathcal{S}$
m sets



$\mathcal{U}$
n elements

Relaxation 1: Random Order (RO)

$\mathcal{S}$
m sets

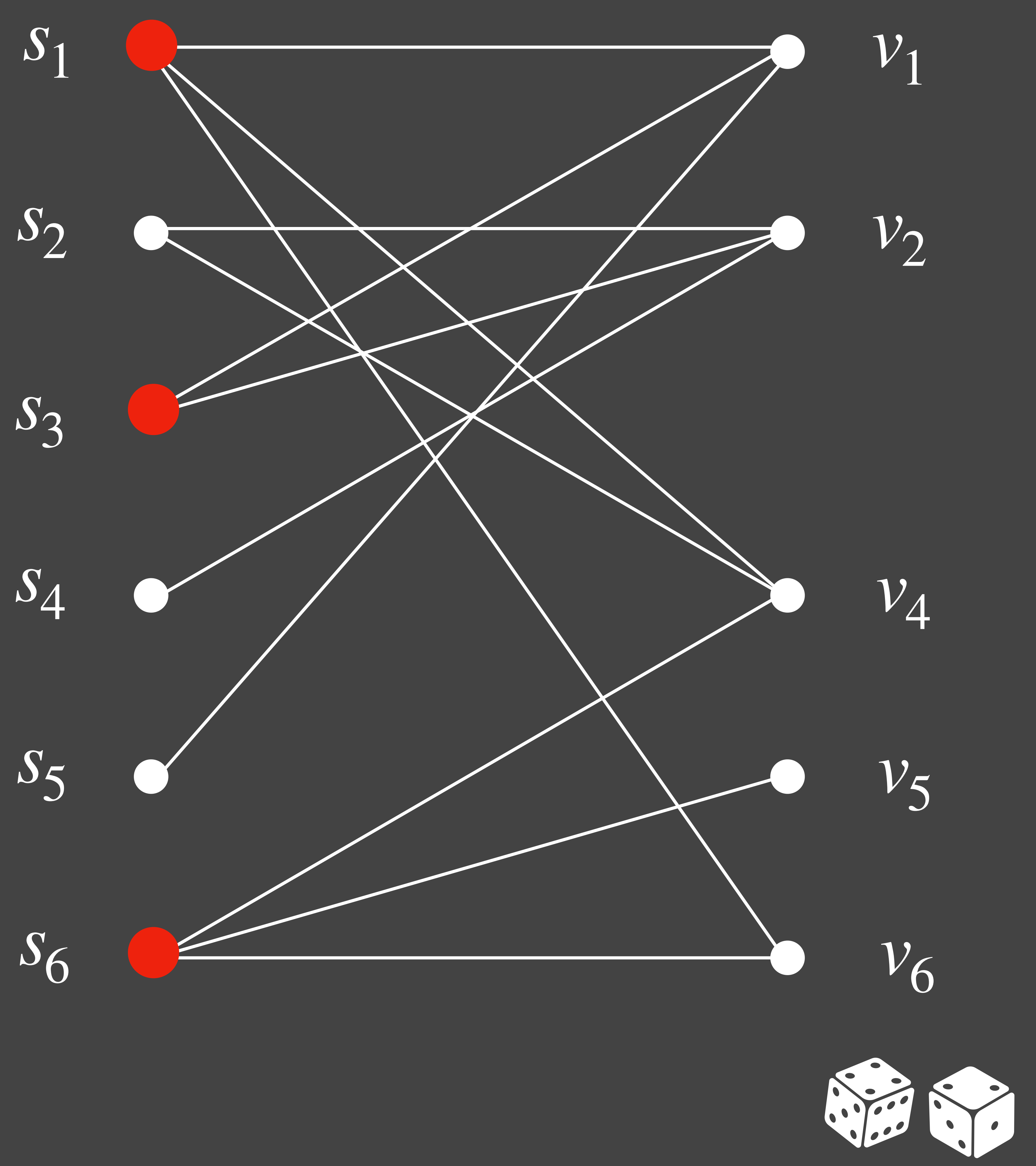


$\mathcal{U}$
n elements

Relaxation 1: Random Order (RO)

$\mathcal{S}$

m sets



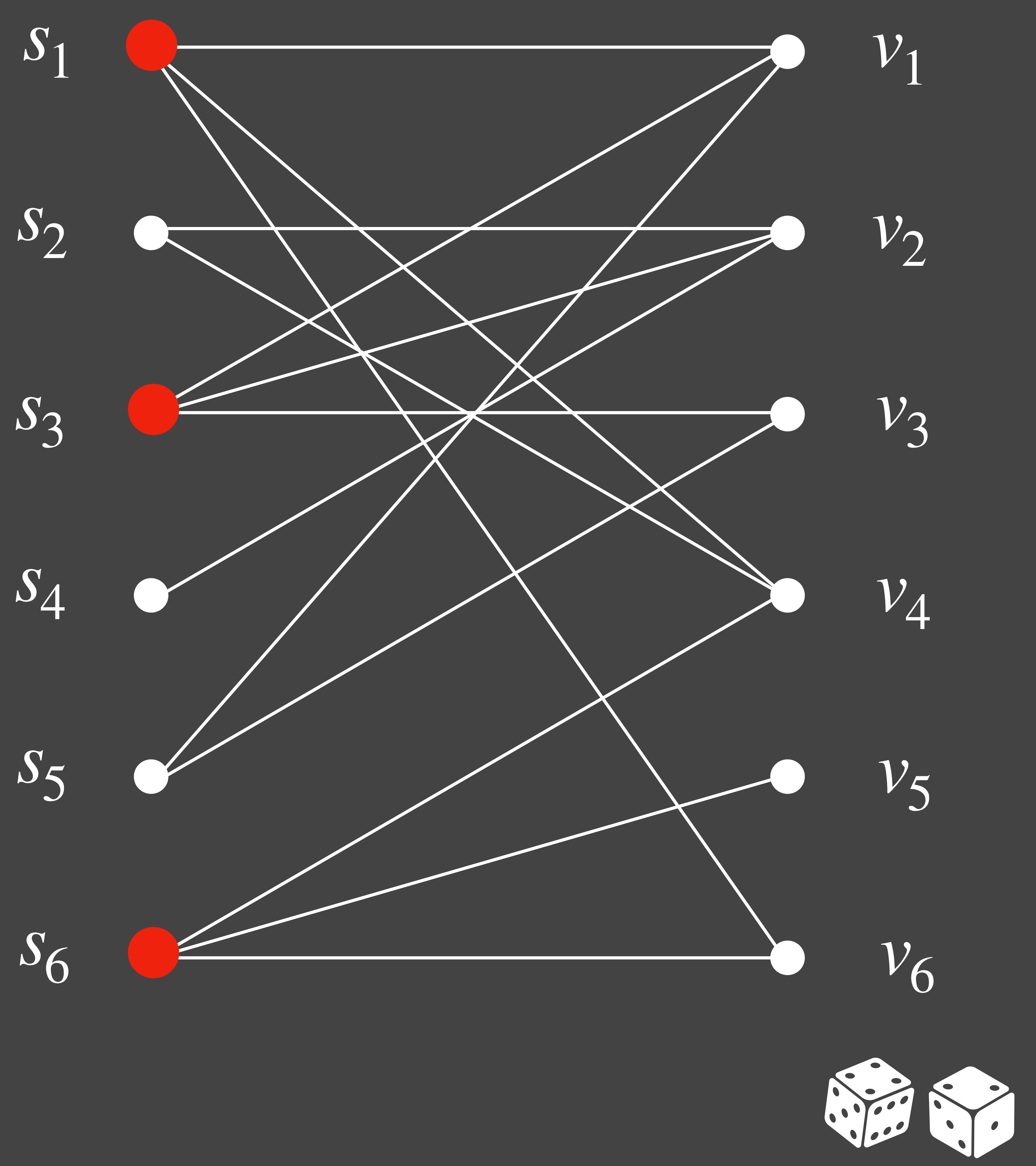
$\mathcal{U}$

n elements

Relaxation 1: Random Order (RO)

$\mathcal{S}$

m sets



$\mathcal{U}$

n elements

Relaxation 2: Random Instance

$\mathcal{S}$

m sets

s_1

s_2

s_3

s_4

s_5

s_6

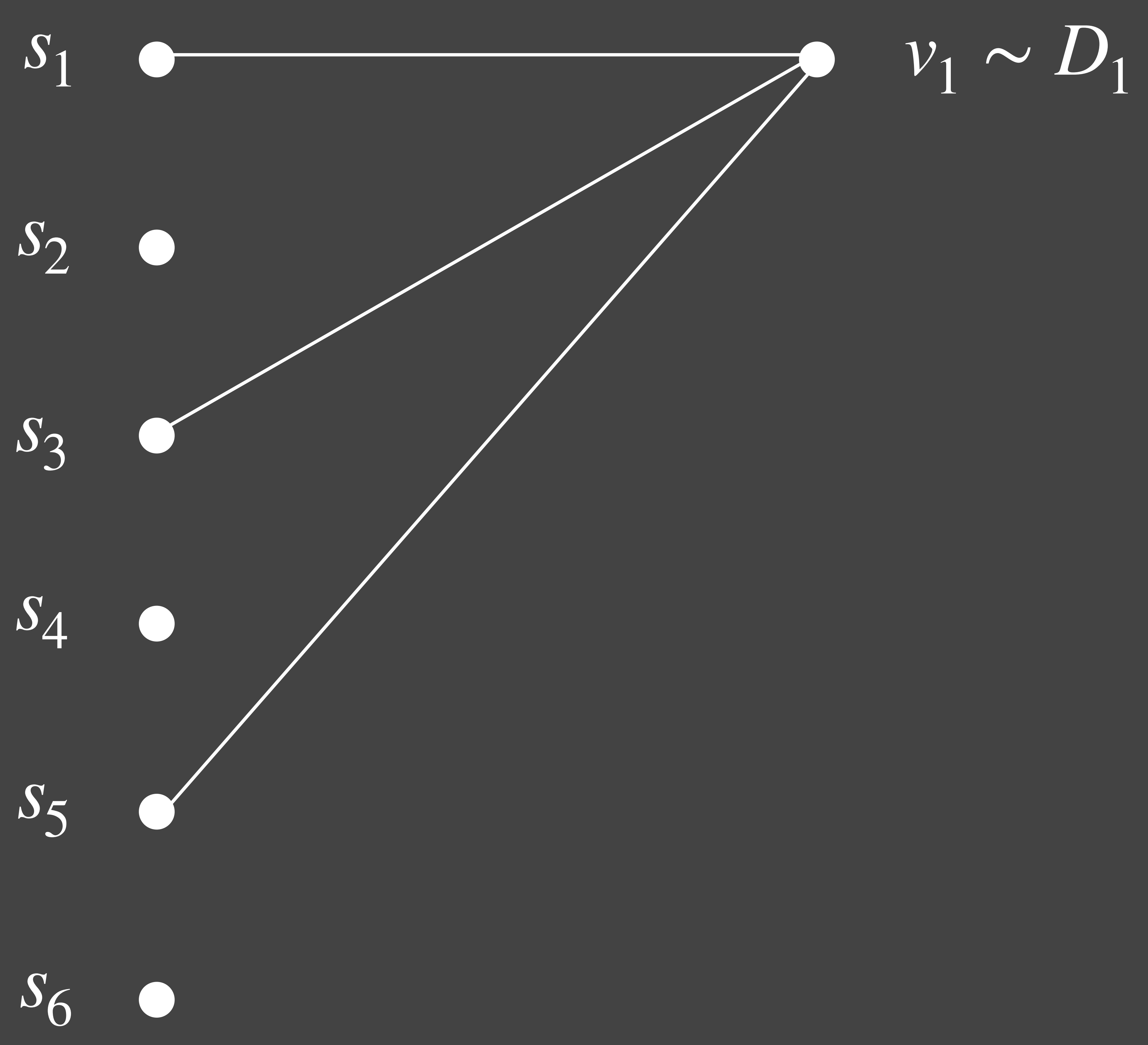
$\mathcal{U}$

n elements

Relaxation 2: Random Instance

$\mathcal{S}$

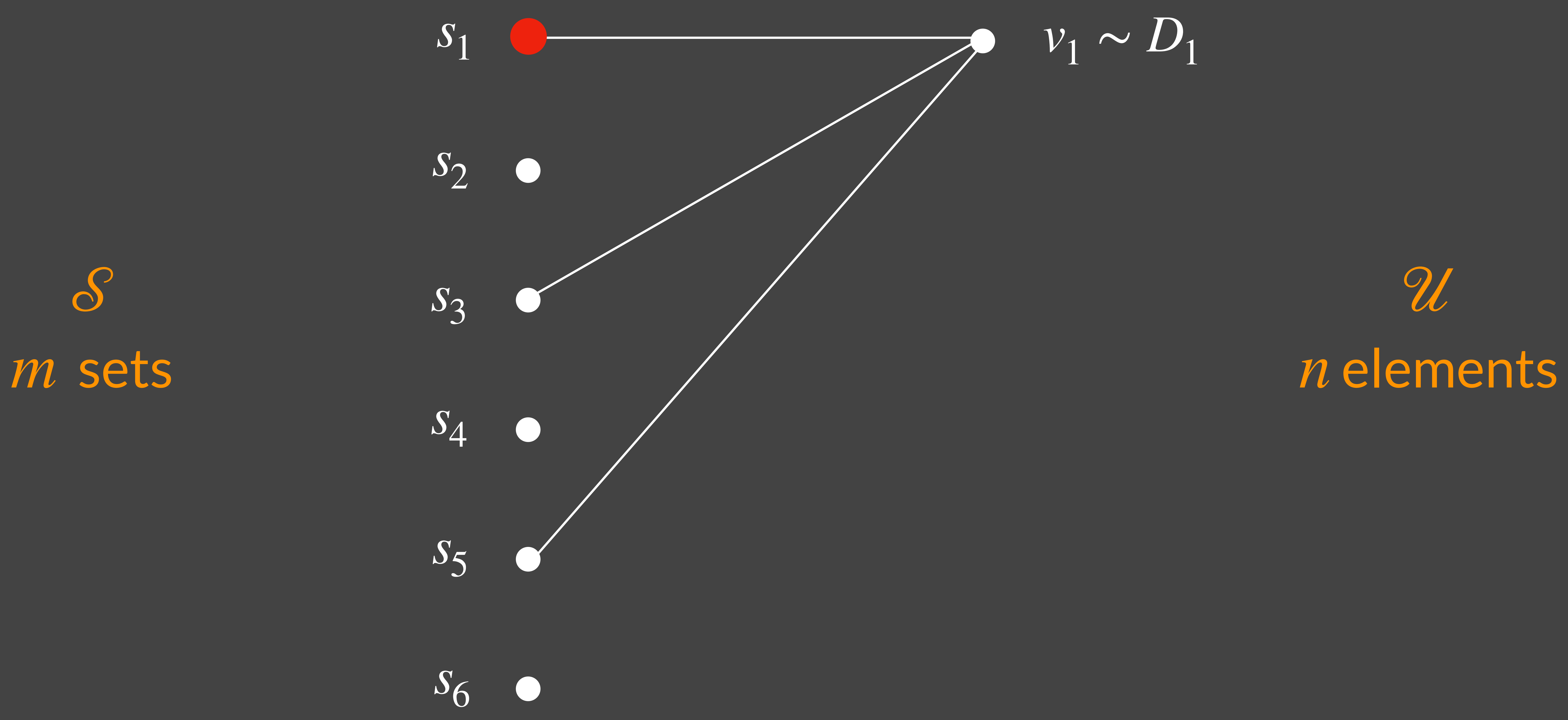
m sets



$\mathcal{U}$

n elements

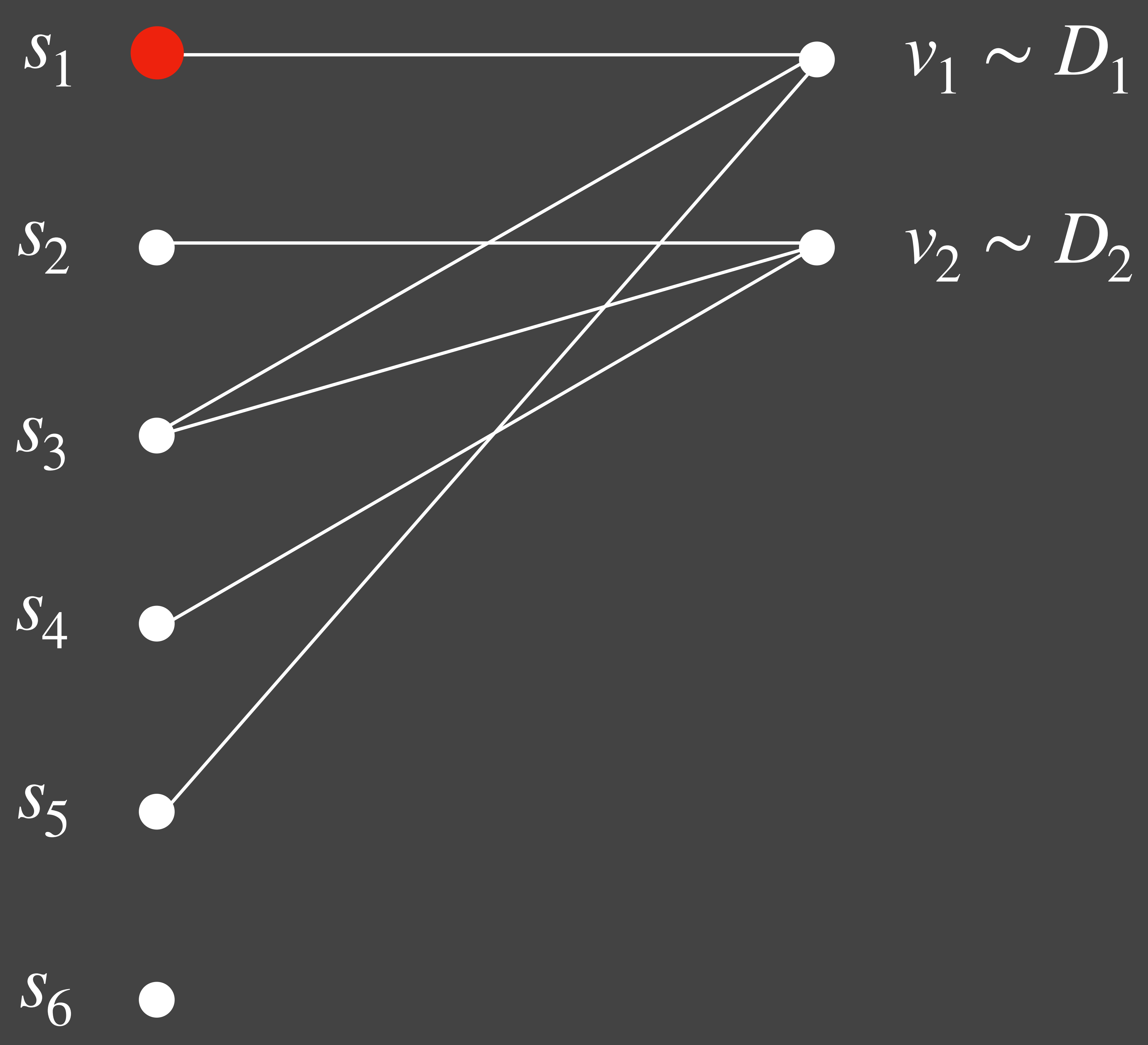
Relaxation 2: Random Instance



Relaxation 2: Random Instance

$\mathcal{S}$

m sets

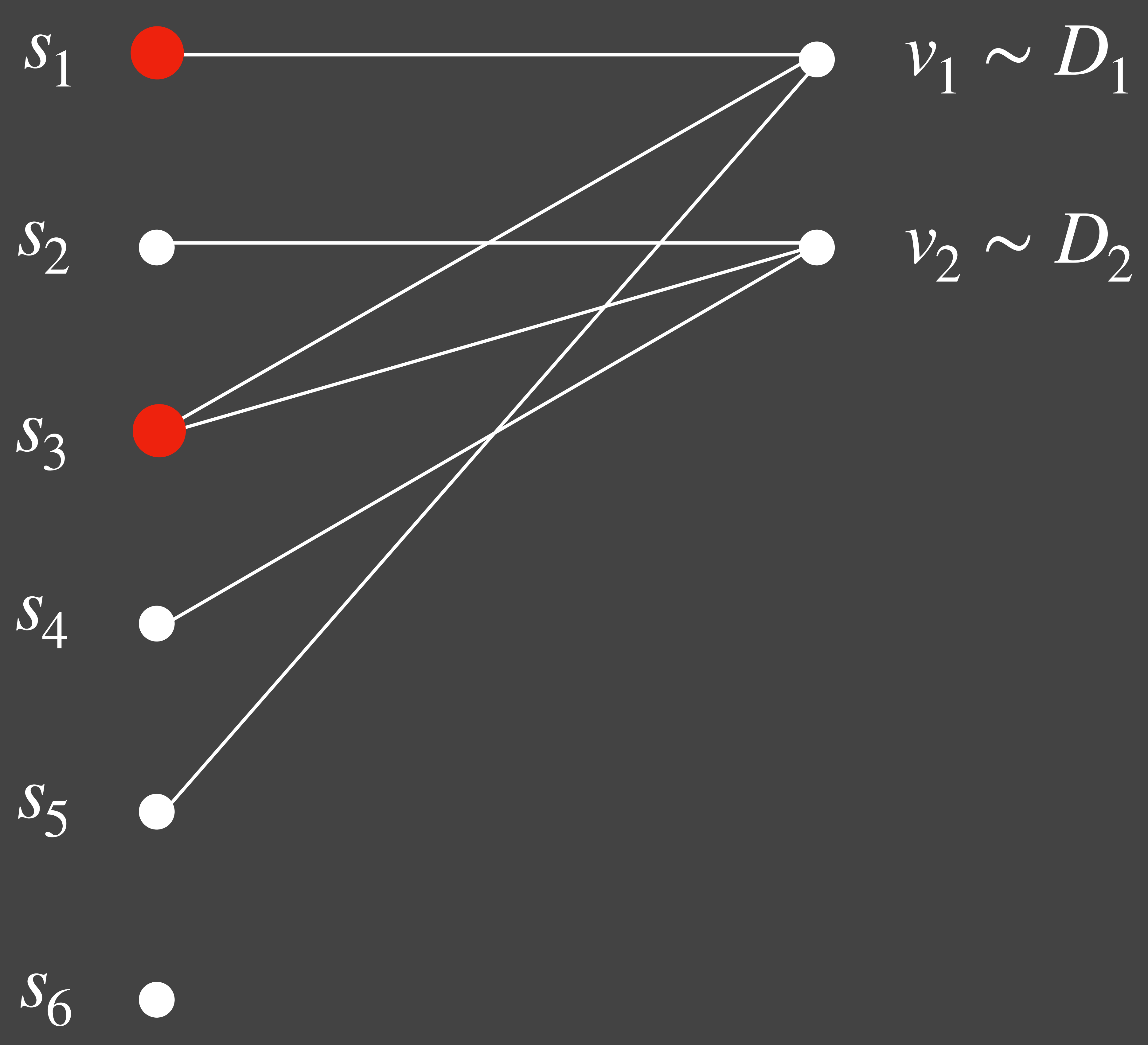


$\mathcal{U}$

n elements

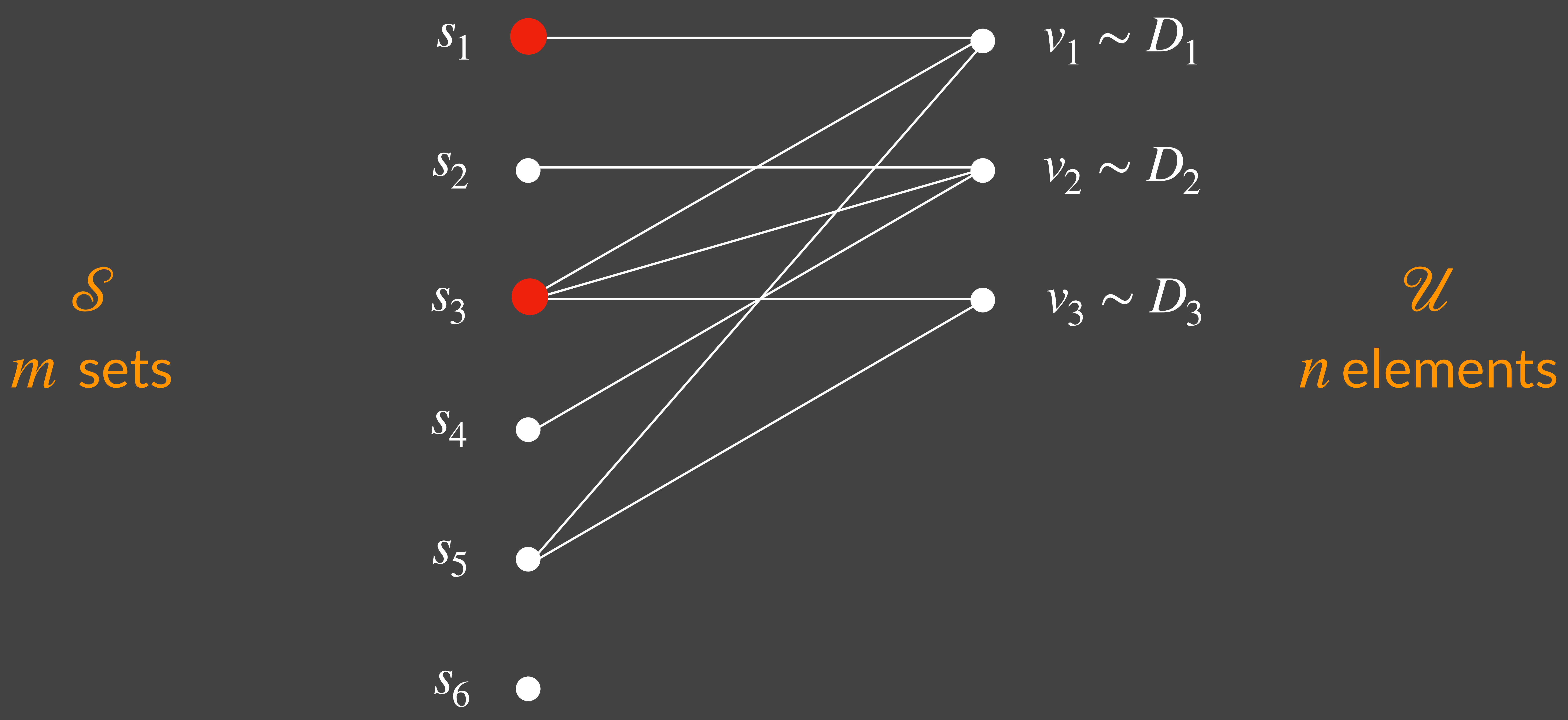
Relaxation 2: Random Instance

$\mathcal{S}$
m sets



$\mathcal{U}$
n elements

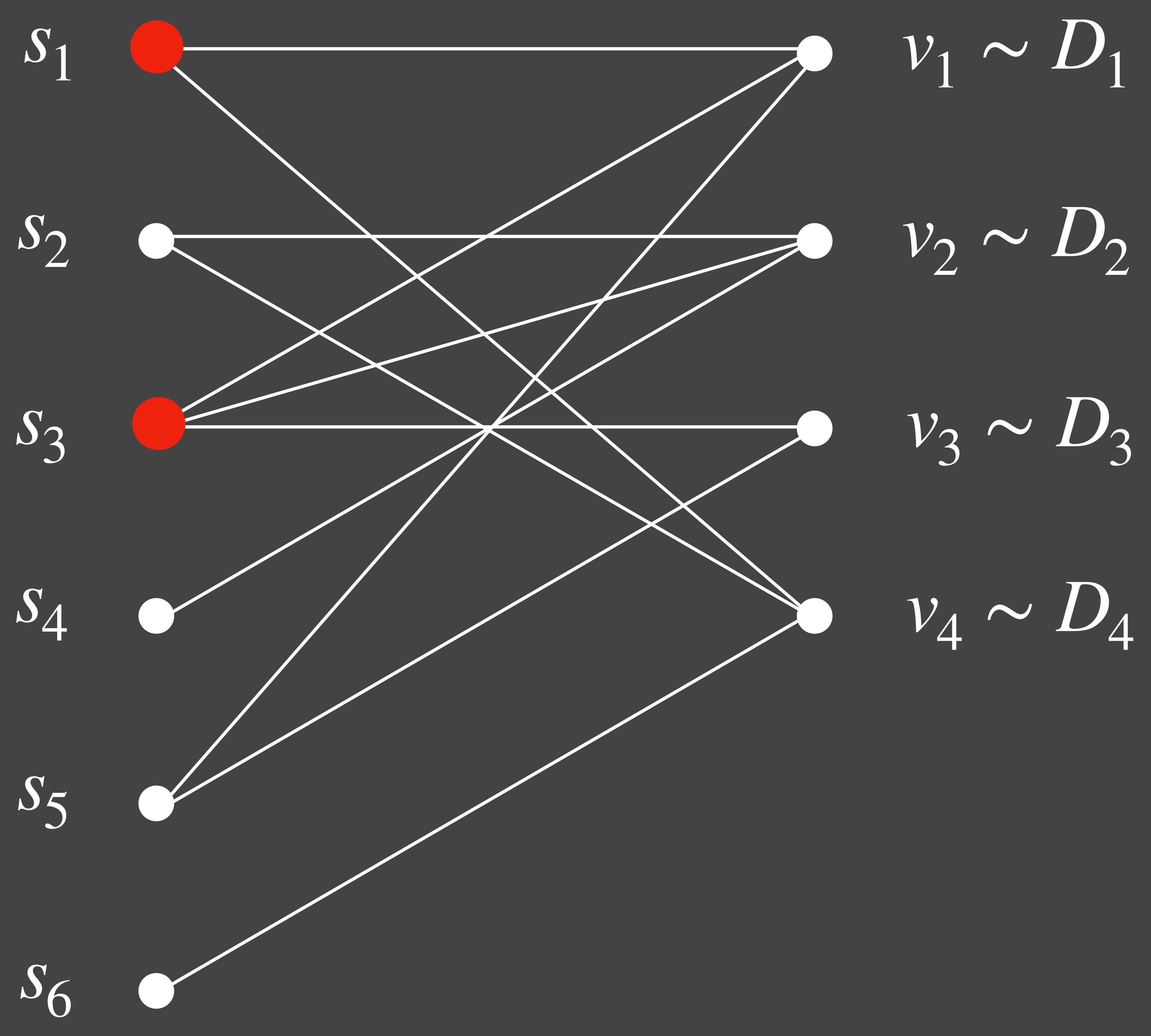
Relaxation 2: Random Instance



Relaxation 2: Random Instance

$\mathcal{S}$

m sets



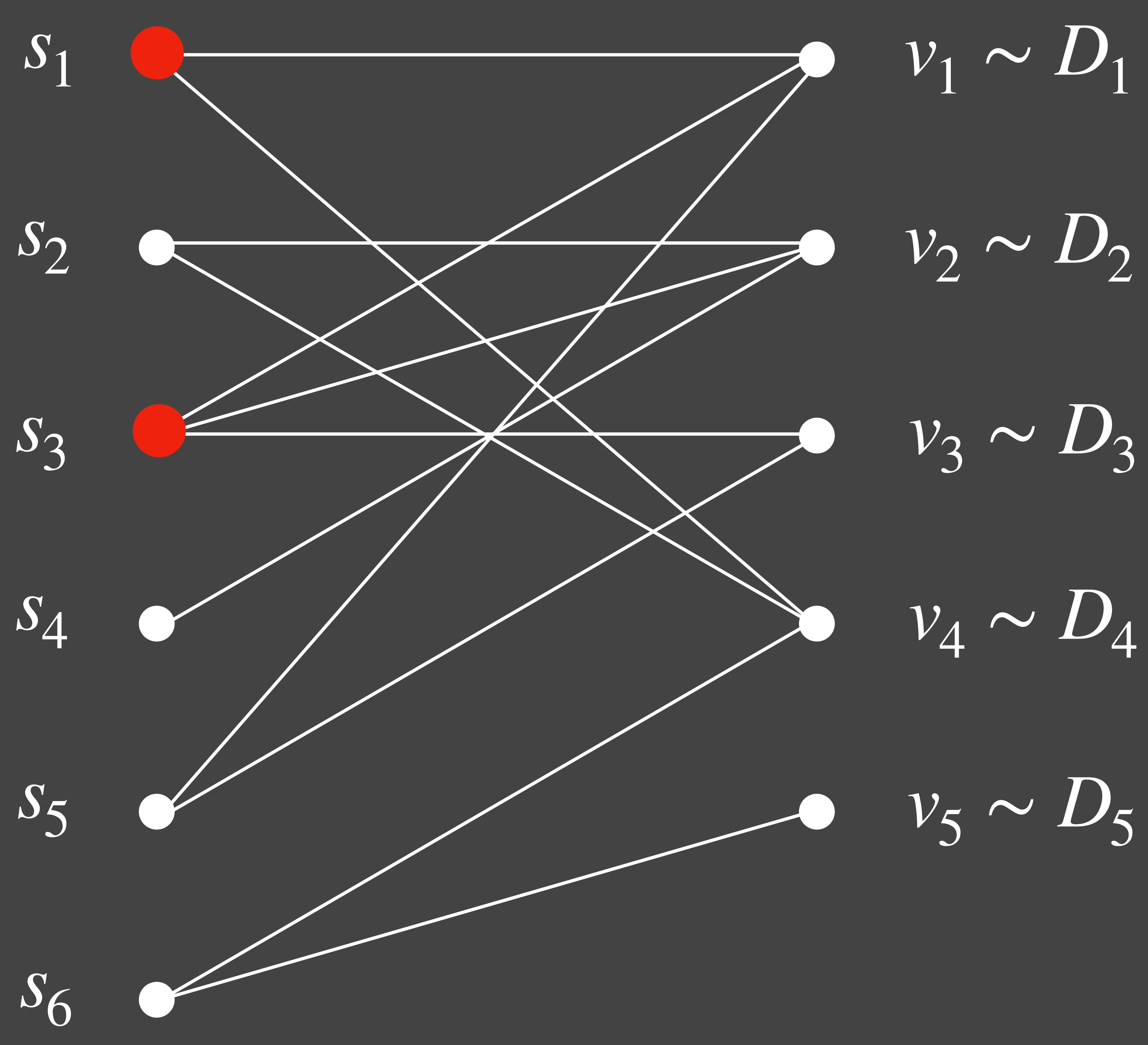
$\mathcal{U}$

n elements

Relaxation 2: Random Instance

$\mathcal{S}$

m sets



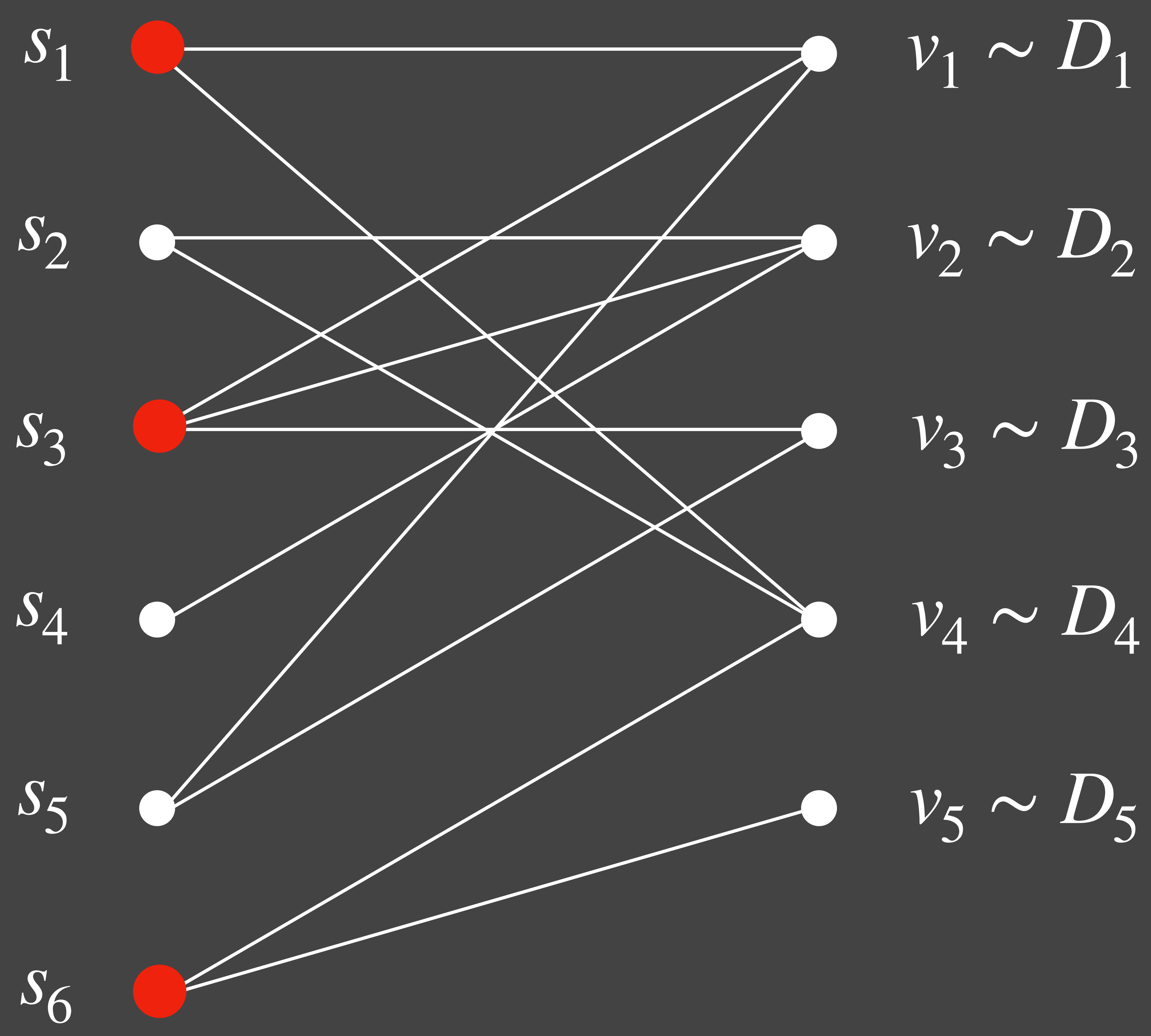
$\mathcal{U}$

n elements

Relaxation 2: Random Instance

$\mathcal{S}$

m sets



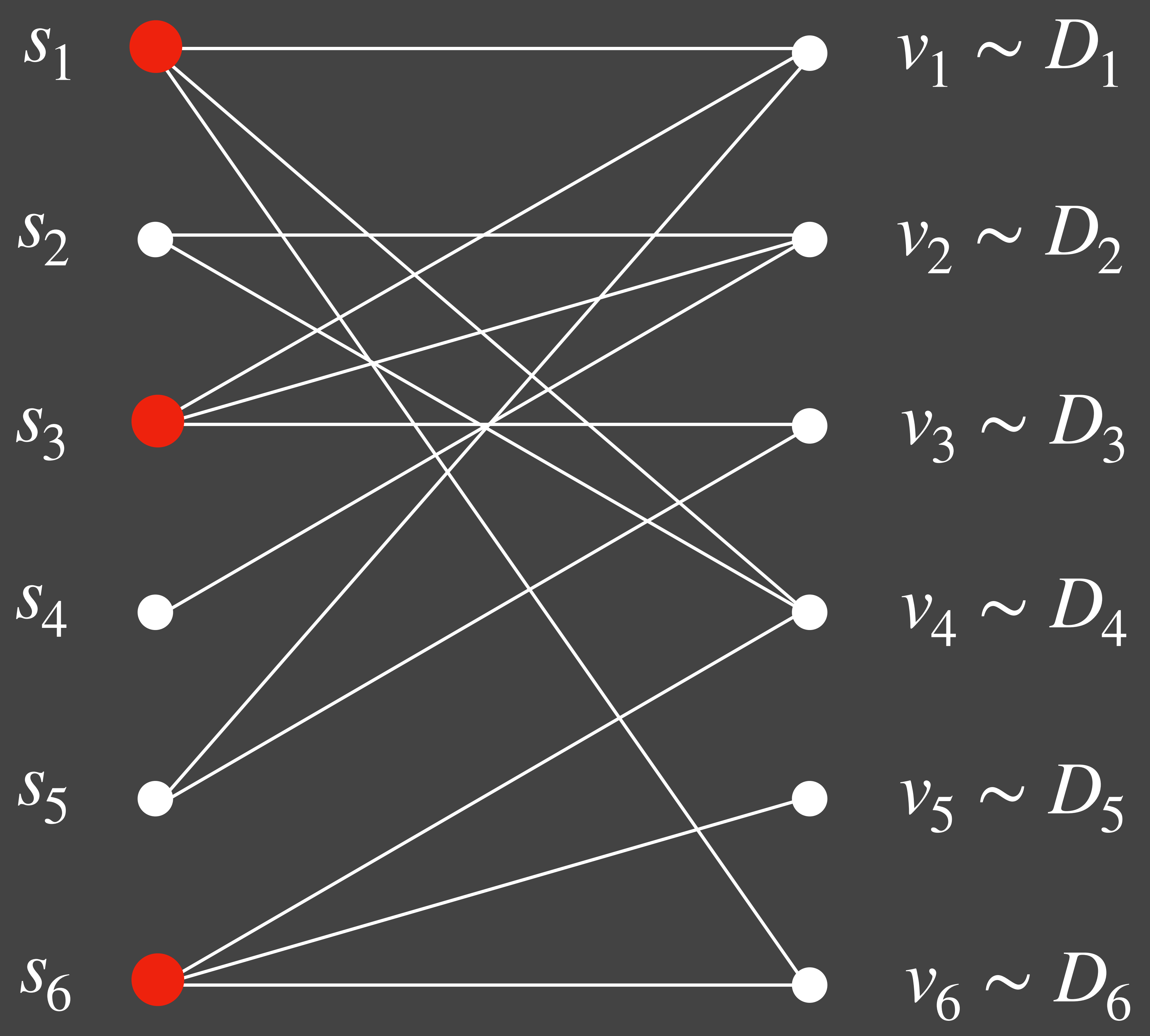
$\mathcal{U}$

n elements

Relaxation 2: Random Instance

$\mathcal{S}$

m sets



$\mathcal{U}$

n elements

The Landscape

$m =$ # sets
$n =$ # elements

The Landscape

$m =$ # sets
$n =$ # elements

Instance

Arrival Order	Random	Adversarial
Random		
Adversarial		O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

The Landscape

$m =$ # sets
$n =$ # elements

Instance

Arrival Order	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	
Adversarial		O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

The Landscape

m = # sets
n = # elements

Instance

Arrival Order	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	Secretary
Adversarial		O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

The Landscape

m = # sets
n = # elements

Arrival Order \ Instance	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	Secretary
Adversarial	Prophet	O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

The Landscape

m = # sets
n = # elements

Instance

Arrival Order	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	Secretary
Adversarial	Prophet	O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

Some reasons to believe $o(\log n \log m)$ not possible...

The Landscape

Instance

Arrival Order	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	Secretary: $\boxed{\text{O(log mn) Our work}}$
Adversarial	Prophet: $\boxed{}$	O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

$m =$ # sets
$n =$ # elements

Theorem [Gupta Kehne **L.** **FOCS** 21]:

There is a poly time algorithm for ***secretary*** *Covering IPs with competitive ratio* $O(\log mn)$.

The Landscape

Instance

Arrival Order

	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	O(log mn) Our work
Adversarial		O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

Secretary

Prophet

m = # sets
n = # elements

Theorem [Gupta Kehne **L.** **FOCS** 21]:

There is a poly time algorithm for ***secretary*** *Covering IPs with competitive ratio* $O(\log mn)$.

New algorithm! We show how to learn distribution & solve at same time.

The Landscape

$m =$ # sets
$n =$ # elements

Instance

Arrival Order	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	Secretary: O(log mn) Our work
Adversarial	Prophet: New! O(log mn) Our work	O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

Theorem [Gupta Kehne **L.** FOCS 21]:

There is a poly time algorithm for ***secretary*** *Covering IPs with competitive ratio* $O(\log mn)$.

Theorem [Gupta Kehne **L.** 22]:

There is a poly time algorithm for ***prophet*** *Covering IPs with competitive ratio* $O(\log mn)$.

The Landscape

$m = $ # sets
$n = $ # elements

Instance

Arrival Order	Random	Adversarial
Random	O(log(m [support size])) [Gupta Grandoni Leonardi Miettinen Sankowski Singh 08]	Secretary: O(log mn) Our work
Adversarial	Prophet: New! O(log mn) Our work	O(log n log m) [Alon+ 03] [Buchbinder Naor 09]

Bonus Properties!

1. Only need single sample from each D_i!
2. Universal! Gives sample complexity bound $O(n)$.

Theorem [Gupta Kehne **L.** FOCS 21]:

There is a poly time algorithm for ***secretary*** *Covering IPs with competitive ratio* $O(\log mn)$.

Theorem [Gupta Kehne **L.** 22]:

There is a poly time algorithm for ***prophet*** *Covering IPs with competitive ratio* $O(\log mn)$.

Talk Outline

➡ Intro

Secretary

LearnOrCover in Exponential Time

LearnOrCover in Poly Time

(Single Sample) Prophet

Conclusion & Extensions

Talk Outline

Intro

➔ Secretary

- LearnOrCover in Exponential Time
- LearnOrCover in Poly Time

(Single Sample) Prophet

Conclusion & Extensions

Set Cover via Random Rounding

Set Cover via Random Rounding

2 Stage algorithm!

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

(II) Round.

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\min \sum_S x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

(II) Round.

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\min \sum_{S} x_S$$

$$\forall v \in \mathscr{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathscr{S} : \quad x_S \geq 0$$

This is relaxation, so $c(x) \leq c(\text{OPT})$.

(II) Round.

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

This is relaxation, so $c(x) \leq c(\text{OPT})$.

(II) Round.

Buy S with probability x_S.

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

This is relaxation, so $c(x) \leq c(\text{OPT})$.

(II) Round.

Buy S with probability x_S.

Expected cost is $c(x)$!

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\boxed{\begin{array}{ll} & \min \sum_{S} x_S \\ \forall v \in \mathcal{U}: & \sum_{S \ni v} x_S \geq 1 \\ \forall S \in \mathcal{S}: & x_S \geq 0 \end{array}}$$

This is relaxation, so $c(x) \leq c(\text{OPT})$.

(II) Round.

Buy S with probability x_S.

Expected cost is $c(x)$!

Can show $\forall v \in \mathcal{U}$, covered with constant prob.

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\min \sum_{S} x_S$$

$$\forall v \in \mathscr{U}: \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathscr{S}: \quad x_S \geq 0$$

This is relaxation, so $c(x) \leq c(\text{OPT})$.

(II) Round.

Buy S with probability x_S.

Expected cost is $c(x)$!

Can show $\forall v \in \mathscr{U}$, covered with constant prob.

Repeat $O(\log n)$ times, union bound.

Set Cover via Random Rounding

2 Stage algorithm!

(I) Solve LP.

$$\min \sum_S x_S$$

$$\forall v \in \mathscr{U}: \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathscr{S}: \quad x_S \geq 0$$

This is relaxation, so $c(x) \leq c(\mathrm{OPT})$.

(II) Round.

Buy S with probability x_S.

Expected cost is $c(x)$!

Can show $\forall v \in \mathscr{U}$, covered with constant prob.

Repeat $O(\log n)$ times, union bound.

Expected Cost: $O(\log n) \cdot \mathrm{OPT}$

How [Alon+ 03] works

How [Alon+ 03] works

Same 2 Stages!

How [Alon+ 03] works

Same 2 Stages!

(I) Solve LP Online.

(II) Round Online.

How [Alon+ 03] works

Same 2 Stages!

(I) Solve LP Online.

$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U}: \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S}: \quad x_S \geq 0$$

(II) Round Online.

How [Alon+ 03] works

Same 2 Stages!

(I) Solve LP Online.

$$\min \sum_S x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

Can guarantee x is $O(\log m)$-apx, and only increases ***monotonically.***

(II) Round Online.

How [Alon+ 03] works

Same 2 Stages!

(I) Solve LP Online.

$$\min \sum_S x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

Can guarantee x is $O(\log m)$-apx, and only increases ***monotonically.***

(II) Round Online.

How [Alon+ 03] works

Same 2 Stages!

(I) Solve LP Online.

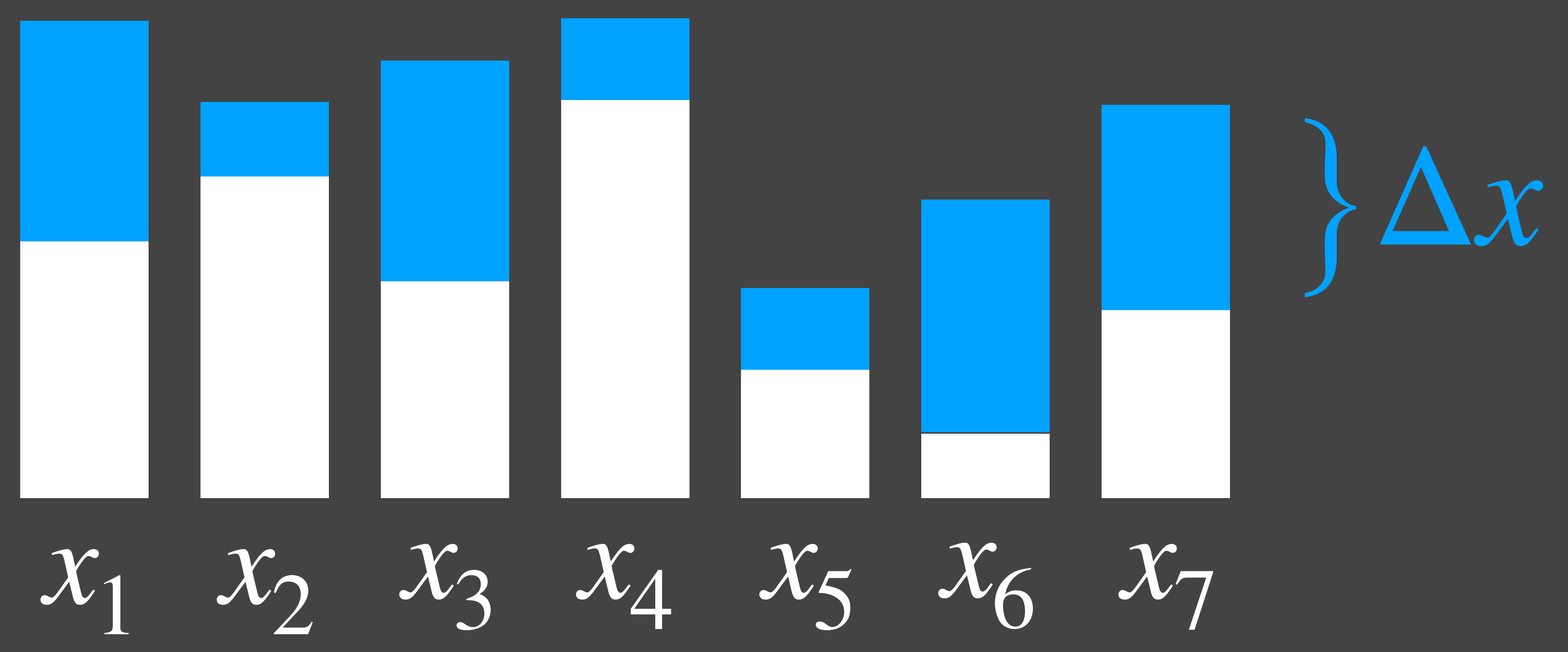
$$\min \sum_S x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

Can guarantee x is $O(\log m)$-apx, and only increases ***monotonically.***

(II) Round Online.



How [Alon+ 03] works

Same 2 Stages!

(I) Solve LP Online.

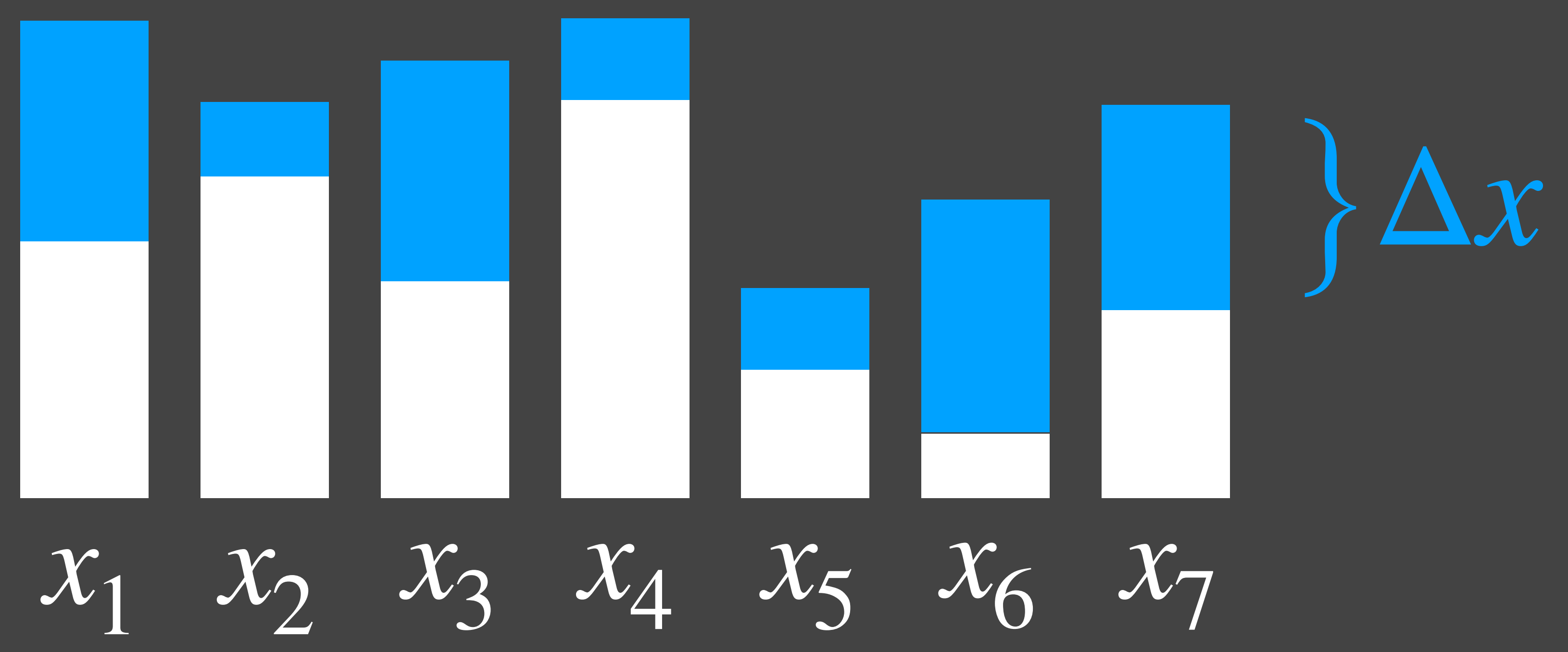
$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

Can guarantee x is $O(\log m)$-apx, and only increases ***monotonically.***

(II) Round Online.



Take S with prob. $\propto \Delta x_S$.

How [Alon+ 03] works

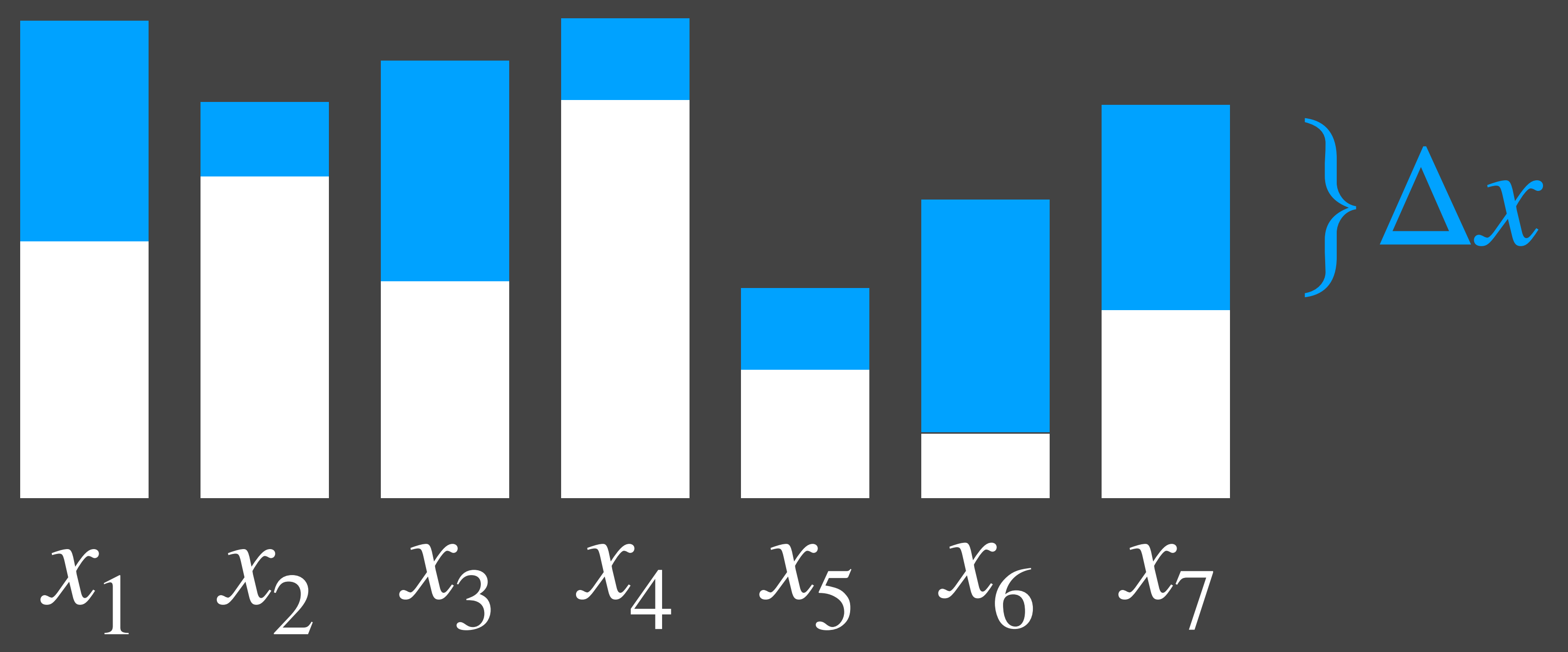
Same 2 Stages!

(I) Solve LP Online.

$$\min \sum_{S} x_S$$
$$\forall v \in \mathcal{U}: \quad \sum_{S \ni v} x_S \geq 1$$
$$\forall S \in \mathcal{S}: \quad x_S \geq 0$$

Can guarantee x is $O(\log m)$-apx, and only increases ***monotonically.***

(II) Round Online.



Take S with prob. $\propto \Delta x_S$.

Suffices to analyze ***offline*** rounding.
Repeat $\log n$ times, union bound.

How [Alon+ 03] works

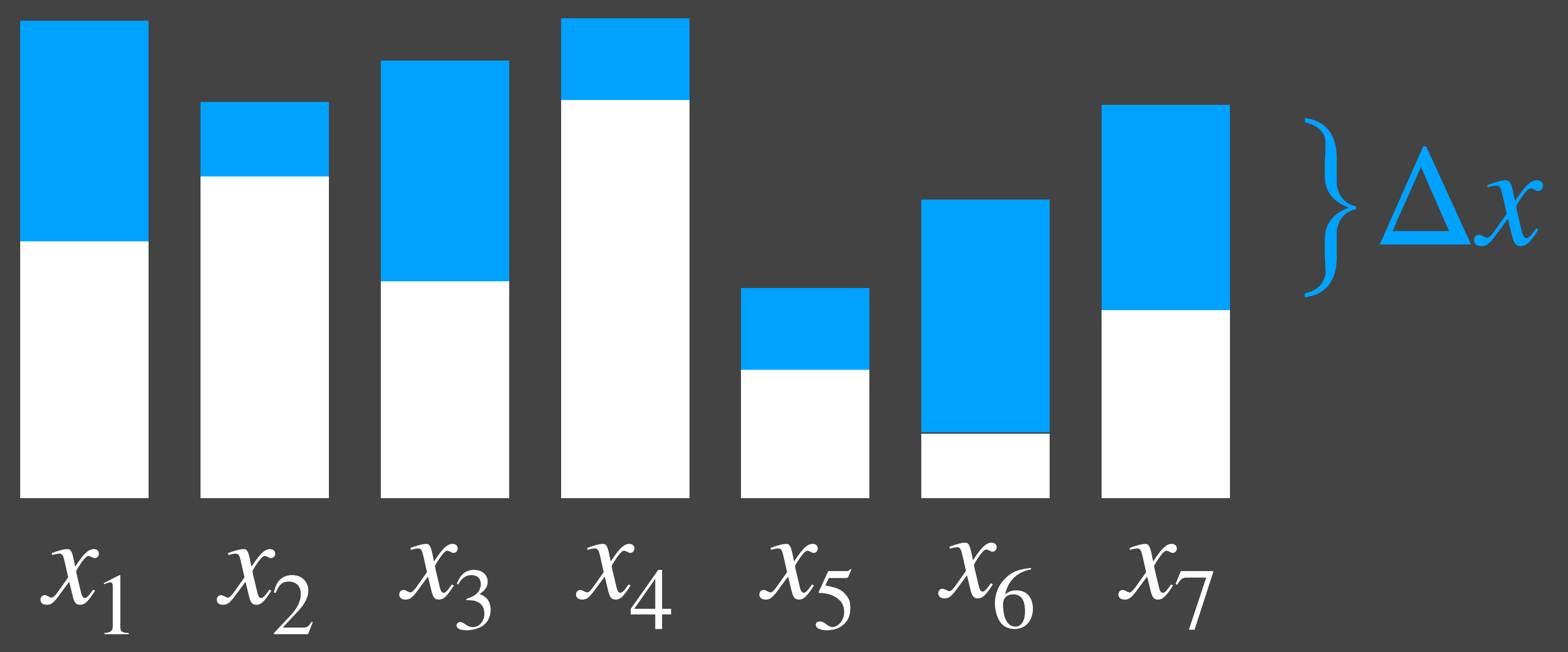
Same 2 Stages!

(I) Solve LP Online.

$$\min \sum_S x_S$$
$$\forall v \in \mathcal{U}: \quad \sum_{S \ni v} x_S \geq 1$$
$$\forall S \in \mathcal{S}: \quad x_S \geq 0$$

Can guarantee x is $O(\log m)$-apx, and only increases ***monotonically.***

(II) Round Online.

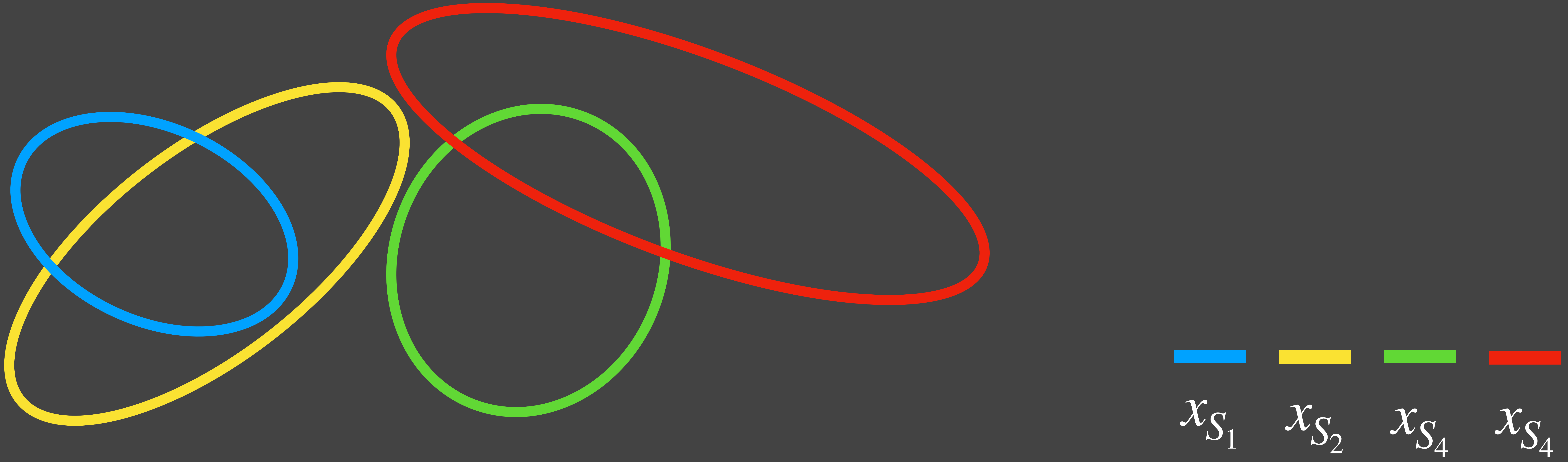


Take S with prob. $\propto \Delta x_S$.

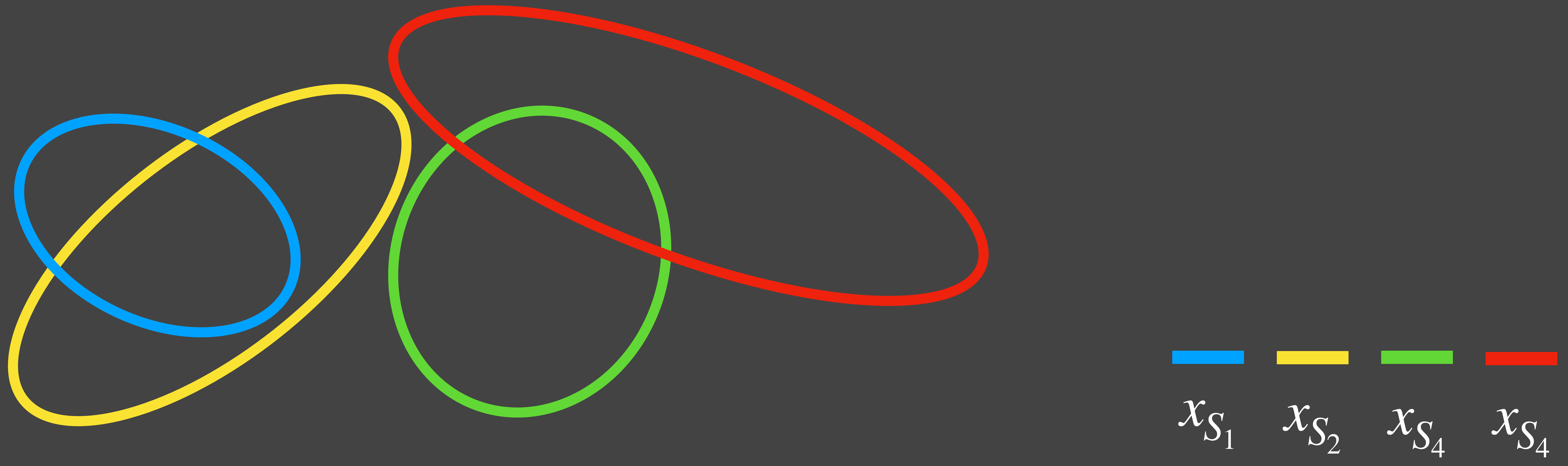
Suffices to analyze ***offline*** rounding.
Repeat $\log n$ times, union bound.

Expected Cost: $O(\log n \log m) \cdot$ OPT

Online LP Solver of [Alon+ 03]



Online LP Solver of [Alon+ 03]

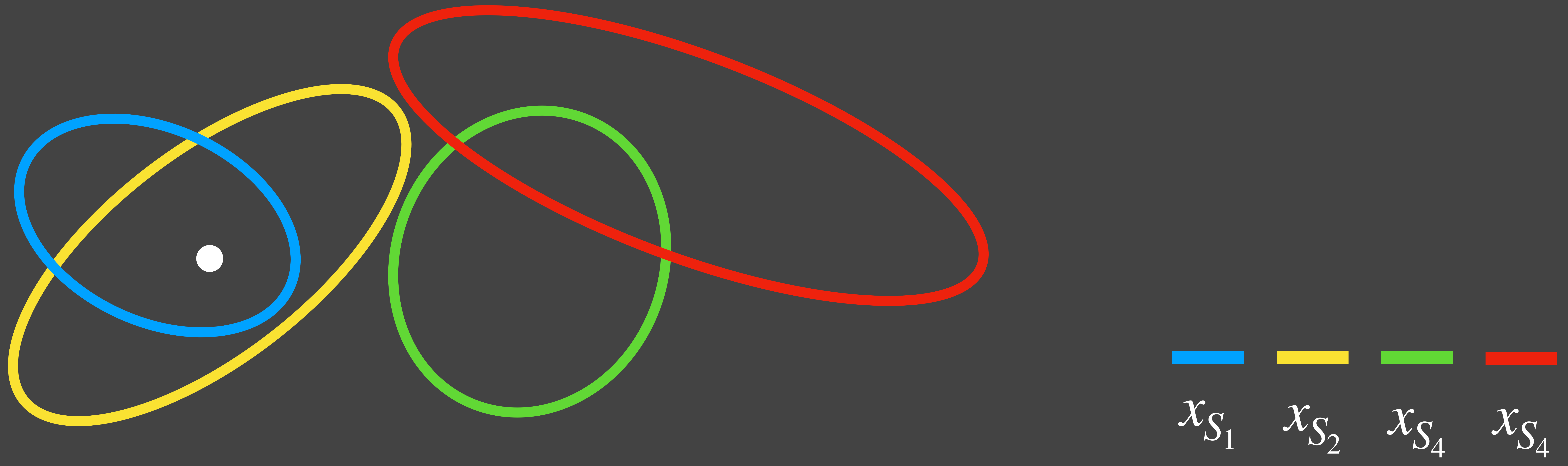


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

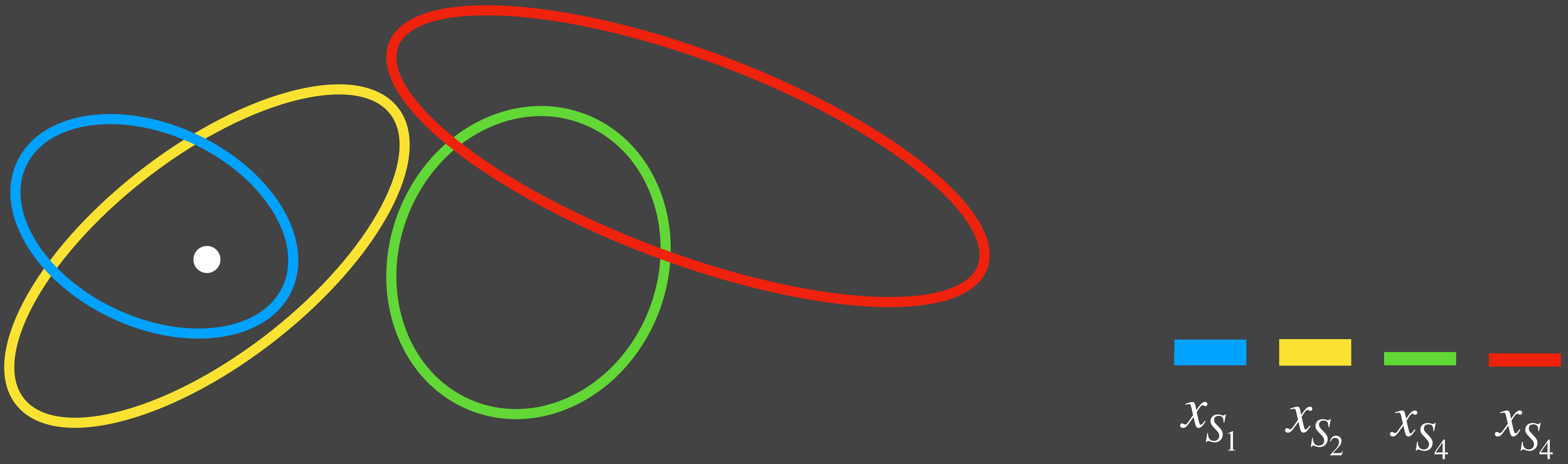


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

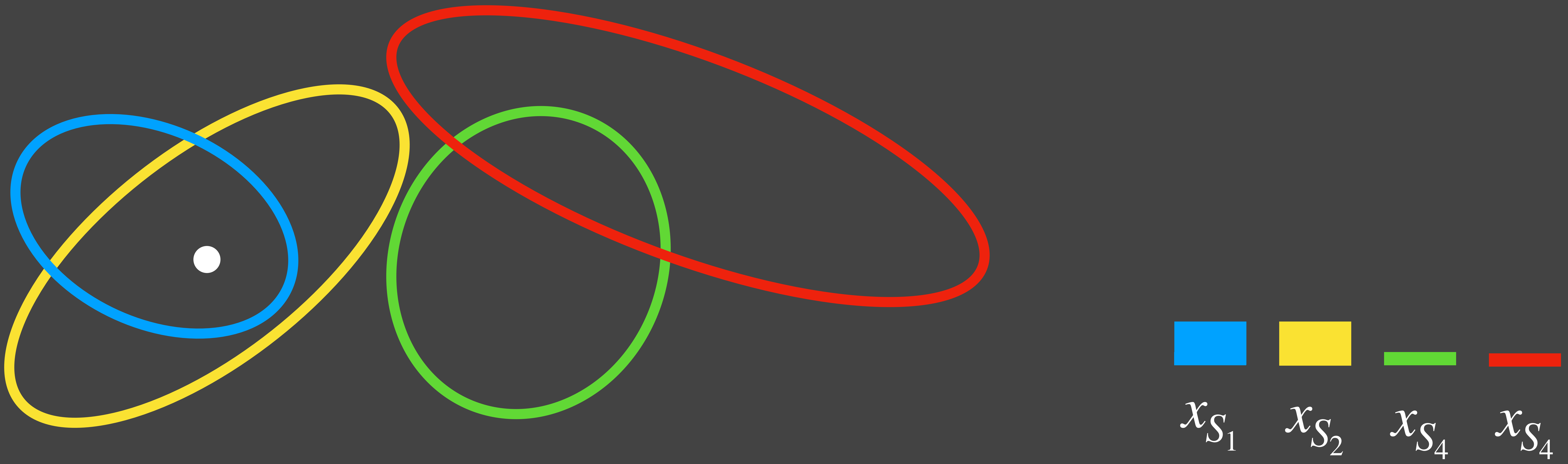


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

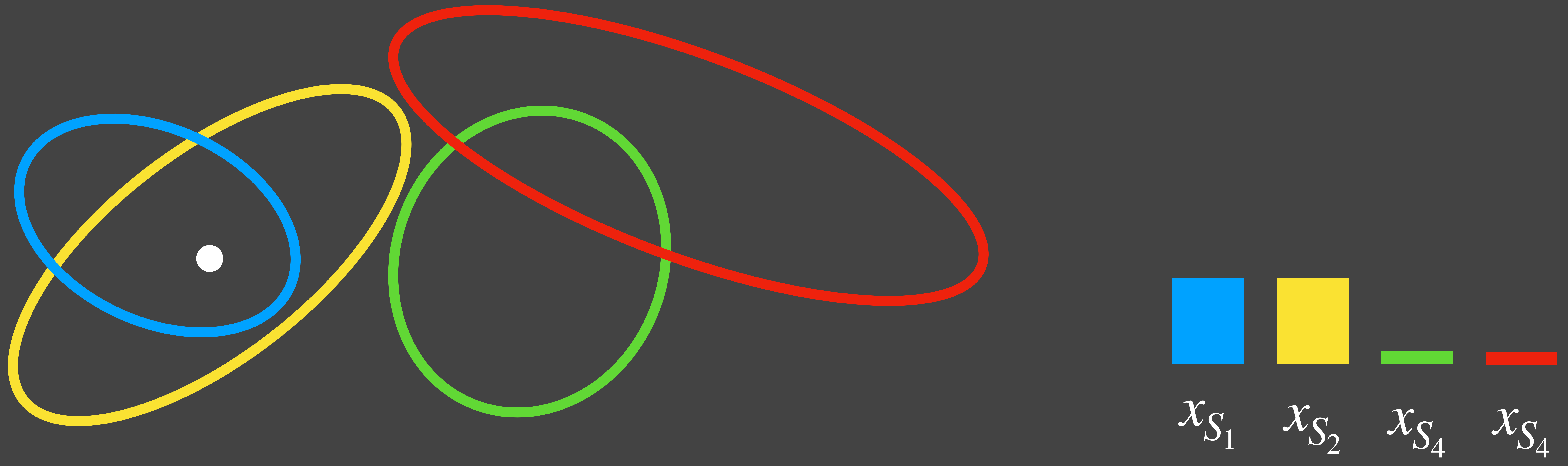


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

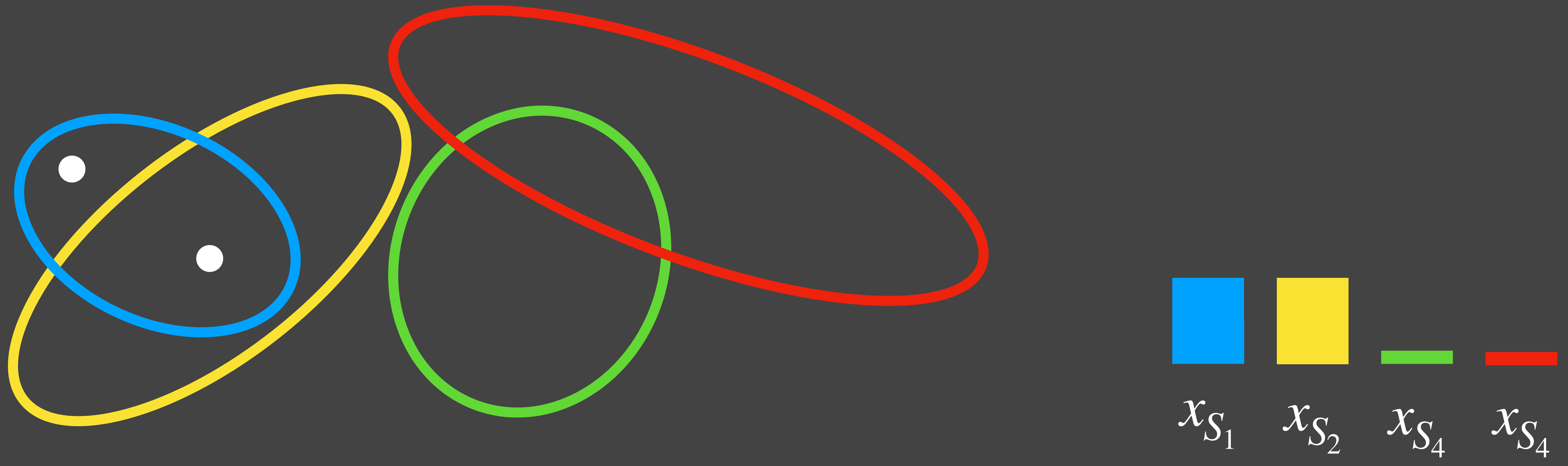


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

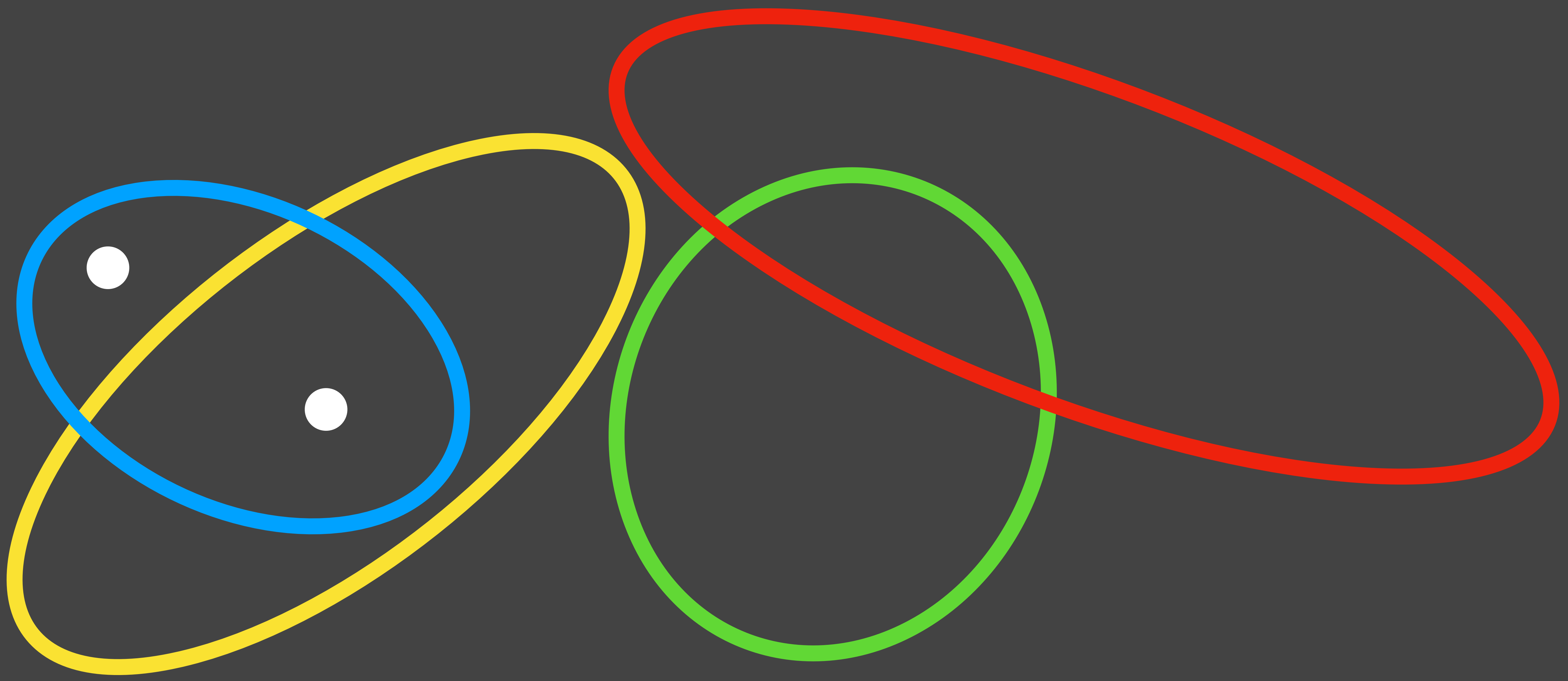


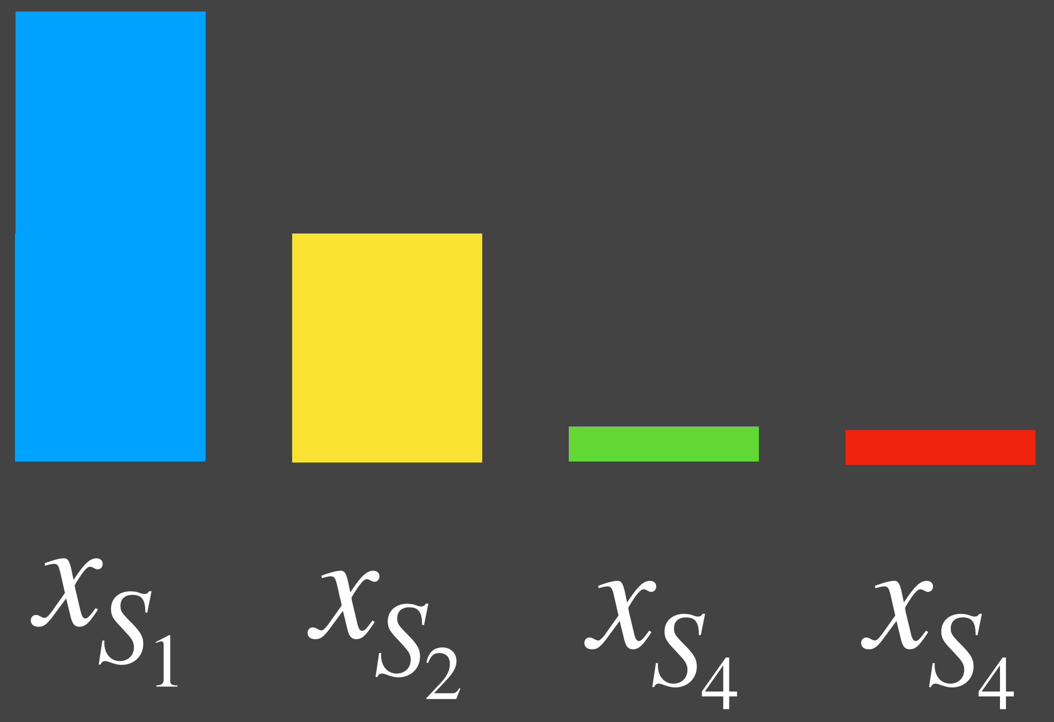
Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times\, 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]



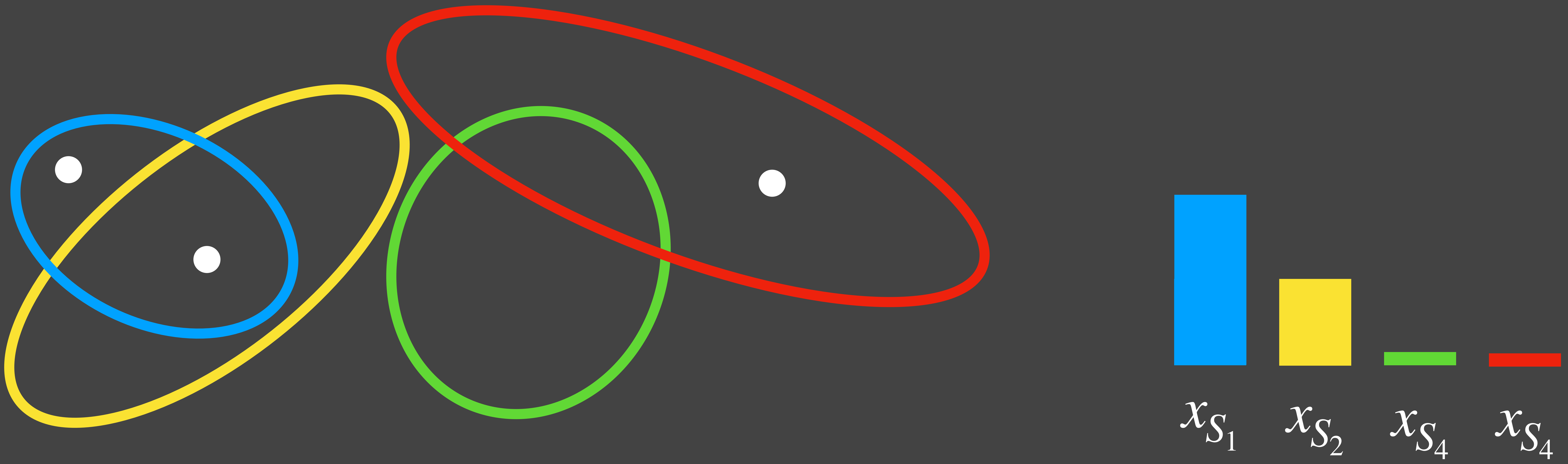


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

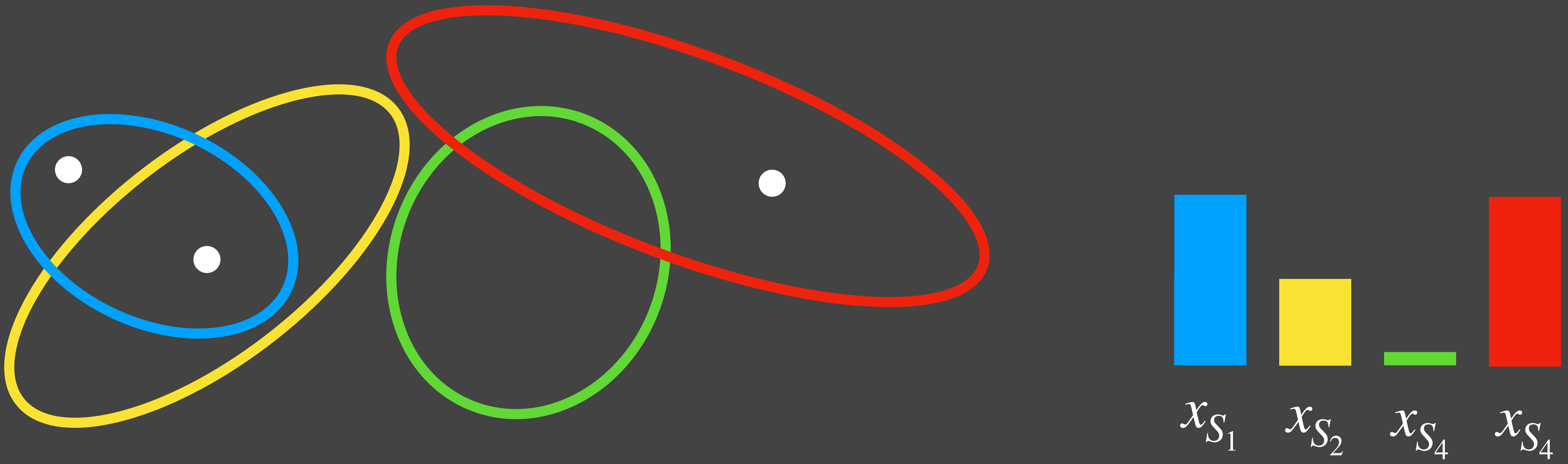


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

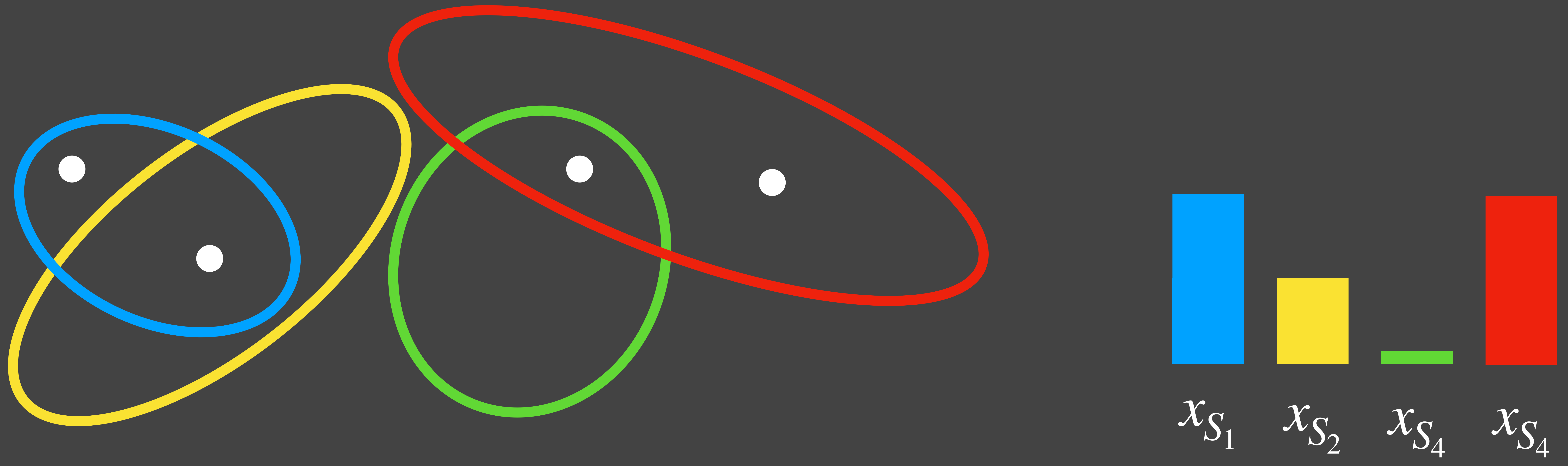


Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]



Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

(P)

$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

(D)

$$\max \sum_{v} y_v$$

$$\forall S \in \mathcal{S} : \quad \sum_{v \in S} y_v \leq 1$$

$$\forall v \in \mathcal{U} : \quad y_v \geq 0$$

Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.

Online LP Solver of [Alon+ 03]

(P)

$$\min \sum_S x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

(D)

$$\max \sum_v y_v$$

$$\forall S \in \mathcal{S} : \quad \sum_{v \in S} y_v \leq 1$$

$$\forall v \in \mathcal{U} : \quad y_v \geq 0$$

Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.
- $+1$ to y_v.

Online LP Solver of [Alon+ 03]

(P)

$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

(D)

$$\max \sum_{v} y_v$$

$$\forall S \in \mathcal{S} : \quad \sum_{v \in S} y_v \leq 1$$

$$\forall v \in \mathcal{U} : \quad y_v \geq 0$$

Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.
- $+1$ to y_v.

Claim 1: x feasible for (**P**).

Online LP Solver of [Alon+ 03]

(P)

$$\min \sum_{S} x_S$$

$$\forall v \in \mathcal{U} : \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S} : \quad x_S \geq 0$$

(D)

$$\max \sum_{v} y_v$$

$$\forall S \in \mathcal{S} : \quad \sum_{v \in S} y_v \leq 1$$

$$\forall v \in \mathcal{U} : \quad y_v \geq 0$$

Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.
- $+1$ to y_v.

Claim 1: x feasible for (**P**).

Claim 2: $c(x) \leq c(y)$

Online LP Solver of [Alon+ 03]

(P)

$$\min \sum_S x_S$$

$$\forall v \in \mathcal{U}: \quad \sum_{S \ni v} x_S \geq 1$$

$$\forall S \in \mathcal{S}: \quad x_S \geq 0$$

(D)

$$\max \sum_v y_v$$

$$\forall S \in \mathcal{S}: \quad \sum_{v \in S} y_v \leq 1$$

$$\forall v \in \mathcal{U}: \quad y_v \geq 0$$

Init $x \leftarrow 1/m$.

While v (fractionally) uncovered:

- $\times 2$ to x_S for all $S \ni v$.
- $+1$ to y_v.

Claim 1: x feasible for (**P**).

Claim 2: $c(x) \leq c(y)$

Claim 3: $y/\log m$ feasible for (**D**).

Neither stage of [Alon+ 03] can be improved!

Neither stage of [Alon+ 03] can be improved!

Independent rounding loses $\Omega(\log n)$.

Neither stage of [Alon+ 03] can be improved!

Independent rounding loses $\Omega(\log n)$.

Theorem [Gupta Kehne **L.**]: $\Omega(\log m)$ for fractional algos in RO.

Neither stage of [Alon+ 03] can be improved!

Independent rounding loses $\Omega(\log n)$.

<u>**Theorem**</u> [Gupta Kehne **L.**]: $\Omega(\log m)$ for <u>fractional</u> algos in RO.

<u>**Theorem**</u> [Gupta Kehne **L.**]: algo of [Alon+ 03] gets $\Omega(\log m \log n)$ in RO.

Neither stage of [Alon+ 03] can be improved!

Independent rounding loses $\Omega(\log n)$.

Theorem [Gupta Kehne **L.**]: $\Omega(\log m)$ for fractional algos in RO.

Theorem [Gupta Kehne **L.**]: algo of [Alon+ 03] gets $\Omega(\log m \log n)$ in RO.

New algorithm needed!

Neither stage of [Alon+ 03] can be improved!

Independent rounding loses $\Omega(\log n)$.

Theorem [Gupta Kehne **L.**]: $\Omega(\log m)$ for fractional algos in RO.

Theorem [Gupta Kehne **L.**]: algo of [Alon+ 03] gets $\Omega(\log m \log n)$ in RO.

New algorithm needed!

We maintain coarse solution x, neither feasible nor monotone,
but round x anyway…

Talk Outline

Intro

➡ Secretary

LearnOrCover in Exponential Time

LearnOrCover in Poly Time

(Single Sample) Prophet

Conclusion & Extensions

Talk Outline

Intro

Secretary

➡ LearnOrCover in Exponential Time

LearnOrCover in Poly Time

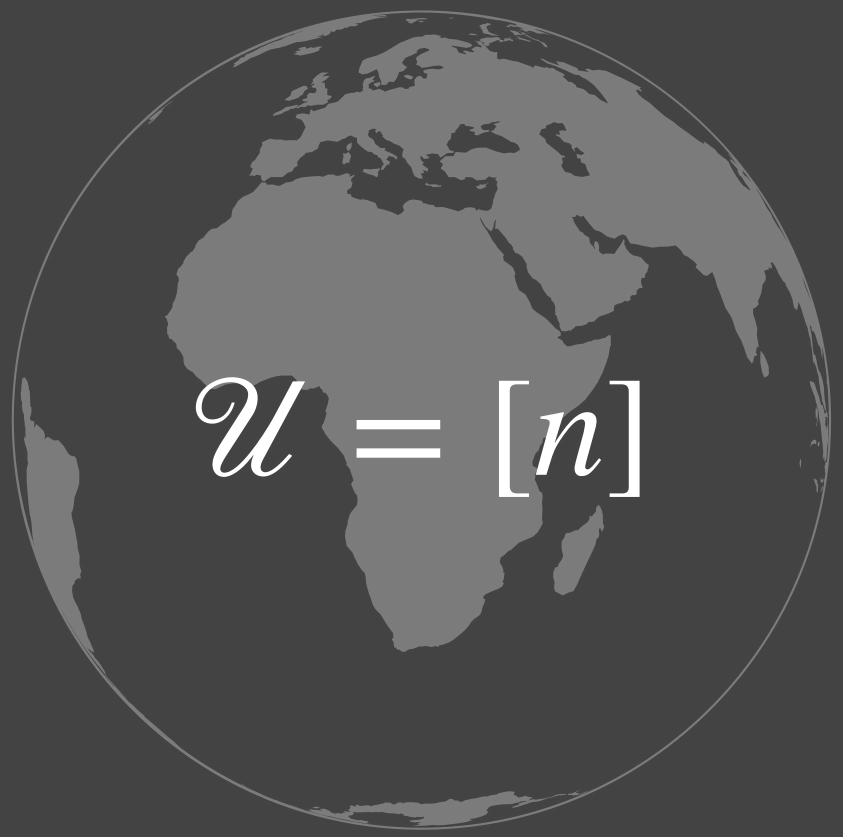
(Single Sample) Prophet

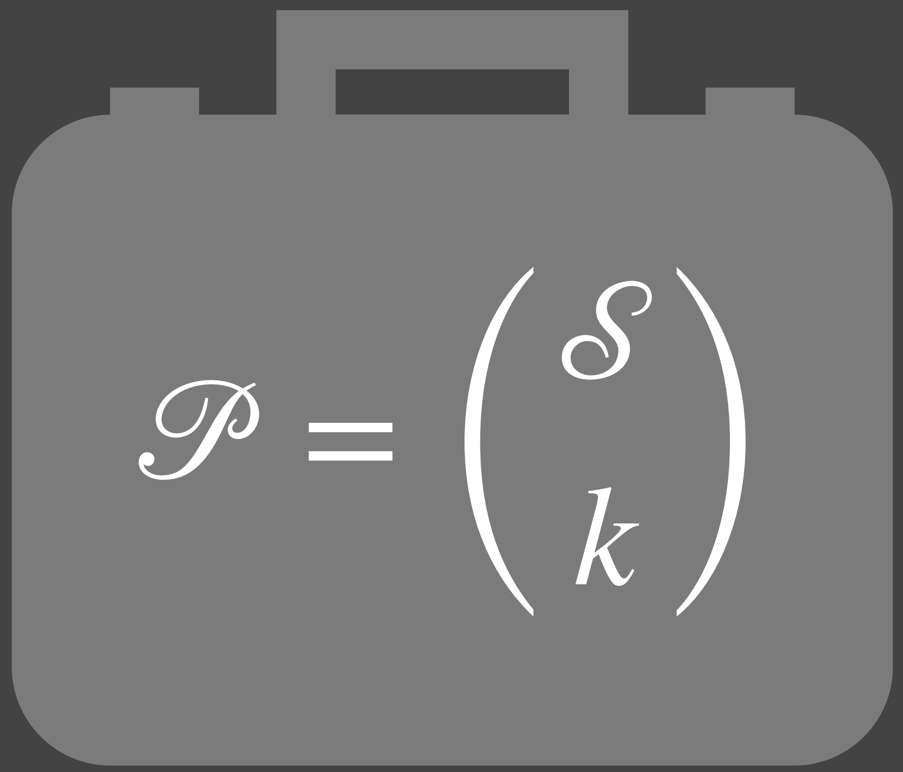
Conclusion & Extensions

LearnOrCover

(Unit cost, exp time warmup)

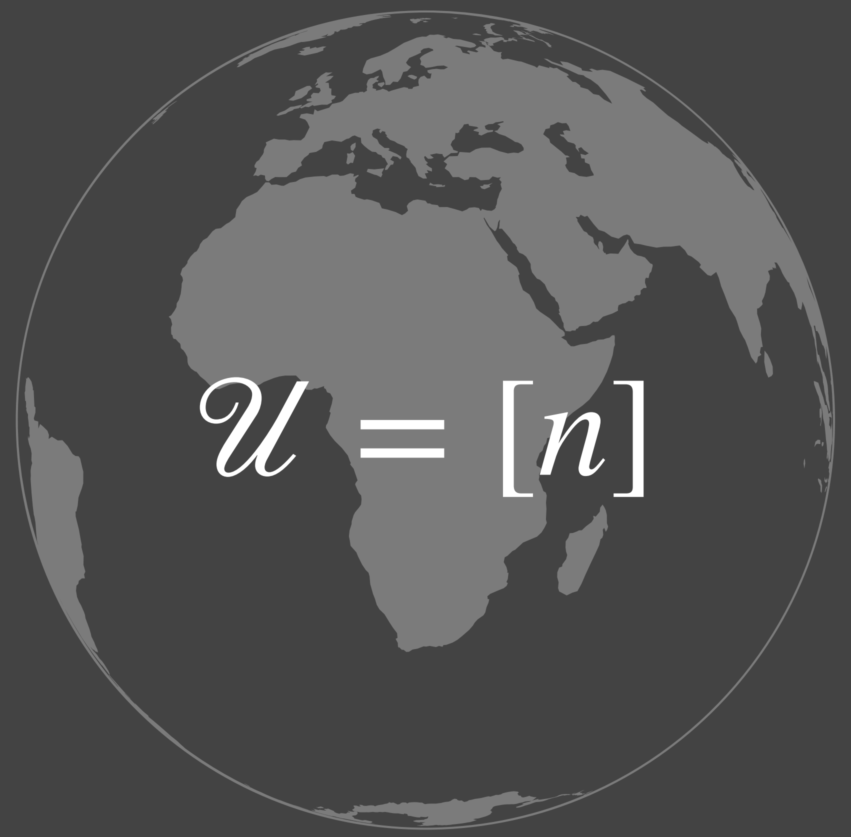
LearnOrCover
(Unit cost, exp time warmup)



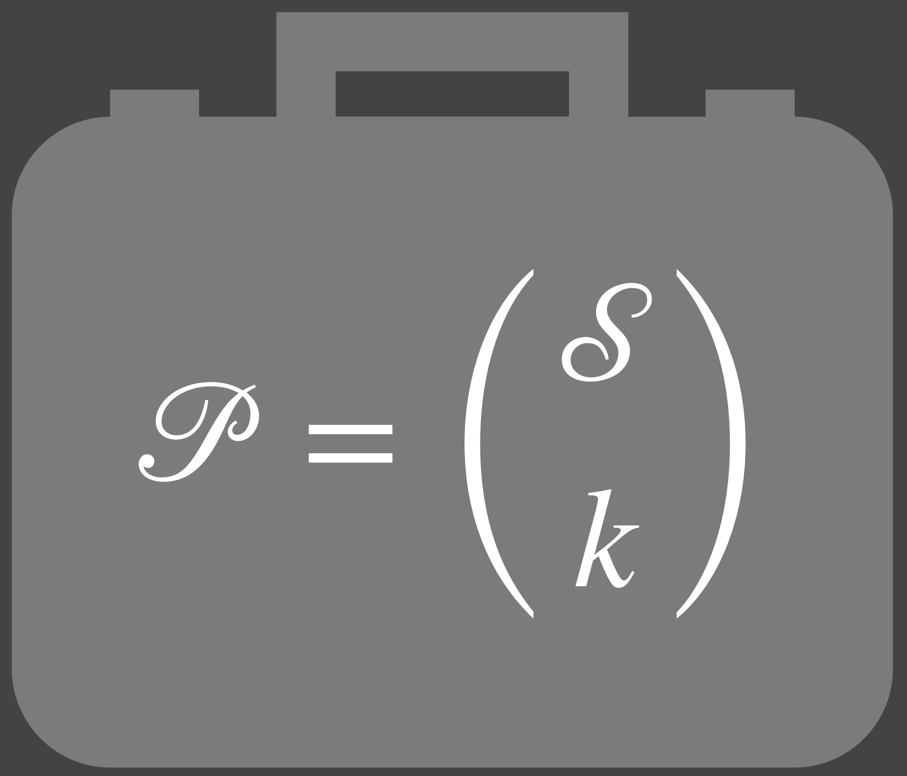


$k := |OPT|$

LearnOrCover
(Unit cost, exp time warmup)



$\mathscr{U} = [n]$

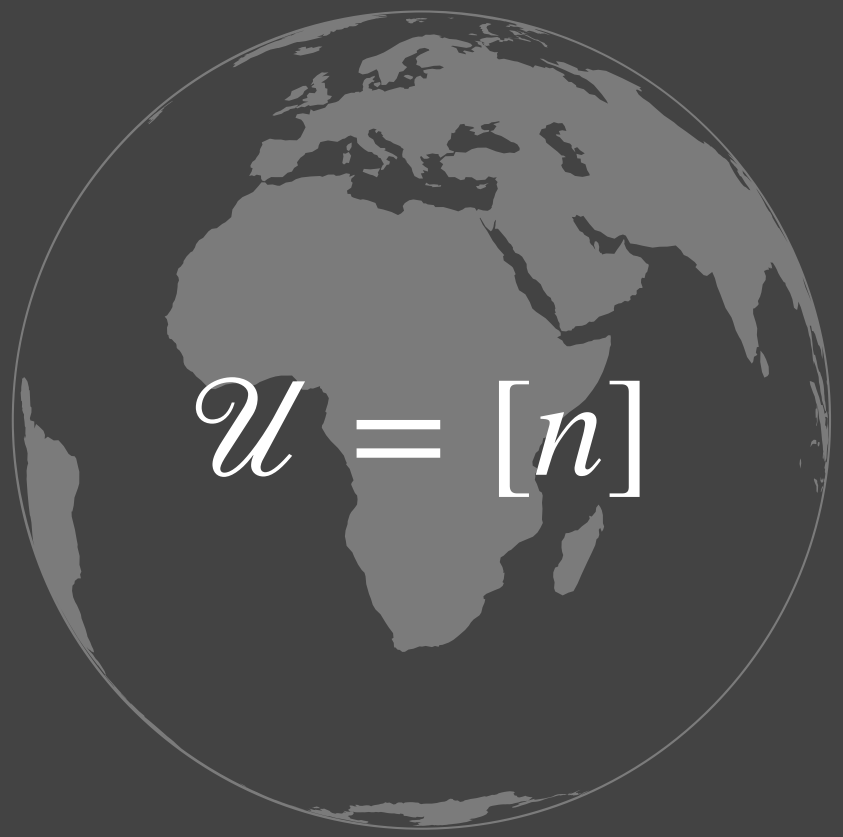


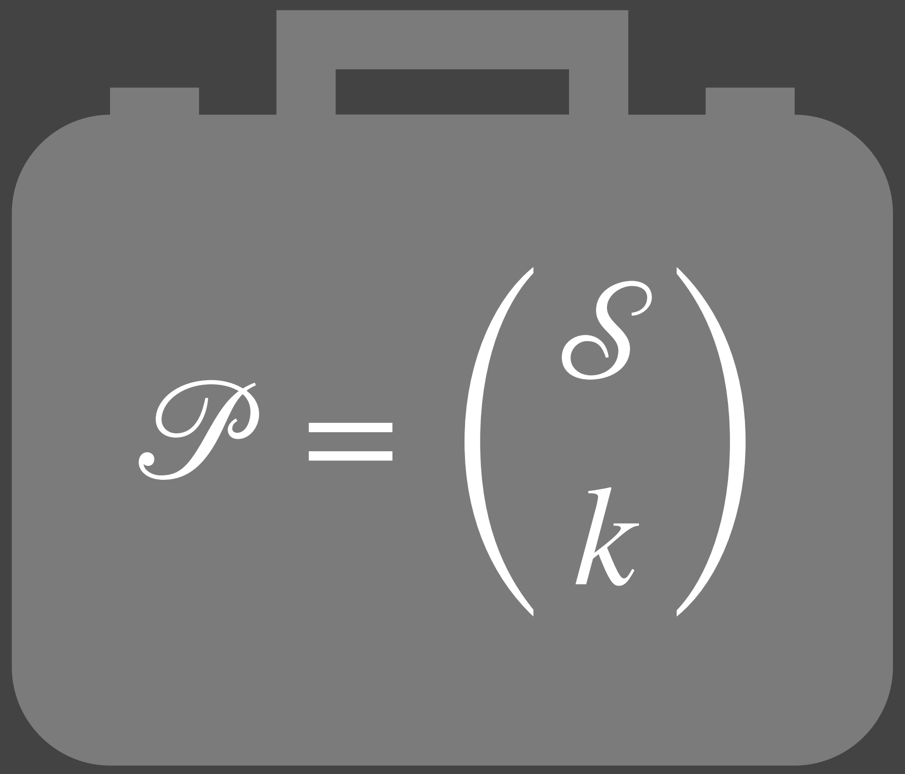
$\mathscr{P} = \binom{\mathscr{S}}{k}$

$k := |OPT|$

@ time t, element v arrives:

LearnOrCover
(Unit cost, exp time warmup)



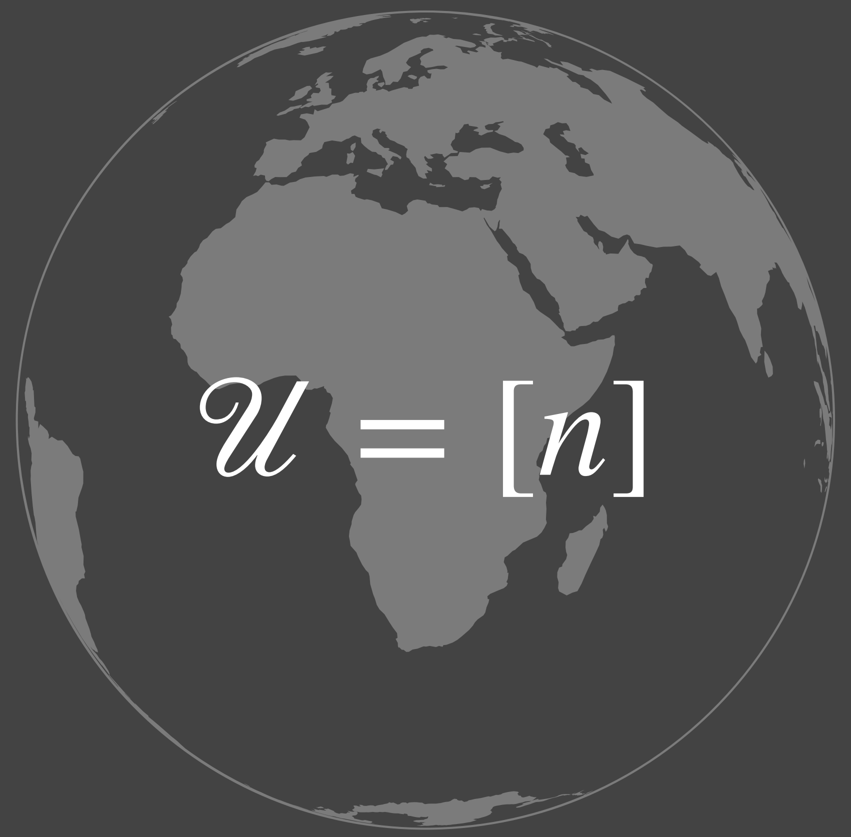


$k := |OPT|$

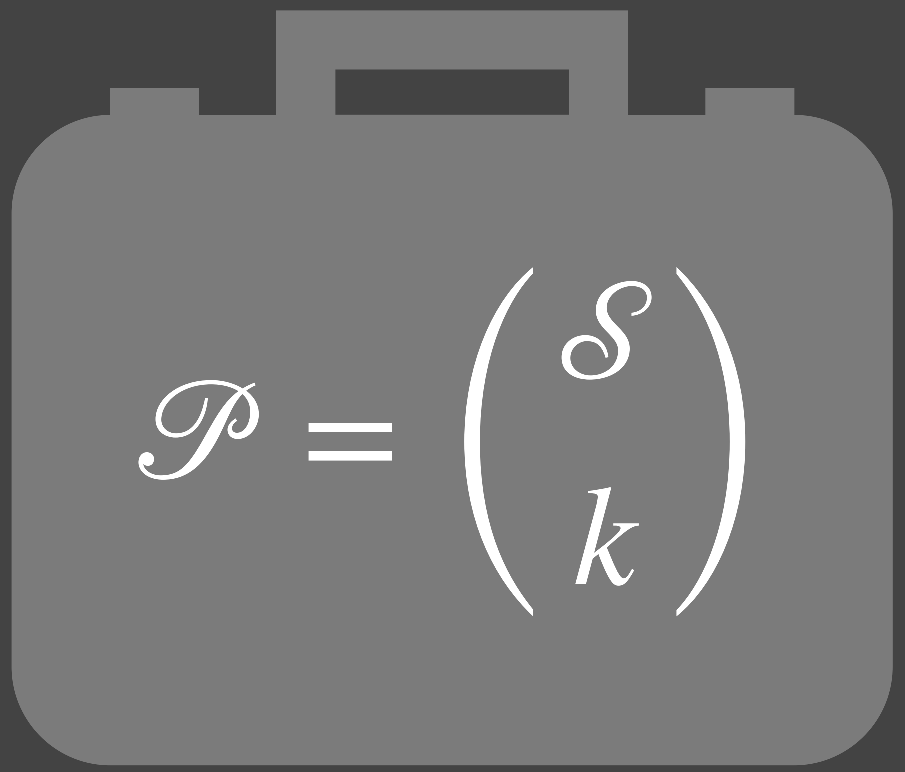
@ time t, element v arrives:

If v covered, do nothing.

LearnOrCover
(Unit cost, exp time warmup)



$\mathcal{U} = [n]$



$\mathscr{P} = \binom{\mathcal{S}}{k}$

$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

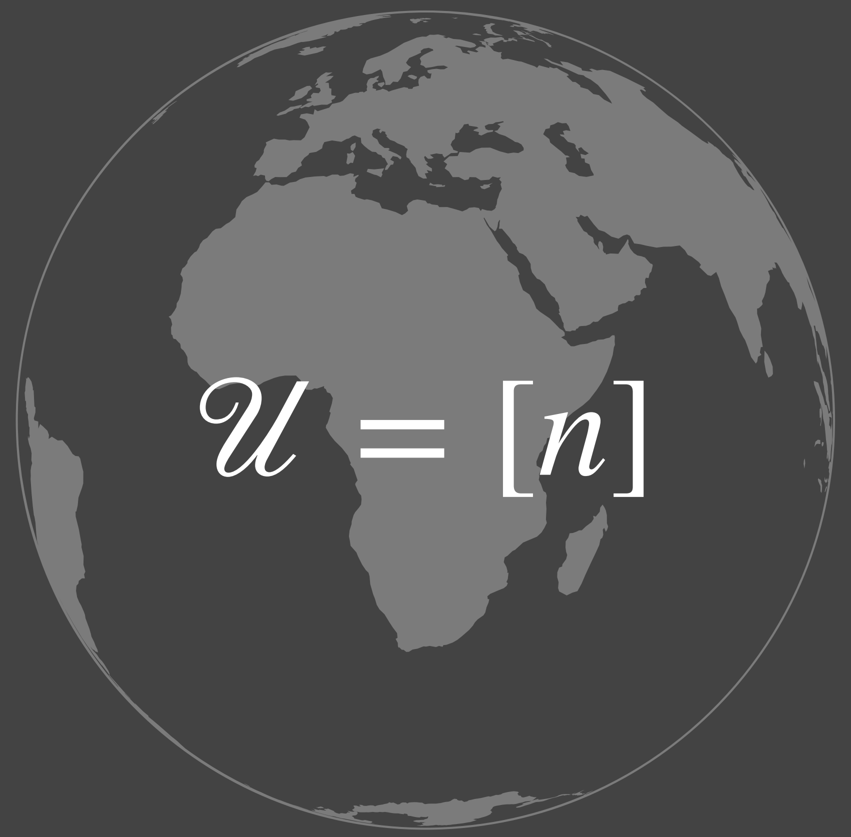
Else:

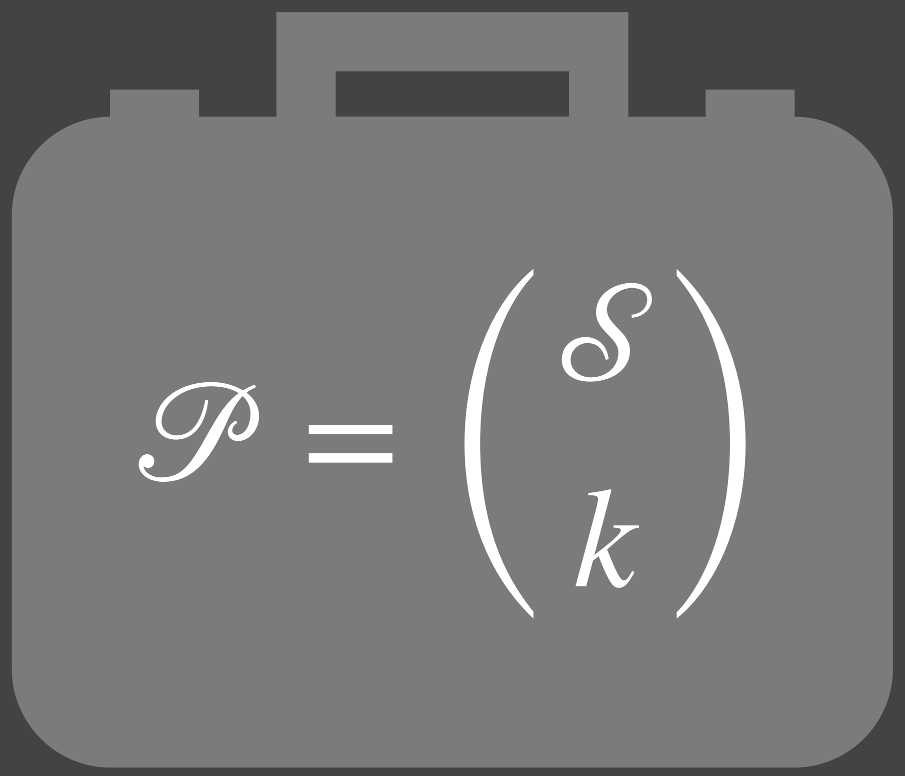
(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

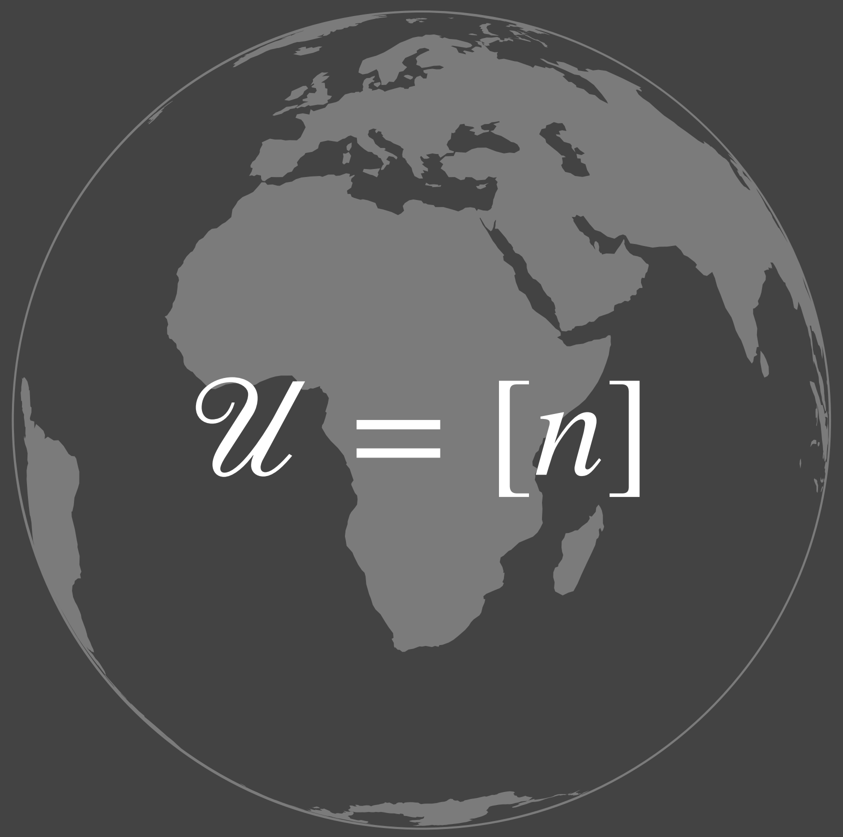
Else:

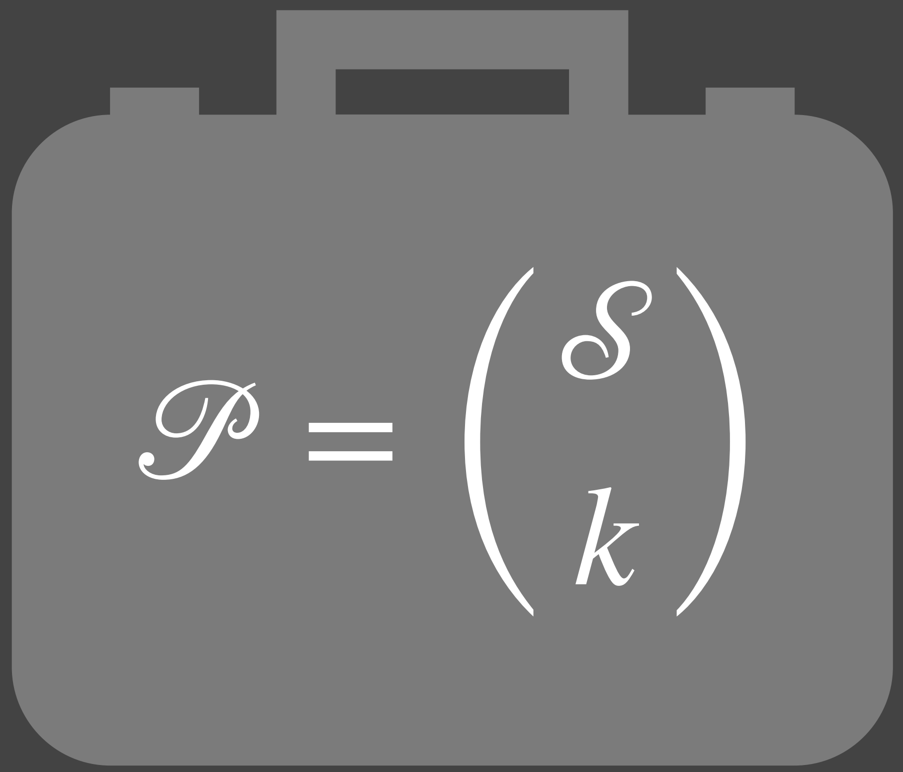
(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

LearnOrCover
(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) “Prune” $T \not\ni v$ from $\mathscr{P}$.

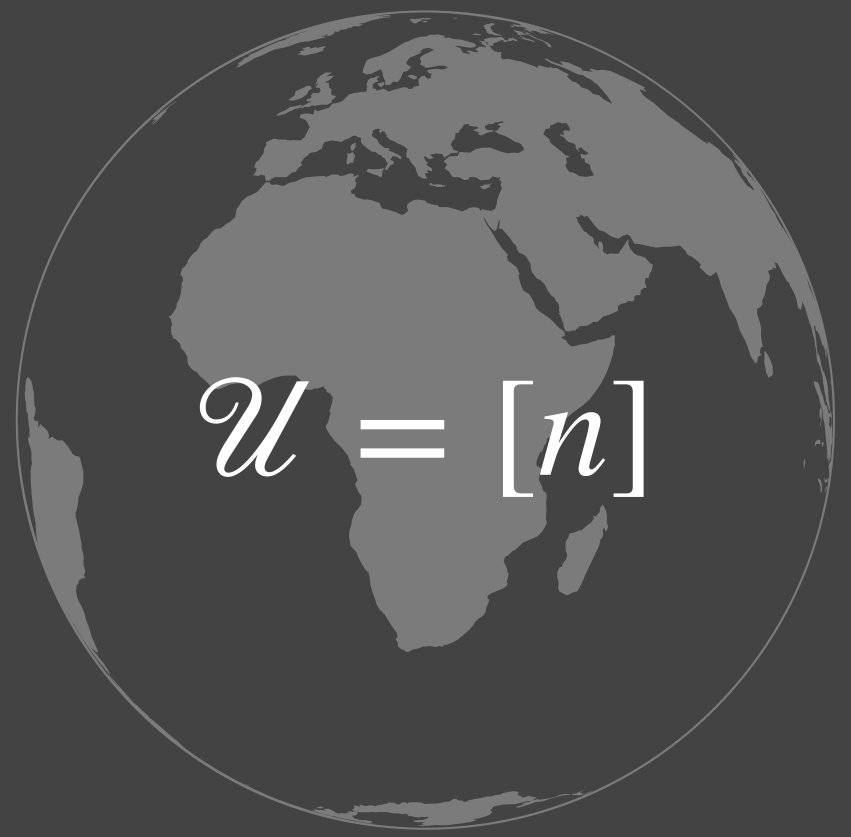
Buy arbitrary set to cover v.

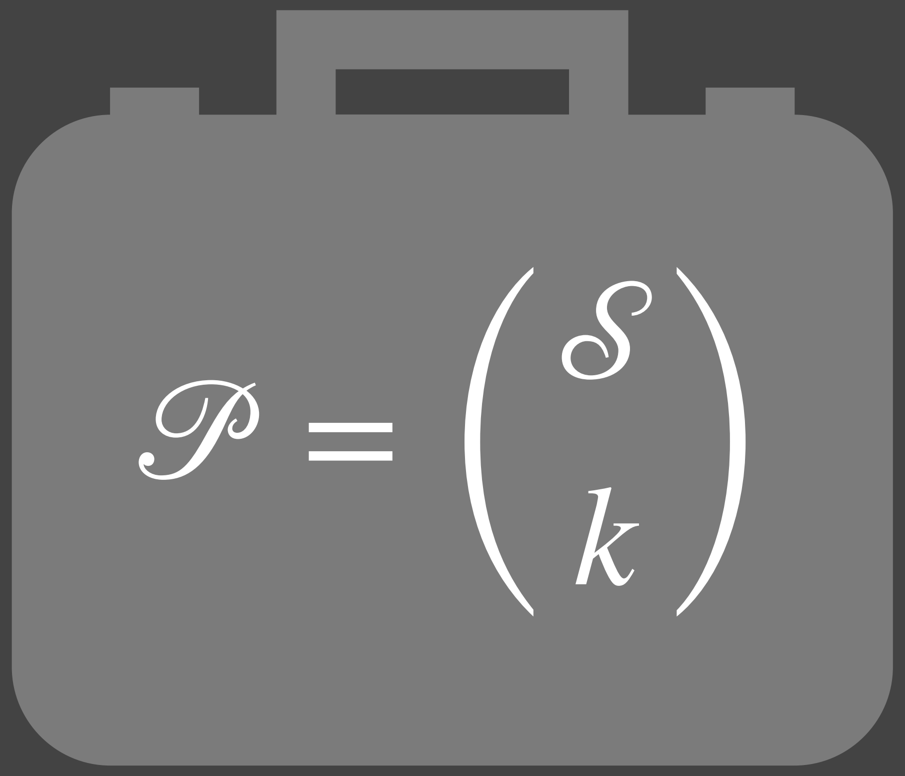
Case 1: $\geq 1/2$ of $T \in \mathscr{P}$ cover $\geq 1/2$ of $\mathscr{U}$.

Case 2: $> 1/2$ of $T \in \mathscr{P}$ cover $< 1/2$ of $\mathscr{U}$.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) “Prune” $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

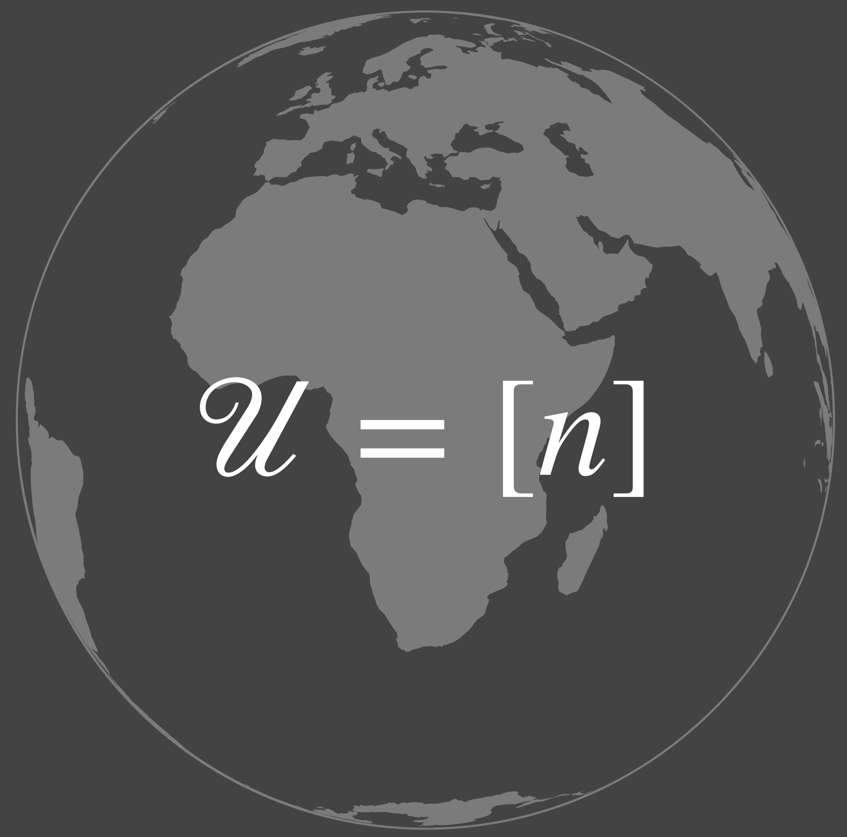
Case 1: $\geq 1/2$ **of** $T \in \mathscr{P}$ **cover** $\geq 1/2$ **of** $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

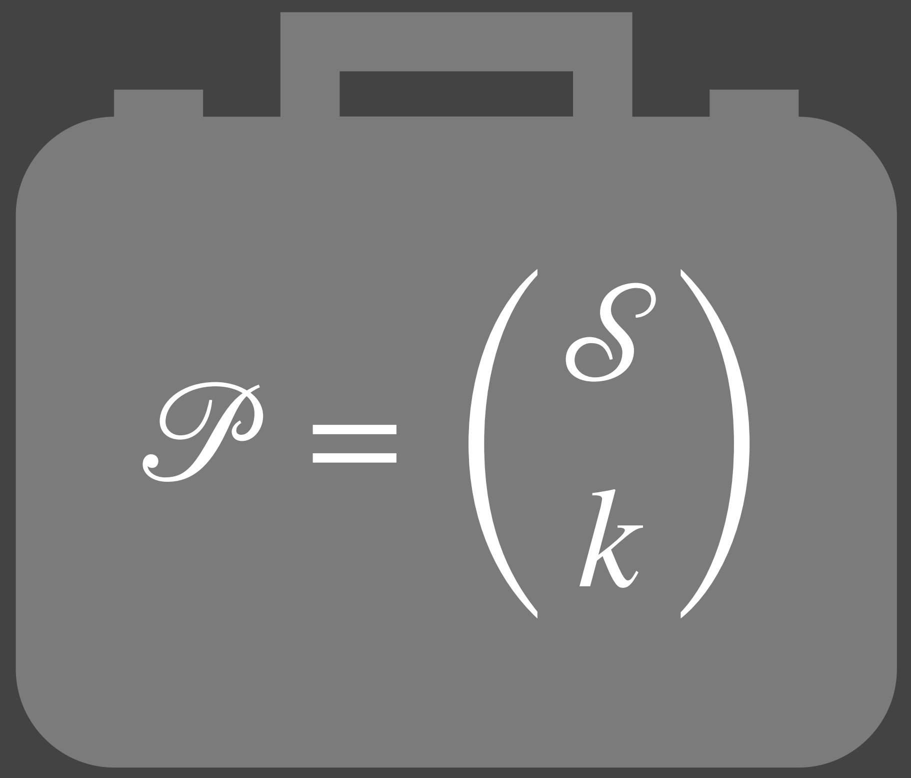
Case 2: $> 1/2$ **of** $T \in \mathscr{P}$ **cover** $< 1/2$ **of** $\mathscr{U}$.

LearnOrCover

(Unit cost, exp time warmup)



$\mathscr{U} = [n]$



$\mathscr{P} = \binom{\mathscr{S}}{k}$

$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

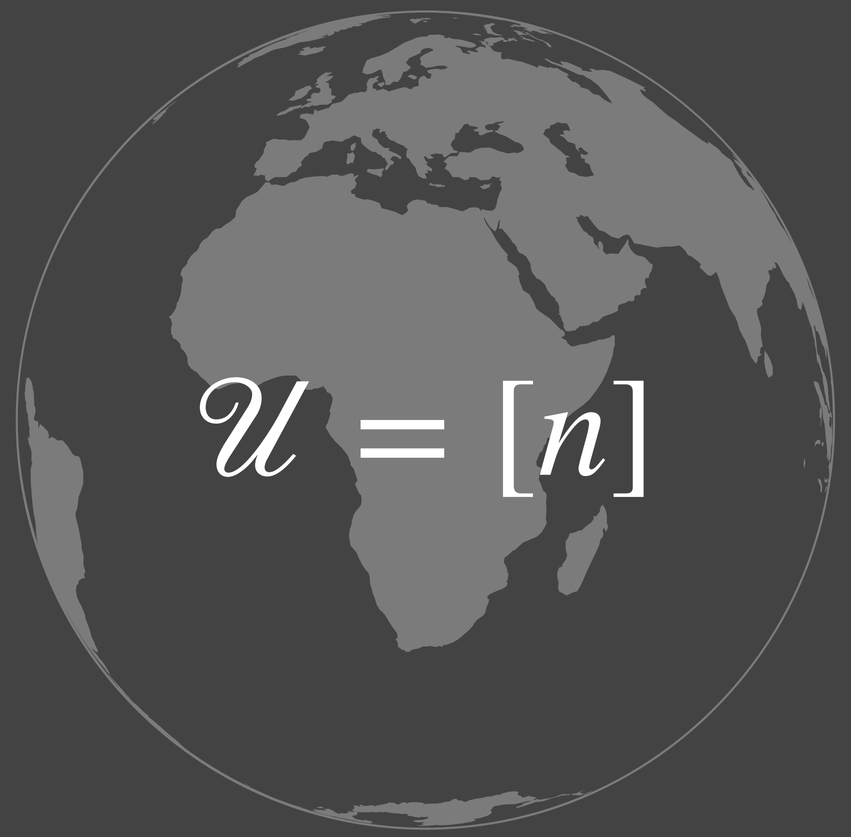
Case 1: $\geq 1/2$ **of** $T \in \mathscr{P}$ **cover** $\geq 1/2$ **of** $\mathscr{U}$.

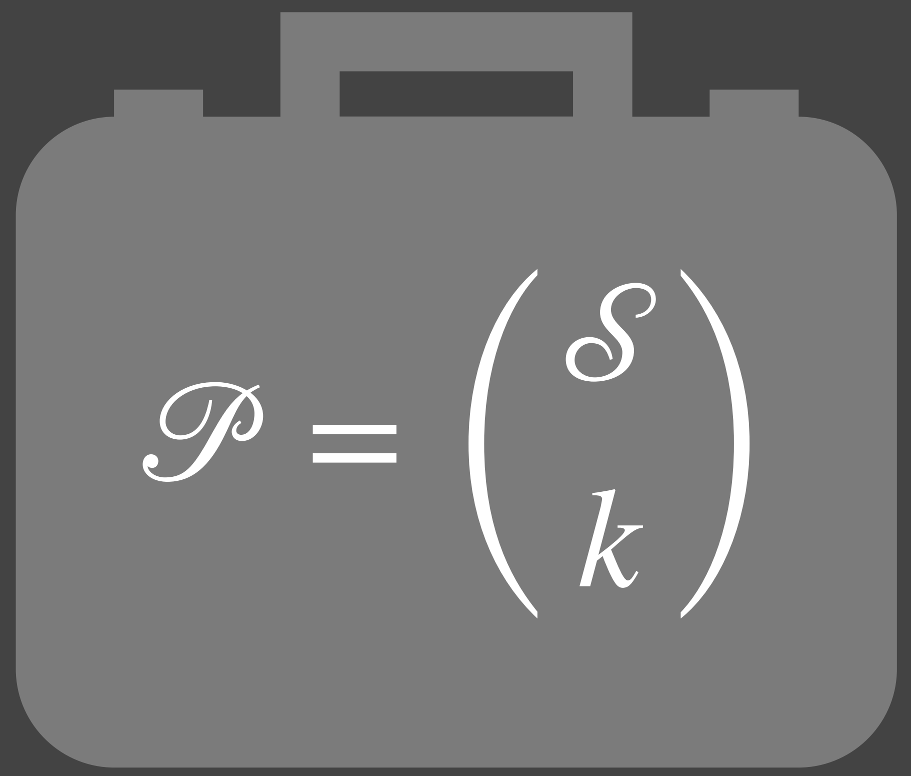
R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

Case 2: $> 1/2$ **of** $T \in \mathscr{P}$ **cover** $< 1/2$ **of** $\mathscr{U}$.

LearnOrCover
(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) “Prune” $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ **of** $T \in \mathscr{P}$ **cover** $\geq 1/2$ **of** $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

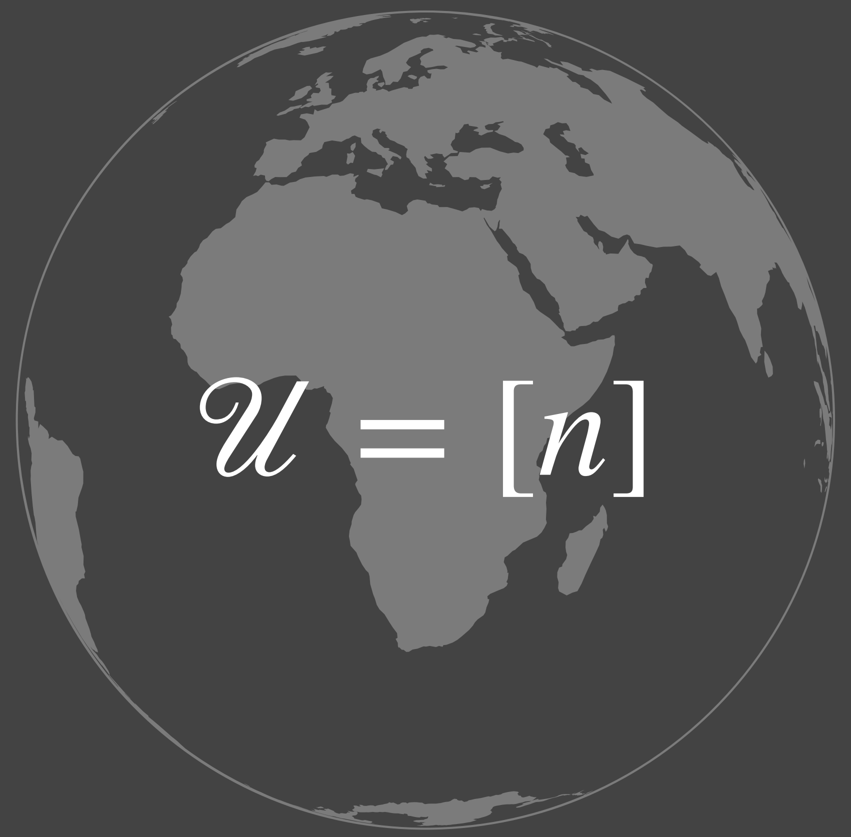
$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

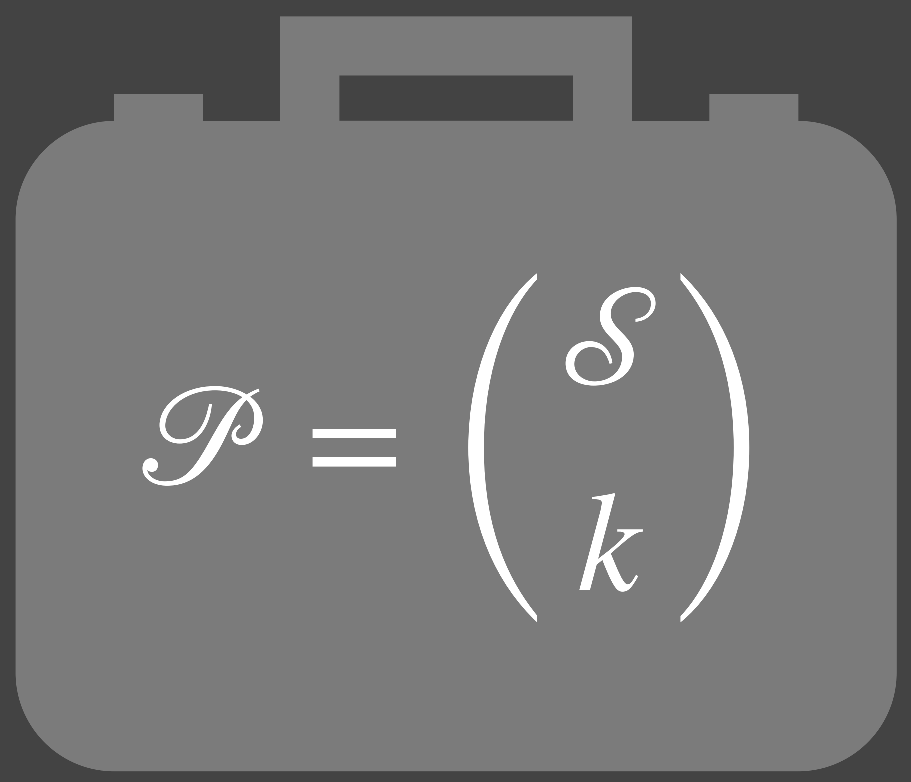
Case 2: $> 1/2$ **of** $T \in \mathscr{P}$ **cover** $< 1/2$ **of** $\mathscr{U}$.

$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ of $T \in \mathscr{P}$ cover $\geq 1/2$ of $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

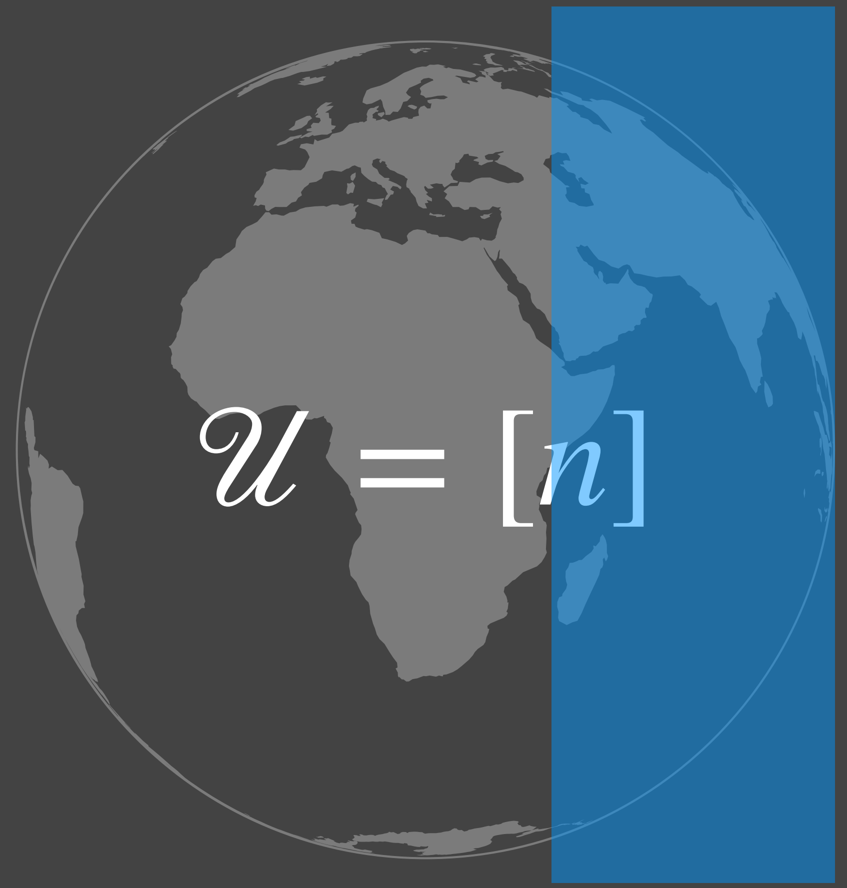
Case 2: $> 1/2$ of $T \in \mathscr{P}$ cover $< 1/2$ of $\mathscr{U}$.

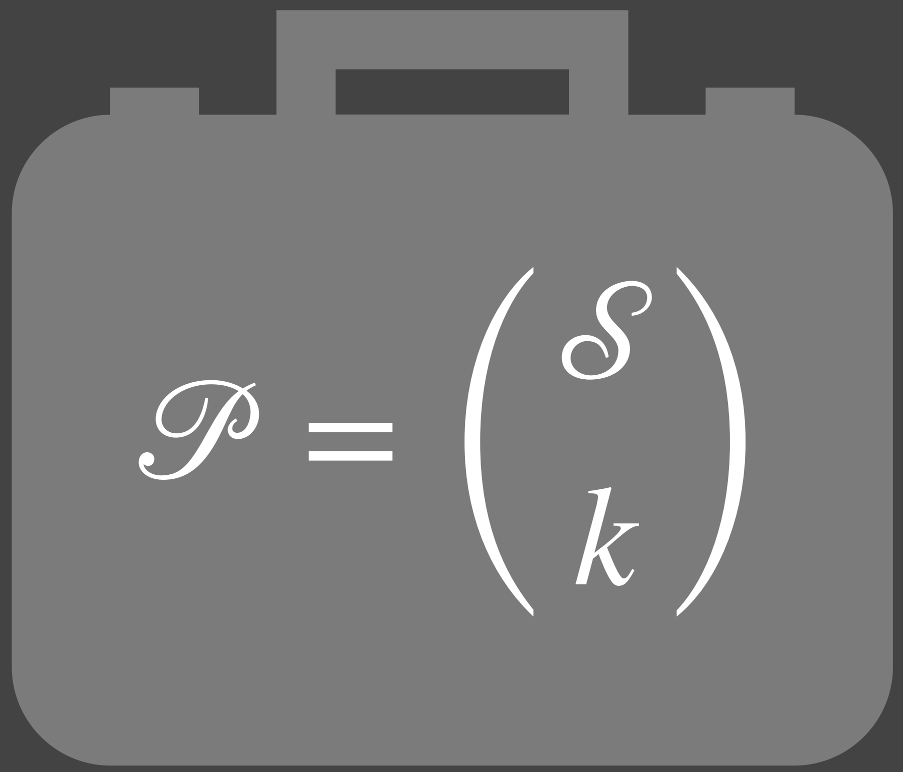
$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

$\mathscr{P}$ shrinks by $3/4$ in expectation.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) “Prune” $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ **of** $T \in \mathscr{P}$ **cover** $\geq 1/2$ **of** $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

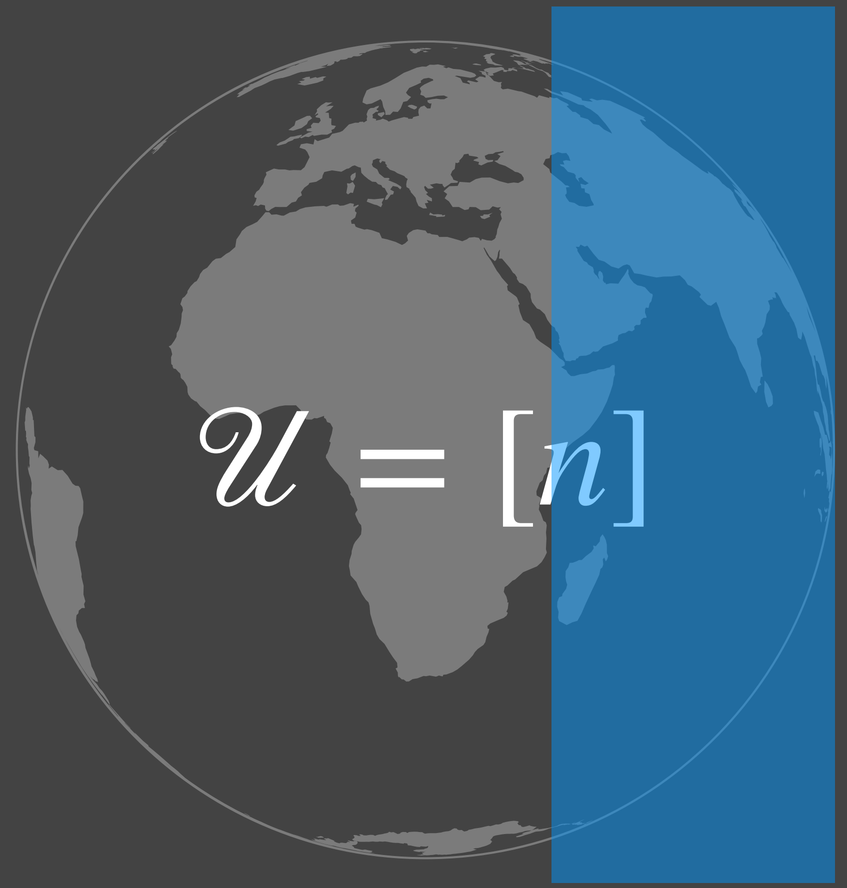
Case 2: $> 1/2$ **of** $T \in \mathscr{P}$ **cover** $< 1/2$ **of** $\mathscr{U}$.

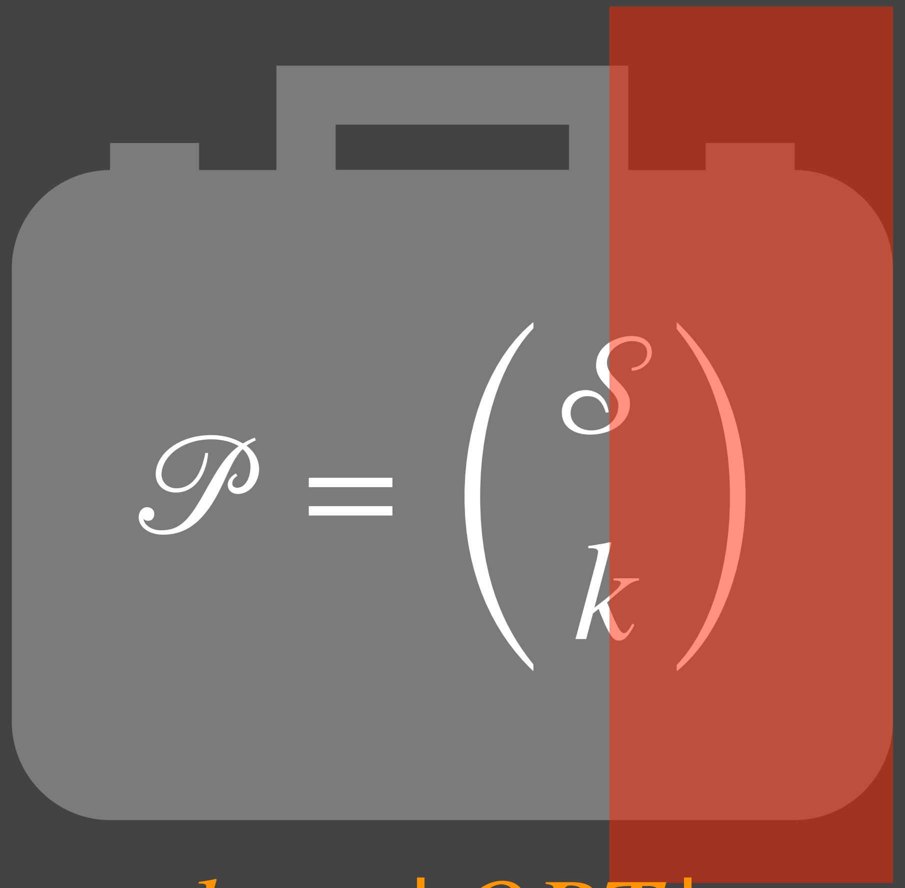
$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

$\mathscr{P}$ shrinks by $3/4$ in expectation.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ of $T \in \mathscr{P}$ cover $\geq 1/2$ of $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

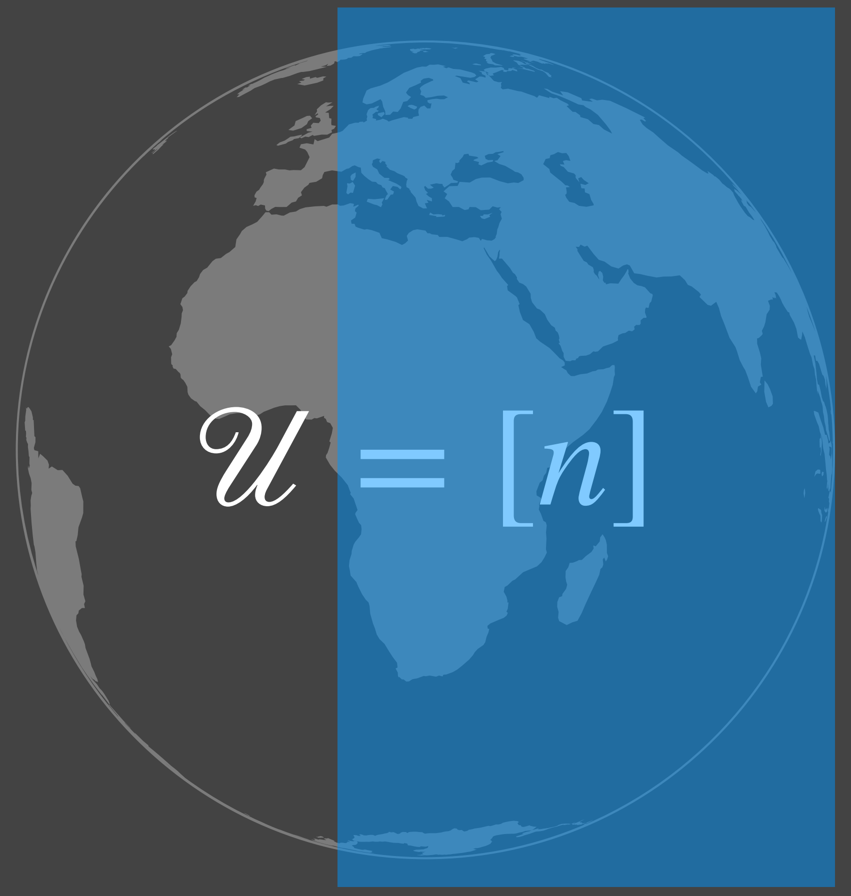
$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

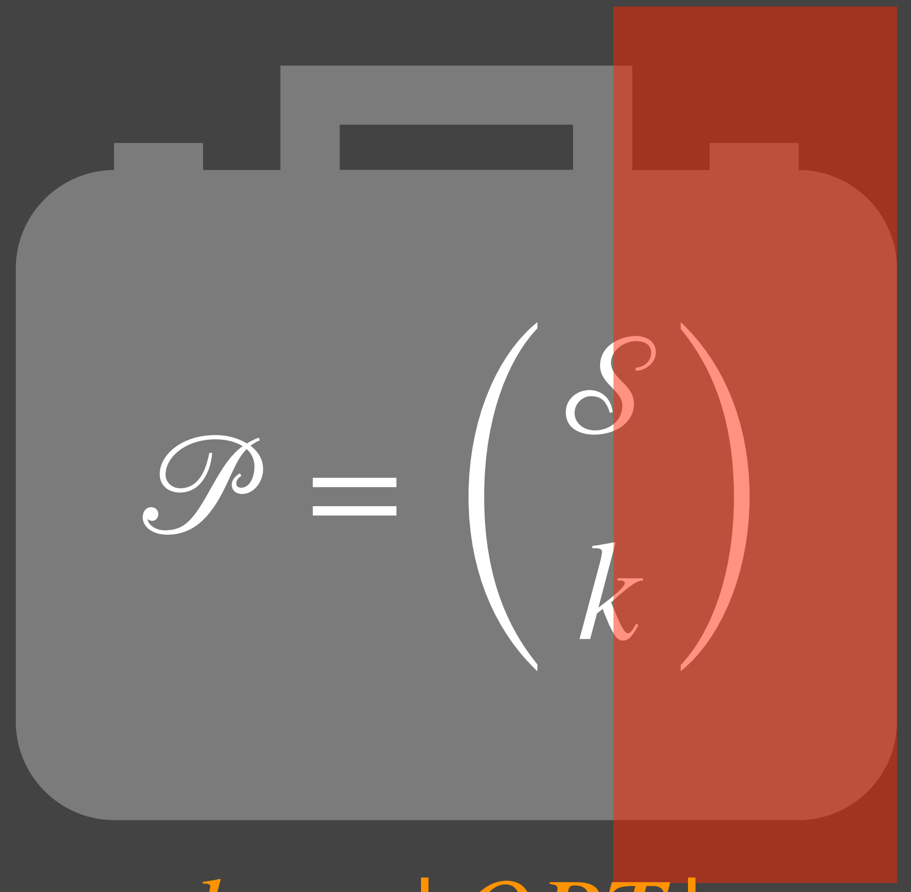
Case 2: $> 1/2$ of $T \in \mathscr{P}$ cover $< 1/2$ of $\mathscr{U}$.

$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

$\mathscr{P}$ shrinks by $3/4$ in expectation.

LearnOrCover
(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ **of** $T \in \mathscr{P}$ **cover** $\geq 1/2$ **of** $\mathscr{U}$.

R covers $\frac{|\mathscr{U}|}{4k}$ in expectation.

$\mathscr{U}$ shrinks by $\left(1 - \frac{1}{4k}\right)$ in expectation.

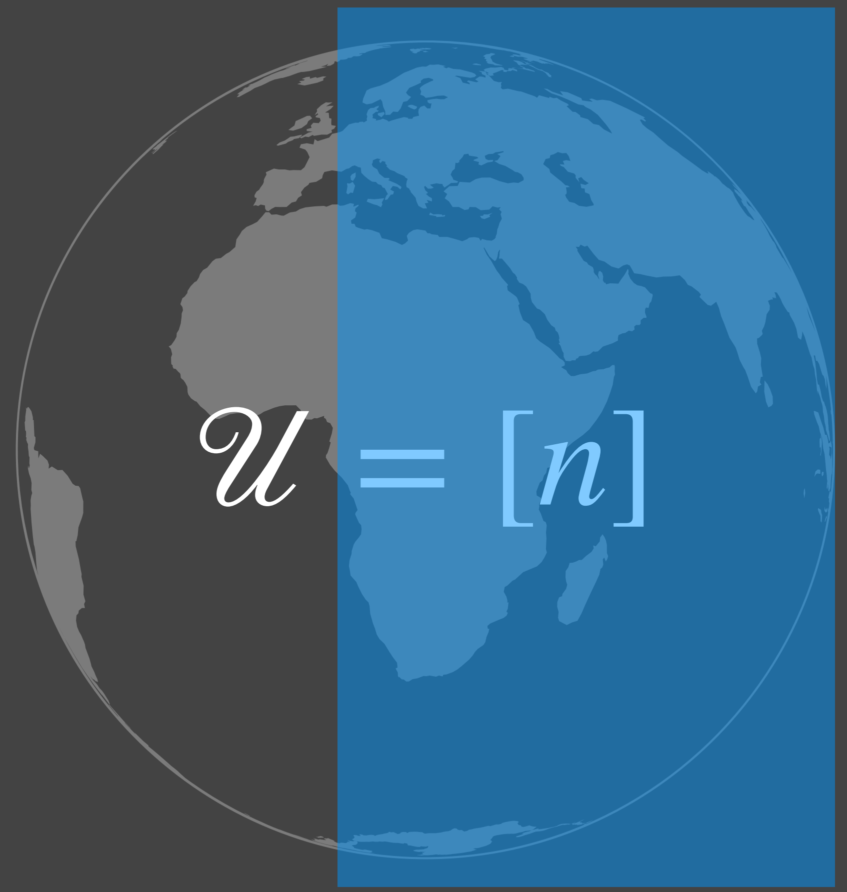
Case 2: $> 1/2$ **of** $T \in \mathscr{P}$ **cover** $< 1/2$ **of** $\mathscr{U}$.

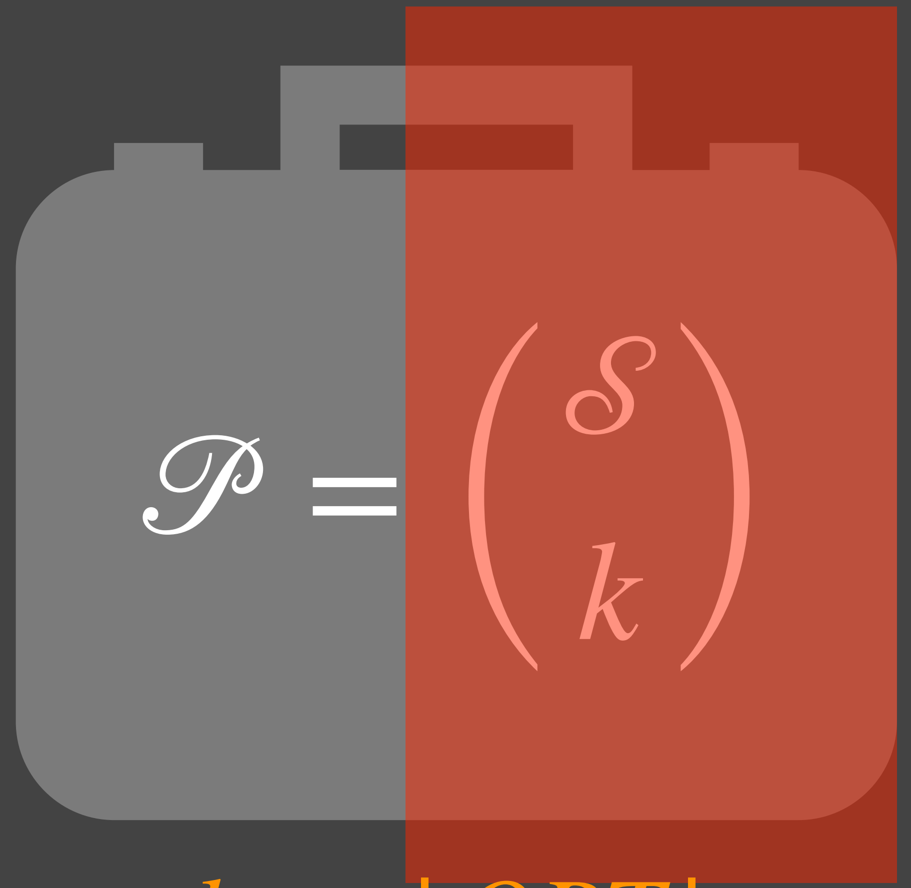
$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

$\mathscr{P}$ shrinks by $3/4$ in expectation.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) “Prune” $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ **of** $T \in \mathscr{P}$ **cover** $\geq 1/2$ **of** $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

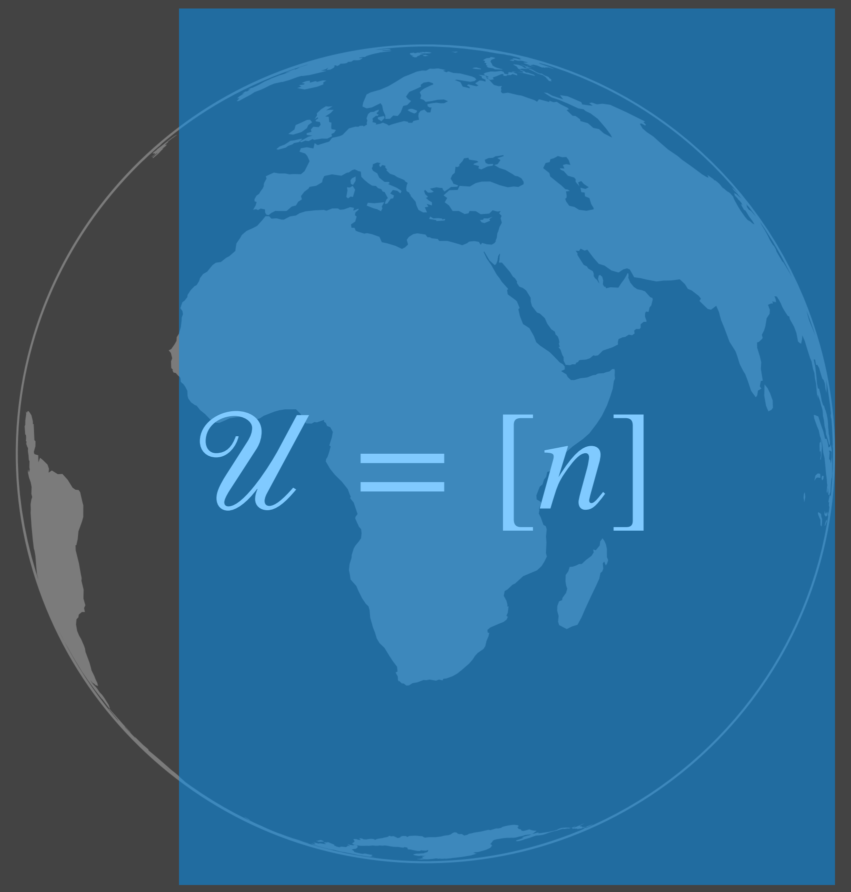
Case 2: $> 1/2$ **of** $T \in \mathscr{P}$ **cover** $< 1/2$ **of** $\mathscr{U}$.

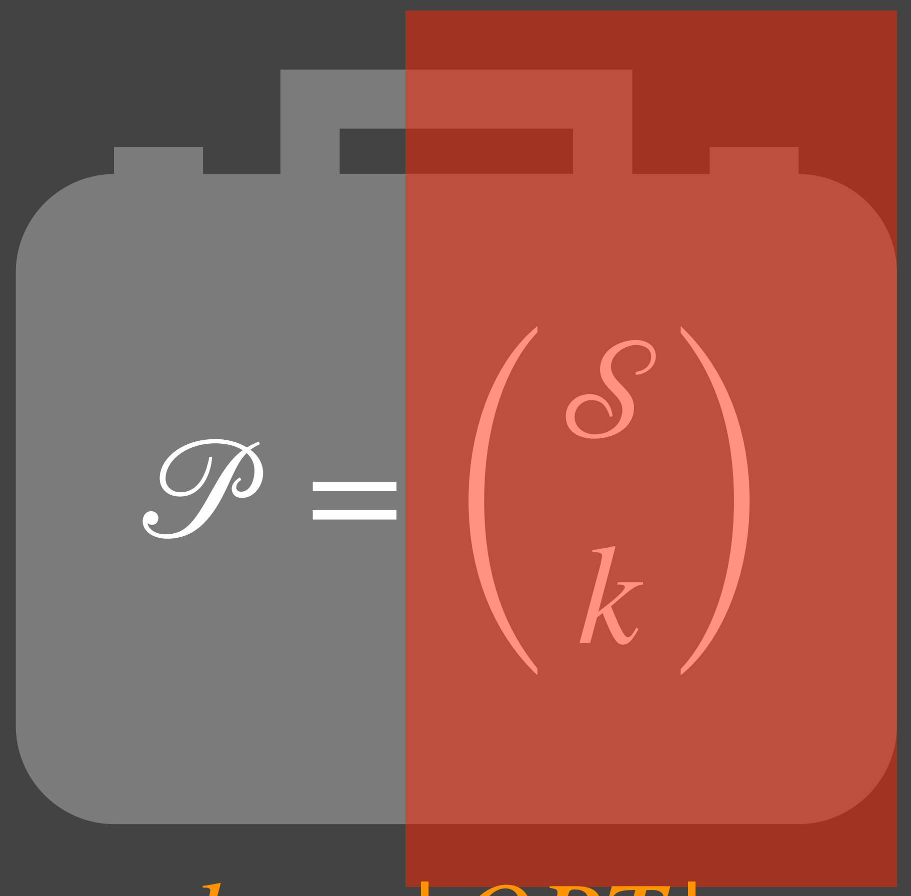
$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

$\mathscr{P}$ shrinks by $3/4$ in expectation.

LearnOrCover

(Unit cost, exp time warmup)





$k := |OPT|$

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) choose $T \sim \mathscr{P}$, buy random $R \sim T$.

(II) "Prune" $T \not\ni v$ from $\mathscr{P}$.

Buy arbitrary set to cover v.

Case 1: $\geq 1/2$ of $T \in \mathscr{P}$ cover $\geq 1/2$ of $\mathscr{U}$.

R covers $\dfrac{|\mathscr{U}|}{4k}$ in expectation.

$\mathscr{U}$ shrinks by $\left(1 - \dfrac{1}{4k}\right)$ in expectation.

Case 2: $> 1/2$ of $T \in \mathscr{P}$ cover $< 1/2$ of $\mathscr{U}$.

$\geq 1/2$ of $T \in \mathscr{P}$ pruned w.p. $1/2$.

$\mathscr{P}$ shrinks by $3/4$ in expectation.

RO Set Cover

(Exponential Time Warmup)

Case 1: (COVER)

$\mathscr{U}$ shrinks by $\left(1 - \frac{1}{4k}\right)$ in expectation.

Case 2: (LEARN)

$\mathscr{P}$ shrinks by $3/4$ in expectation.

RO Set Cover
(Exponential Time Warmup)

<u>Case 1:</u> (COVER)

$\mathscr{U}$ shrinks by $\left(1 - \frac{1}{4k}\right)$ in expectation.

<u>Case 2:</u> (LEARN)

$\mathscr{P}$ shrinks by $3/4$ in expectation.

$|\mathscr{U}|$ initially n, $\Rightarrow$ $O(k \log n)$ COVER steps suffice.

RO Set Cover
(Exponential Time Warmup)

Case 1: (COVER)

$\mathscr{U}$ shrinks by $\left(1 - \frac{1}{4k}\right)$ in expectation.

Case 2: (LEARN)

$\mathscr{P}$ shrinks by $3/4$ in expectation.

$|\mathscr{U}|$ initially n, $\Rightarrow$ $O(k \log n)$ COVER steps suffice.

$|\mathscr{P}|$ initially $\binom{m}{k} \approx m^k$, $\Rightarrow$ $O(k \log m)$ LEARN steps suffice.

RO Set Cover

(Exponential Time Warmup)

Case 1: (COVER)

$\mathscr{U}$ shrinks by $\left(1 - \frac{1}{4k}\right)$ in expectation.

Case 2: (LEARN)

$\mathscr{P}$ shrinks by $3/4$ in expectation.

$|\mathscr{U}|$ initially n, $\Rightarrow$ $O(k \log n)$ COVER steps suffice.

$|\mathscr{P}|$ initially $\binom{m}{k} \approx m^k$, $\Rightarrow$ $O(k \log m)$ LEARN steps suffice.

$\Rightarrow O(k \log mn)$ steps suffice.

RO Set Cover

(Exponential Time Warmup)

Case 1: (COVER)

$\mathscr{U}$ shrinks by $\left(1 - \frac{1}{4k}\right)$ in expectation.

Case 2: (LEARN)

$\mathscr{P}$ shrinks by $3/4$ in expectation.

$|\mathscr{U}|$ initially n, $\Rightarrow$ $O(k \log n)$ COVER steps suffice.

$|\mathscr{P}|$ initially $\binom{m}{k} \approx m^k$, $\Rightarrow$ $O(k \log m)$ LEARN steps suffice.

$\Rightarrow O(k \log mn)$ steps suffice.

But how to make polytime?

Can we reuse LEARN/COVER intuition?

Talk Outline

Intro

Secretary

➡ LearnOrCover in Exponential Time

LearnOrCover in Poly Time

(Single Sample) Prophet

Conclusion & Extensions

Talk Outline

Intro

Secretary

LearnOrCover in Exponential Time

LearnOrCover in Poly Time

(Single Sample) Prophet

Conclusion & Extensions

LearnOrCover
(Unit cost)

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

  If v covered, do nothing.

LearnOrCover

(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/\|x\|_1$.

LearnOrCover

(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/\|x\|_1$.

Buy arbitrary set to cover v.

LearnOrCover
(Unit cost)

Idea! Measure convergence with potential function:

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/||x||_1$.

Buy arbitrary set to cover v.

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/\|x\|_1$.

Buy arbitrary set to cover v.

Idea! Measure convergence with potential function:

$$\boxed{\Phi(t) = c_1\,\mathrm{KL}(x^* \,||\, x^t) + c_2 \log|\mathcal{U}^t|}$$

$\mathcal{U}^t$:= uncovered elements @ time t

x^* := uniform distribution on OPT

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/\|x\|_1$.

Buy arbitrary set to cover v.

$$\sum_S x_S^* \log \frac{x_S^*}{x_S^t}$$

Idea! Measure convergence with potential function:

$$\boxed{\Phi(t) = c_1 \, \mathrm{KL}(x^* \,||\, x^t) + c_2 \log |\mathcal{U}^t|}$$

$\mathcal{U}^t$:= uncovered elements @ time t

x^* := uniform distribution on OPT

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/\|x\|_1$.

Buy arbitrary set to cover v.

$$\sum_S x_S^* \log \frac{x_S^*}{x_S^t}$$

Idea! Measure convergence with potential function:

$$\boxed{\Phi(t) = c_1 \, \mathrm{KL}(x^* \,||\, x^t) + c_2 \log |\mathcal{U}^t|}$$

$\mathcal{U}^t$:= uncovered elements @ time t

x^* := uniform distribution on OPT

Claim 1: $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.
@ time t, element v arrives:
If v covered, do nothing.
Else:
(I) Buy random $R \sim x$.
(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.
Renormalize $x \leftarrow x/\|x\|_1$.
Buy arbitrary set to cover v.

$\sum_S x_S^* \log \frac{x_S^*}{x_S^t}$

Idea! Measure convergence with potential function:

$$\boxed{\Phi(t) = c_1 \, \mathrm{KL}(x^* \,||\, x^t) + c_2 \log |\mathcal{U}^t|}$$

$\mathcal{U}^t$:= uncovered elements @ time t
x^* := uniform distribution on OPT

Claim 1: $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

Claim 2: If v uncovered, then $E[\Delta\Phi] \leq -\frac{1}{k}$.

(Recall $k = |OPT|$)

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.
@ time t, element v arrives:
If v covered, do nothing.
Else:
(I) Buy random $R \sim x$.
(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.
Renormalize $x \leftarrow x/\|x\|_1$.
Buy arbitrary set to cover v.

$$\sum_S x_S^* \log \frac{x_S^*}{x_S^t}$$

Idea! Measure convergence with potential function:

$$\boxed{\Phi(t) = c_1 \,\mathrm{KL}(x^* \,||\, x^t) + c_2 \log |\mathcal{U}^t|}$$

$\mathcal{U}^t$:= uncovered elements @ time t
x^* := uniform distribution on OPT

Claim 1: $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

Claim 2: If v uncovered, then $E[\Delta\Phi] \leq -\frac{1}{k}$.

(Recall $k = |OPT|$)

Bound $E_R[\Delta \log |\mathcal{U}^t|]$ over randomness of R.
Bound $E_v[\Delta KL]$ over randomness of v.

LearnOrCover
(Unit cost)

Init. $x \leftarrow 1/m$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy random $R \sim x$.

(II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

Renormalize $x \leftarrow x/||x||_1$.

Buy arbitrary set to cover v.

$\sum_S x_S^* \log \frac{x_S^*}{x_S^t}$

Idea! Measure convergence with potential function:

$$\boxed{\Phi(t) = c_1 \, \mathrm{KL}(x^* \,||\, x^t) + c_2 \log |\mathcal{U}^t|}$$

$\mathcal{U}^t$:= uncovered elements @ time t

x^* := uniform distribution on OPT

Claim 1: $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

Claim 2: If v uncovered, then $E[\Delta\Phi] \leq -\frac{1}{k}$.

(Recall $k = |OPT|$)

Bound $E_R[\Delta \log |\mathcal{U}^t|]$ over randomness of R.

Bound $E_v[\Delta KL]$ over randomness of v. $\longleftarrow$ This is where we use RO!

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathscr{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \le -E_v\left[\sum_{S \ni v} x_S\right].$$

$$E[\Delta\Phi] = E_v[\Delta \mathrm{KL}] + (e-1) \cdot E_R[\Delta \log |\mathcal{U}^t|] \le -\frac{1}{k}$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

$$E[\Delta\Phi] = E_v[\Delta \mathrm{KL}] + (e-1)\cdot E_R[\Delta \log|\mathcal{U}^t|] \le -\frac{1}{k}$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \le -E_v\left[\sum_{S \ni v} x_S\right].$$

$$E[\Delta\Phi] = E_v[\Delta \mathrm{KL}] + (e-1) \cdot E_R[\Delta \log |\mathcal{U}^t|] \le -\frac{1}{k}$$

Since $\Phi(0) = O(\log(mn))$, expected total cost is $k \log(mn)$.

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\mathbf{KL}(x^* \,||\, x^t) - \mathbf{KL}(x^* \,||\, x^{t-1})$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \le -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \sum_S x_S^* \log \|x\|_1 - \sum_{S \ni v} x_S^* \log e$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathscr{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\underbrace{\sum_S x_S^*}_{=1}} \log \|x\|_1 - \sum_{S \ni v} x_S^* \log e$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \le -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \text{KL}] \le (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\underbrace{\sum_S x_S^*}_{=1}} \log \|x\|_1 - \sum_{S\ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathscr{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \underbrace{\cancel{\sum_S x_S^*}}_{=1} \log \|x\|_1 - \sum_{S\ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

$$= \log\left(\sum_S x_S^{t-1} + \sum_{S\ni v}(e-1)\cdot x_S^{t-1}\right) - \sum_{S\ni v} x_S^*$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\sum_S x_S^*}_{=1} \log \|x\|_1 - \sum_{S \ni v} x_S^* \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S \ni v} (e-1) \cdot x_S^{t-1}\right) - \sum_{S \ni v} x_S^*$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \underbrace{\cancel{\sum_S x_S^*}}_{=1} \log \|x\|_1 - \sum_{S \ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S \ni v} (e-1) \cdot x_S^{t-1}\right) - \underbrace{\sum_{S \ni v} x_S^*}_{\geq 1/k}$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathscr{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\sum_S x_S^*}_{=1} \log \|x\|_1 - \sum_{S\ni v} x_S^* \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S\ni v}(e-1)\cdot x_S^{t-1}\right) - \underbrace{\sum_{S\ni v} x_S^*}_{\geq 1/k}$$

$$\leq \log\left(1 + \sum_{S\ni v}(e-1)\cdot x_S\right) - \frac{1}{k}.$$

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \leq -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\sum_S x_S^*}_{=1} \log \|x\|_1 - \sum_{S \ni v} x_S^* \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S \ni v} (e-1) \cdot x_S^{t-1}\right) - \underbrace{\sum_{S \ni v} x_S^*}_{\ge 1/k}$$

$$\le \log\left(1 + \sum_{S \ni v} (e-1) \cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \le z$, take expectation over v, ■.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \le -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \underbrace{\cancel{\sum_S x_S^*}}_{=1} \log \|x\|_1 - \sum_{S\ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S\ni v}(e-1)\cdot x_S^{t-1}\right) - \underbrace{\sum_{S\ni v} x_S^*}_{\geq 1/k}$$

$$\leq \log\left(1 + \sum_{S\ni v}(e-1)\cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \leq z$, take expectation over v, ■.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \leq -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

$$\log|\mathcal{U}^t| - \log|\mathcal{U}^{t-1}|$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\sum_S x_S^*}_{=1} \log \|x\|_1 - \sum_{S \ni v} x_S^* \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S \ni v} (e-1)\cdot x_S^{t-1}\right) - \underbrace{\sum_{S \ni v} x_S^*}_{\ge 1/k}$$

$$\le \log\left(1 + \sum_{S \ni v} (e-1)\cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \le z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \le -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

$$\log|\mathcal{U}^t| - \log|\mathcal{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathcal{U}^{t-1}| - |\mathcal{U}^t|}{|\mathcal{U}^{t-1}|}\right)$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x^*_S \log\left(\frac{x^*_S}{x^t_S}\right) - \sum_S x^*_S \log\left(\frac{x^*_S}{x^{t-1}_S}\right)$$

$$= \sum_S x^*_S \log\left(\frac{x^{t-1}_S}{x^t_S}\right)$$

$$= \cancel{\sum_S x^*_S}_{=1} \log \|x\|_1 - \sum_{S\ni v} x^*_S \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x^{t-1}_S}_{=1} + \sum_{S\ni v}(e-1)\cdot x^{t-1}_S\right) - \underbrace{\sum_{S\ni v} x^*_S}_{\ge 1/k}$$

$$\le \log\left(1 + \sum_{S\ni v}(e-1)\cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \le z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathscr{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

$$\log|\mathscr{U}^t| - \log|\mathscr{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathscr{U}^{t-1}| - |\mathscr{U}^t|}{|\mathscr{U}^{t-1}|}\right)$$

Use $\log(1-z) \le -z$.

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \underbrace{\cancel{\sum_S x_S^*}}_{=1} \log \|x\|_1 - \sum_{S \ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S \ni v} (e-1) \cdot x_S^{t-1}\right) - \underbrace{\sum_{S \ni v} x_S^*}_{\geq 1/k}$$

$$\leq \log\left(1 + \sum_{S \ni v} (e-1) \cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \leq z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

$$\log |\mathcal{U}^t| - \log |\mathcal{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathcal{U}^{t-1}| - |\mathcal{U}^t|}{|\mathcal{U}^{t-1}|}\right)$$

Use $\log(1-z) \leq -z$.

$$\leq -\frac{1}{|\mathcal{U}^{t-1}|} \sum_{v \in \mathcal{U}^{t-1}} \mathbb{1}\{R \ni v\}.$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \leq (e-1) \cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \underbrace{\cancel{\sum_S x_S^*}}_{=1} \log \|x\|_1 - \sum_{S \ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S \ni v} (e-1) \cdot x_S^{t-1}\right) - \underbrace{\sum_{S \ni v} x_S^*}_{\geq 1/k}$$

$$\leq \log\left(1 + \sum_{S \ni v} (e-1) \cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \leq z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log |\mathcal{U}^t|] \leq -E_v\left[\sum_{S \ni v} x_S\right].$$

Proof:

$$\log |\mathcal{U}^t| - \log |\mathcal{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathcal{U}^{t-1}| - |\mathcal{U}^t|}{|\mathcal{U}^{t-1}|}\right)$$

Use $\log(1-z) \leq -z$.

$$\leq -\frac{1}{|\mathcal{U}^{t-1}|} \sum_{v \in \mathcal{U}^{t-1}} \mathbb{1}\{R \ni v\}.$$

Take expectation over R.

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S \ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\sum_S x_S^*}_{=1} \log \|x\|_1 - \sum_{S\ni v} x_S^* \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S\ni v}(e-1)\cdot x_S^{t-1}\right) - \underbrace{\sum_{S\ni v} x_S^*}_{\ge 1/k}$$

$$\le \log\left(1 + \sum_{S\ni v}(e-1)\cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \le z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

$$\log|\mathcal{U}^t| - \log|\mathcal{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathcal{U}^{t-1}| - |\mathcal{U}^t|}{|\mathcal{U}^{t-1}|}\right)$$

Use $\log(1-z) \le -z$.

$$\le -\frac{1}{|\mathcal{U}^{t-1}|}\sum_{v\in\mathcal{U}^{t-1}} \mathbb{1}\{R \ni v\}.$$

Take expectation over R.

$$E_R[\Delta \log|\mathcal{U}^t|] \le -\frac{1}{|\mathcal{U}^{t-1}|}\sum_R x_R \sum_{v\in\mathcal{U}^{t-1}} \mathbb{1}\{R\ni v\}$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \underbrace{\cancel{\sum_S x_S^*}}_{=1} \log \|x\|_1 - \sum_{S\ni v} x_S^* \underbrace{\cancel{\log e}}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S\ni v}(e-1)\cdot x_S^{t-1}\right) - \underbrace{\sum_{S\ni v} x_S^*}_{\ge 1/k}$$

$$\le \log\left(1 + \sum_{S\ni v}(e-1)\cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \le z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathcal{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

$$\log|\mathcal{U}^t| - \log|\mathcal{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathcal{U}^{t-1}| - |\mathcal{U}^t|}{|\mathcal{U}^{t-1}|}\right)$$

Use $\log(1-z) \le -z$.

$$\le -\frac{1}{|\mathcal{U}^{t-1}|}\sum_{v\in\mathcal{U}^{t-1}} \mathbb{1}\{R \ni v\}.$$

Take expectation over R.

$$E_R[\Delta \log|\mathcal{U}^t|] \le -\frac{1}{|\mathcal{U}^{t-1}|}\sum_R x_R \sum_{v\in\mathcal{U}^{t-1}} \mathbb{1}\{R\ni v\}$$

$$= -\frac{1}{|\mathcal{U}^{t-1}|}\sum_{v\in\mathcal{U}^{t-1}}\sum_{R\ni v} x_R.$$

Claim 2a: If v^t uncovered,

$$E_v[\Delta \mathrm{KL}] \le (e-1)\cdot E_v\left[\sum_{S\ni v} x_S\right] - \frac{1}{k}.$$

Proof:

$$\sum_S x_S^* \log\left(\frac{x_S^*}{x_S^t}\right) - \sum_S x_S^* \log\left(\frac{x_S^*}{x_S^{t-1}}\right)$$

$$= \sum_S x_S^* \log\left(\frac{x_S^{t-1}}{x_S^t}\right)$$

$$= \cancel{\sum_S x_S^*}_{=1} \log \|x\|_1 - \sum_{S\ni v} x_S^* \cancel{\log e}_{=1}$$

$$= \log\left(\underbrace{\sum_S x_S^{t-1}}_{=1} + \sum_{S\ni v}(e-1)\cdot x_S^{t-1}\right) - \underbrace{\sum_{S\ni v} x_S^*}_{\ge 1/k}$$

$$\le \log\left(1 + \sum_{S\ni v}(e-1)\cdot x_S\right) - \frac{1}{k}.$$

Use $\log(1+z) \le z$, take expectation over v, $\blacksquare$.

Claim 2b: If v^t uncovered,

$$E_R[\Delta \log|\mathscr{U}^t|] \le -E_v\left[\sum_{S\ni v} x_S\right].$$

Proof:

$$\log|\mathscr{U}^t| - \log|\mathscr{U}^{t-1}|$$

$$= \log\left(1 - \frac{|\mathscr{U}^{t-1}| - |\mathscr{U}^t|}{|\mathscr{U}^{t-1}|}\right)$$

Use $\log(1-z) \le -z$.

$$\le -\frac{1}{|\mathscr{U}^{t-1}|}\sum_{v\in\mathscr{U}^{t-1}} \mathbb{1}\{R\ni v\}.$$

Take expectation over R.

$$E_R[\Delta \log|\mathscr{U}^t|] \le -\frac{1}{|\mathscr{U}^{t-1}|}\sum_R x_R \sum_{v\in\mathscr{U}^{t-1}} \mathbb{1}\{R\ni v\}$$

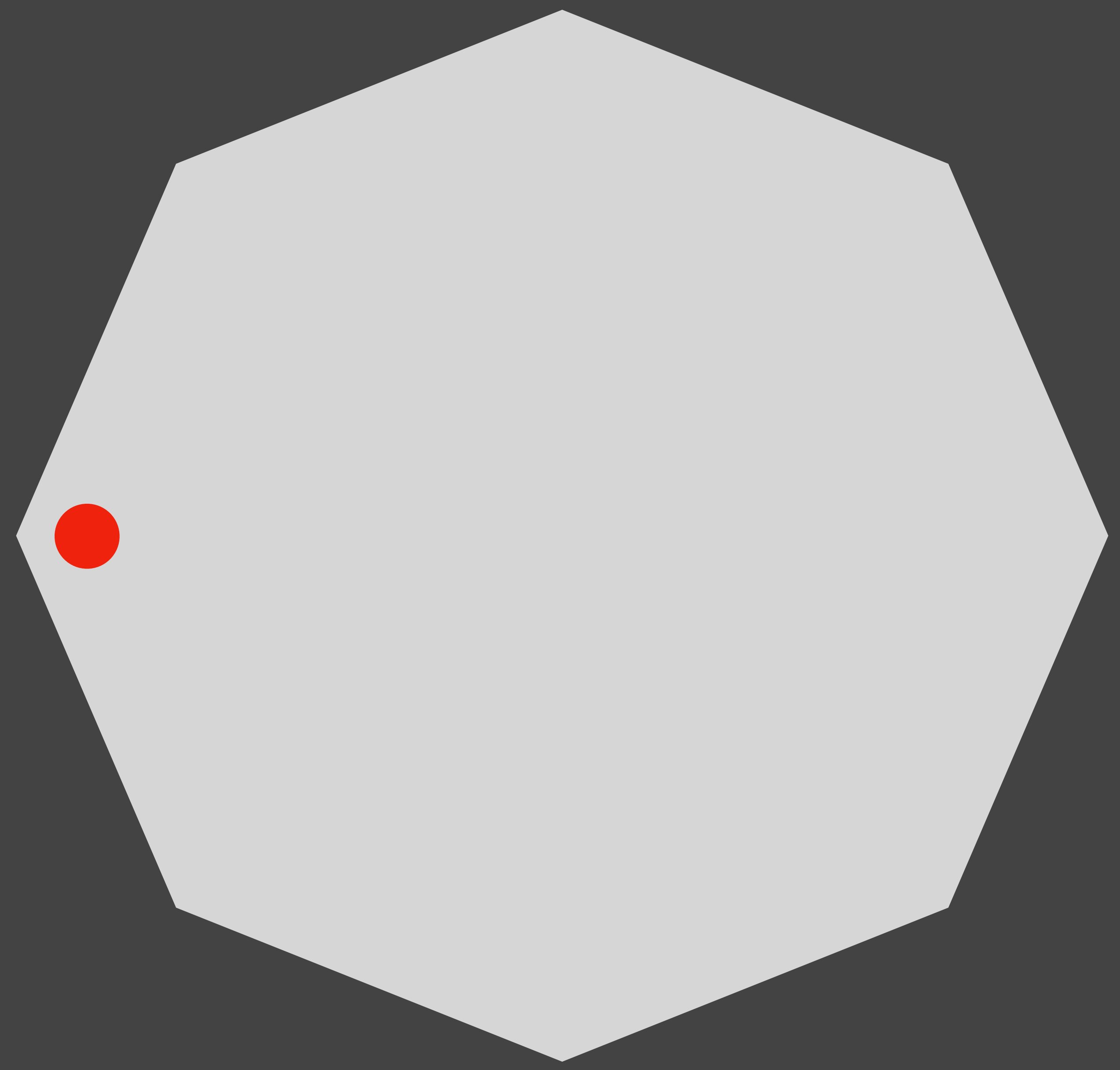
$$= -\frac{1}{|\mathscr{U}^{t-1}|}\sum_{v\in\mathscr{U}^{t-1}}\sum_{R\ni v} x_R. \quad \blacksquare$$

LearnOrCover

(Some philosophy)

LearnOrCover
(Some philosophy)

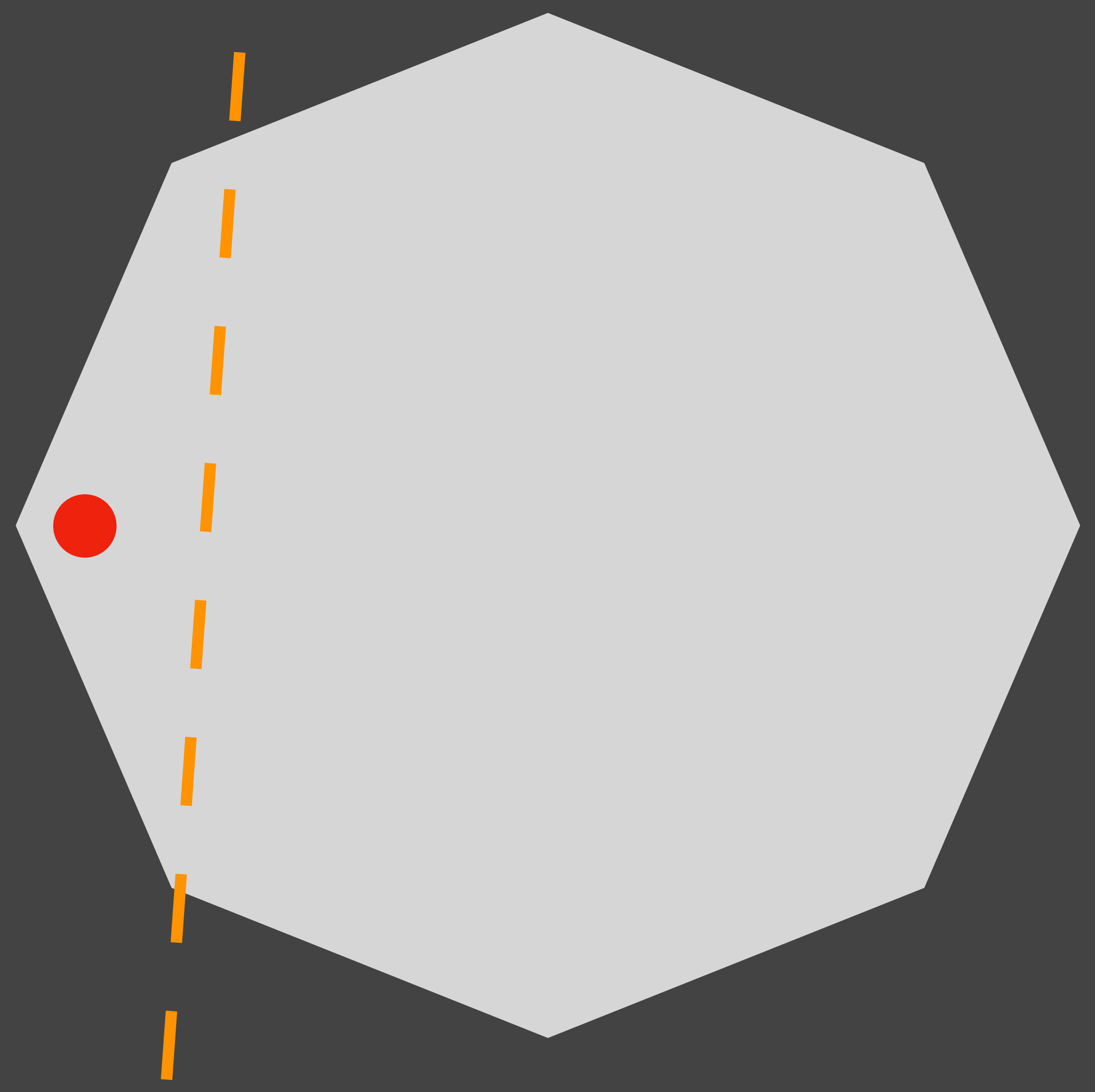
Perspective 1:



[Alon+ 03]
[Buchbinder Gupta Molinaro Naor 19]

LearnOrCover
(Some philosophy)

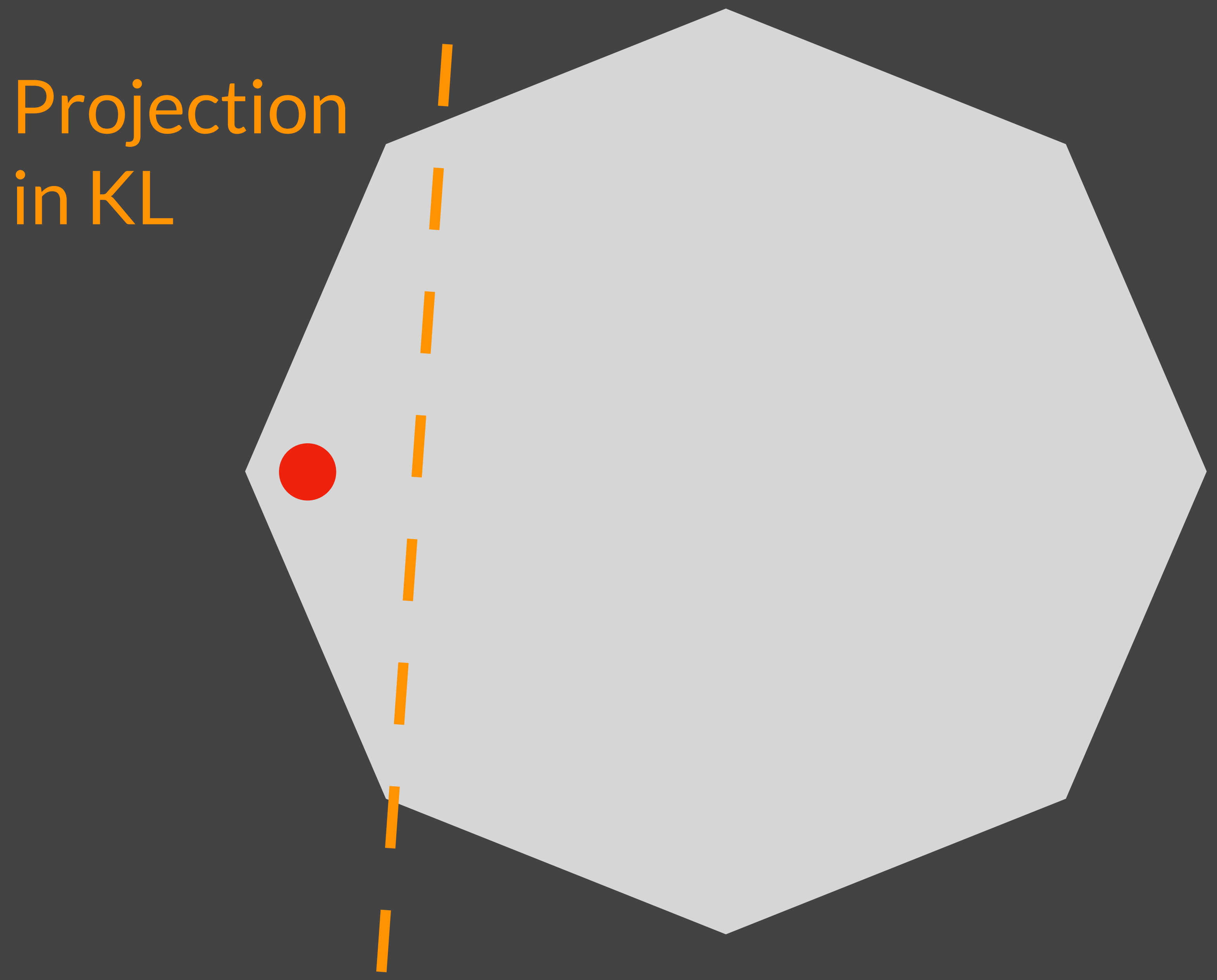
Perspective 1:



[Alon+ 03]
[Buchbinder Gupta Molinaro Naor 19]

LearnOrCover
(Some philosophy)

Perspective 1:

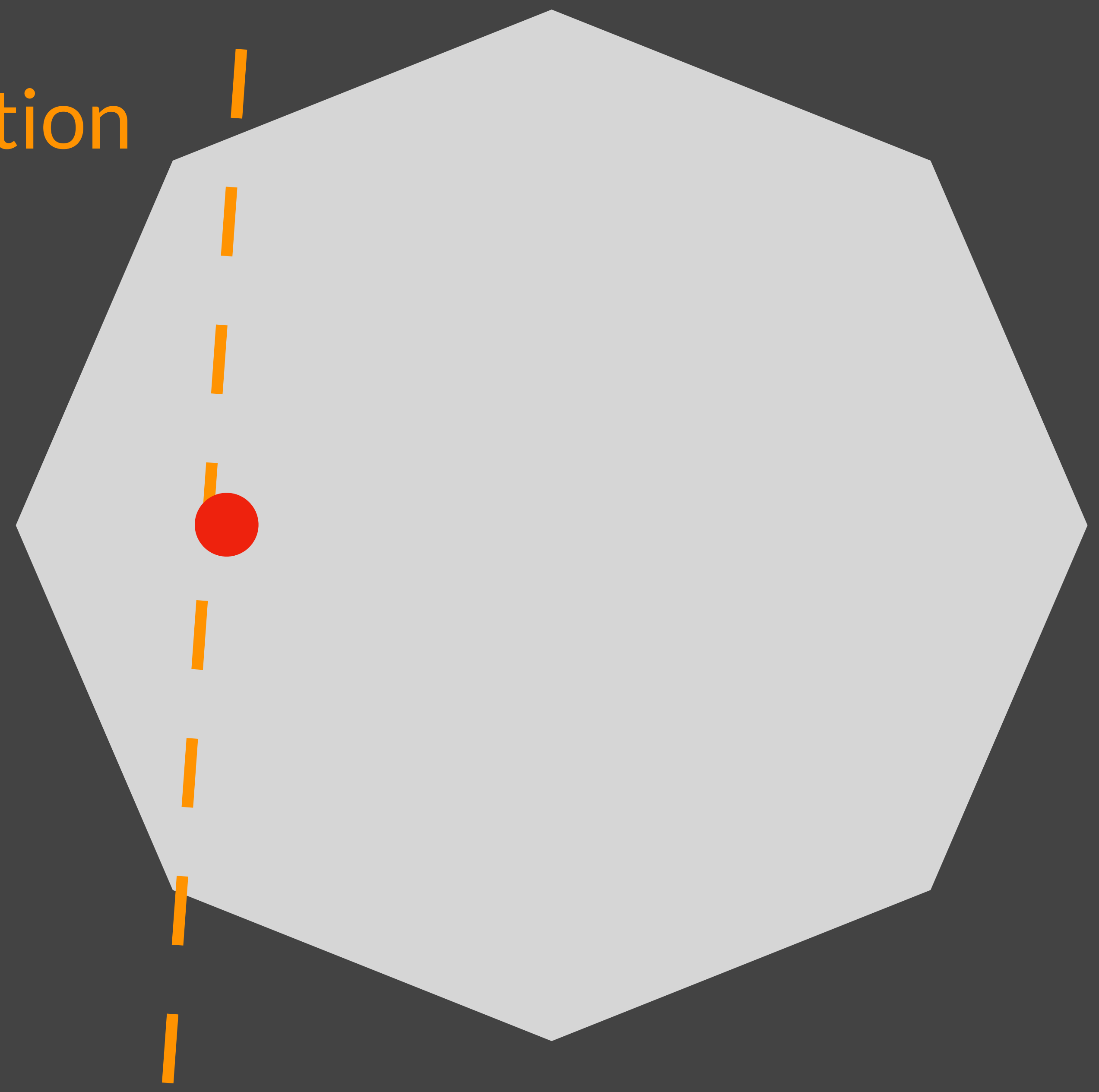


[Alon+ 03]
[Buchbinder Gupta Molinaro Naor 19]

LearnOrCover

(Some philosophy)

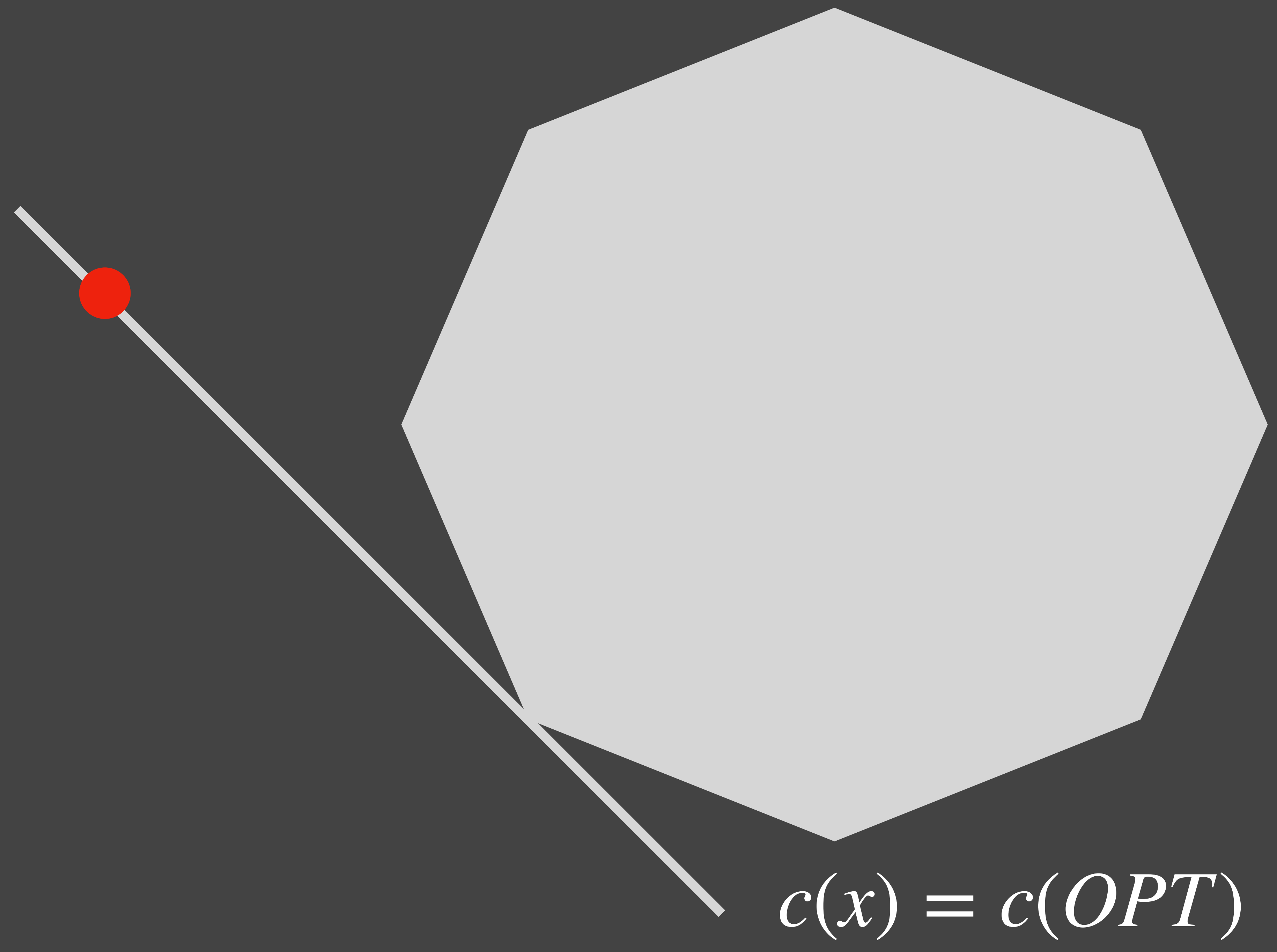
Perspective 1:



[Alon+ 03]
[Buchbinder Gupta Molinaro Naor 19]

LearnOrCover
(Some philosophy)

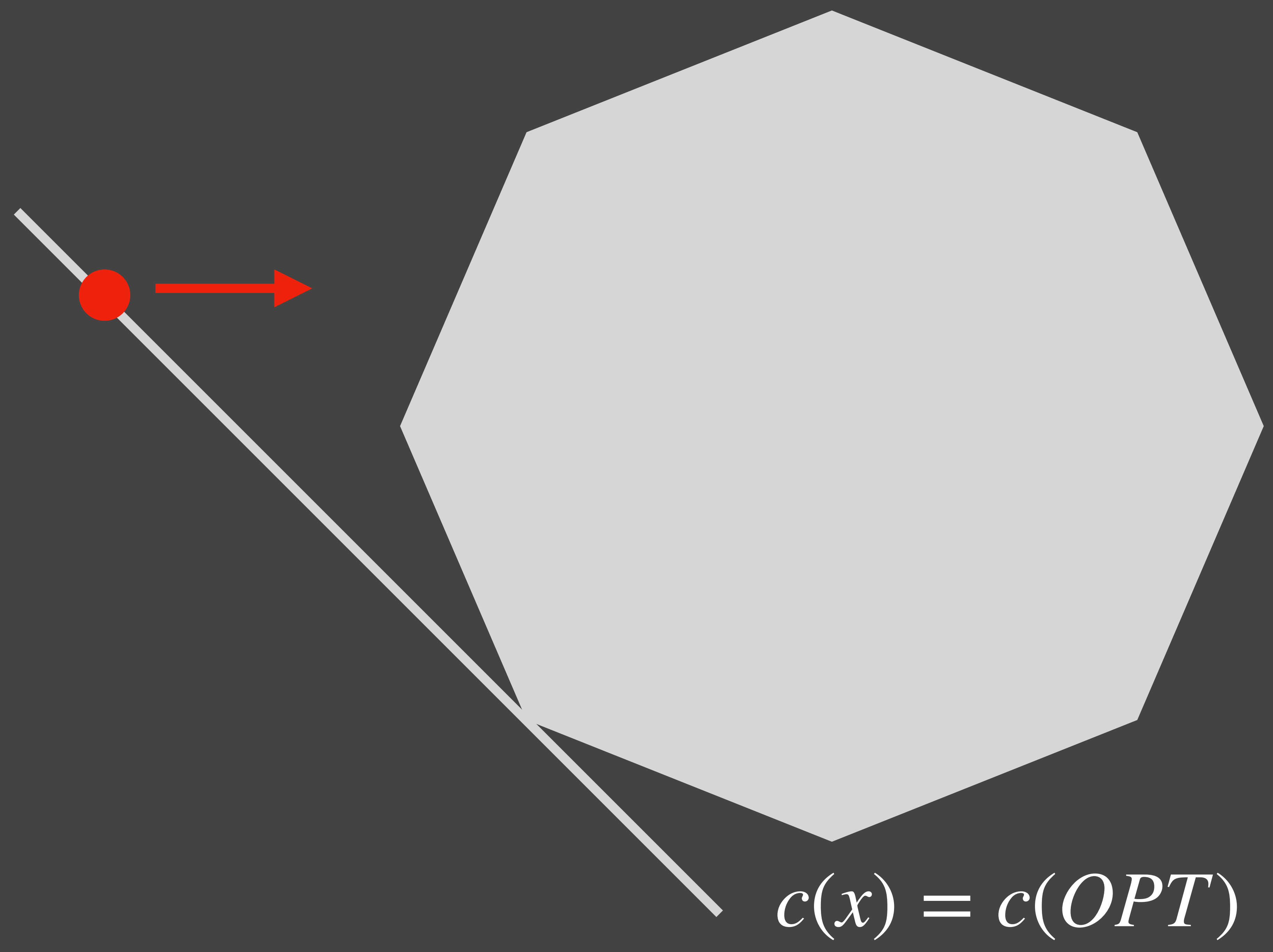
Perspective 1:



LearnOrCover

(Some philosophy)

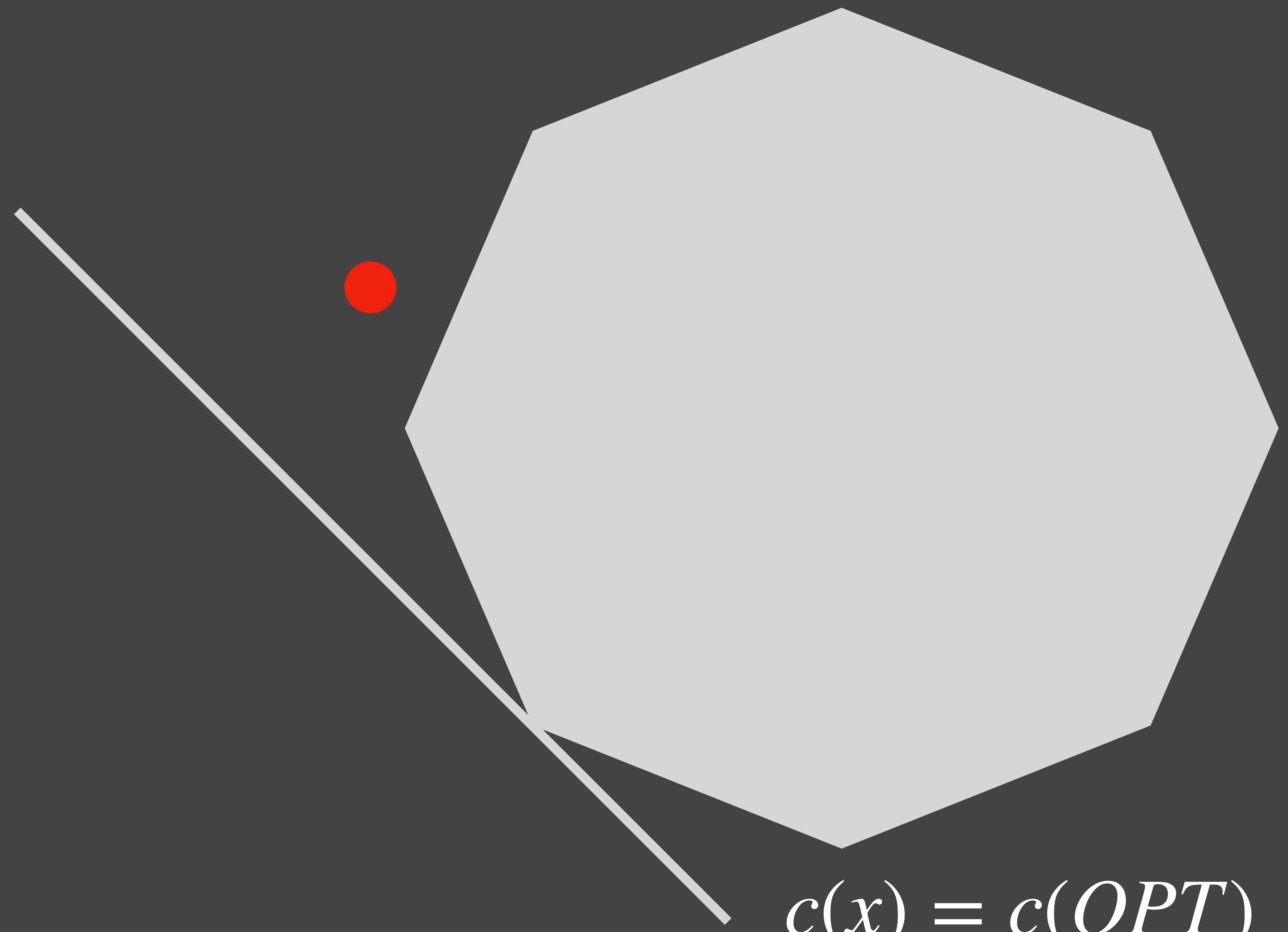
Perspective 1:



LearnOrCover

(Some philosophy)

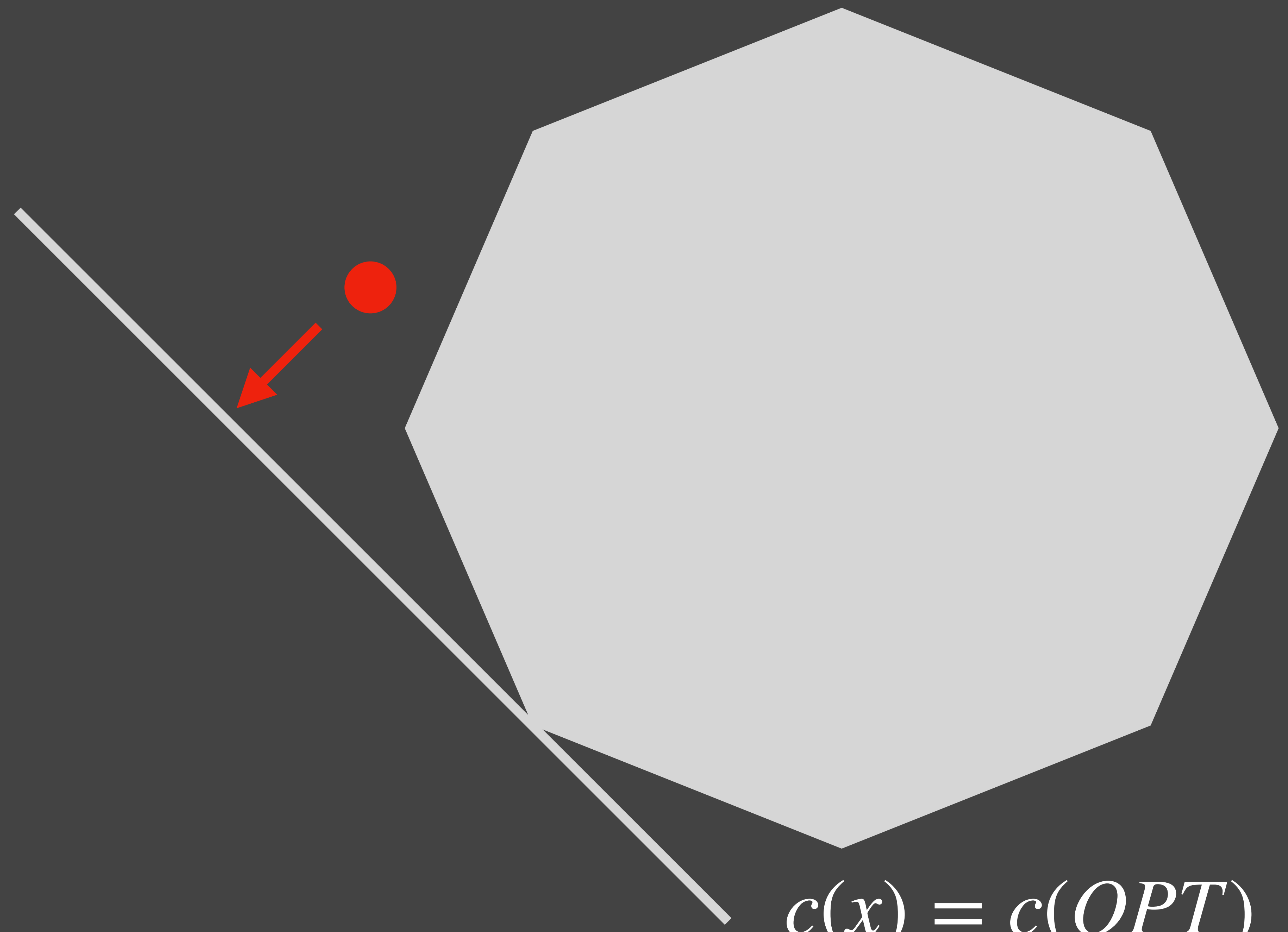
Perspective 1:



$c(x) = c(OPT)$

LearnOrCover
(Some philosophy)

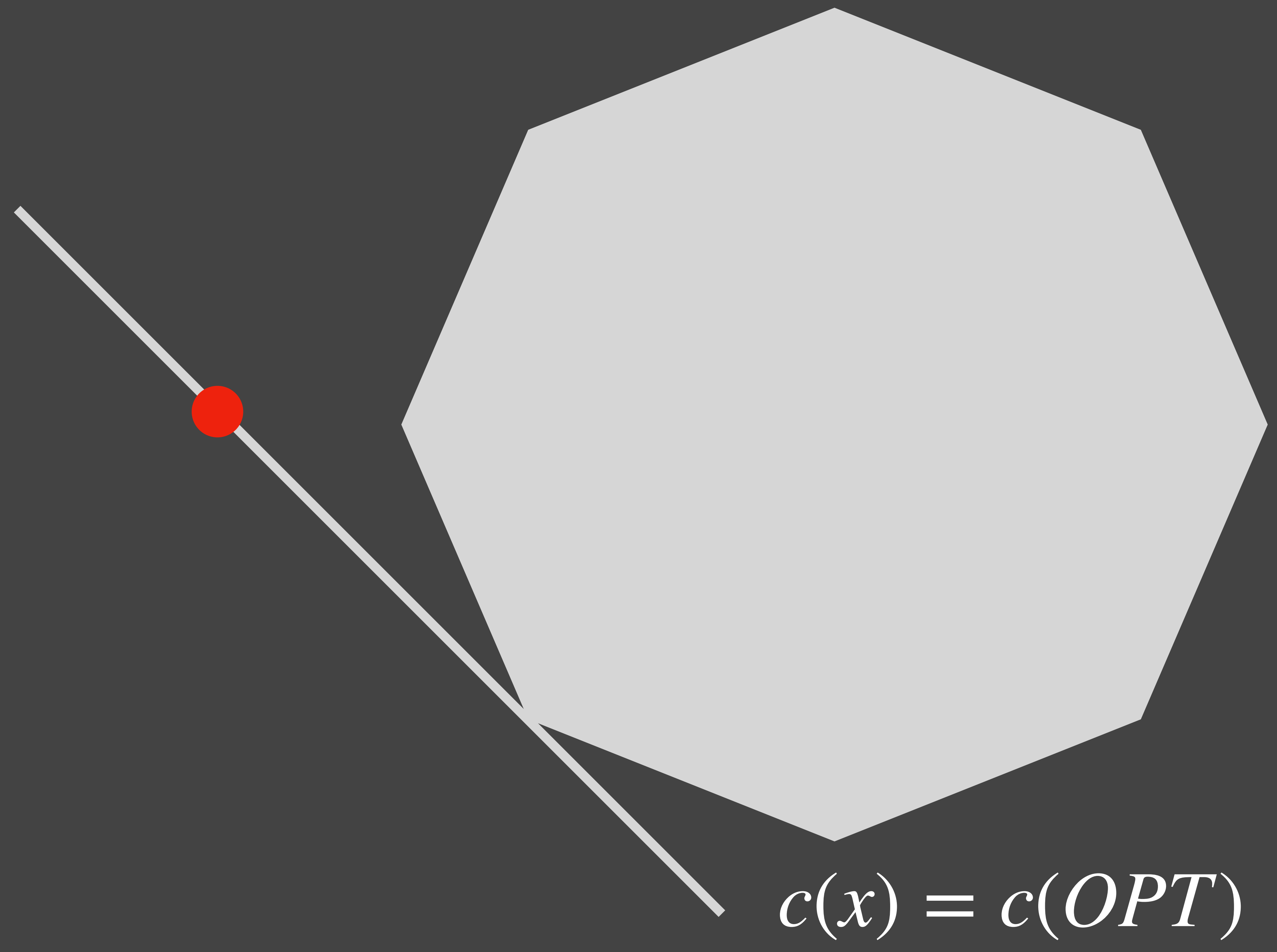
Perspective 1:



$c(x) = c(OPT)$

LearnOrCover
(Some philosophy)

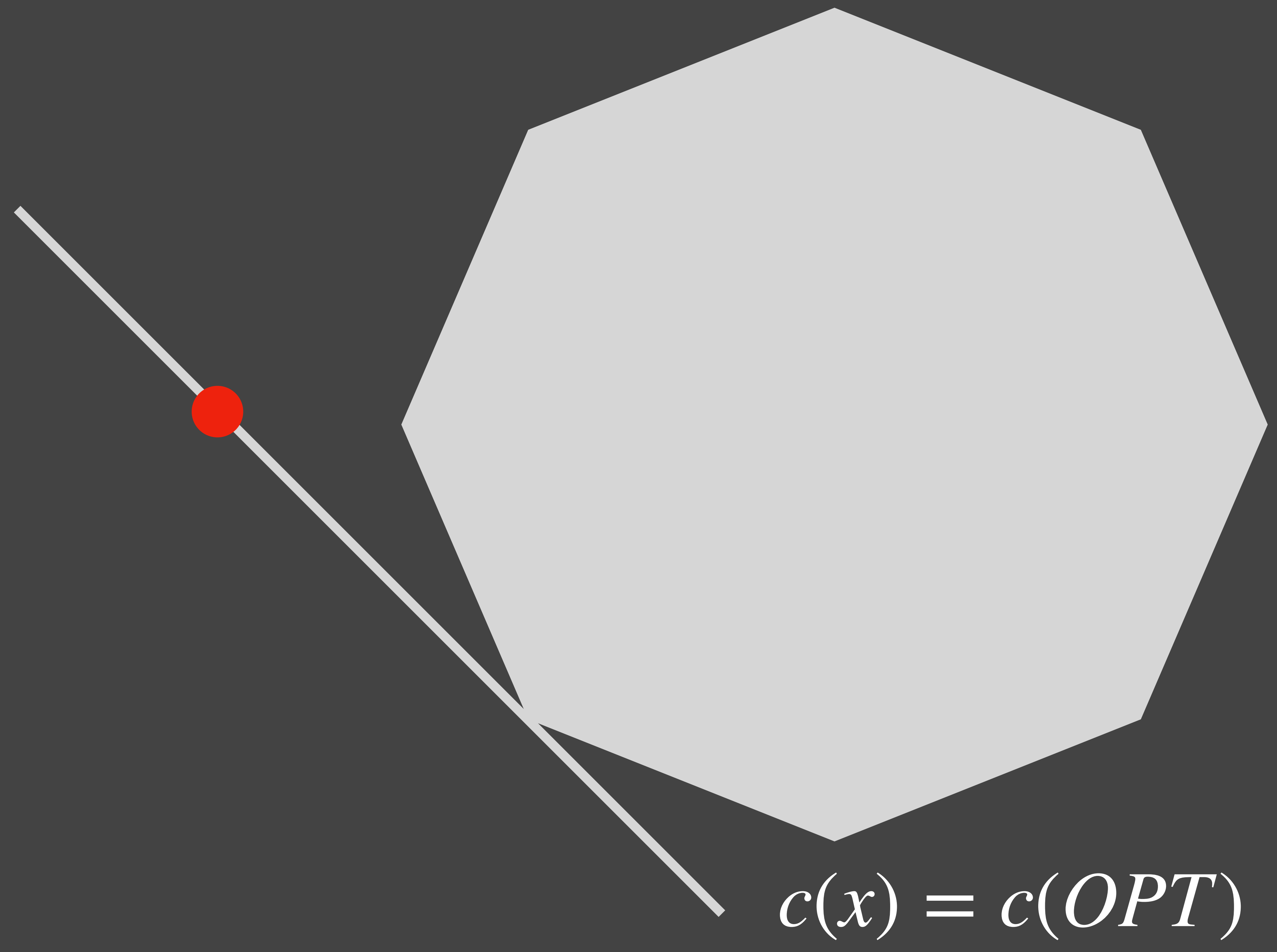
Perspective 1:



LearnOrCover

(Some philosophy)

Perspective 1:



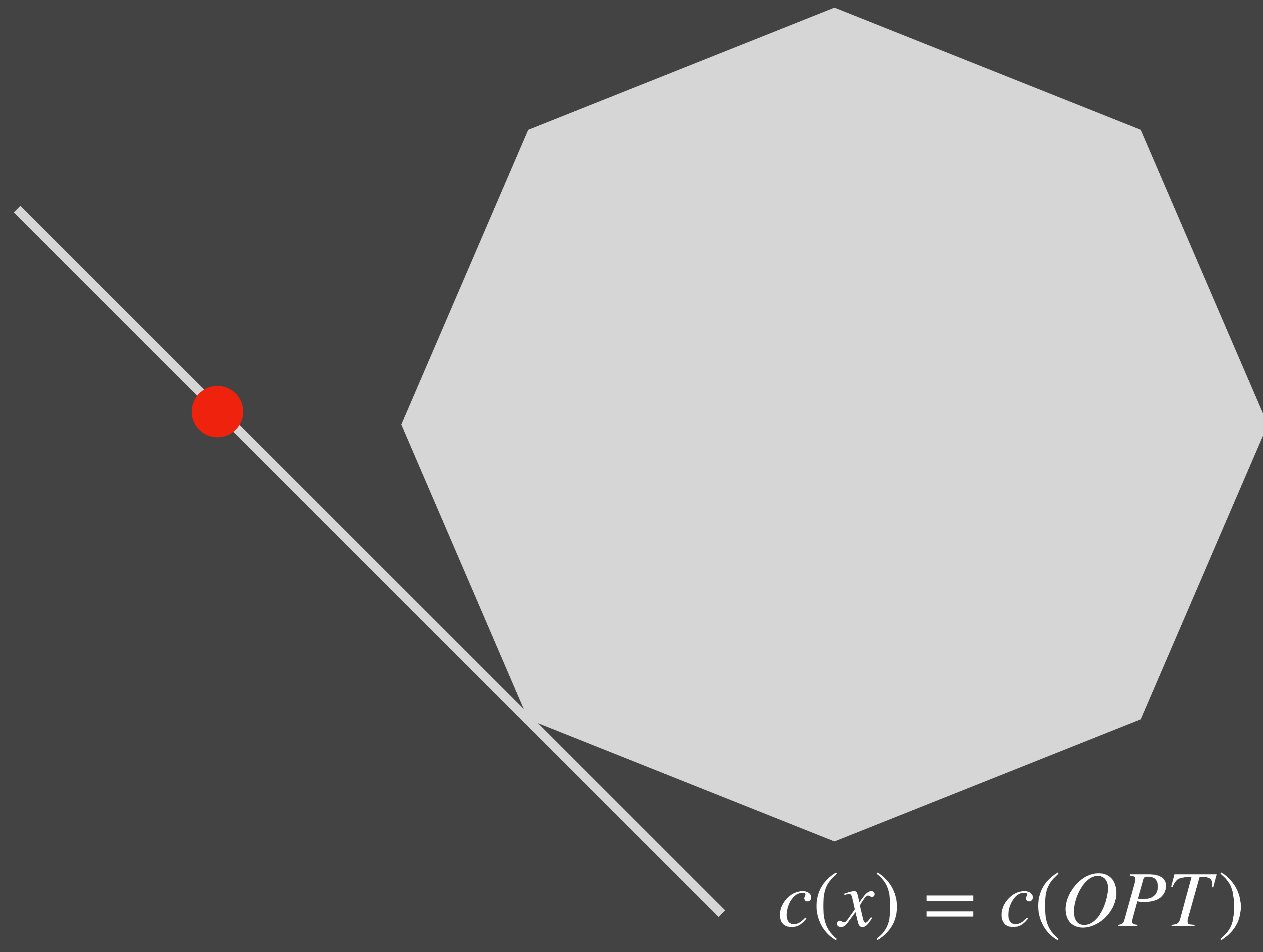
LearnOrCover

Perspective 2:

LearnOrCover

(Some philosophy)

<u>Perspective 1:</u>



LearnOrCover

<u>Perspective 2:</u>

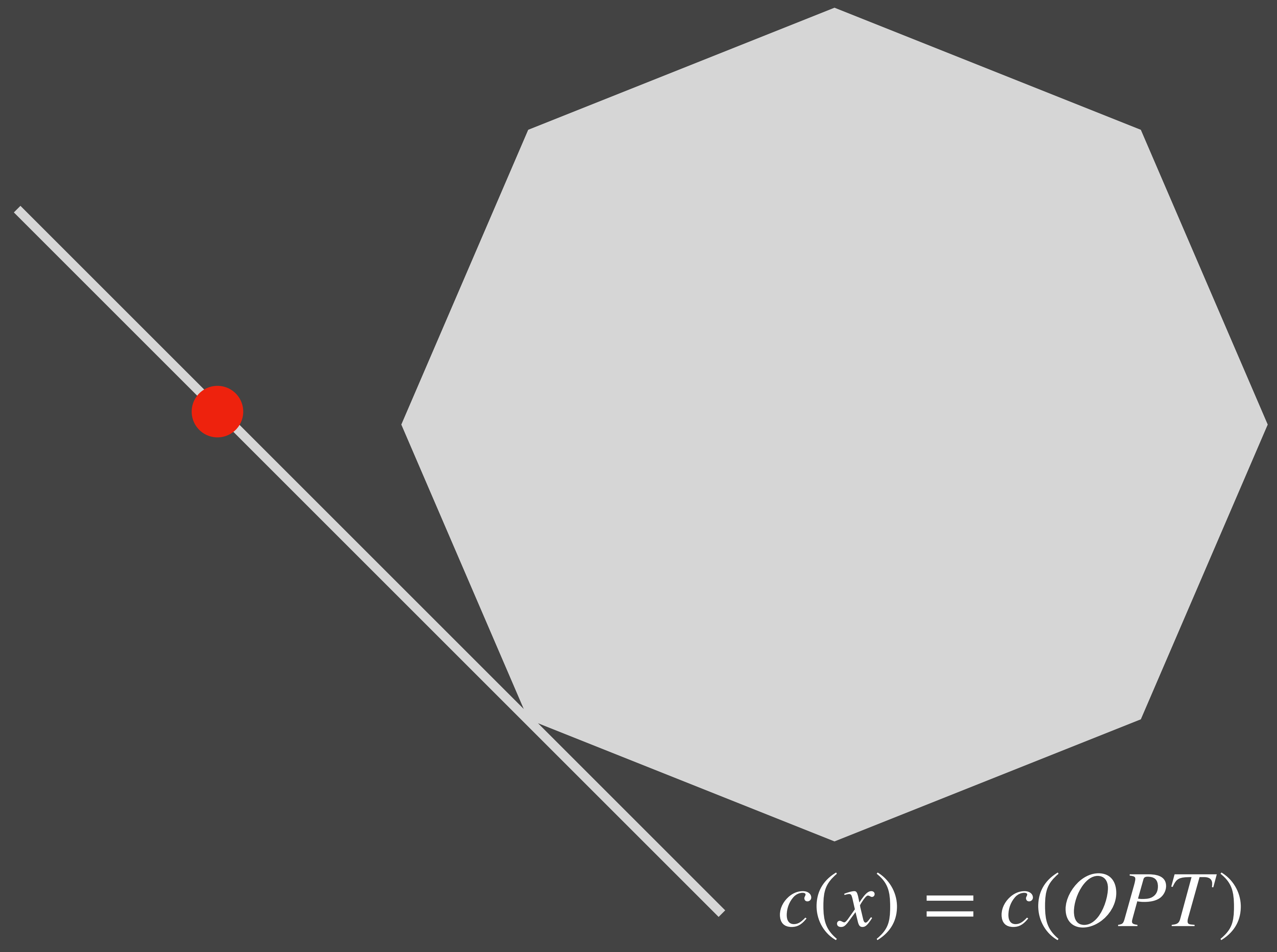
Define

$$f(x) := \sum_{v} \max\left(0, 1 - \sum_{S \ni v} x_S\right)$$

LearnOrCover

(Some philosophy)

Perspective 1:



LearnOrCover

Perspective 2:

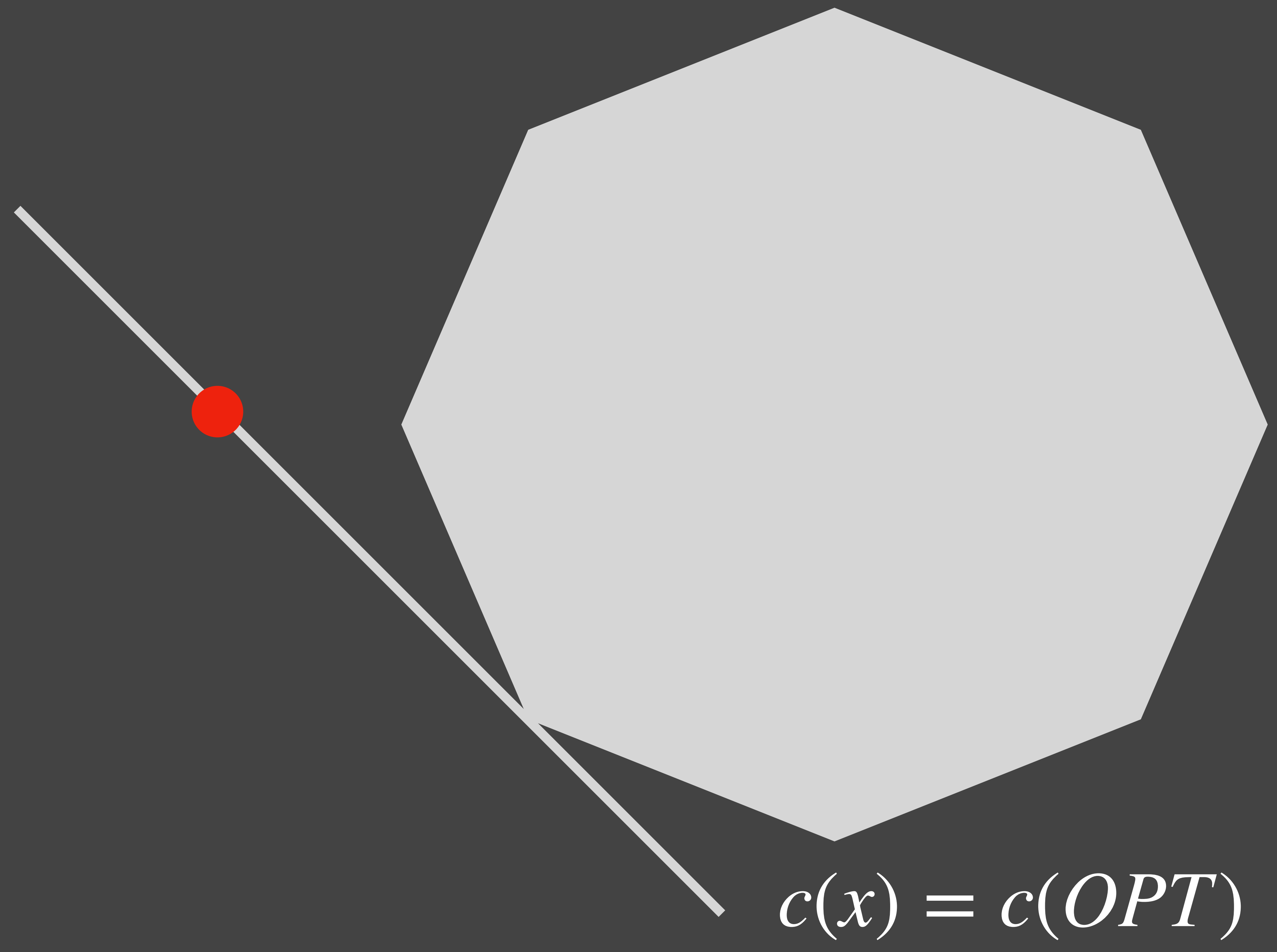
Define

$$f(x) := \sum_{v} \max\left(0, 1 - \sum_{S \ni v} x_S\right)$$

(Goal is to minimize f in smallest # of steps)

LearnOrCover
(Some philosophy)

Perspective 1:



LearnOrCover

Perspective 2:

Define

$$f(x) := \sum_{v} \max\left(0, 1 - \sum_{S \ni v} x_S\right)$$

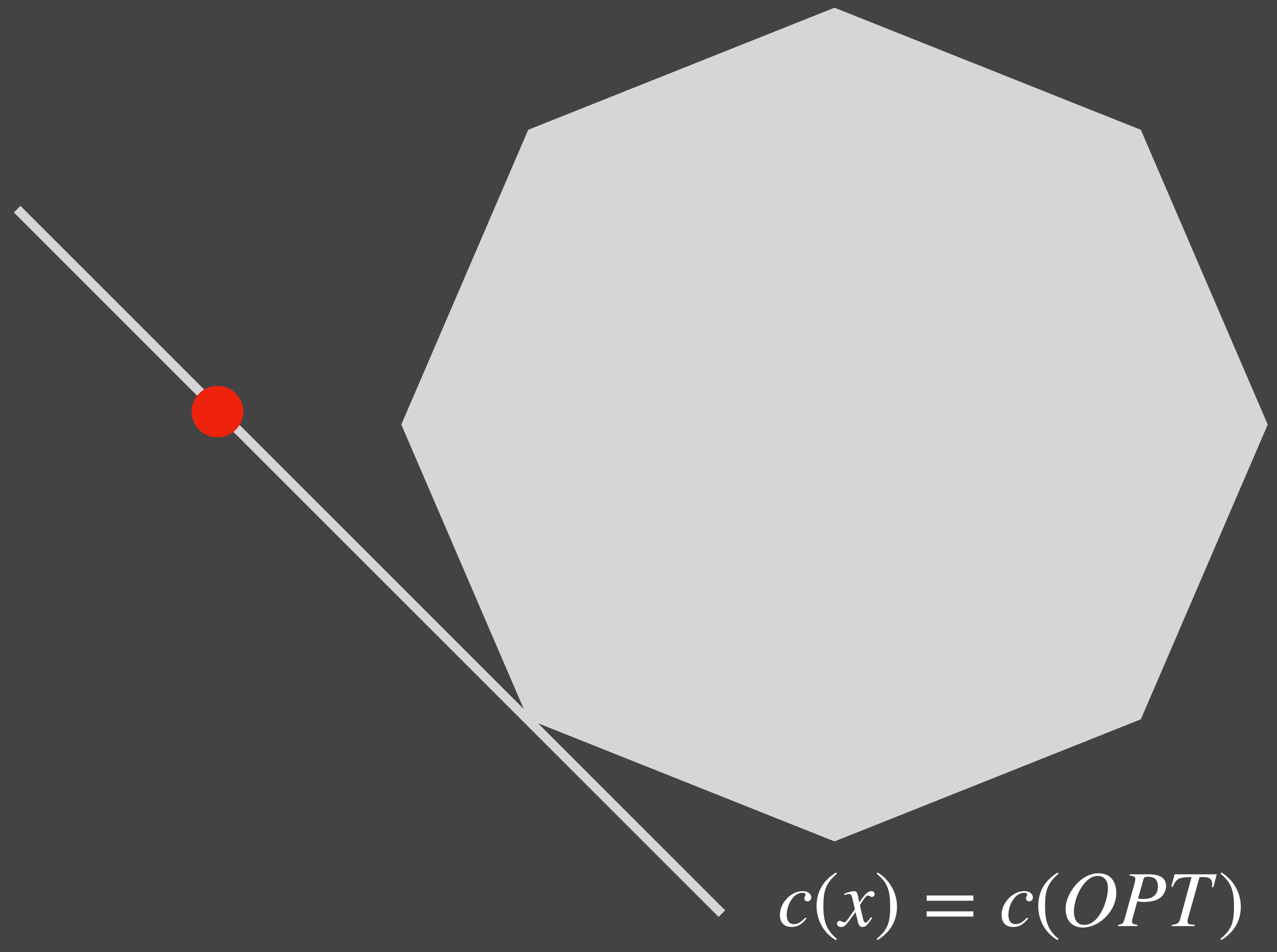
(Goal is to minimize f in smallest # of steps)

$\nabla f|_S(x) =$ # uncovered elements in S

LearnOrCover

(Some philosophy)

Perspective 1:



LearnOrCover

Perspective 2:

Define

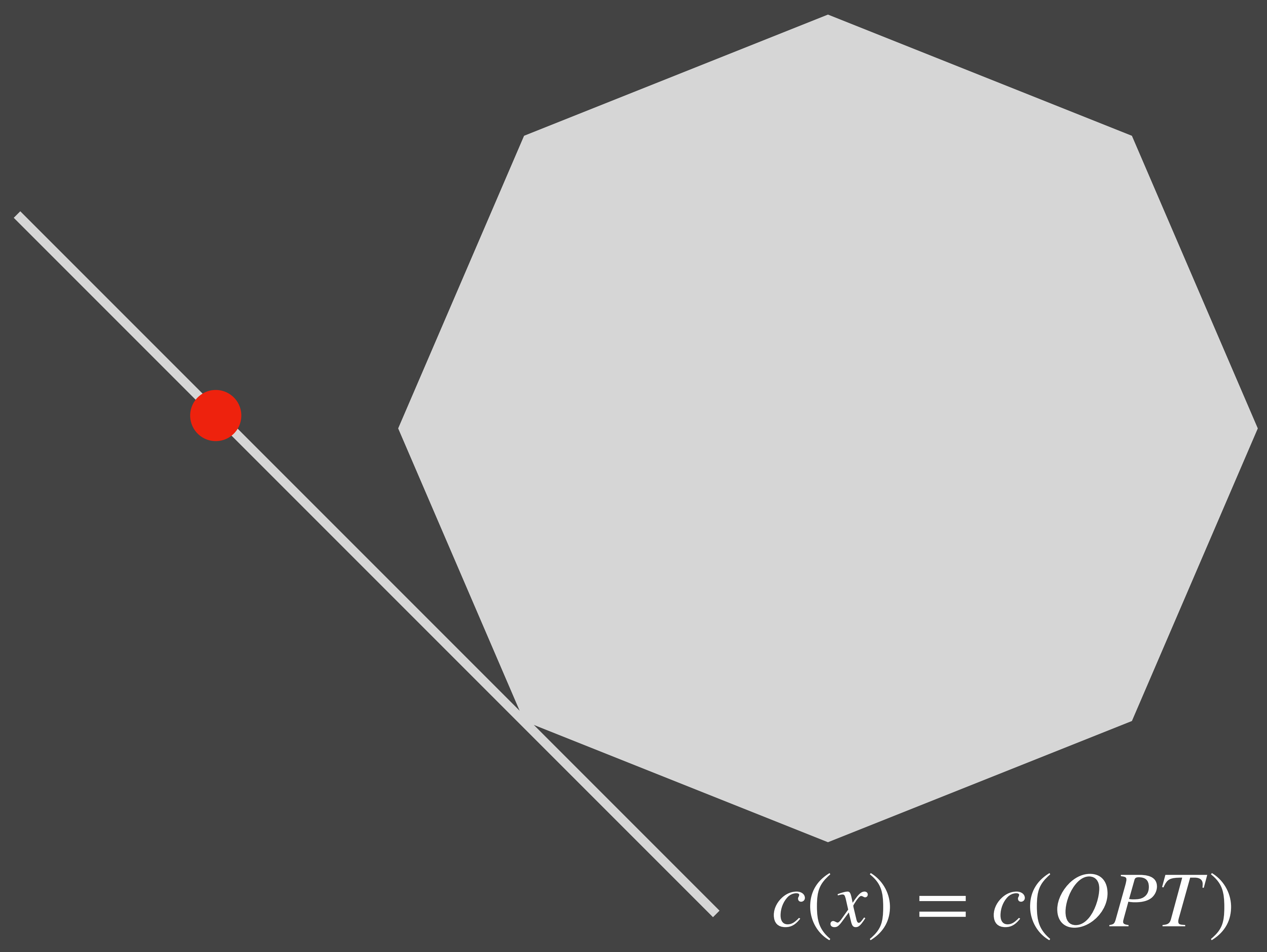
$$f(x) := \sum_{v} \max\left(0, 1 - \sum_{S \ni v} x_S\right)$$

(Goal is to minimize f in smallest # of steps)

$$\nabla f|_S(x) = \text{\# uncovered elements in } S$$
$$\propto E[\mathbb{1}\{v \in S \mid v \text{ uncovered}\}]$$

LearnOrCover
(Some philosophy)

Perspective 1:



LearnOrCover

Perspective 2:

Define

$$f(x) := \sum_v \max\left(0, 1 - \sum_{S \ni v} x_S\right)$$

(Goal is to minimize f in smallest # of steps)

$$\nabla f|_S(x) = \text{ \# uncovered elements in } S$$
$$\propto E[\mathbb{1}\{v \in S \mid v \text{ uncovered}\}]$$

RO reveals stochastic gradient…

Talk Outline

Intro

Secretary

LearnOrCover in Exponential Time

LearnOrCover in Poly Time

(Single Sample) Prophet

Conclusion & Extensions

Talk Outline

Intro

Secretary

LearnOrCover in Exponential Time

LearnOrCover in Poly Time

➡ (Single Sample) Prophet

Conclusion & Extensions

Special Case: the With-a-Sample model

Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

Remaining fraction revealed in <u>adversarial order.</u>

Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

Remaining fraction revealed in adversarial order.

S_1 ●

S_2 ●

S_3 ●

S_4 ●

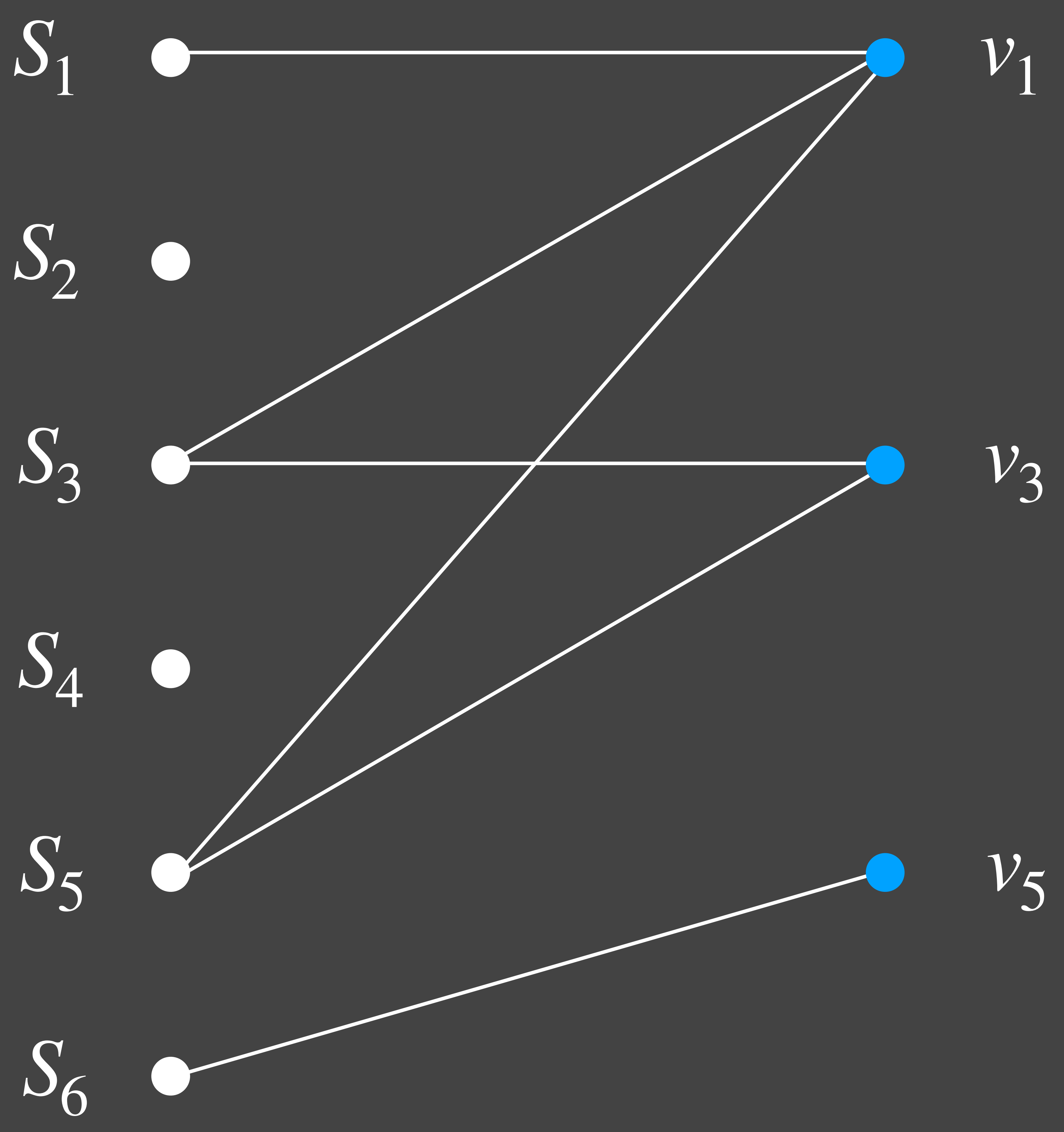
S_5 ●

S_6 ●

Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

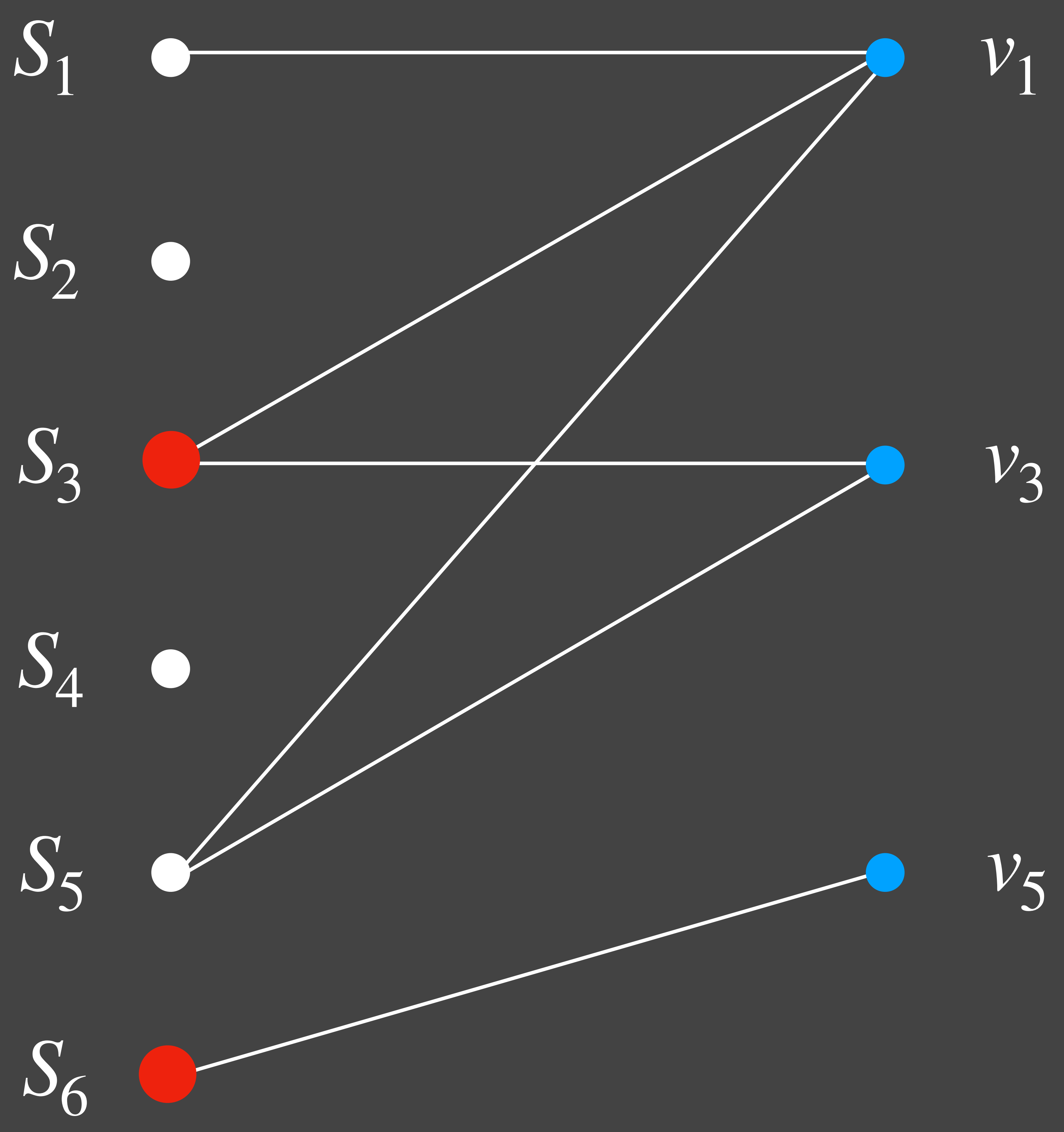
Remaining fraction revealed in adversarial order.



Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

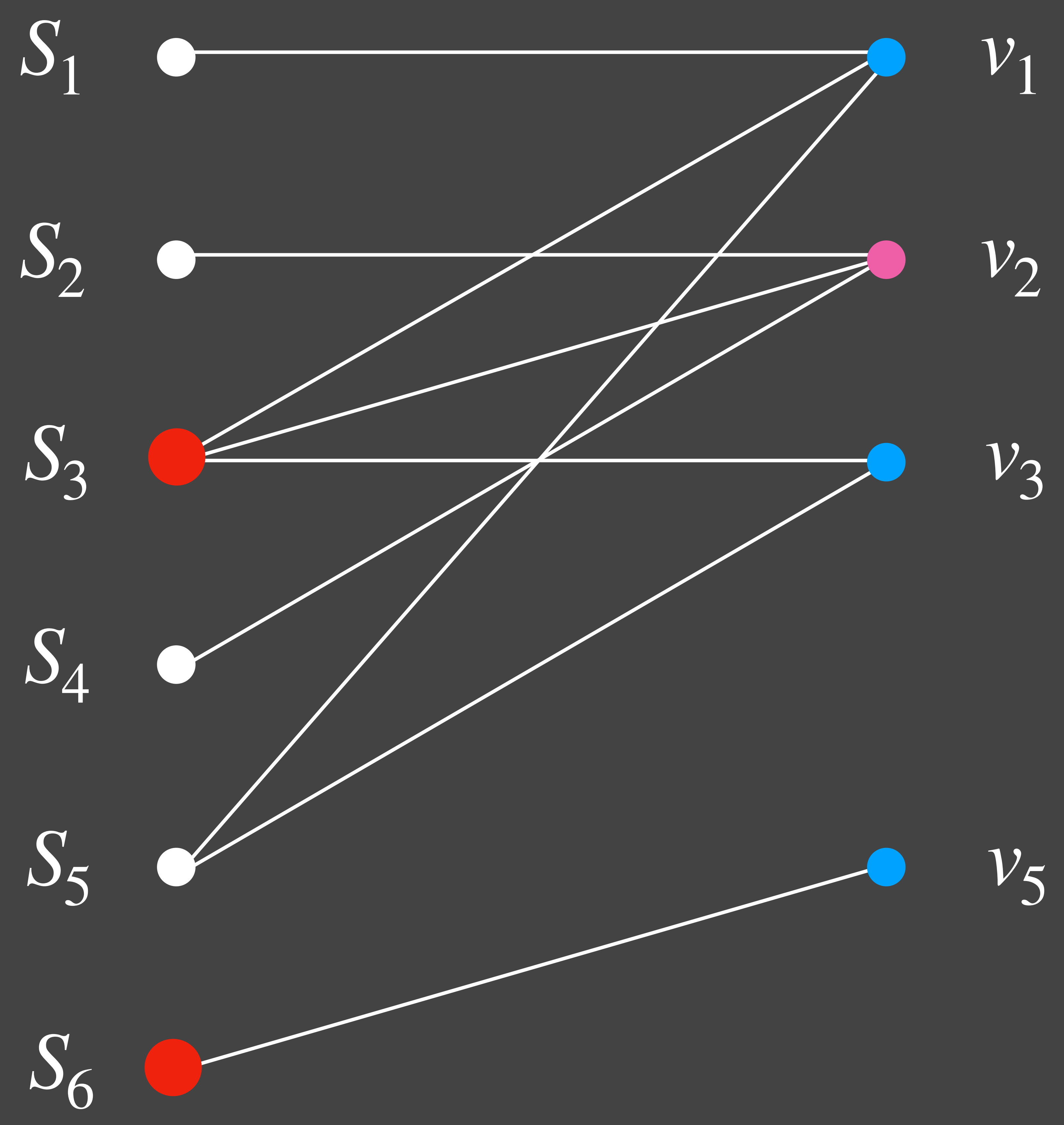
Remaining fraction revealed in adversarial order.



Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

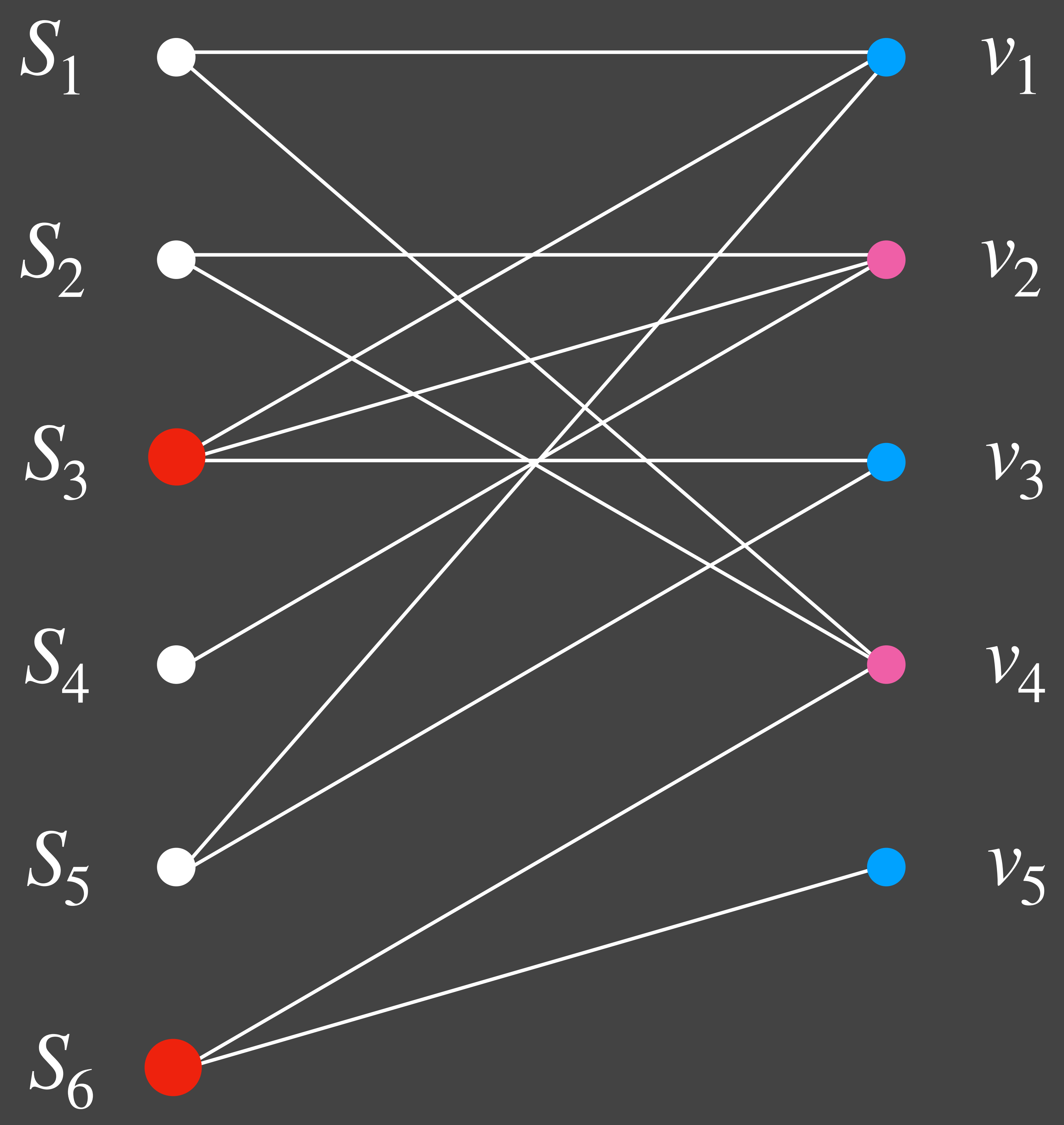
Remaining fraction revealed in adversarial order.



Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

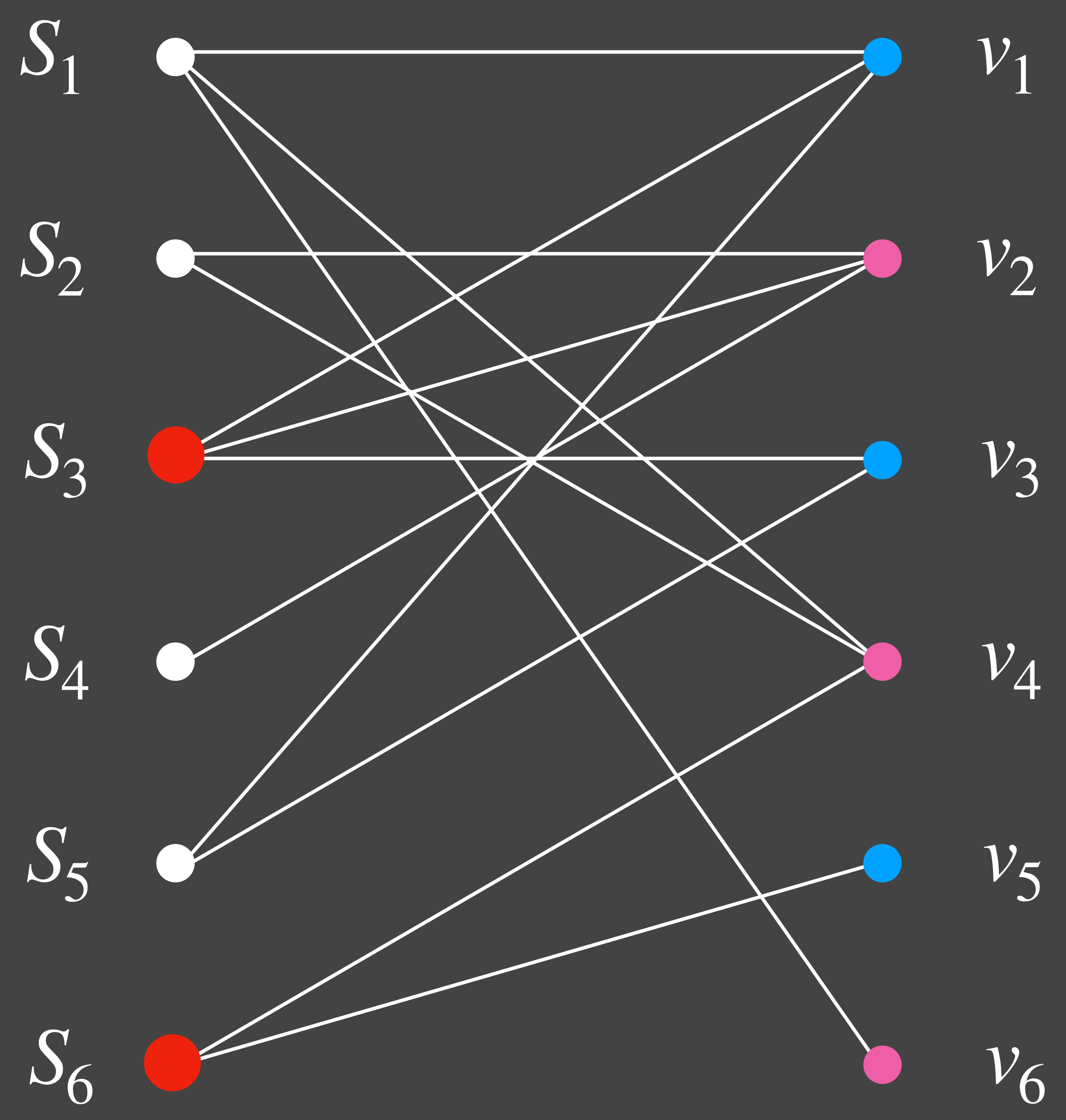
Remaining fraction revealed in adversarial order.



Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

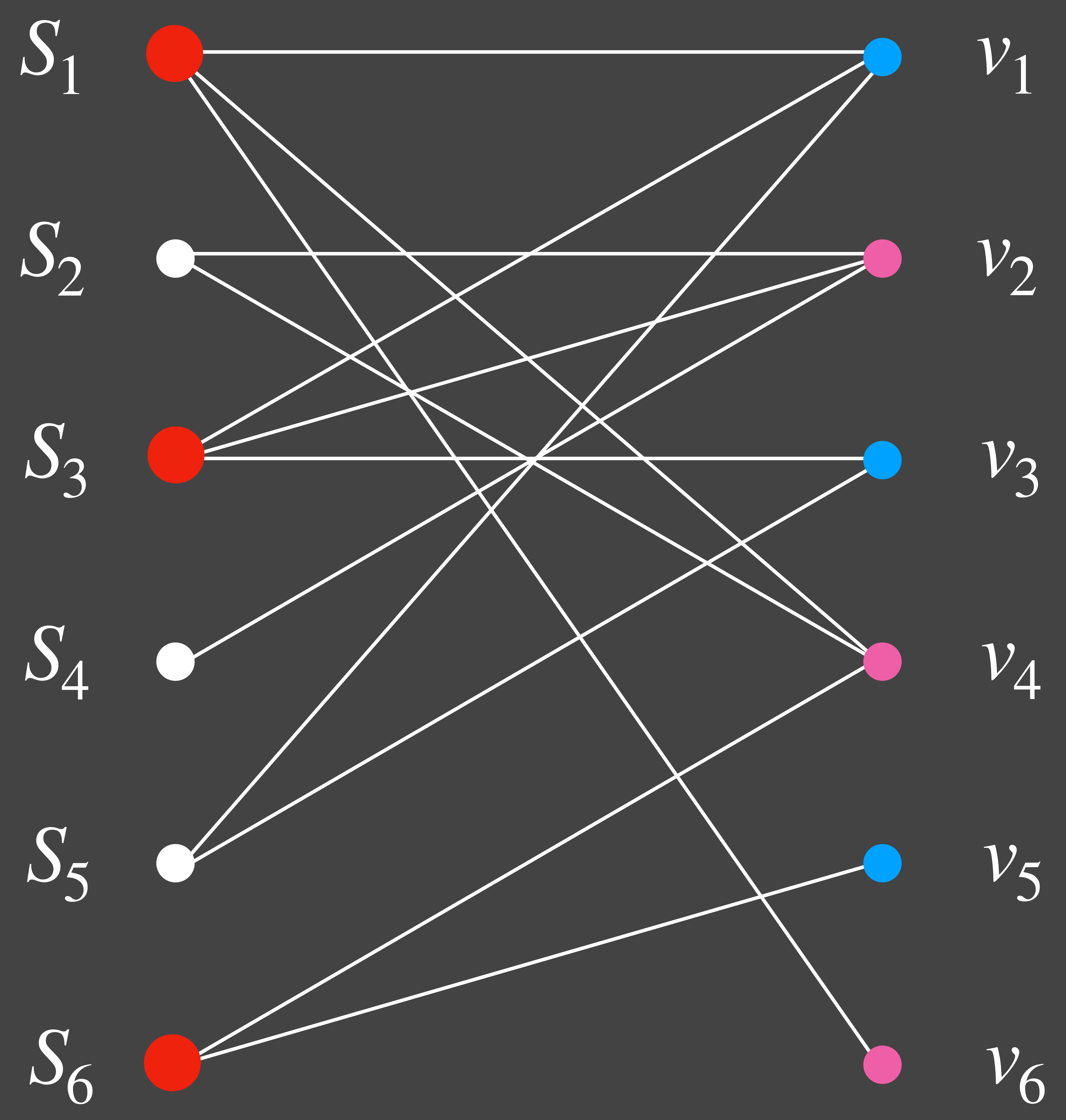
Remaining fraction revealed in <u>adversarial order.</u>



Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

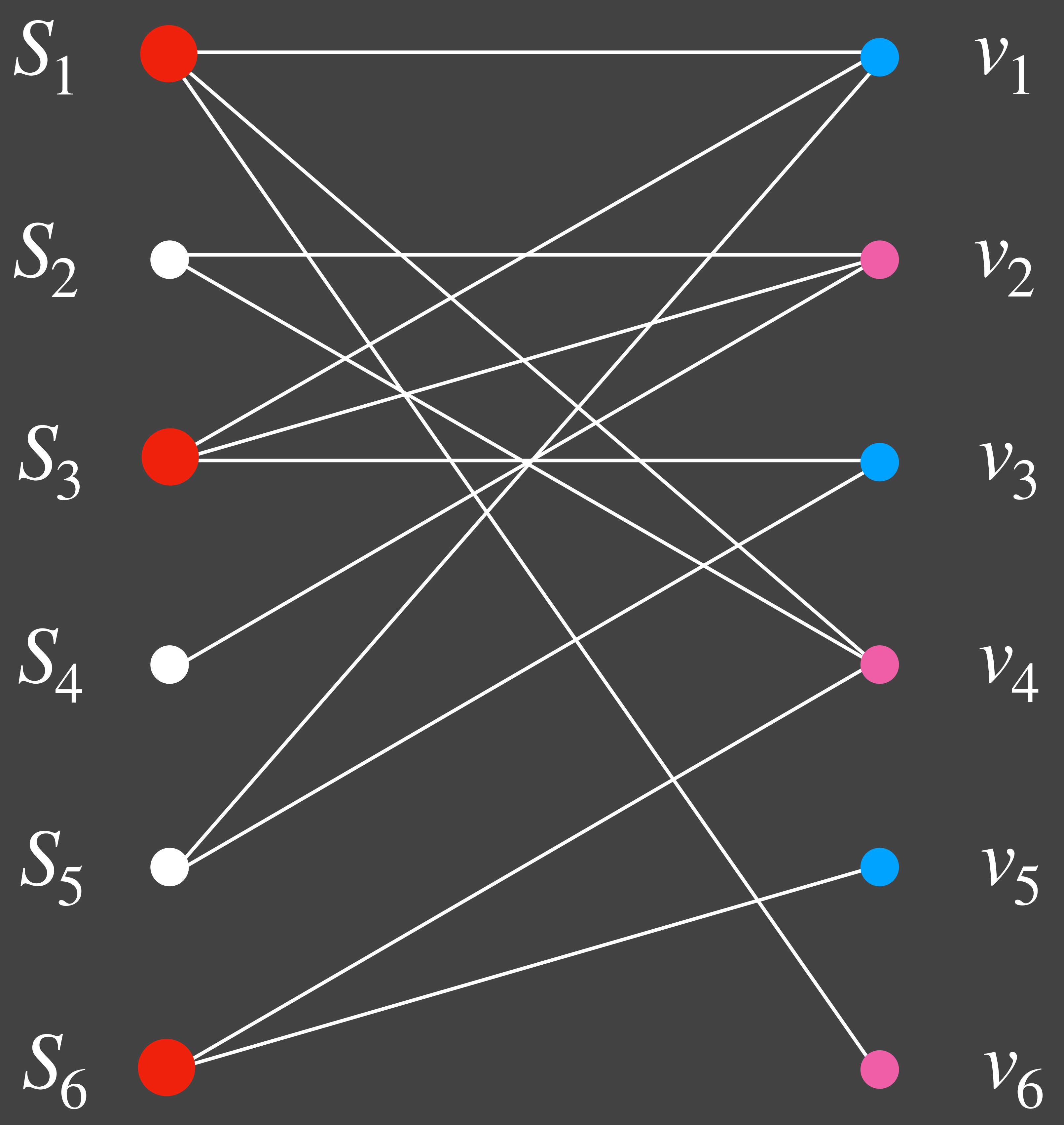
Remaining fraction revealed in <u>adversarial order.</u>



Special Case: the With-a-Sample model

Online set cover, but random 1/2 of elements known upfront (see [Kaplan Naori Raz 21]).

Remaining fraction revealed in adversarial order.



Theorem:

There is a poly time algorithm for Online ***Set*** *Cover* ***With-a-Sample*** *with competitive ratio* $O(\log(mn))$.

Reduction to LearnOrCover!

S_1 •

S_2 •

S_3 •

S_4 •

S_5 •

S_6 •

Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

S_1 ●

S_2 ●

S_3 ●

S_4 ●

S_5 ●

S_6 ●

Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

S_1 ●

S_2 ●

S_3 ●

S_4 ●

S_5 ●

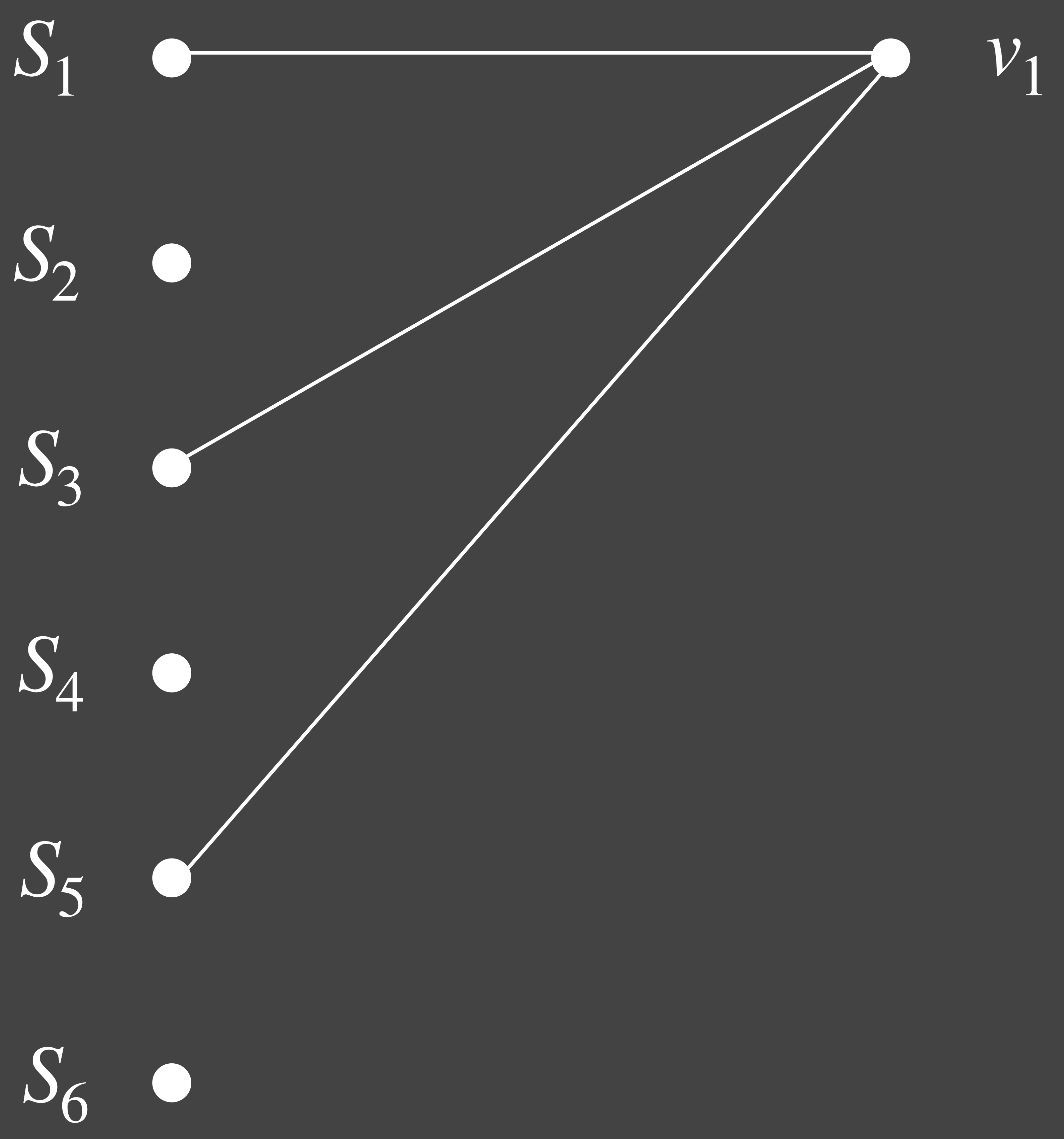
S_6 ●

Reduction to LearnOrCover!

<u>Idea</u>:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

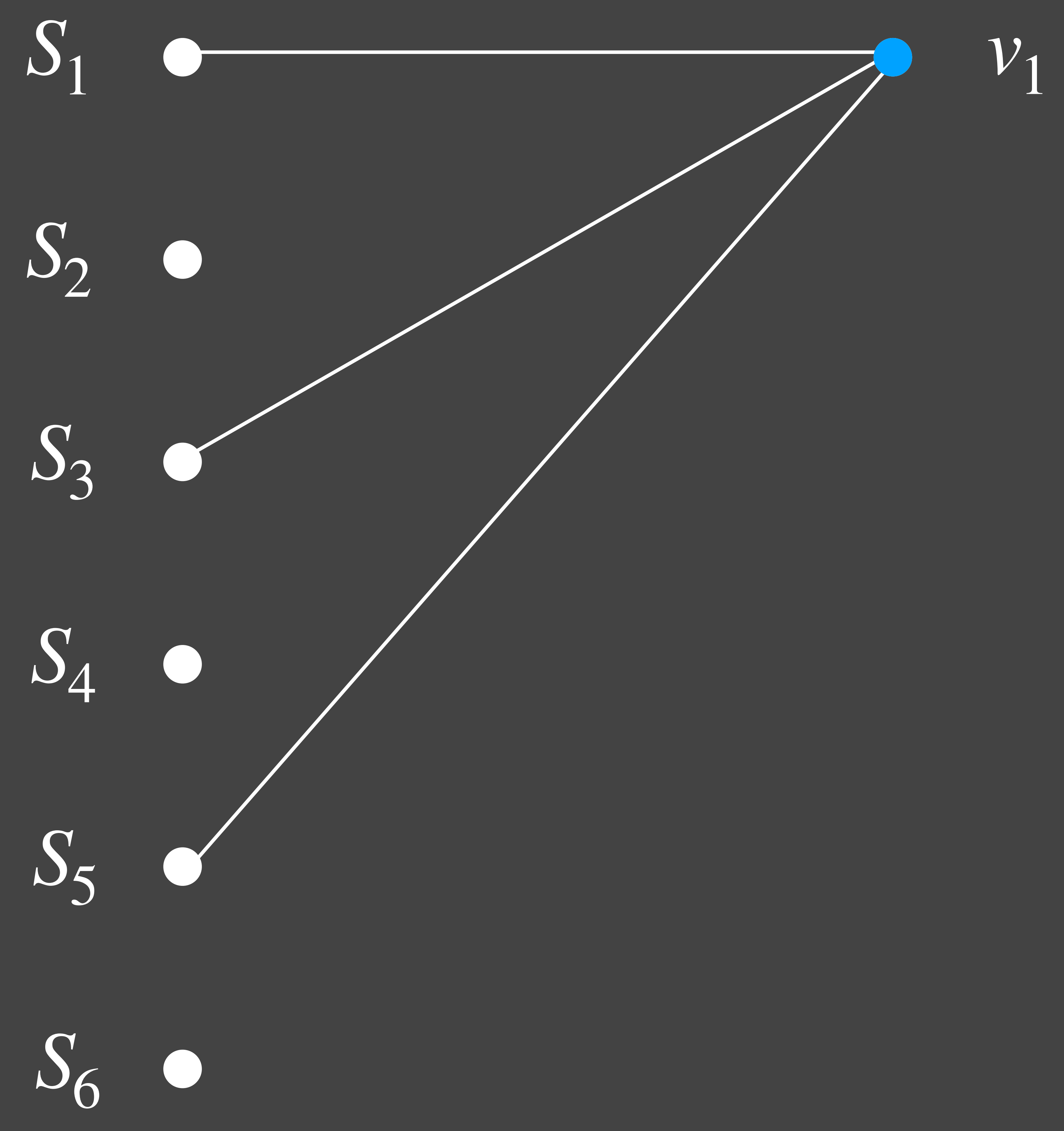


Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

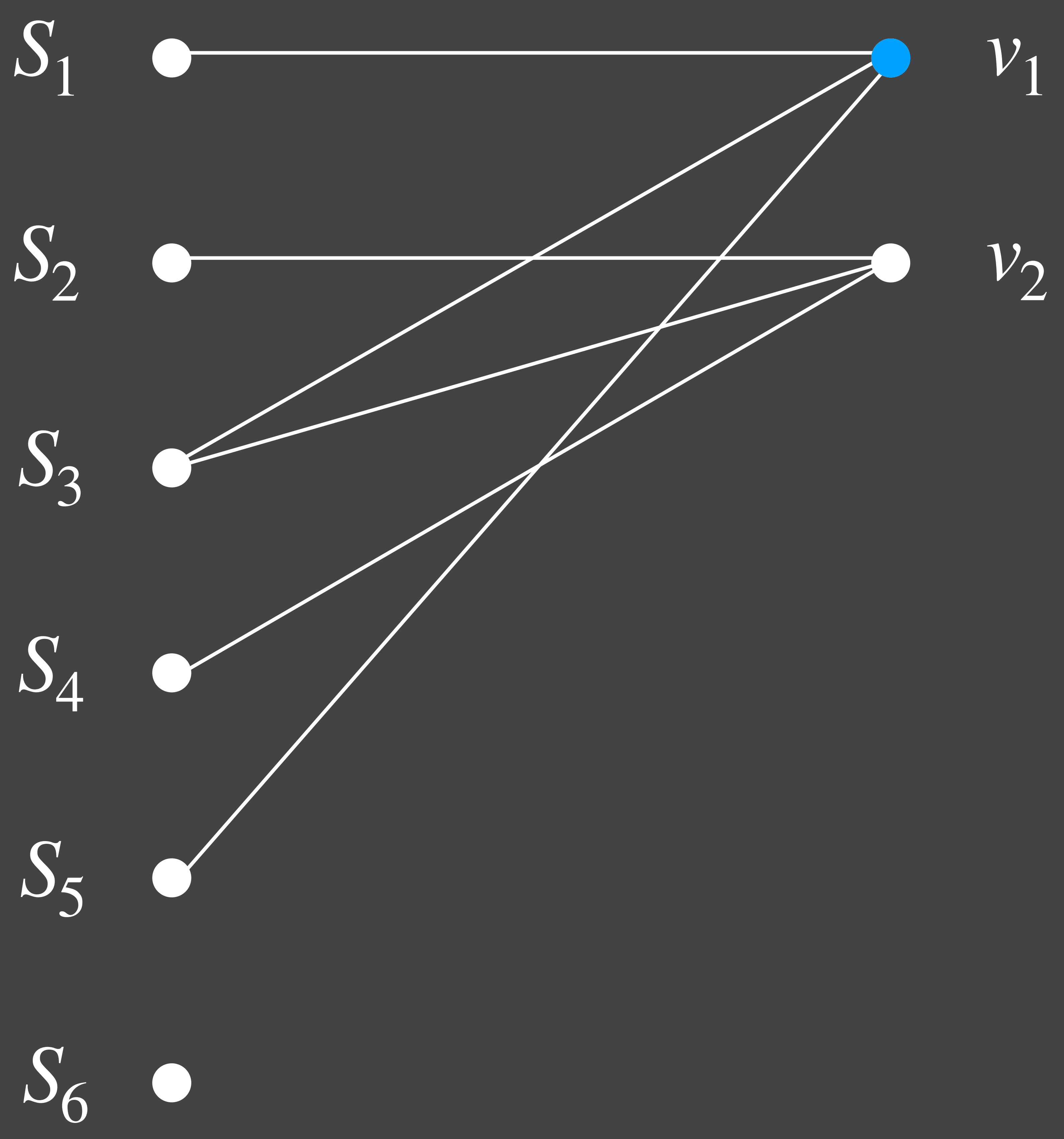


Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

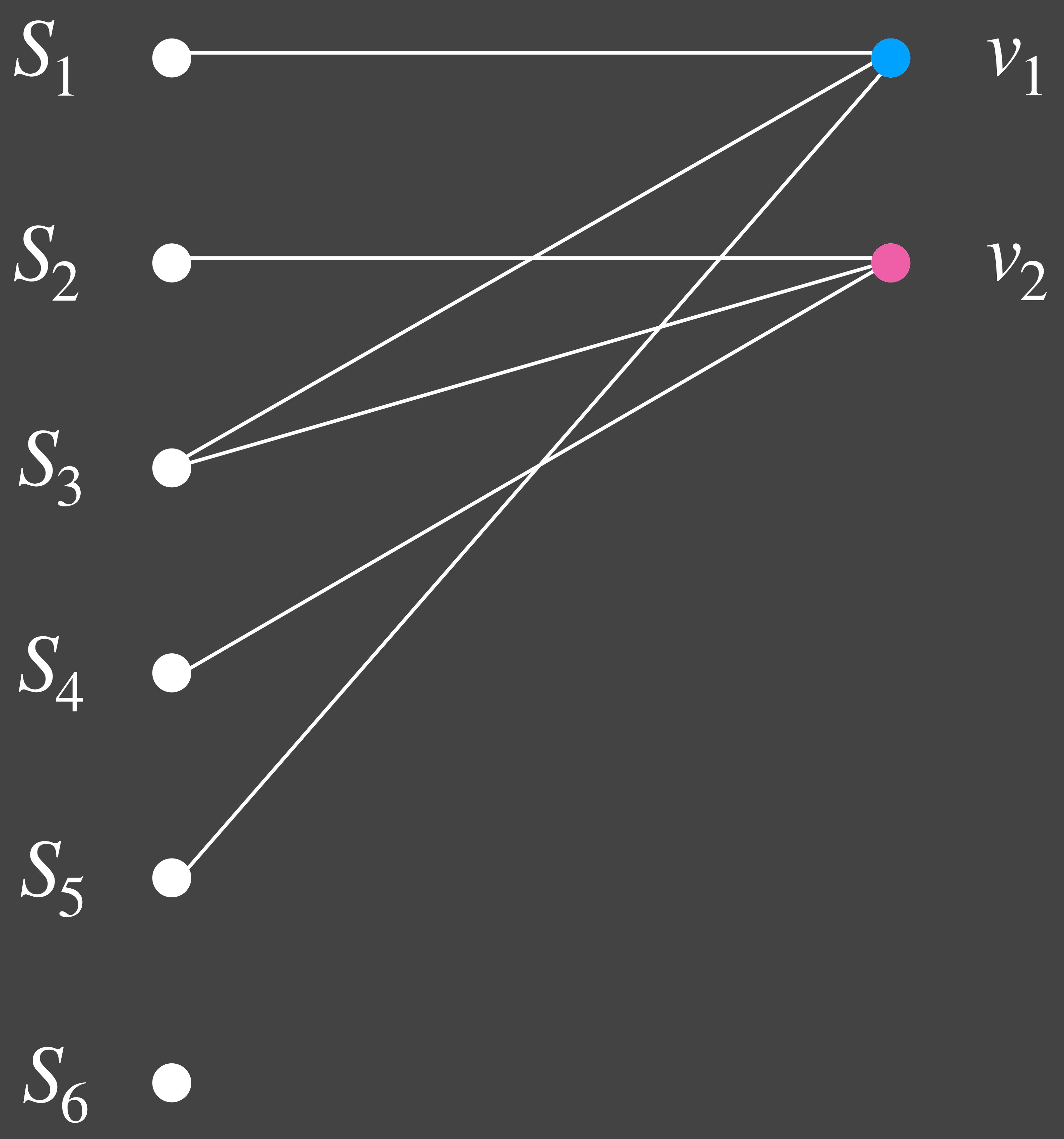


Reduction to LearnOrCover!

<u>**Idea**</u>:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

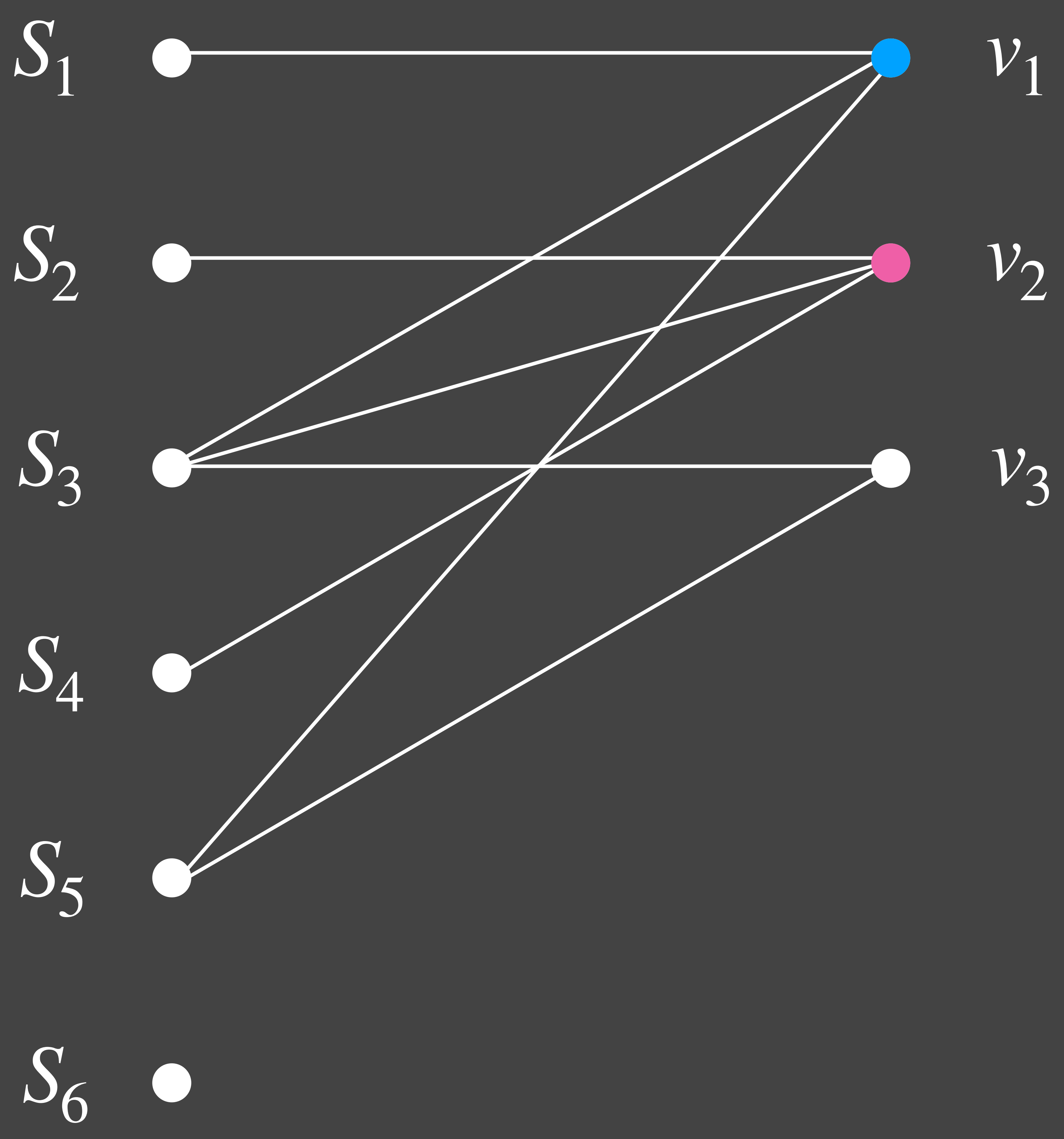


Reduction to LearnOrCover!

<u>**Idea**</u>:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

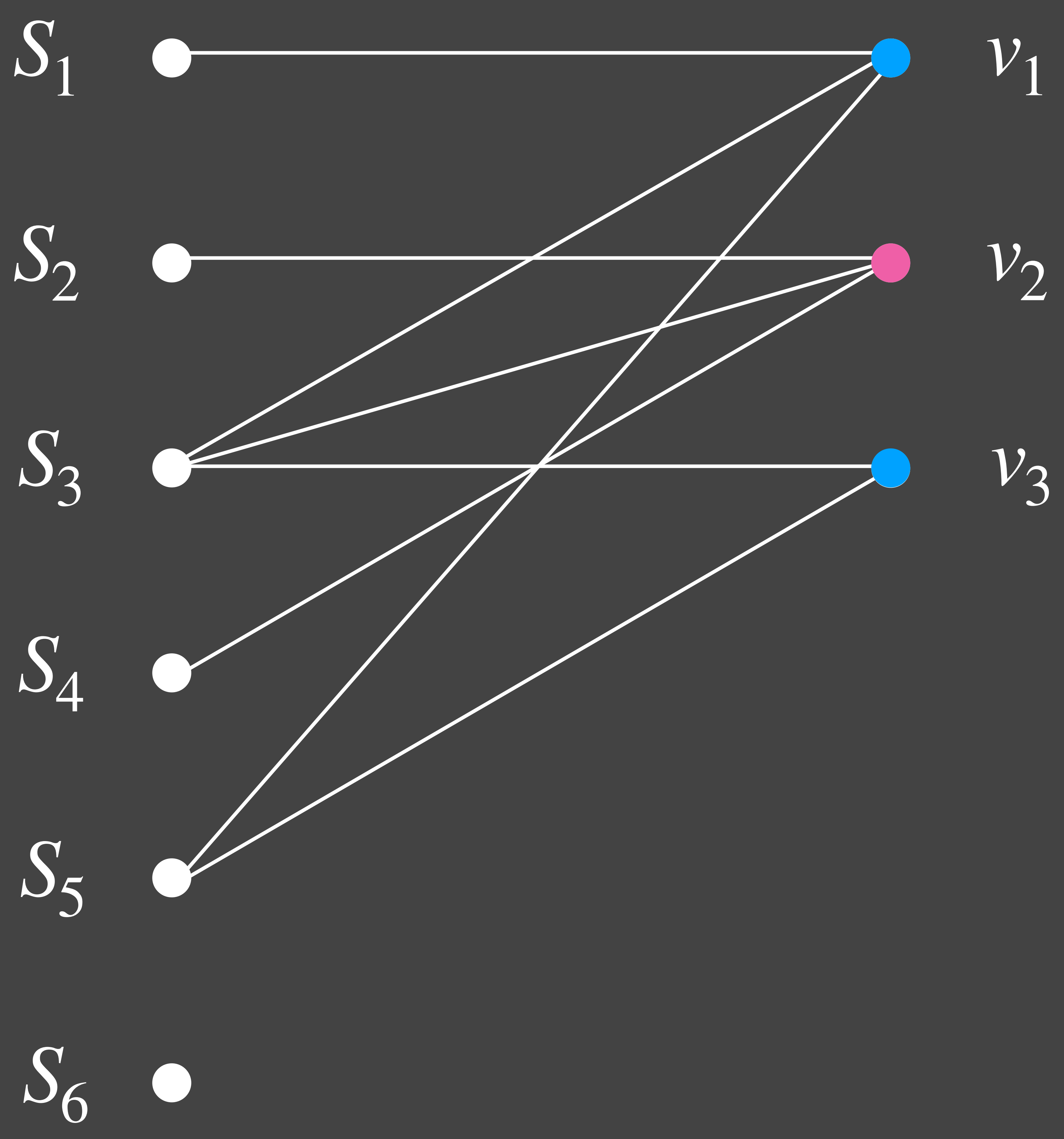


Reduction to LearnOrCover!

<u>Idea</u>:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

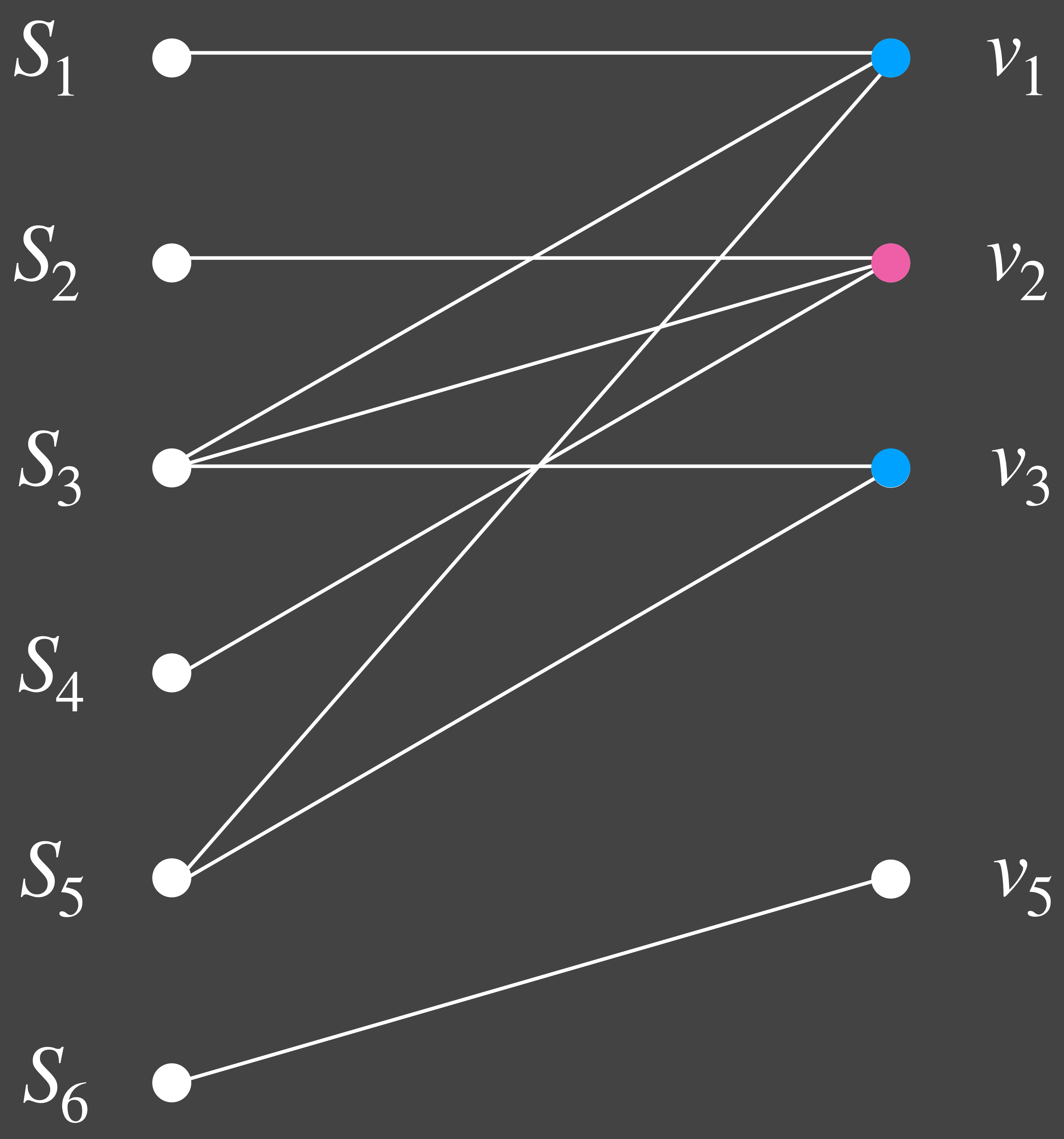
Pretend colored pink (sampled)/blue (adversarial) on arrival.



Reduction to LearnOrCover!

Idea:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

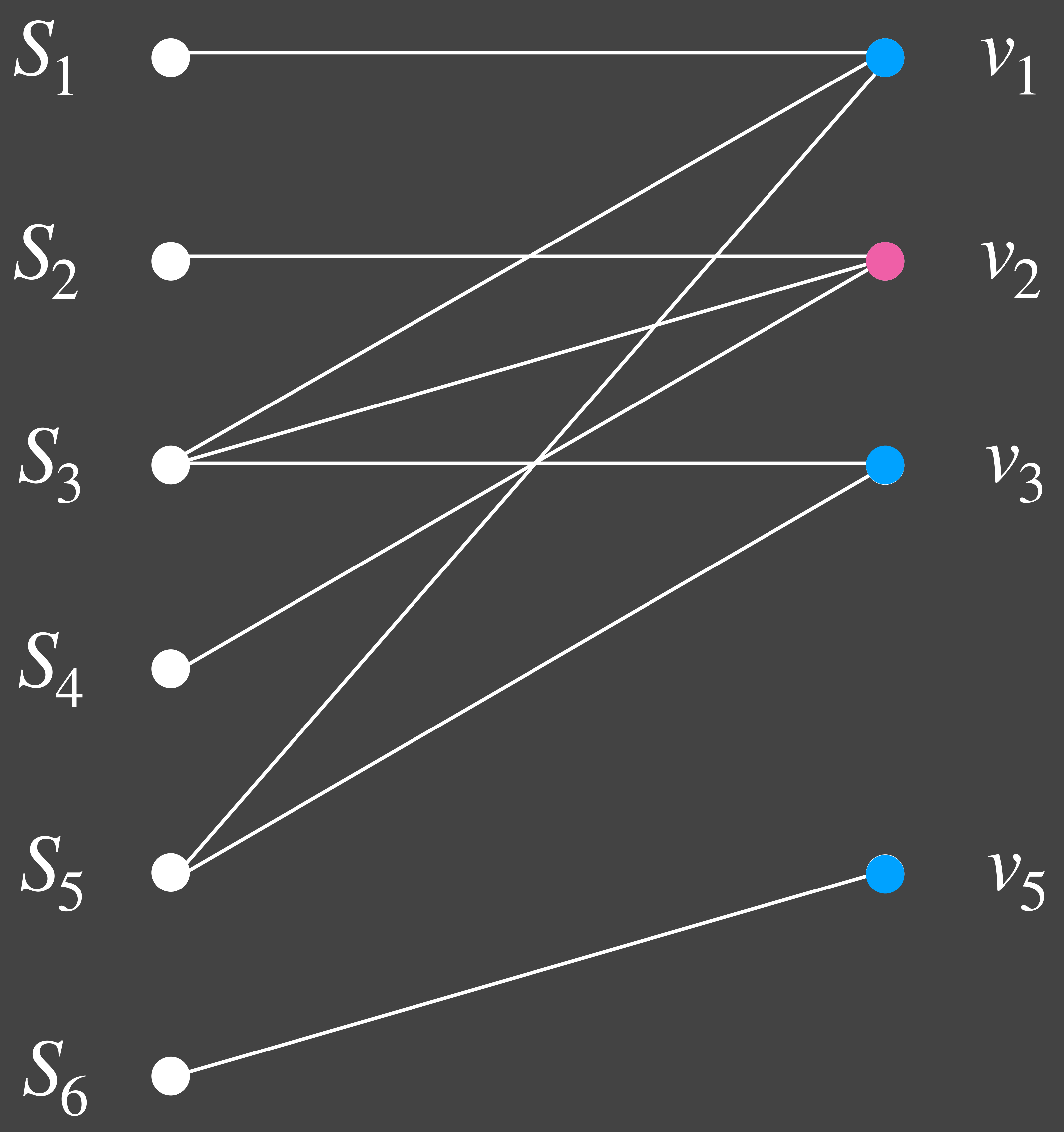


Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

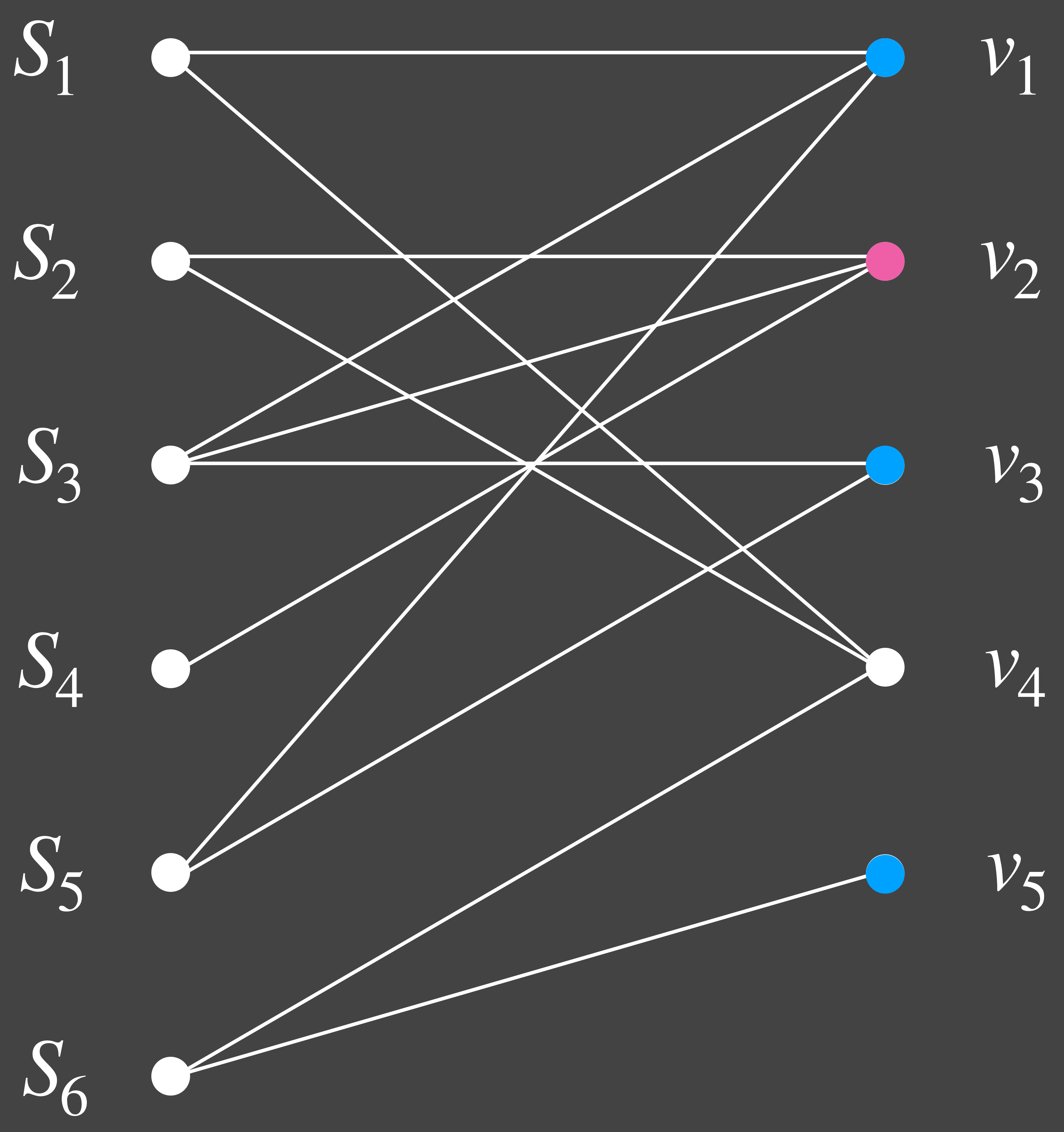


Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

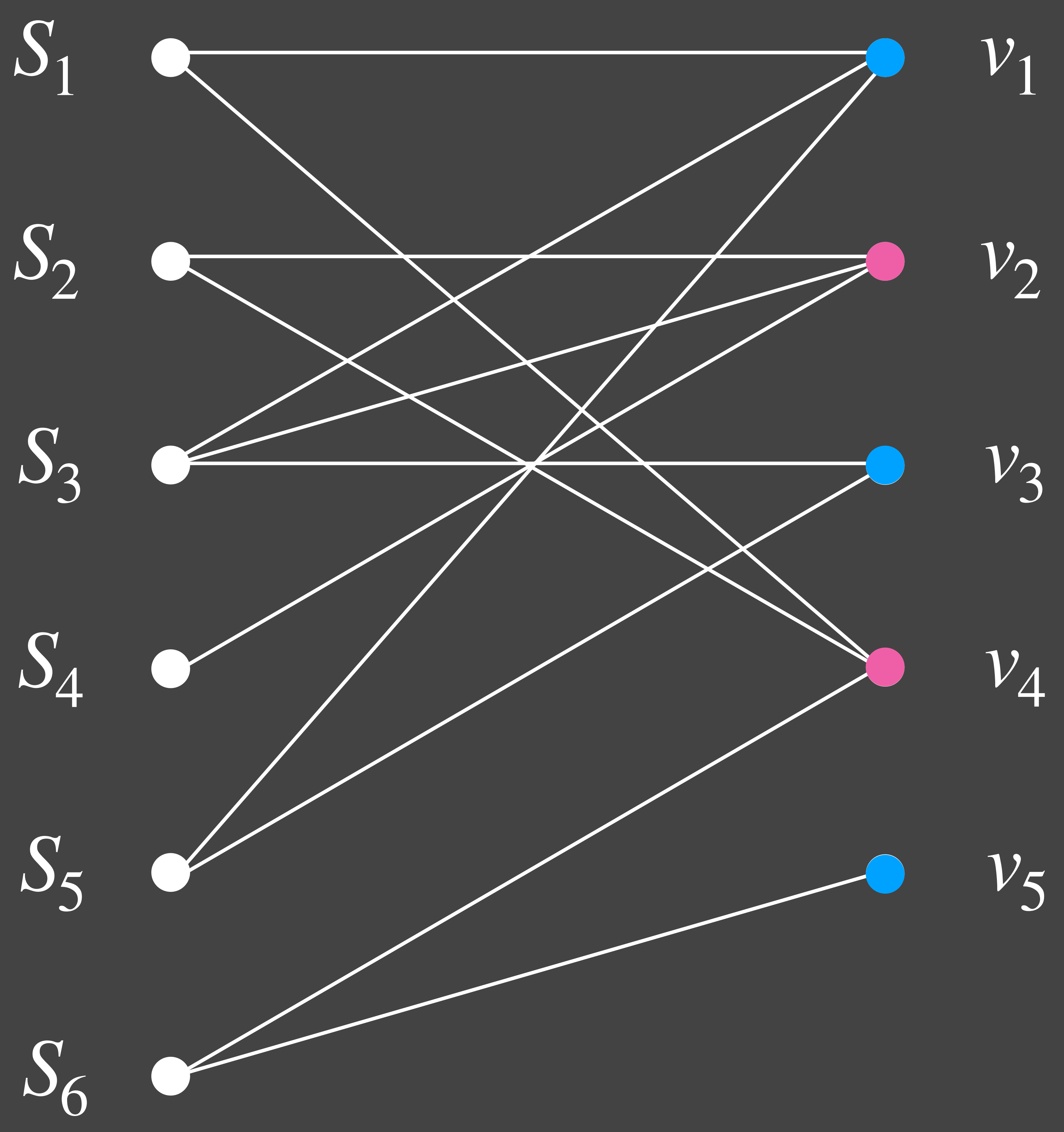


Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.

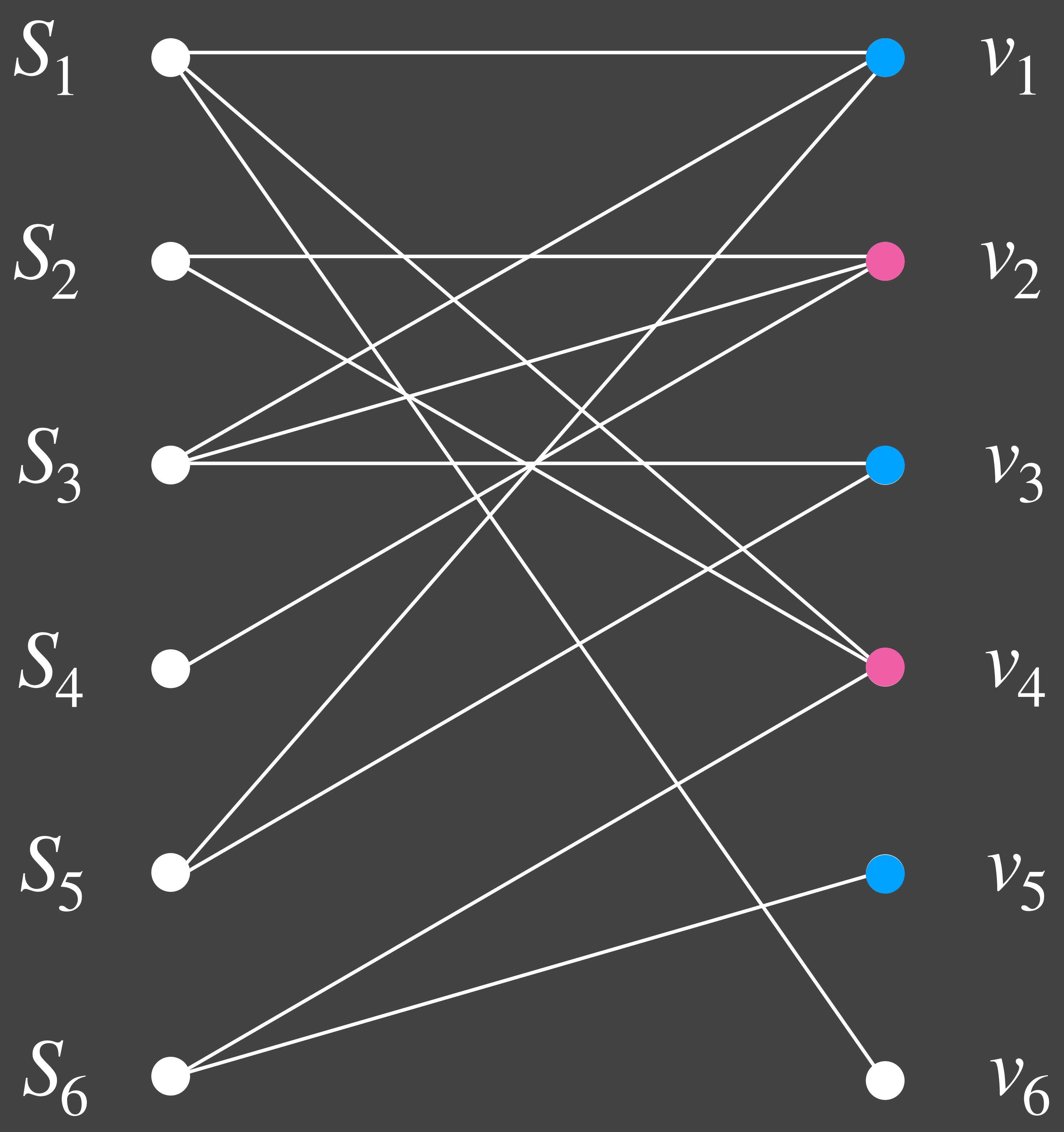


Reduction to LearnOrCover!

<u>Idea</u>:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

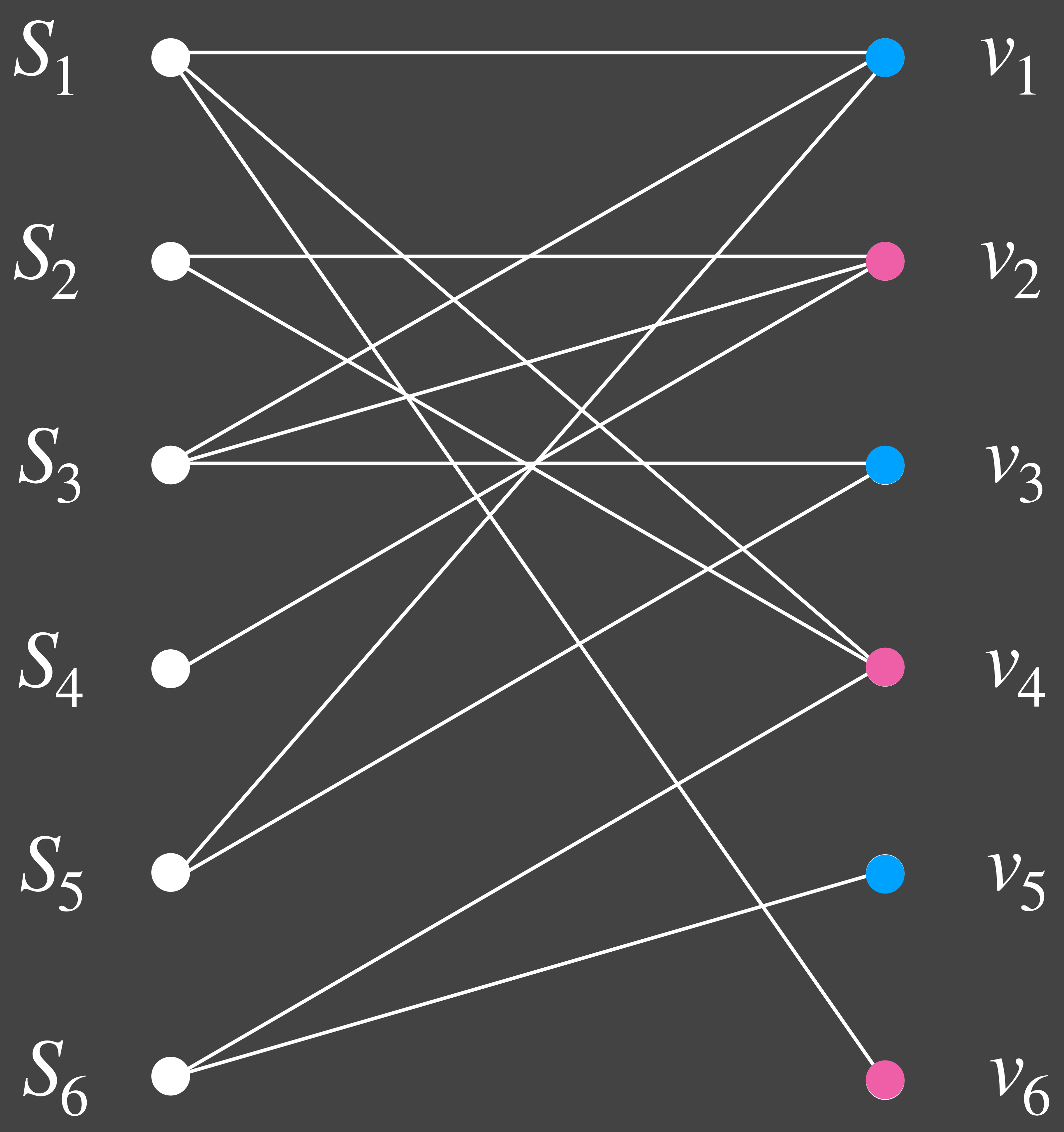
Pretend colored pink (sampled)/blue (adversarial) on arrival.



Reduction to LearnOrCover!

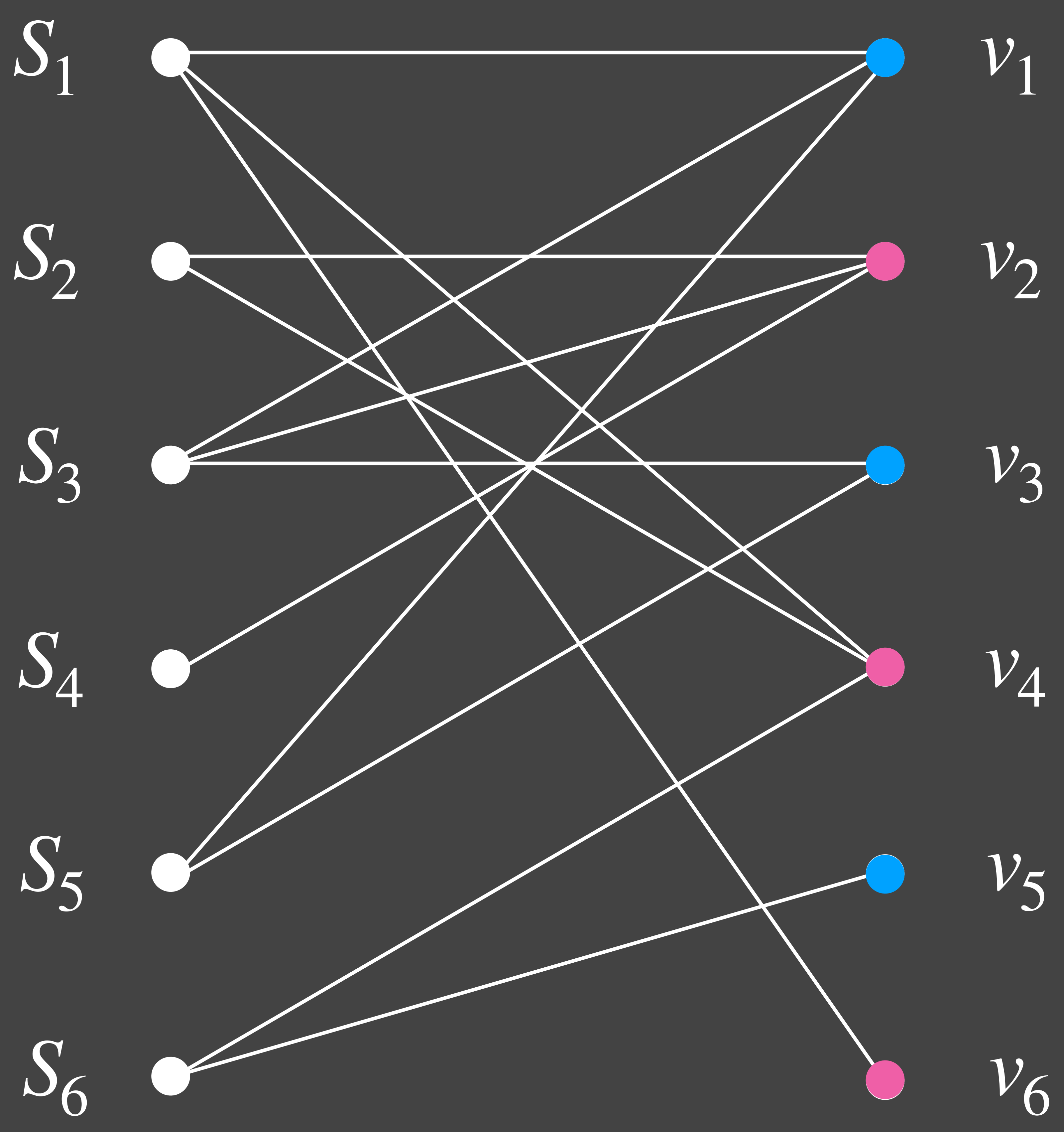
Idea:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

Pretend colored pink (sampled)/blue (adversarial) on arrival.



Reduction to LearnOrCover!

Idea:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

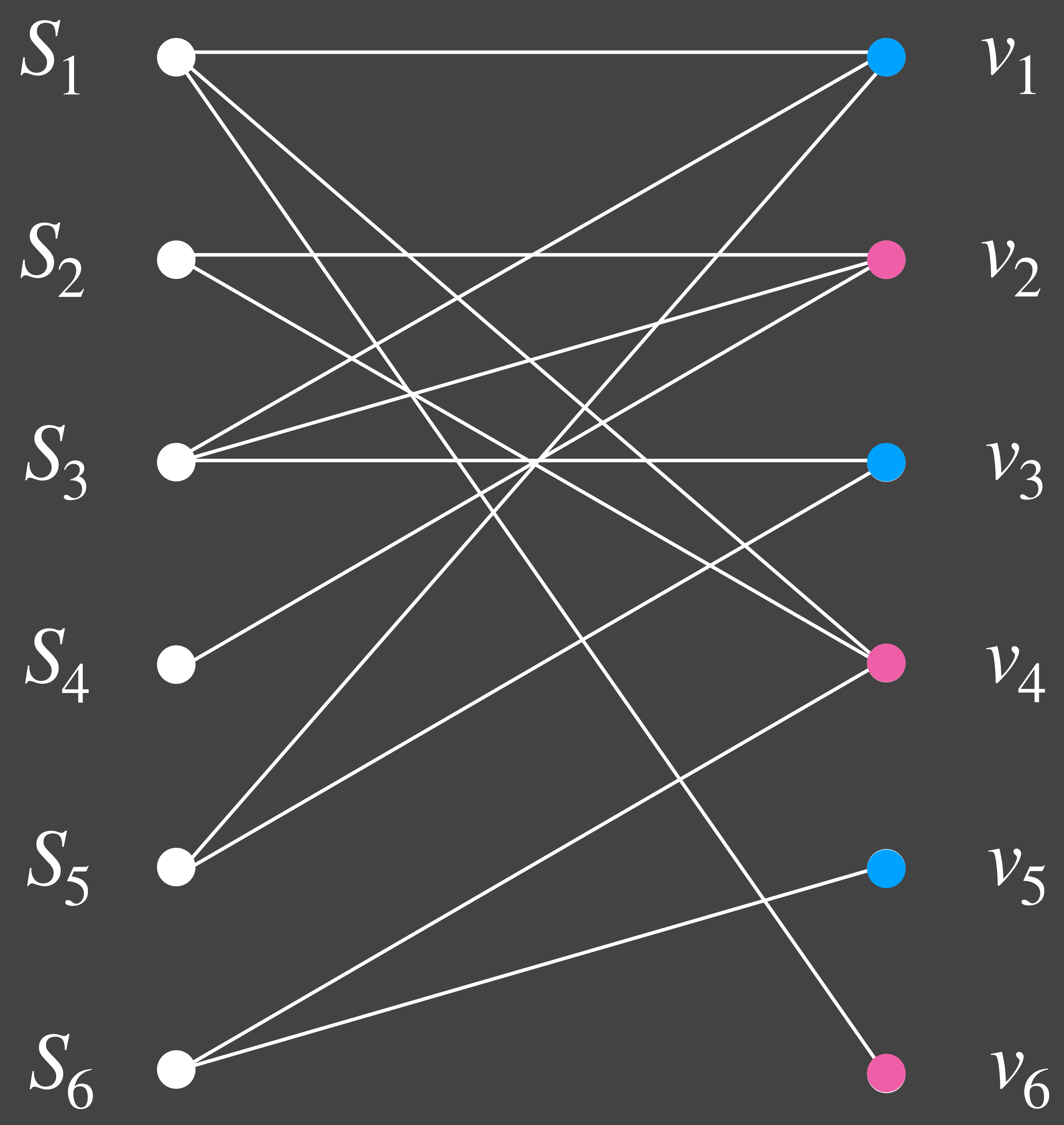


Pretend colored pink (sampled)/blue (adversarial) on arrival.

@ time t:

Reduction to LearnOrCover!

Idea:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

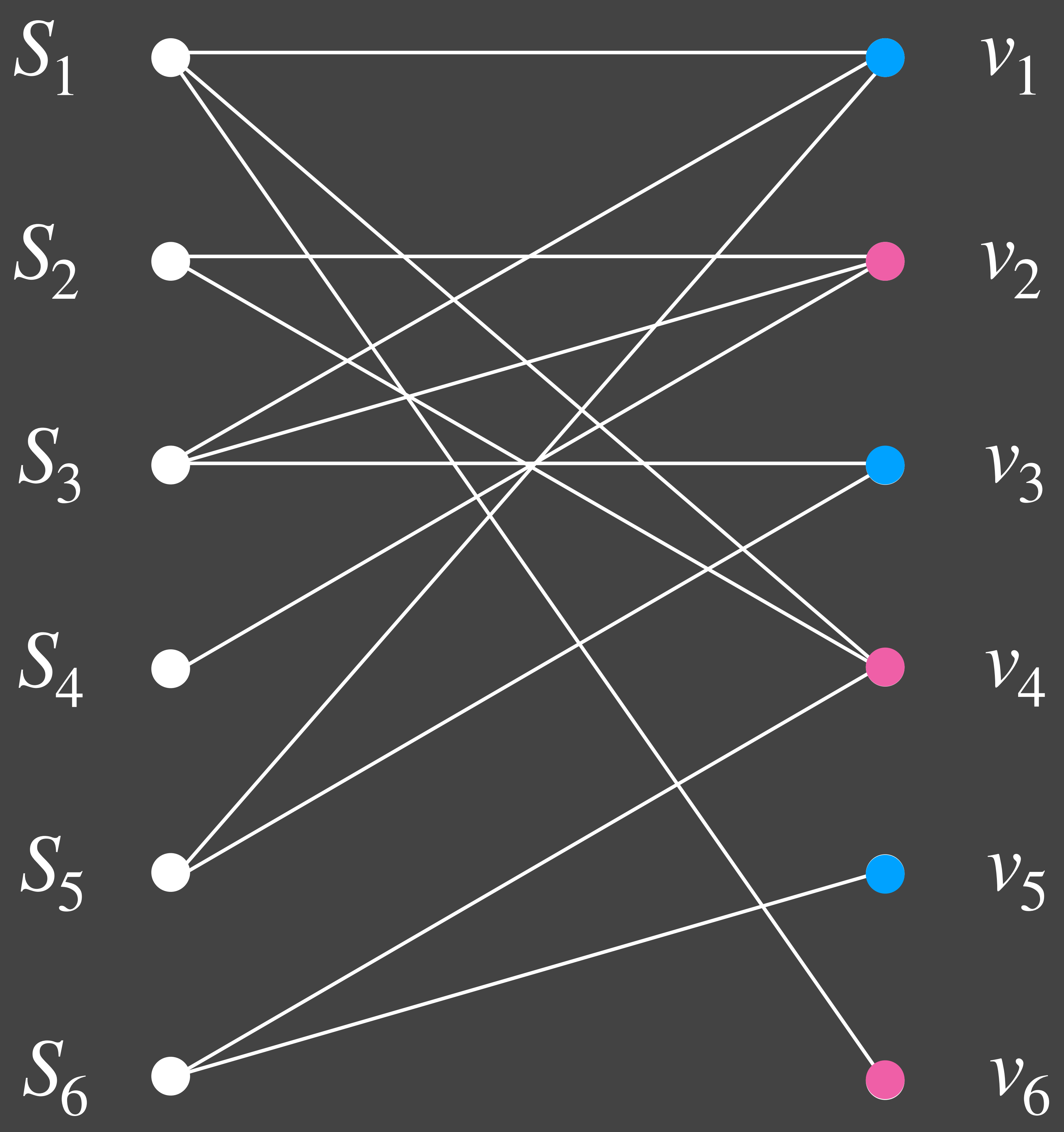
@ time t:

If v^t pink, feed to LearnOrCover.

Reduction to LearnOrCover!

Idea:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

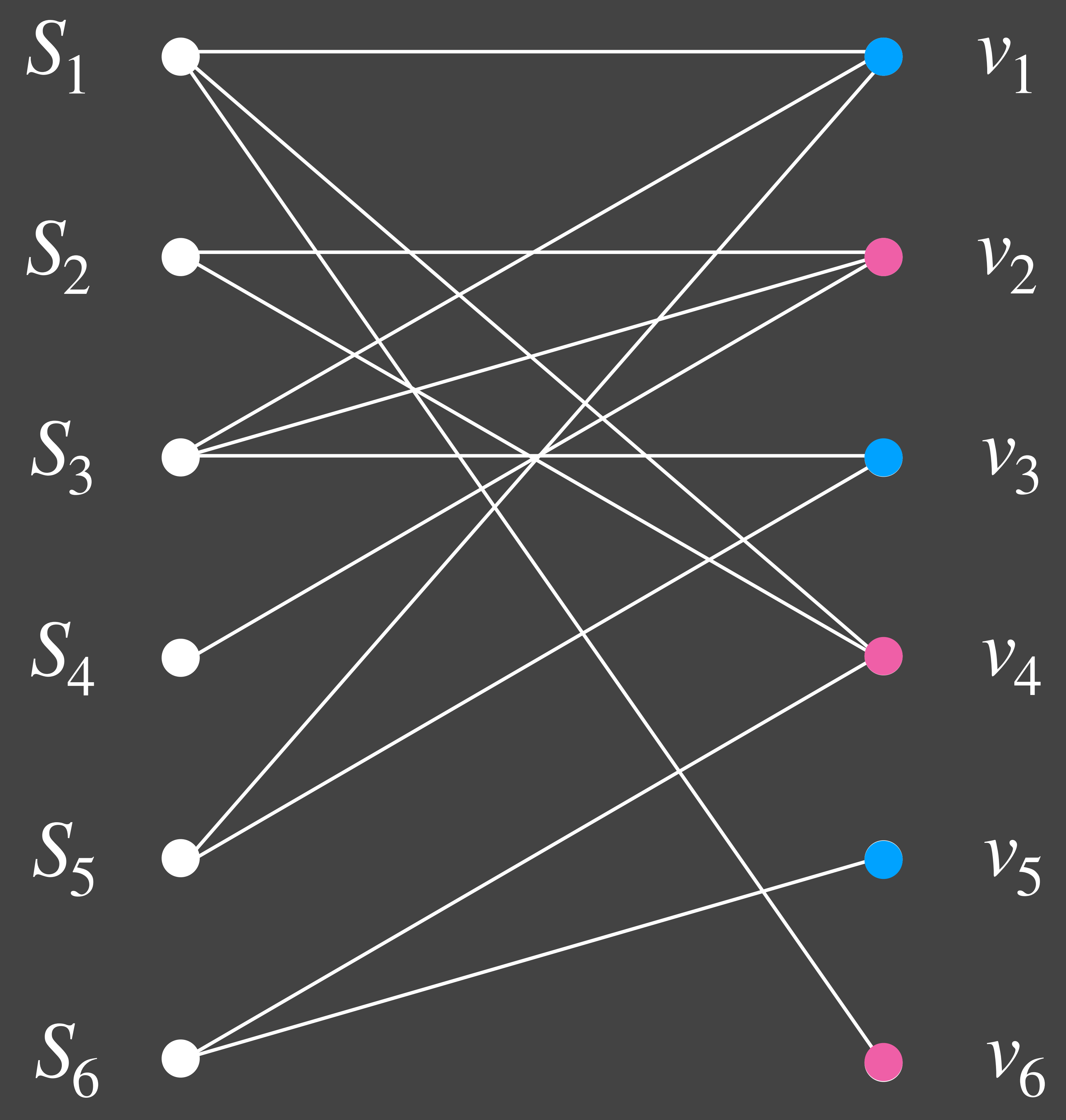
@ time t:

If v^t pink, feed to LearnOrCover.

If v^t blue, buy arbitrary set to cover.

Reduction to LearnOrCover!

<u>**Idea**</u>:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

@ time t:

If v^t pink, feed to LearnOrCover.

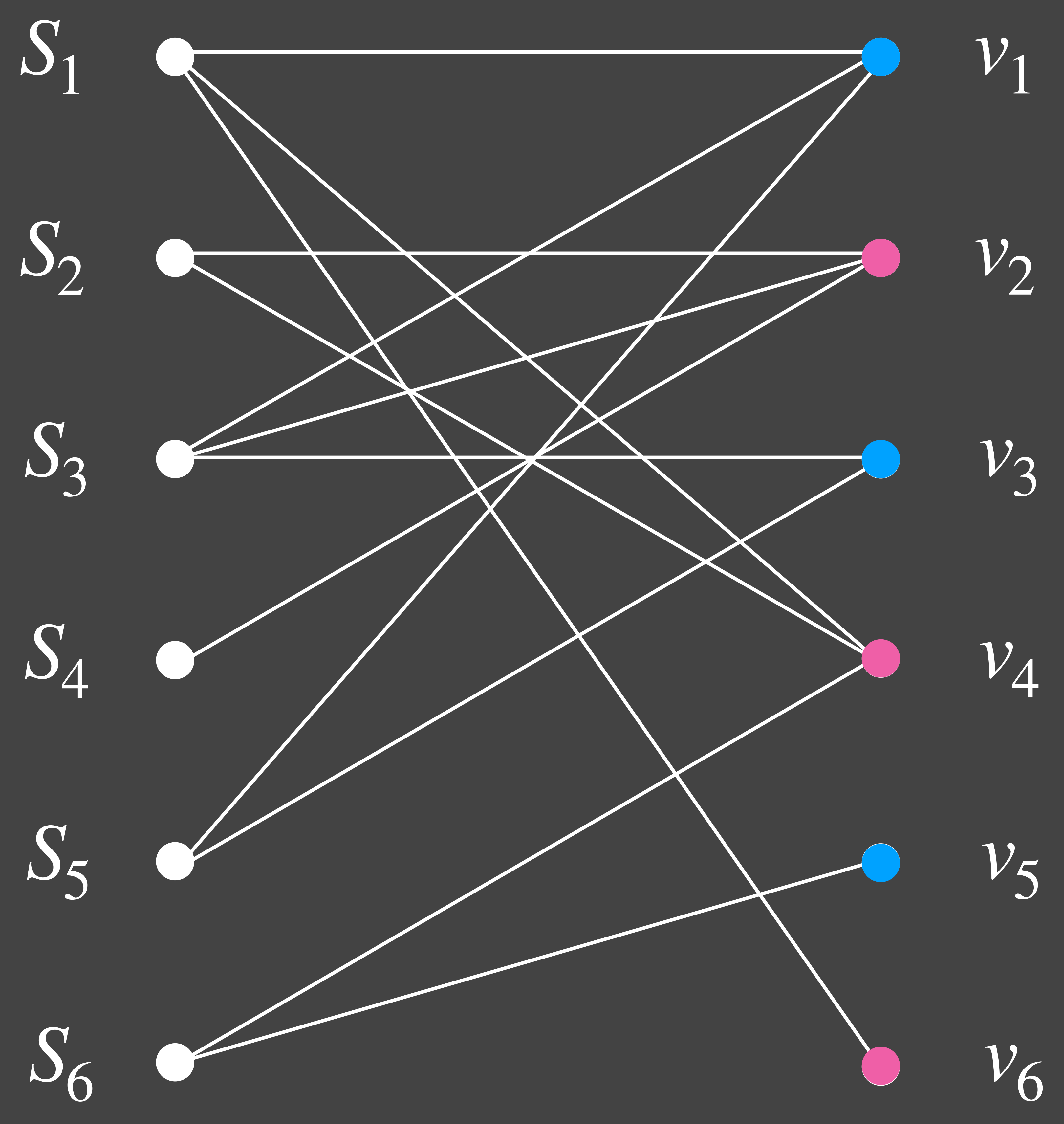
If v^t blue, buy arbitrary set to cover.

Recall LearnOrCover proof template:

Reduction to LearnOrCover!

<u>**Idea**</u>:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

@ time t:

If v^t pink, feed to LearnOrCover.

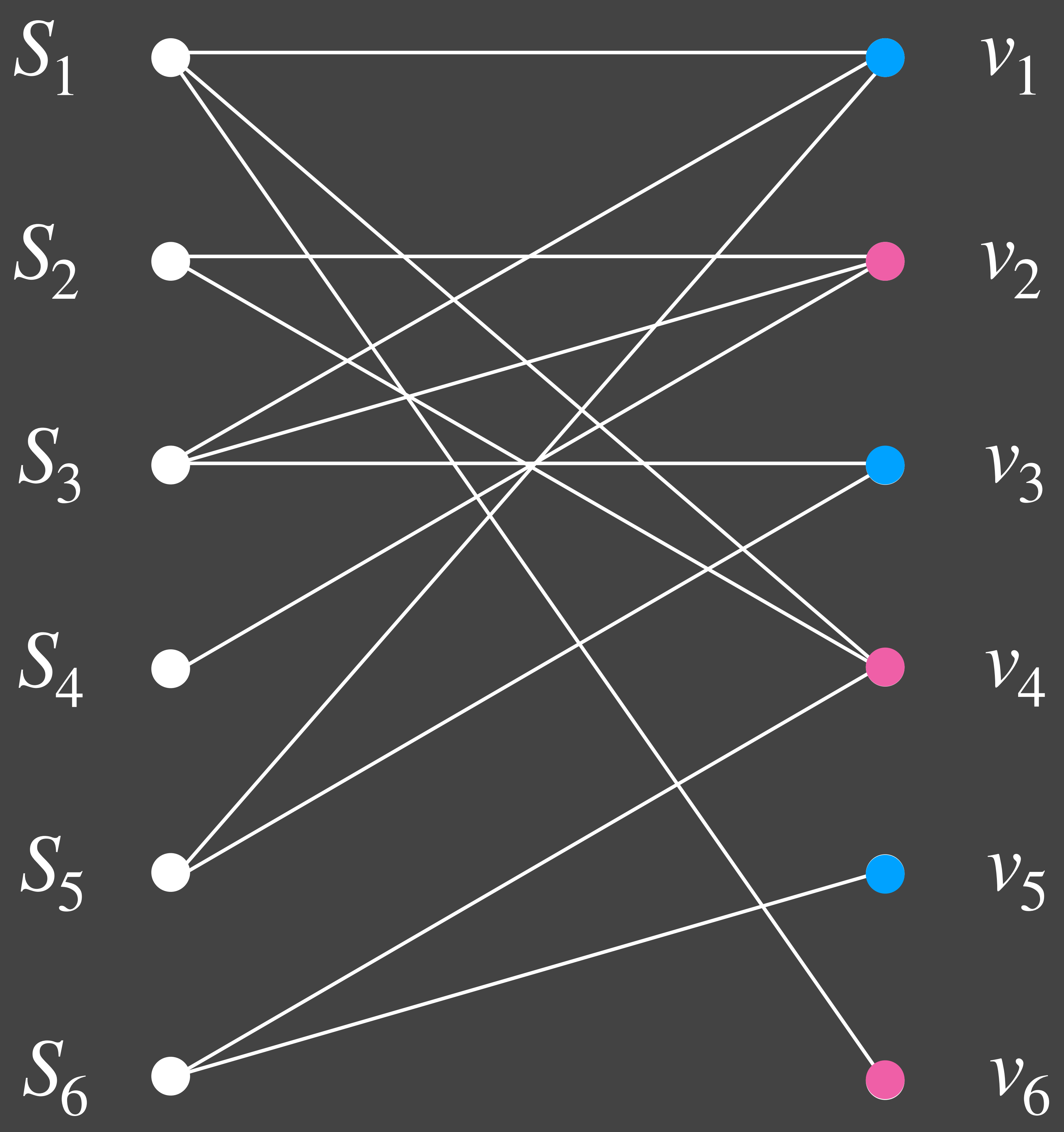
If v^t blue, buy arbitrary set to cover.

Recall LearnOrCover proof template:

<u>Claim 1:</u> $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

Reduction to LearnOrCover!

Idea:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

@ time t:

If v^t pink, feed to LearnOrCover.

If v^t blue, buy arbitrary set to cover.

Recall LearnOrCover proof template:

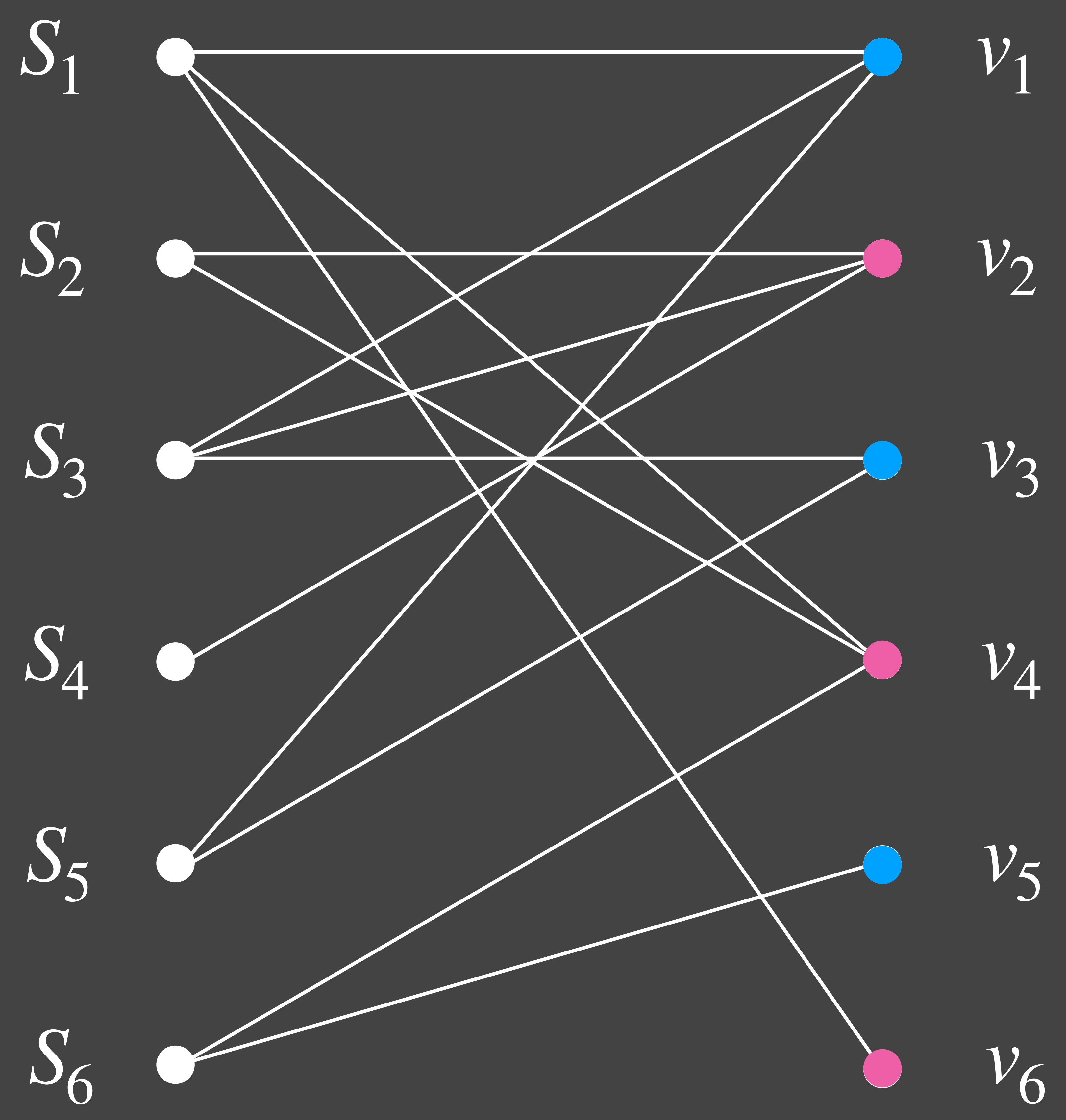
Claim 1: $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

Claim 2: If v^t uncovered, then $E[\Delta\Phi] \leq -\Omega\left(\frac{1}{k}\right)$.

Reduction to LearnOrCover!

__Idea__:

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

@ time t:

If v^t pink, feed to LearnOrCover.

If v^t blue, buy arbitrary set to cover.

Recall LearnOrCover proof template:

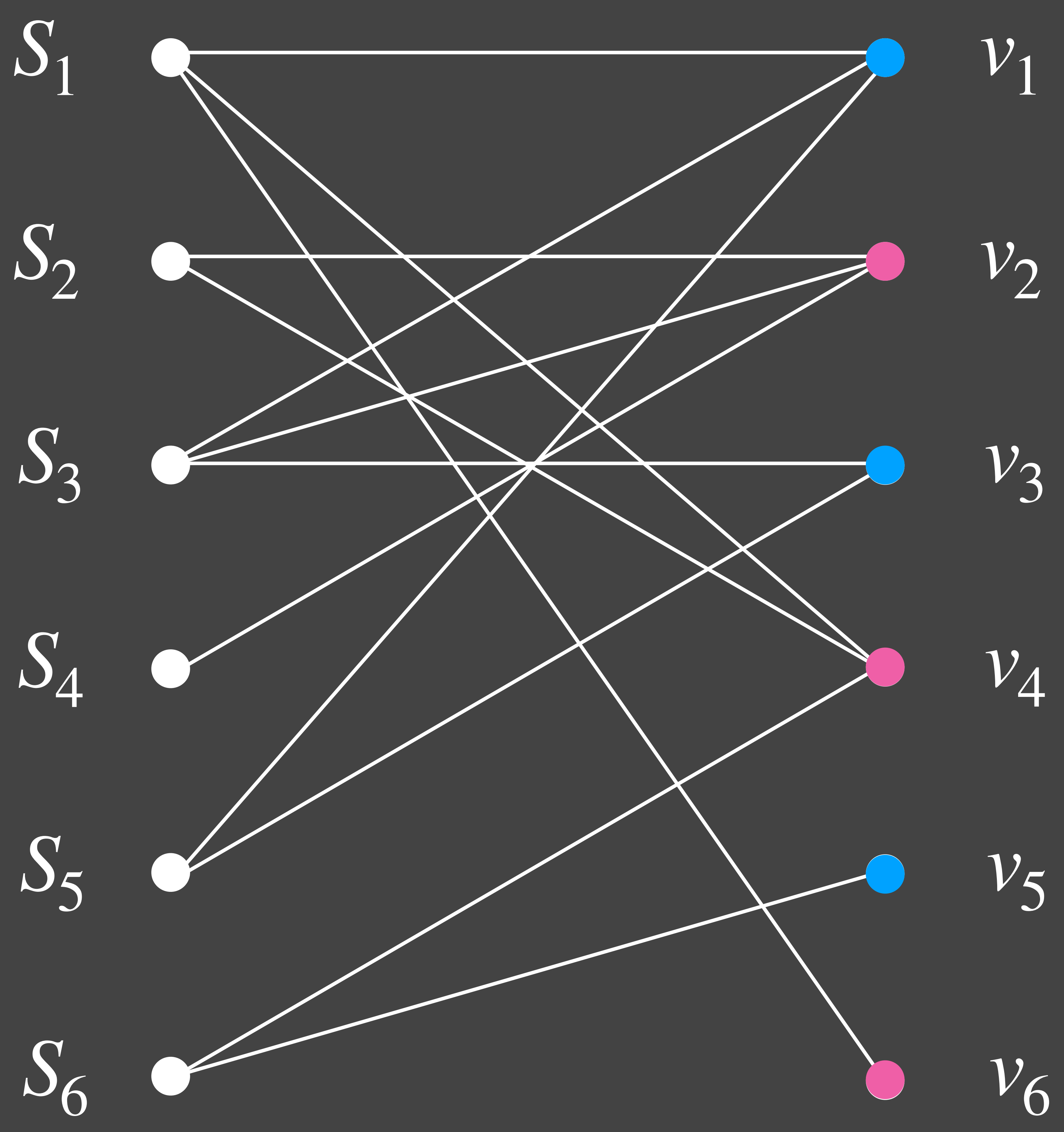
__Claim 1:__ $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

__Claim 2:__ If v^t uncovered, then $E[\Delta\Phi] \leq -\Omega\left(\frac{1}{k}\right)$.

Φ only deceases during pink steps (so with prob. 1/2), but still $E[\Delta\Phi] \leq -\Omega\left(\frac{1}{k}\right)$.

Reduction to LearnOrCover!

<u>Idea</u>:
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Pretend colored pink (sampled)/blue (adversarial) on arrival.

@ time t:

If v^t pink, feed to LearnOrCover.

If v^t blue, buy arbitrary set to cover.

Recall LearnOrCover proof template:

<u>Claim 1:</u> $\Phi(0) = O(\log mn)$, and $\Phi(t) \geq 0$.

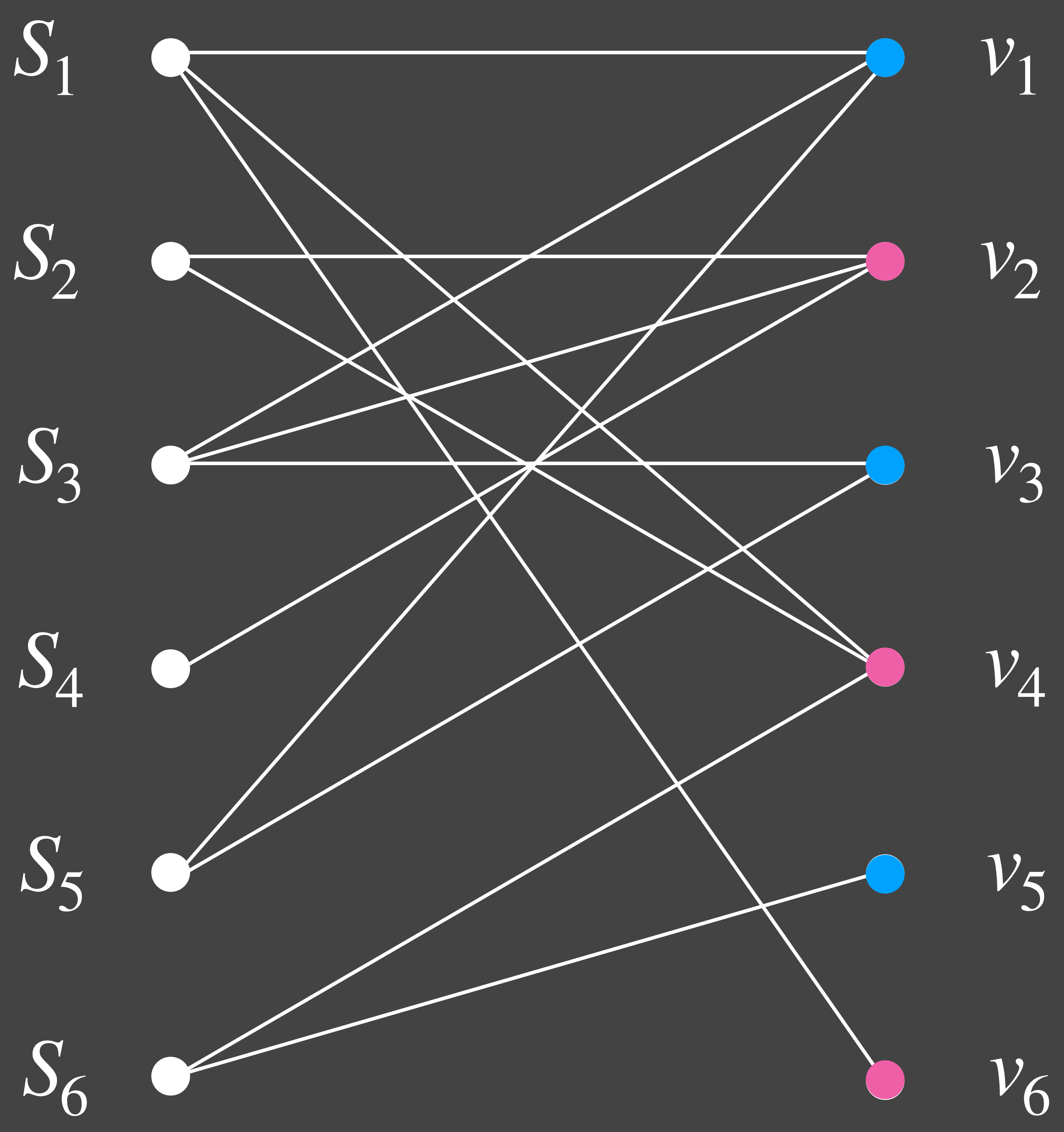
<u>Claim 2:</u> If v^t uncovered, then $E[\Delta\Phi] \leq -\Omega\left(\frac{1}{k}\right)$.

Φ only deceases during pink steps (so with prob. 1/2), but still $E[\Delta\Phi] \leq -\Omega\left(\frac{1}{k}\right)$. 🥳

Universality

<u>Idea</u>: Reduction!

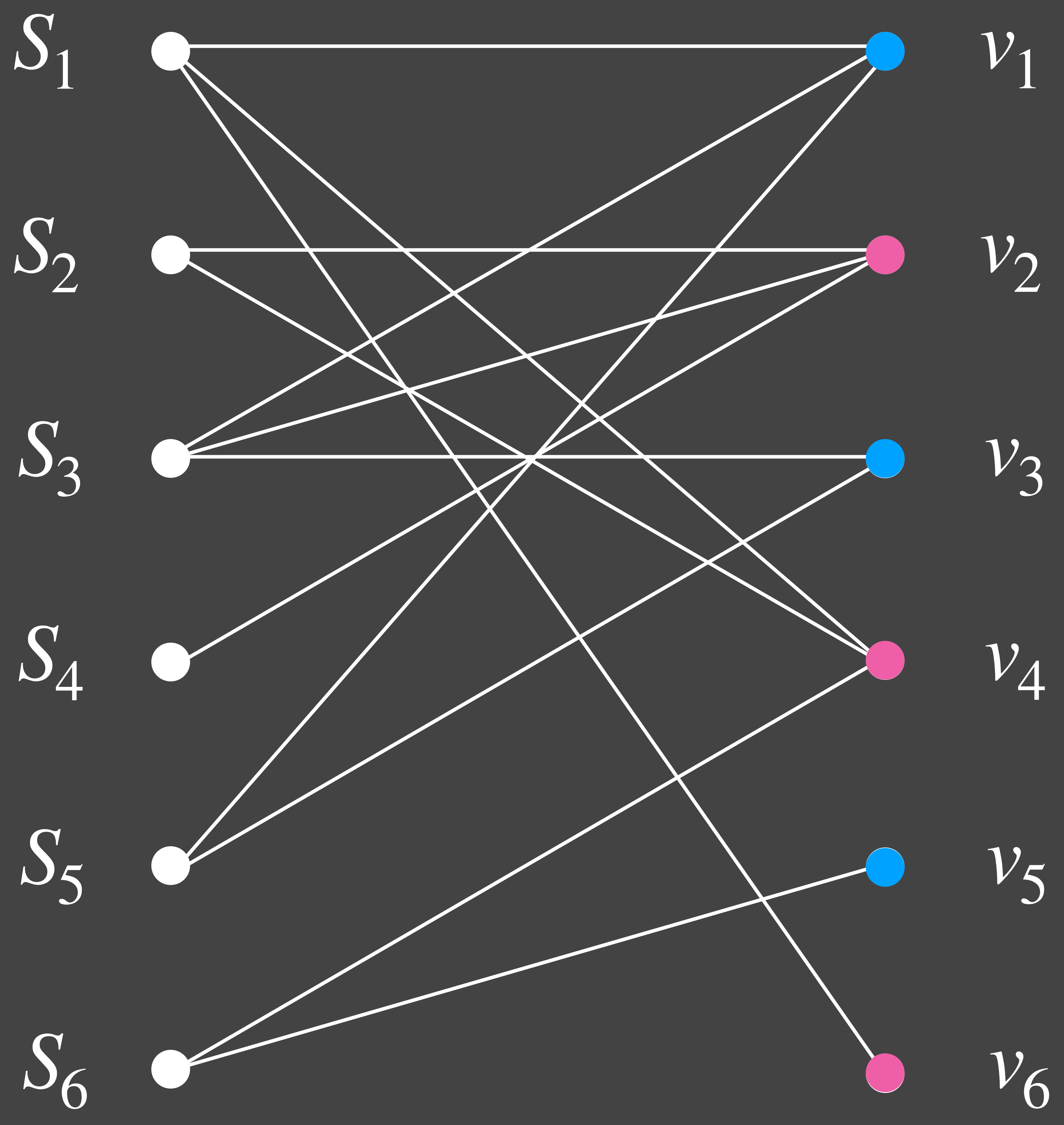
1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Universality

Idea: Reduction!

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.

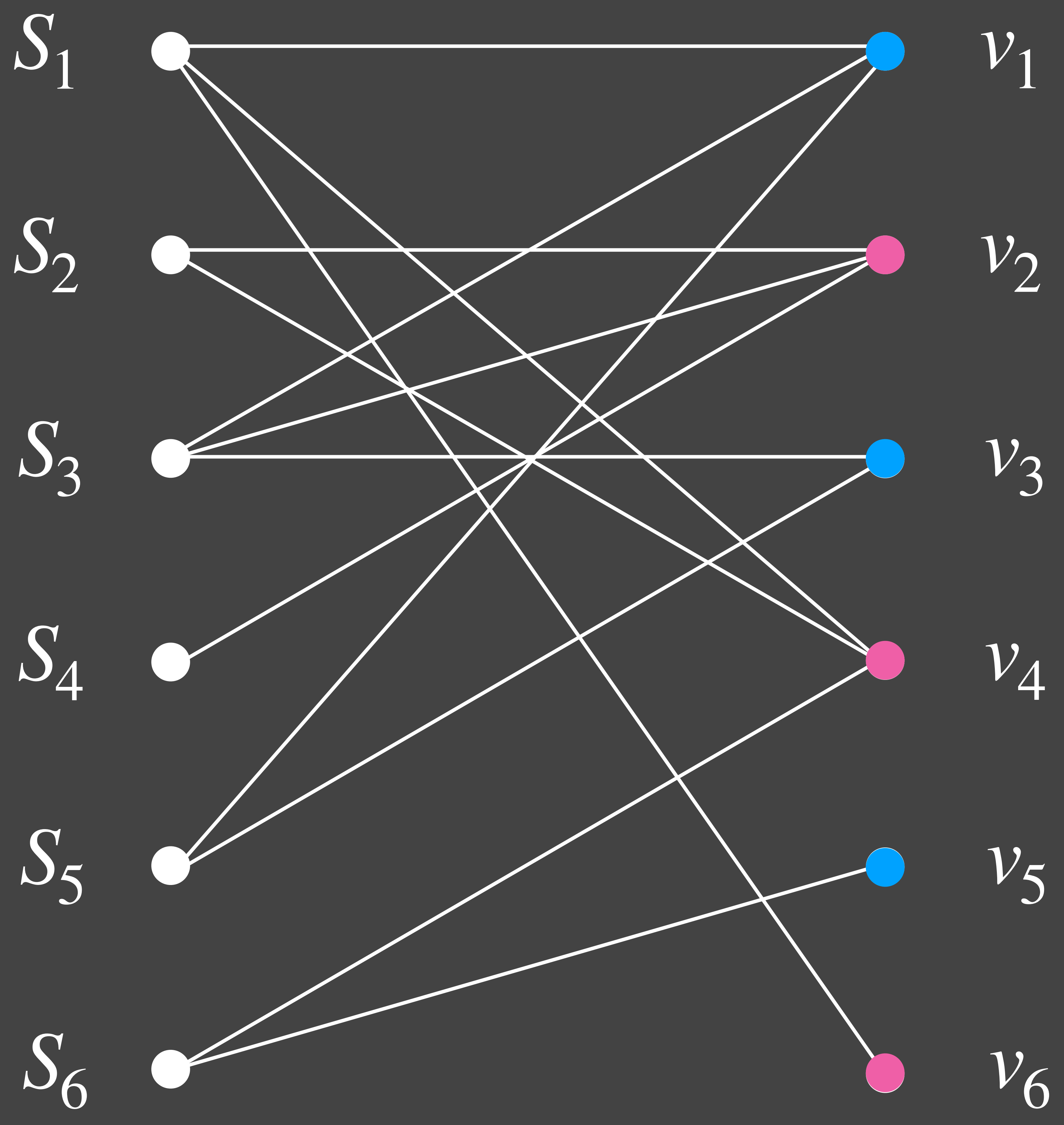


Can build map $f : \mathscr{U} \to \mathscr{S}$ ***before*** we see any actual elements.

Universality

Idea: Reduction!

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



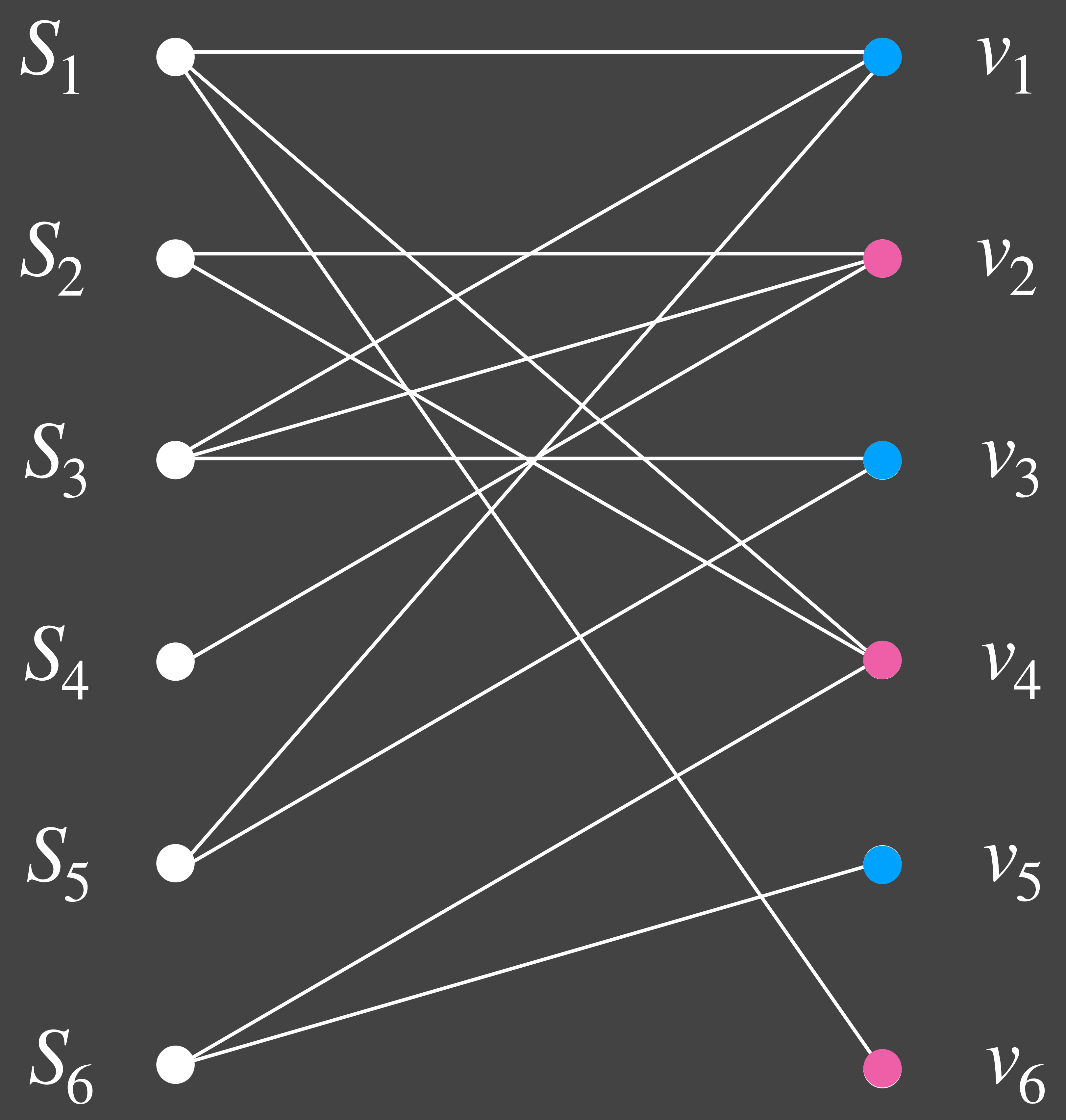
Can build map $f : \mathcal{U} \to \mathcal{S}$ **before** we see any actual elements.

When $u \in \mathcal{U}$ arrives, commit to buying $f(u)$!

Universality

<u>Idea</u>: Reduction!

1. Run LearnOrCover on samples.
2. Buy arbitrary sets for remaining elements.



Can build map $f : \mathscr{U} \to \mathscr{S}$ ***<u>before</u>*** we see any actual elements.

When $u \in \mathscr{U}$ arrives, commit to buying $f(u)$!

Our result shows only need $O(n)$ samples to build this map.

Talk Outline

Intro

Secretary

- LearnOrCover in Exponential Time
- LearnOrCover in Poly Time

➡ (Single Sample) Prophet

Conclusion & Extensions

Talk Outline

Intro

Secretary

- LearnOrCover in Exponential Time
- LearnOrCover in Poly Time

(Single Sample) Prophet

➡ Conclusion & Extensions

Conclusion

Conclusion

<u>Theorem</u>: $O(\log mn)$*-comp. algo for RO Covering IPs.*

Conclusion

Theorem: $O(\log mn)$*-comp. algo for RO Covering IPs.*

Theorem: $O(\log mn)$*-comp. algo for Prophet Covering IPs.*

Conclusion

<u>Theorem</u>: *$O(\log mn)$-comp. algo for RO Covering IPs.*

<u>Theorem</u>: *$O(\log mn)$-comp. algo for Prophet Covering IPs.*

\+ Single-Sample!
\+ Universal!

Conclusion

<u>Theorem</u>: $O(\log mn)$*-comp. algo for RO Covering IPs.*

<u>Theorem</u>: $O(\log mn)$*-comp. algo for Prophet Covering IPs.*

<u>Theorem</u>: Same results *for Non-metric facility location.*

+ Single-Sample!
+ Universal!

Conclusion

Theorem: $O(\log mn)$*-comp. algo for RO Covering IPs.*

Theorem: $O(\log mn)$*-comp. algo for Prophet Covering IPs.*

Theorem: Same results *for Non-metric facility location.*

Theorem: $\Omega(\log m \log f(\mathcal{N}))$ for RO submodular cover.

+ Single-Sample!
+ Universal!

Conclusion

Theorem: $O(\log mn)$*-comp. algo for RO Covering IPs.*

Theorem: $O(\log mn)$*-comp. algo for Prophet Covering IPs.*

+ Single-Sample!
+ Universal!

Theorem: Same results *for Non-metric facility location.*

Theorem: $\Omega(\log m \log f(\mathcal{N}))$ for RO submodular cover.

Open Questions:

Conclusion

Theorem: $O(\log mn)$-*comp. algo for RO Covering IPs.*

Theorem: $O(\log mn)$-*comp. algo for Prophet Covering IPs.*

+ Single-Sample!
+ Universal!

Theorem: Same results *for Non-metric facility location.*

Theorem: $\Omega(\log m \log f(\mathcal{N}))$ for RO submodular cover.

Open Questions:

Does the LearnOrCover idea lend itself to other problems?

Conclusion

Theorem: $O(\log mn)$-*comp. algo for RO Covering IPs.*

Theorem: $O(\log mn)$-*comp. algo for Prophet Covering IPs.*

+ Single-Sample!
+ Universal!

Theorem: Same results *for Non-metric facility location.*

Theorem: $\Omega(\log m \log f(\mathcal{N}))$ for RO submodular cover.

Open Questions:

Does the LearnOrCover idea lend itself to other problems?
Harder covering problems? Covering IPs w/ box constraints?

Conclusion

Theorem: $O(\log mn)$*-comp. algo for RO Covering IPs.*

Theorem: $O(\log mn)$*-comp. algo for Prophet Covering IPs.*

+ Single-Sample!
+ Universal!

Theorem: Same results *for Non-metric facility location.*

Theorem: $\Omega(\log m \log f(\mathcal{N}))$ for RO submodular cover.

Open Questions:

Does the LearnOrCover idea lend itself to other problems?
Harder covering problems? Covering IPs w/ box constraints?
Unified theory? Reinterpret old RO results as LearnOrCover?

Thanks!

Backup Slides

Online Covering IPs

$$\min \; c^\intercal x$$

$$a_1^\intercal x \geq 1$$
$$a_2^\intercal x \geq 1$$
$$a_3^\intercal x \geq 1$$
$$a_4^\intercal x \geq 1$$
$$a_5^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Online Covering IPs

$$\min \quad c^\intercal x$$

$$x \in \mathbb{Z}^m_{\geq 0}$$

Online Covering IPs

$$\min \; c^\intercal x$$

$$a_1^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Online Covering IPs

$$\min \; c^\intercal x$$

$$a_1^\intercal x \geq 1$$

$$a_2^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Online Covering IPs

$$\min \quad c^\intercal x$$

$$a_1^\intercal x \geq 1$$
$$a_2^\intercal x \geq 1$$
$$a_3^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Online Covering IPs

$$\min \; c^\intercal x$$

$$a_1^\intercal x \geq 1$$

$$a_2^\intercal x \geq 1$$

$$a_3^\intercal x \geq 1$$

$$a_4^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Online Covering IPs

$$\min \; c^\intercal x$$

$$a_1^\intercal x \geq 1$$

$$a_2^\intercal x \geq 1$$

$$a_3^\intercal x \geq 1$$

$$a_4^\intercal x \geq 1$$

$$a_5^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Online Covering IPs

$$\min \quad c^\intercal x$$

$$a_1^\intercal x \geq 1$$
$$a_2^\intercal x \geq 1$$
$$a_3^\intercal x \geq 1$$
$$a_4^\intercal x \geq 1$$
$$a_5^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

Goal: Maintain feasible solution x that is *monotonically* increasing.

Online Covering IPs

$$\min \quad c^\intercal x$$

$$a_1^\intercal x \geq 1$$
$$a_2^\intercal x \geq 1$$
$$a_3^\intercal x \geq 1$$
$$a_4^\intercal x \geq 1$$
$$a_5^\intercal x \geq 1$$

$$x \in \mathbb{Z}_{\geq 0}^m$$

<ins>Goal</ins>: Maintain feasible solution x that is *monotonically* increasing.

Set Cover is the special case where constraint matrix A is 0/1.

LearnOrCover for non-unit costs

LearnOrCover for non-unit costs

Main issue: # uncovered elements not good proxy for cost.

LearnOrCover for non-unit costs

Main issue: # uncovered elements <u>not</u> good proxy for cost.

LearnOrCover

Init. $x_S \leftarrow 1/m$.

@ time t, element v arrives:

  If v covered, do nothing.

  Else:

    (I) Buy every set R w.p. x_R.

    (II) $\forall S \ni v$, set $x_S \leftarrow e \cdot x_S$.

    Renormalize $x = x/\|x\|_1$.

    Buy arbitrary set to cover v.

LearnOrCover for non-unit costs

Main issue: # uncovered elements <u>not</u> good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v/c_S} \cdot x_S$.

Renormalize $x \leftarrow x/\langle c, x \rangle$.

Buy cheapest set to cover v.

LearnOrCover for non-unit costs

Main issue: # uncovered elements not good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune learning & sampling rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v / c_S} \cdot x_S$.

Renormalize $x \leftarrow x / \langle c, x \rangle$.

Buy cheapest set to cover v.

LearnOrCover for non-unit costs

Main issue: # uncovered elements not good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune learning & sampling rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v / c_S} \cdot x_S$.

Renormalize $x \leftarrow x / \langle c, x \rangle$.

Buy cheapest set to cover v.

LearnOrCover for non-unit costs

Main issue: # uncovered elements <u>not</u> good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune <u>learning</u> & <u>sampling</u> rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v / c_S} \cdot x_S$.

Renormalize $x \leftarrow x / \langle c, x \rangle$.

Buy cheapest set to cover v.

<u>Claim 1:</u> $\Phi(0) = c(\text{OPT}) \cdot O(\log mn)$, and $\Phi(t) \geq 0$.

LearnOrCover for non-unit costs

Main issue: # uncovered elements <u>not</u> good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune <u>learning</u> & <u>sampling</u> rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v / c_S} \cdot x_S$.

Renormalize $x \leftarrow x / \langle c, x \rangle$.

Buy cheapest set to cover v.

<u>**Claim 1:**</u> $\Phi(0) = c(\text{OPT}) \cdot O(\log mn)$, and $\Phi(t) \geq 0$.

<u>**Claim 2**</u>: $E[\Delta\Phi] = -\Omega(\kappa_v)$.

LearnOrCover for non-unit costs

Main issue: # uncovered elements <u>not</u> good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune <u>learning</u> & <u>sampling</u> rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v / c_S} \cdot x_S$.

Renormalize $x \leftarrow x / \langle c, x \rangle$.

Buy cheapest set to cover v.

<u>**Claim 1:**</u> $\Phi(0) = c(\text{OPT}) \cdot O(\log mn)$, and $\Phi(t) \geq 0$.

<u>**Claim 2**</u>: $E[\Delta\Phi] = -\Omega(\kappa_v)$.

<u>**Claim 3**</u>: $E[\Delta \text{cost(ALG)}] = O(\kappa_v)$.

LearnOrCover for non-unit costs

Main issue: # uncovered elements not good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune learning & sampling rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v / c_S} \cdot x_S$.

Renormalize $x \leftarrow x / \langle c, x \rangle$.

Buy cheapest set to cover v.

Claim 1: $\Phi(0) = c(\text{OPT}) \cdot O(\log mn)$, and $\Phi(t) \geq 0$.

Claim 2: $E[\Delta\Phi] = -\Omega(\kappa_v)$.

Claim 3: $E[\Delta \text{cost(ALG)}] = O(\kappa_v)$.

$$\Rightarrow \quad E[\Delta\Phi + \Delta\text{cost(ALG)}] = 0.$$

LearnOrCover for non-unit costs

Main issue: # uncovered elements not good proxy for cost.

(Assuming WLOG $c(OPT) = 1$)

κ_v := cost of cheapest set covering v

Main Idea: tune learning & sampling rates as a function of κ_v.

LearnOrCover

Init. $x_S \leftarrow 1/(c_S \cdot m)$.

@ time t, element v arrives:

If v covered, do nothing.

Else:

(I) Buy every set R w.p. $\kappa_v x_R$.

(II) $\forall S \ni v$, set $x_S \leftarrow e^{\kappa_v/c_S} \cdot x_S$.

Renormalize $x \leftarrow x/\langle c, x\rangle$.

Buy cheapest set to cover v.

Claim 1: $\Phi(0) = c(\text{OPT}) \cdot O(\log mn)$, and $\Phi(t) \geq 0$.

Claim 2: $E[\Delta\Phi] = -\Omega(\kappa_v)$.

Claim 3: $E[\Delta\text{cost(ALG)}] = O(\kappa_v)$.

$$\Rightarrow \quad E[\Delta\Phi + \Delta\text{cost(ALG)}] = 0.$$

$$E[\text{cost(ALG)}] \leq \Phi(0).$$