Pivotal Sampling for Online Stochastic Matching


Mark Braverman


Mahsa Derakhshan


Tristan Pollner

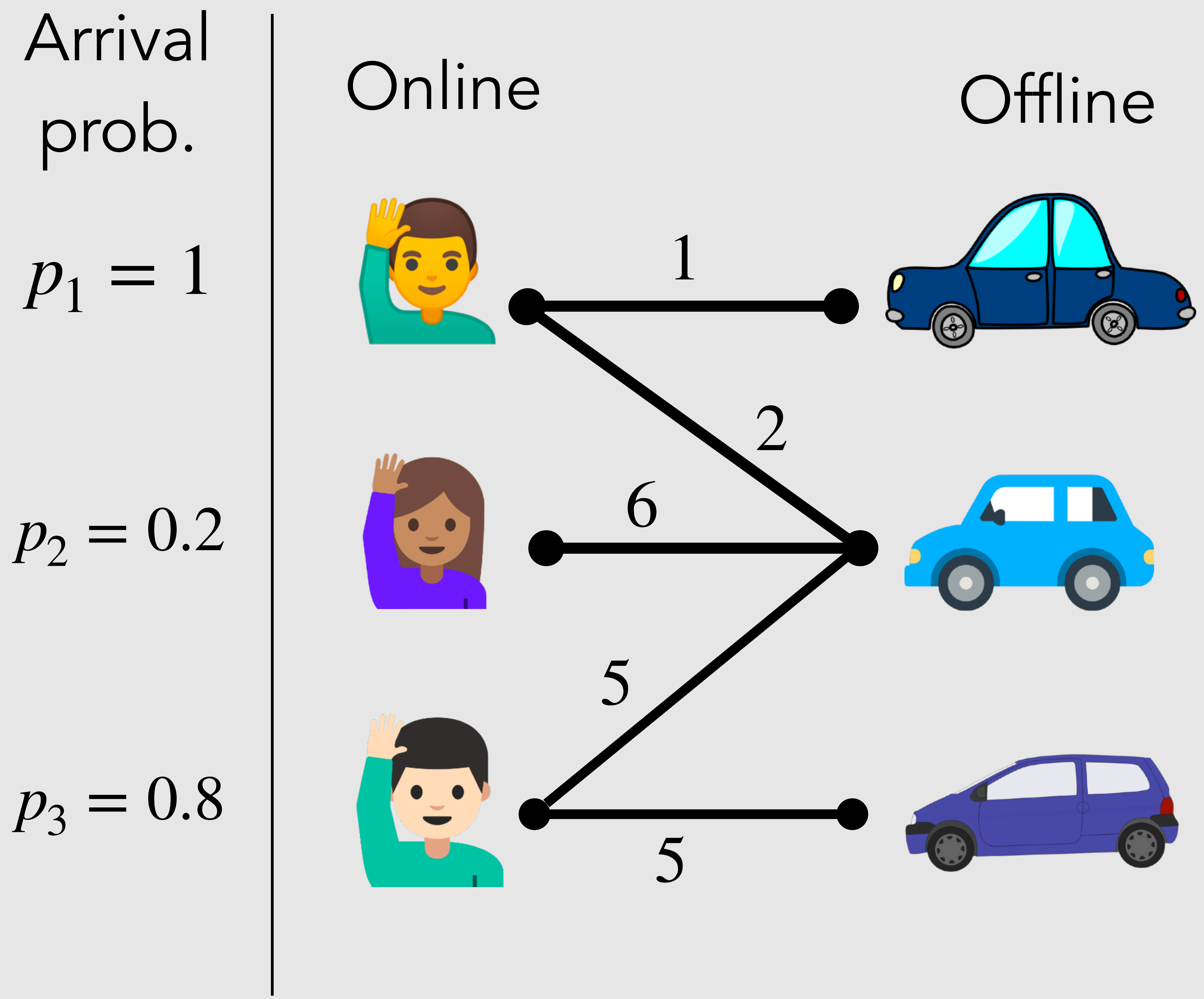

Amin Saberi


David Wajc

Online Stochastic Matching

- Build a matching online in a weighted bipartite graph between **online nodes** and **offline nodes**

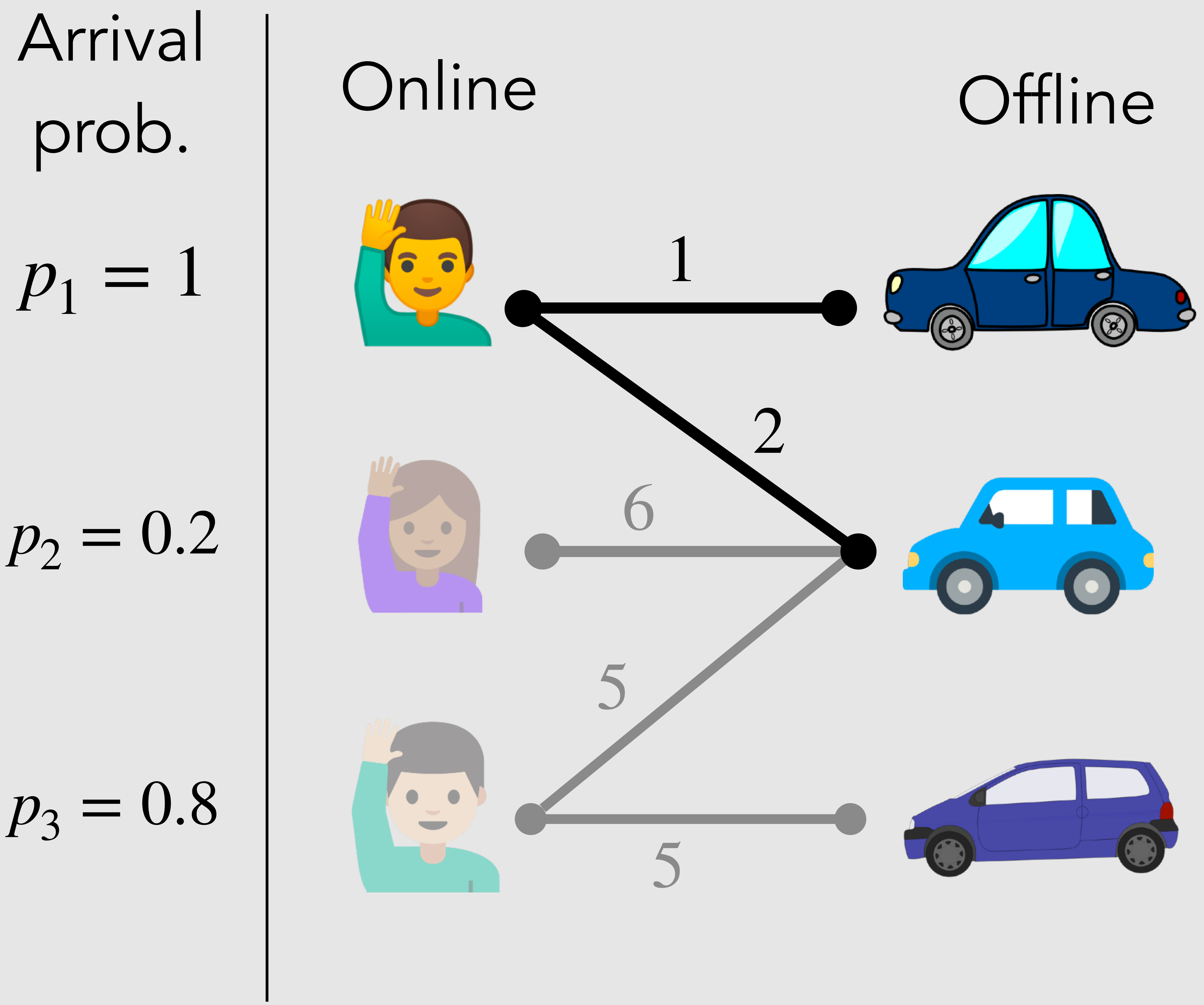
- Each online node $t \in \{1, 2, \ldots, T\}$ arrives with known probability p_t



Online Stochastic Matching

- Build a matching online in a weighted bipartite graph between **online nodes** and **offline nodes**

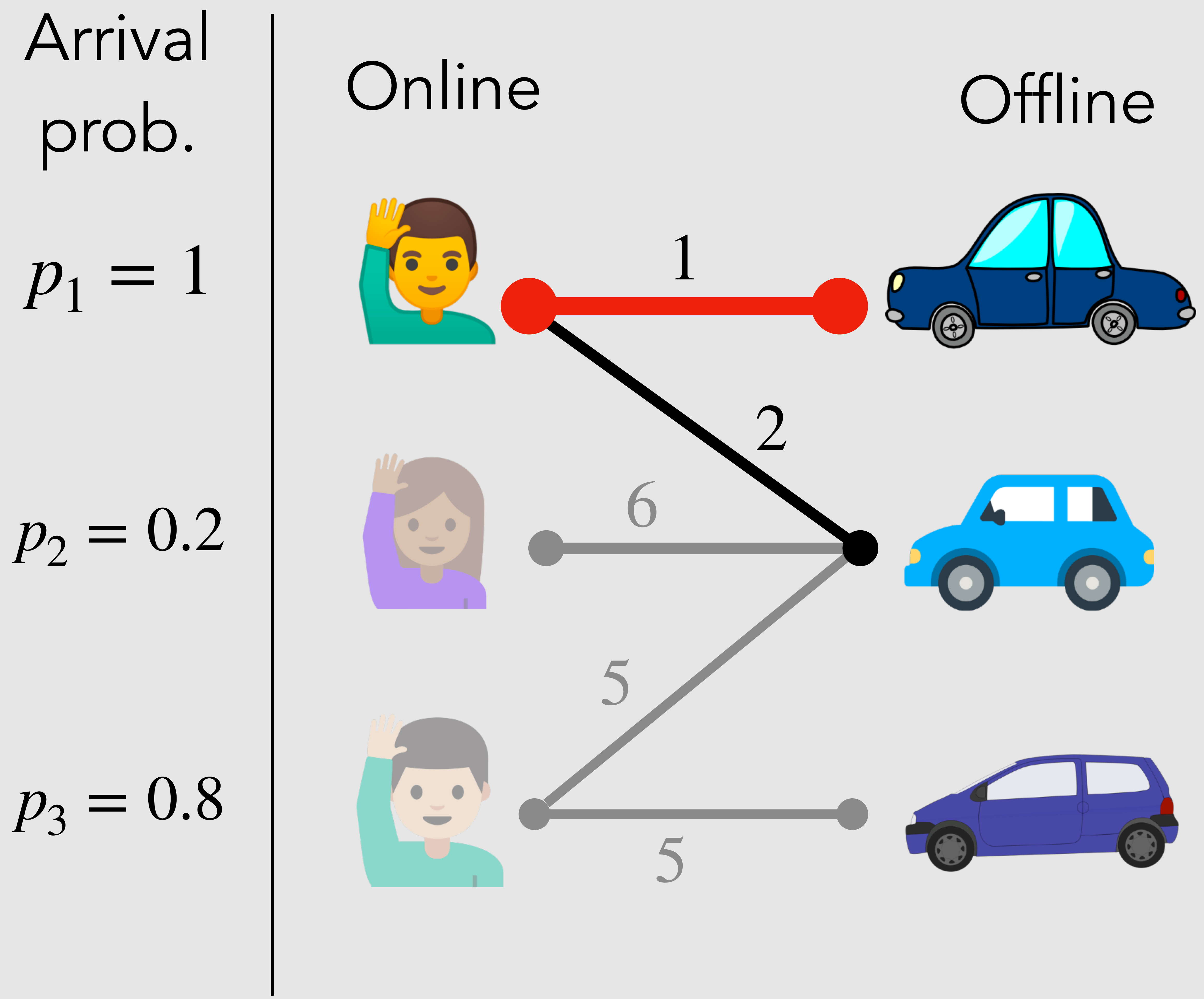
- Each online node $t \in \{1,2,\ldots,T\}$ arrives with known probability p_t



Online Stochastic Matching

- Build a matching online in a weighted bipartite graph between **online nodes** and **offline nodes**

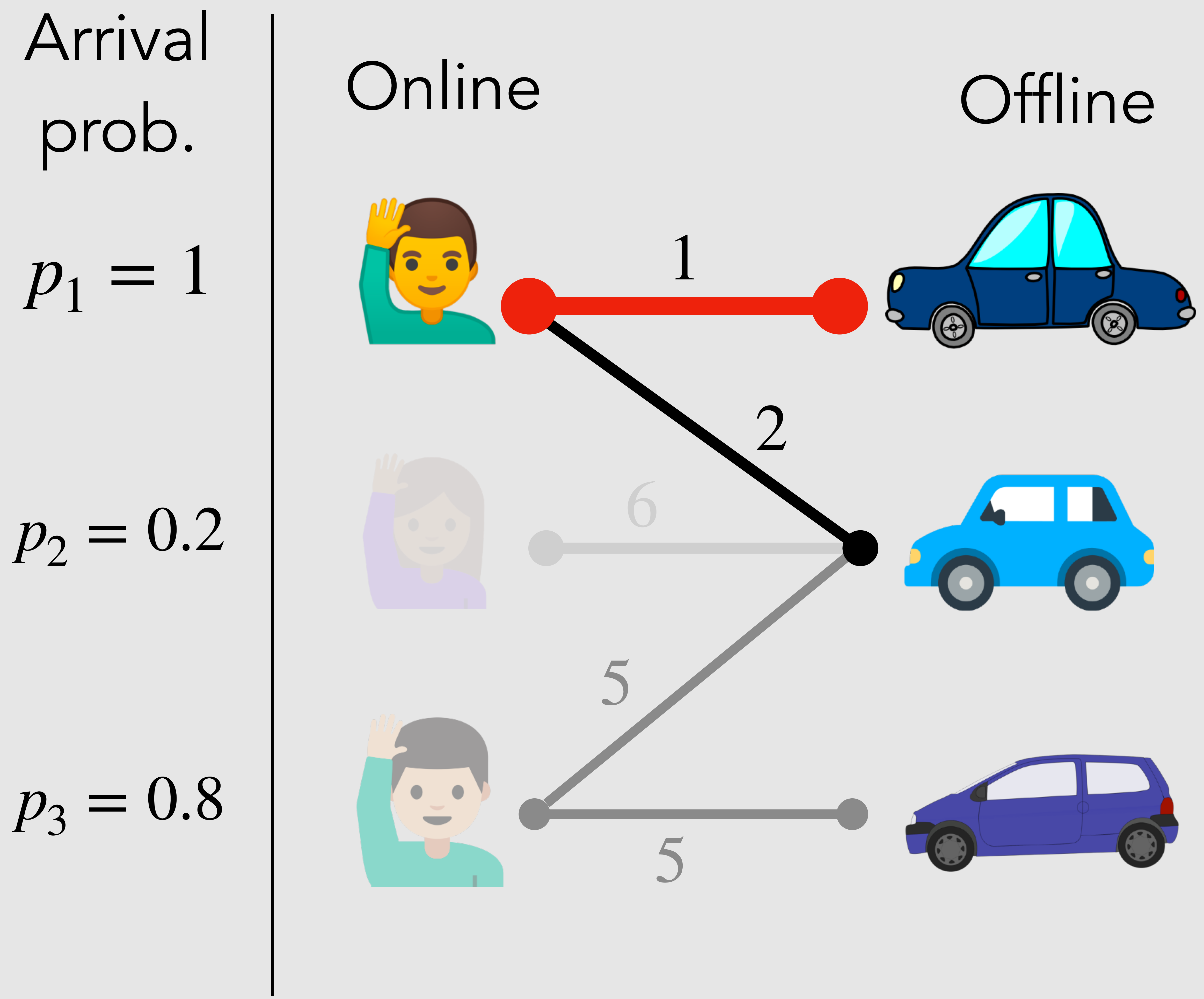
- Each online node $t \in \{1,2,\ldots,T\}$ arrives with known probability p_t



Online Stochastic Matching

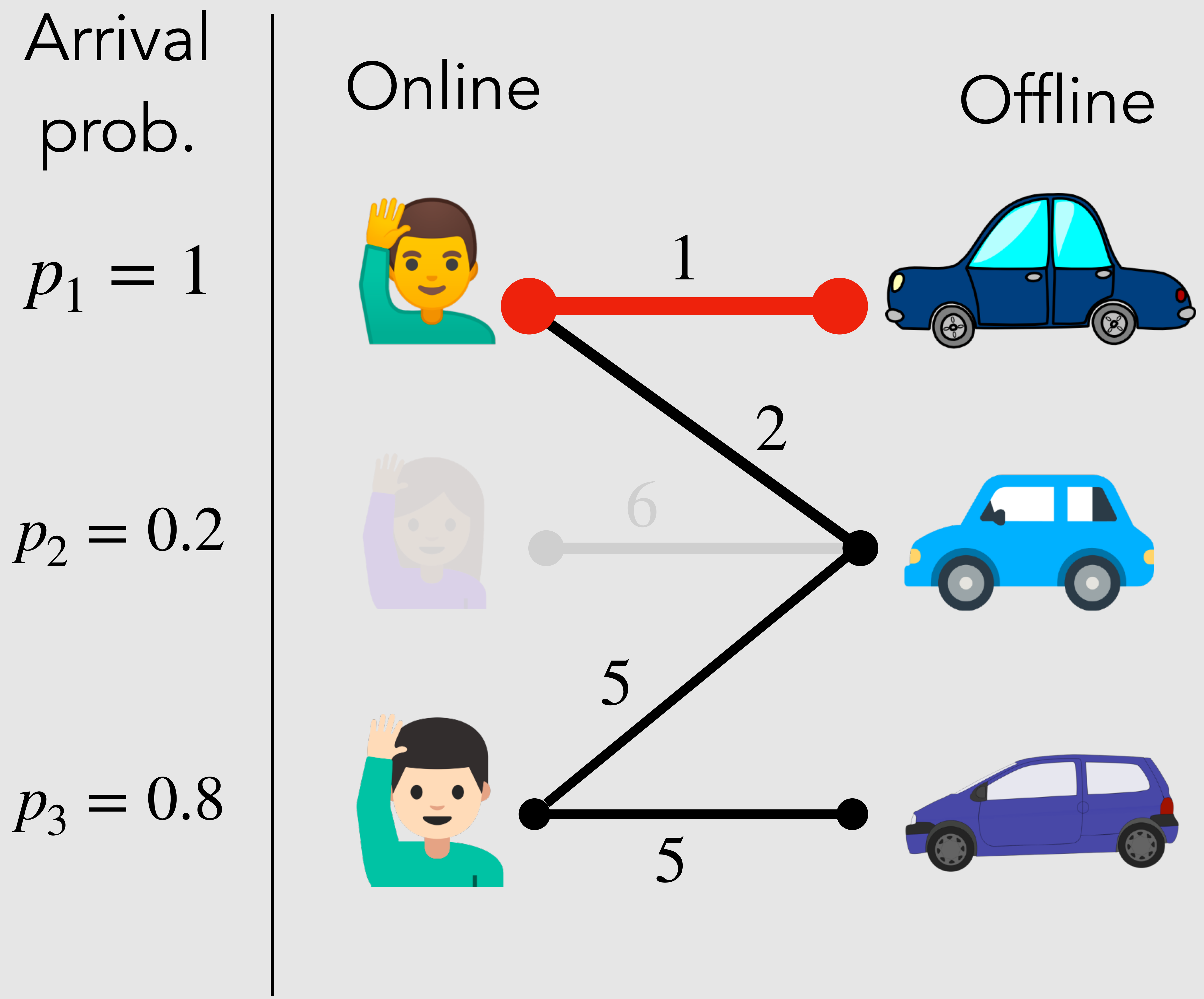
- Build a matching online in a weighted bipartite graph between **online nodes** and **offline nodes**

- Each online node $t \in \{1,2,\ldots,T\}$ arrives with known probability p_t



Online Stochastic Matching

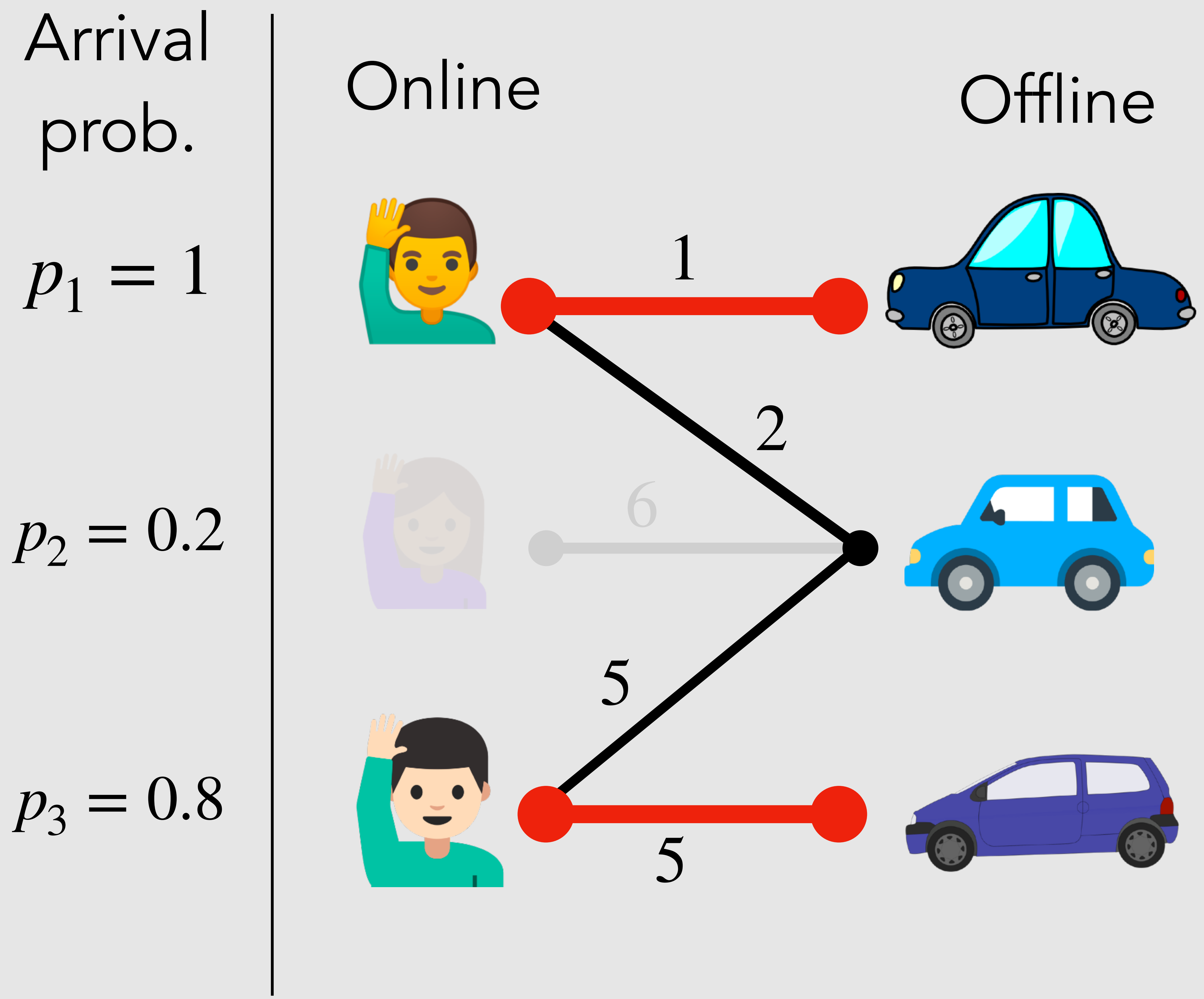
- Build a matching online in a weighted bipartite graph between **online nodes** and **offline nodes**
- Each online node $t \in \{1,2,\ldots,T\}$ arrives with known probability p_t



Online Stochastic Matching

- Build a matching online in a weighted bipartite graph between **online nodes** and **offline nodes**

- Each online node $t \in \{1,2,\ldots,T\}$ arrives with known probability p_t



Prior Work

0.5-competitive online algorithms

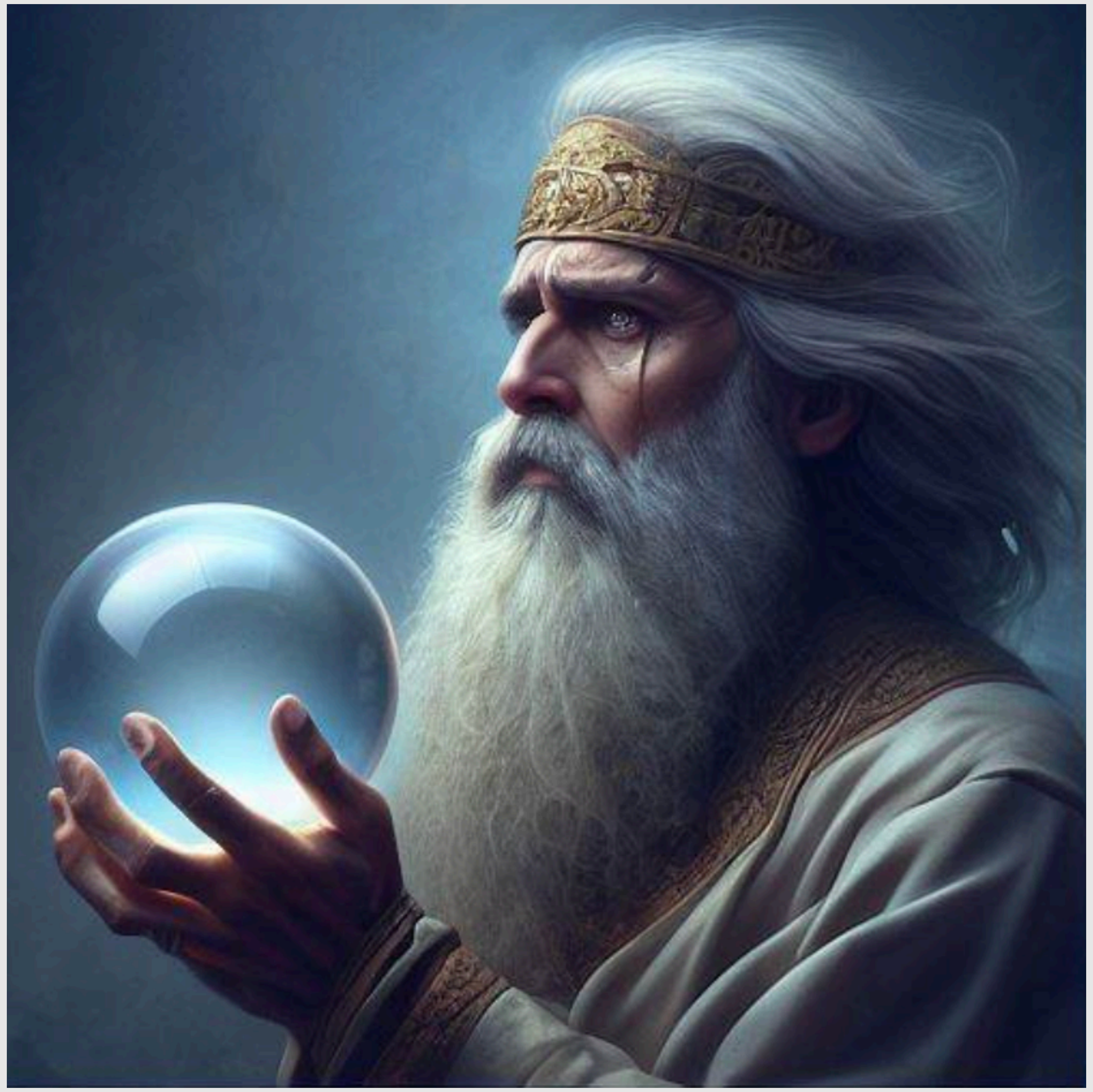
- One offline node: [KSG'78]
- Multiple offline nodes: [FGL'14, EFGT'20]

Prior Work

0.5-competitive online algorithms

- One offline node: [KSG'78]
- Multiple offline nodes: [FGL'14, EFGT'20]

Often called **prophet inequalities**



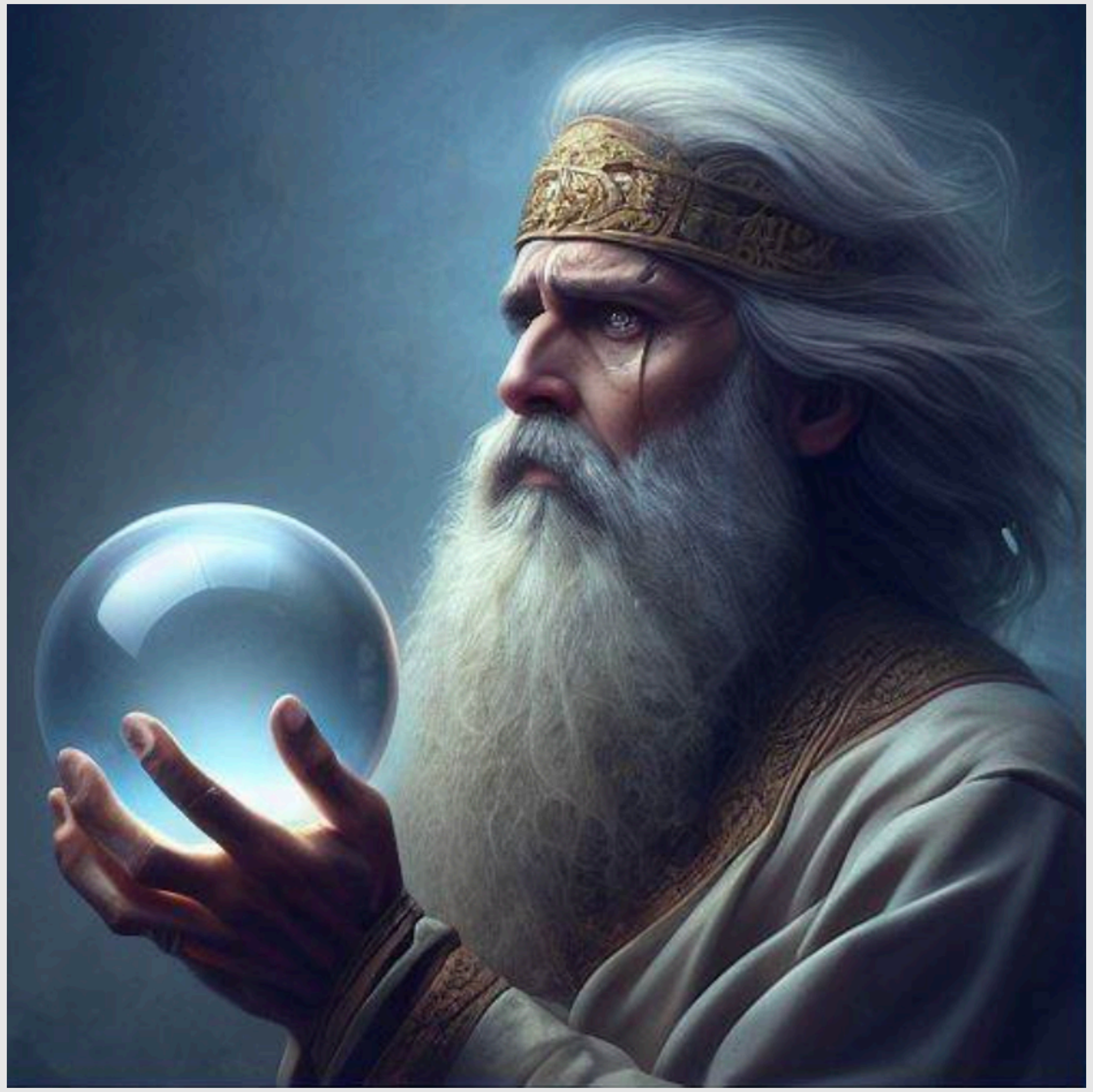
Prior Work

0.5-competitive online algorithms

- One offline node: [KSG'78]
- Multiple offline nodes: [FGL'14, EFGT'20]

Often called **prophet inequalities**

- Also for k-unit allocation [Alaei'11], Matroids [KW'12], General Downwards-Closed [Rubinstein'16], I.I.D. Matching [FMMM'2009, MOS'10, …], …



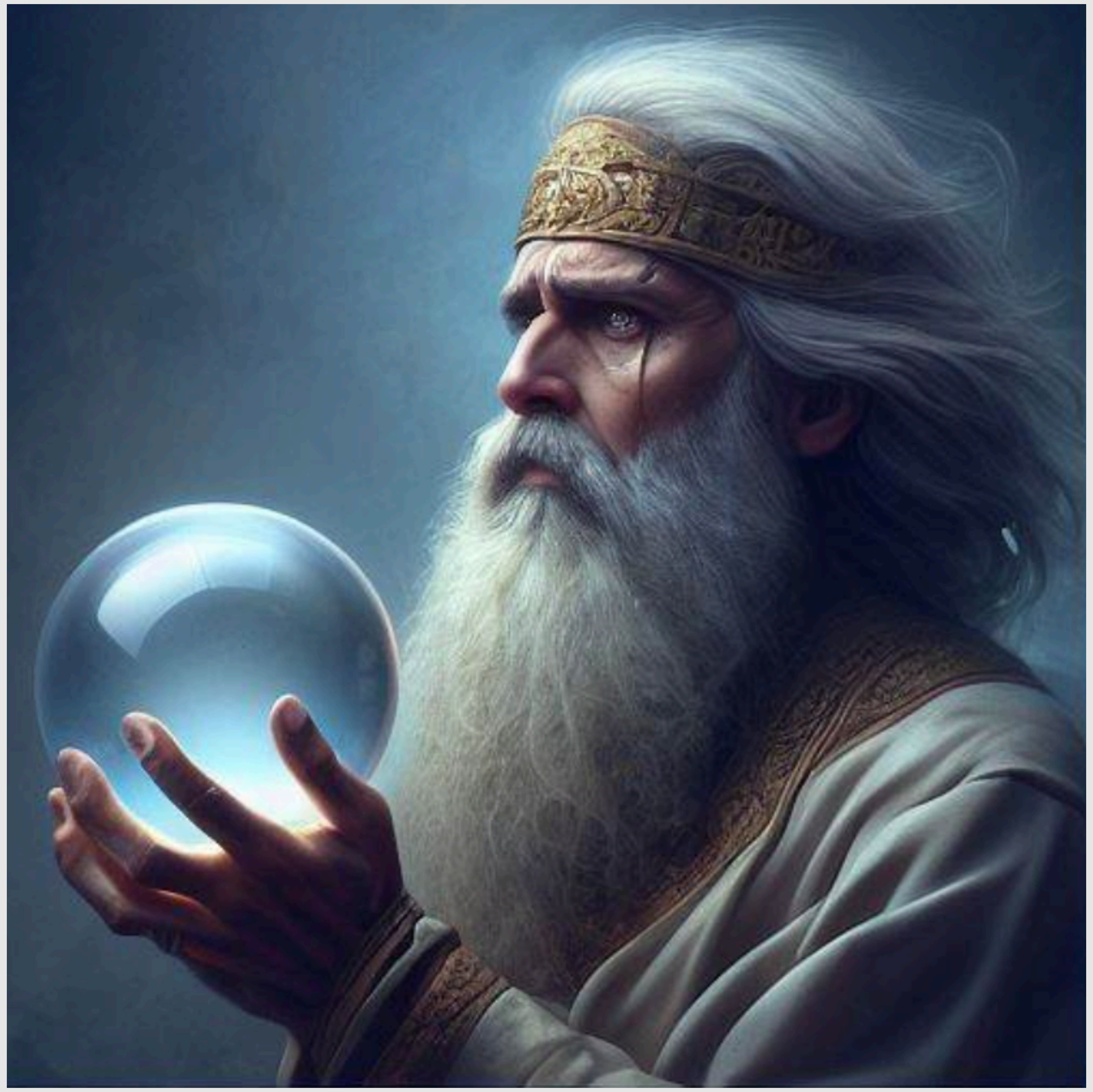
Prior Work

0.5-competitive online algorithms

- One offline node: [KSG'78]
- Multiple offline nodes: [FGL'14, EFGT'20]

Often called **prophet inequalities**

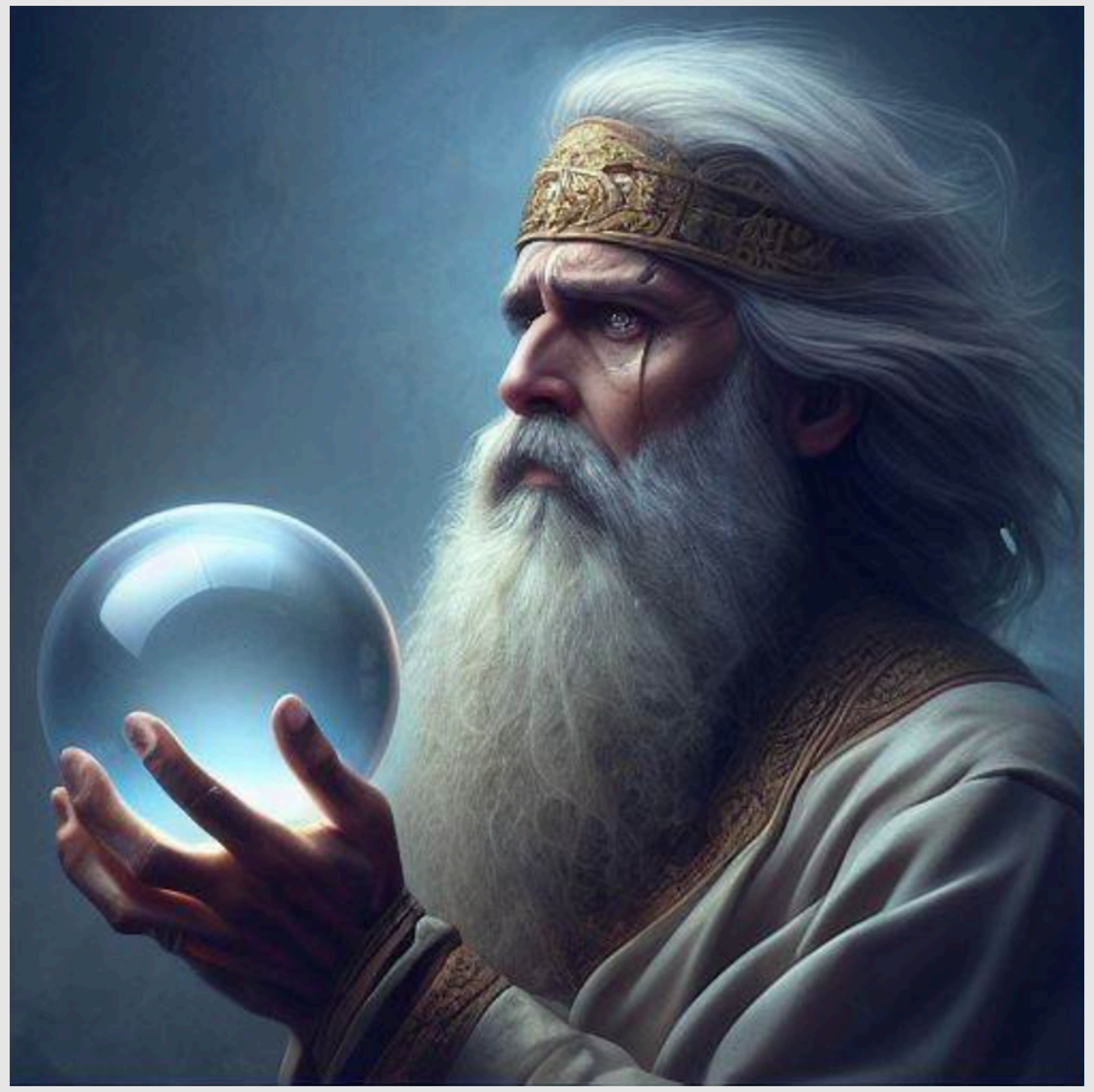
- Also for k-unit allocation [Alaei'11], Matroids [KW'12], General Downwards-Closed [Rubinstein'16], I.I.D. Matching [FMMM'2009, MOS'10, …], …



Observation. 0.5 is tight in the worst case.

Our Benchmark

Competitive analysis has a strong benchmark.



Our Benchmark

Competitive analysis has a strong benchmark.

Our Benchmark. OPT_{on} := the optimal online algorithm

Our Benchmark

Competitive analysis has a strong benchmark.

Our Benchmark. OPT_{on} := **the optimal online algorithm**

Can compute exactly by solving the Bellman equation (**exponential time**)

Our Results

Hardness [PPSW'21]. There exists $\epsilon > 0$ such that it is PSPACE-hard to $(1 - \epsilon)$-approximate OPT_{on}.

Algorithm. New best approximation ratio for polynomial-time algorithms.

Our Results

Hardness [PPSW'21]. There exists $\epsilon > 0$ such that it is PSPACE-hard to $(1-\epsilon)$-approximate OPT_{on}.

Algorithm. New best approximation ratio for polynomial-time algorithms.

Line of Work. $0.51 \to 0.526 \to 0.632 \to 0.65 \to 0.671$

PPSW'21 SW'21 BDM'22 NSW'22 BDPSW'23

Our Results

Hardness [PPSW'21]. There exists $\epsilon > 0$ such that it is PSPACE-hard to $(1-\epsilon)$-approximate OPT_{on}.

Algorithm. New best approximation ratio for polynomial-time algorithms.

Yesterday

Line of Work. $0.51 \rightarrow 0.526 \dashrightarrow 0.632 \dashrightarrow 0.65 \rightarrow 0.671$

PPSW'21 SW'21 BDM'22 NSW'22 BDPSW'23

Our Results

Hardness [PPSW'21]. There exists $\epsilon > 0$ such that it is PSPACE-hard to $(1 - \epsilon)$-approximate OPT_{on}.

Algorithm. New best approximation ratio for polynomial-time algorithms.

	Yesterday		Today
Line of Work. $0.51 \rightarrow 0.526 \rightarrow$	0.632	$\rightarrow 0.65 \rightarrow$	0.671
PPSW'21 SW'21	BDM'22	NSW'22	BDPSW'23

Our Results

Hardness [PPSW'21]. There exists $\epsilon > 0$ such that it is PSPACE-hard to $(1-\epsilon)$-approximate OPT_{on}.

Algorithm. New best approximation ratio for polynomial-time algorithms.

Yesterday | Today

Line of Work. $0.51 \rightarrow 0.526 \rightarrow 0.632 \rightarrow 0.65 \rightarrow 0.671$

PPSW'21 | SW'21 | BDM'22 | NSW'22 | BDPSW'23

Remainder of talk: Edges have $\{0,1\}$ weights

Main Ingredients

Pivotal Sampling

- Technique for **correlating** proposals from offline nodes

Analysis of Correlation of Offline Nodes

- Show offline nodes satisfy **negative cylinder dependence (NCD)**

New Tail Expectation Bounds

- For sums of NCD random variables

Matching with Independent Proposals

Solve **Linear Programming Relaxation**

For each online t:

For each free i, **propose independently** w.p. $\dfrac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)}$

$\text{Prop}_t \leftarrow$ set of proposing offline nodes

For $i^* := \max(\text{Prop}_t)$, match to t iff t arrives

For $i \in \text{Prop}_t \setminus \{i^*\}$, discard independently w.p. p_t

Linear Programming Relaxation for OPT_{on}

Pr[offline node i matched to online node t]

$$\max \sum_{i,t} w_{i,t} \cdot x_{i,t}$$

$$\text{s.t. } x_{i,t} \geq 0 \qquad \text{for all } i, t$$

$$\sum_i x_{i,t} \leq p_t \qquad \text{for all } t$$

$$\sum_t x_{i,t} \leq 1 \qquad \text{for all } i$$

$$x_{i,t} \leq p_t \cdot \left(1 - \sum_{t' < t} x_{i,t'}\right) \qquad \text{for all } i, t$$

Pr[t arrives]

Pr[i free when t arrives]

Matching with Independent Proposals

Solve **Linear Programming Relaxation**

For each online t:

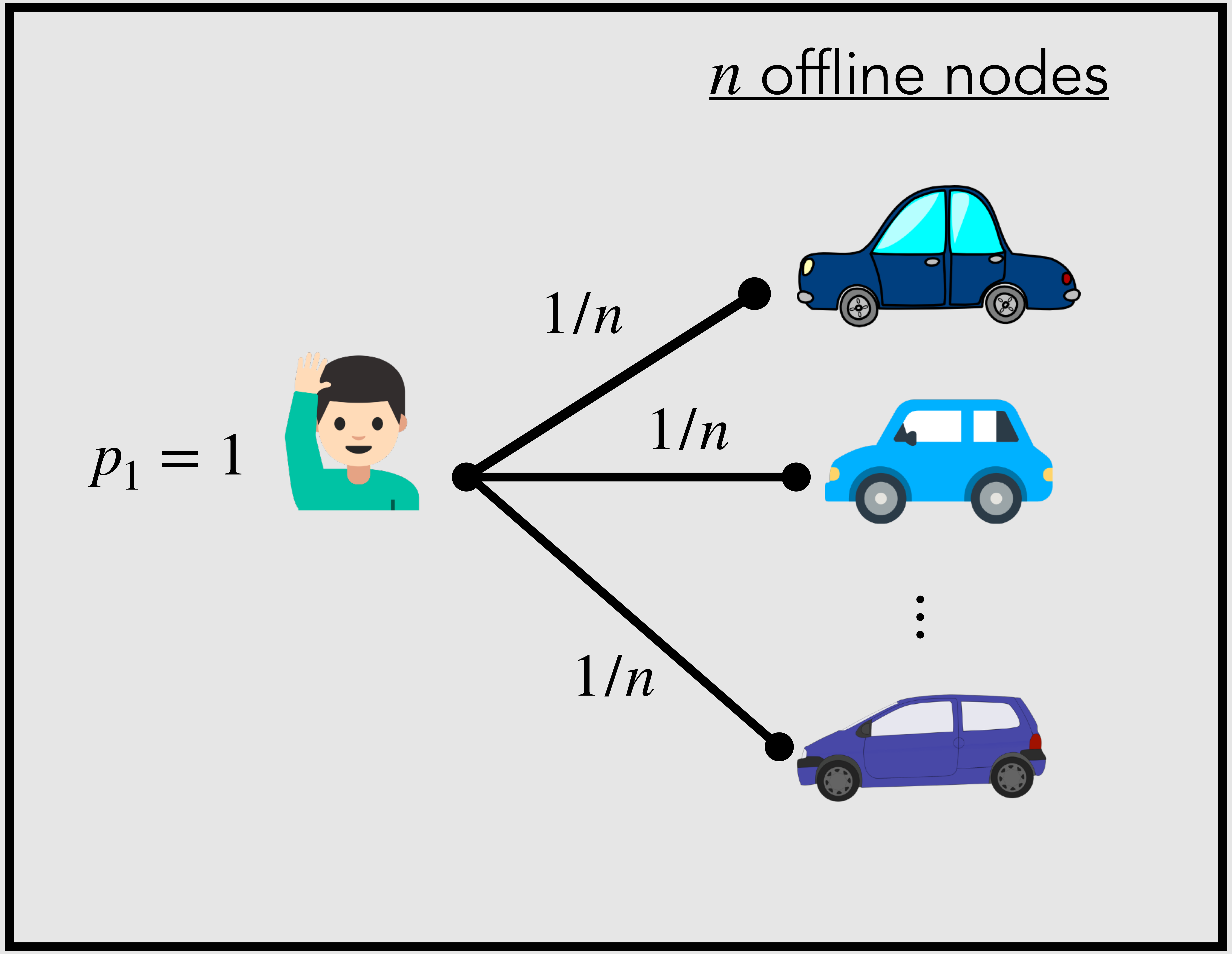
For each free i, **propose independently** w.p. $\dfrac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)}$

$\text{Prop}_t \leftarrow$ set of proposing offline nodes

For $i^* := \max(\text{Prop}_t)$, match to t iff t arrives

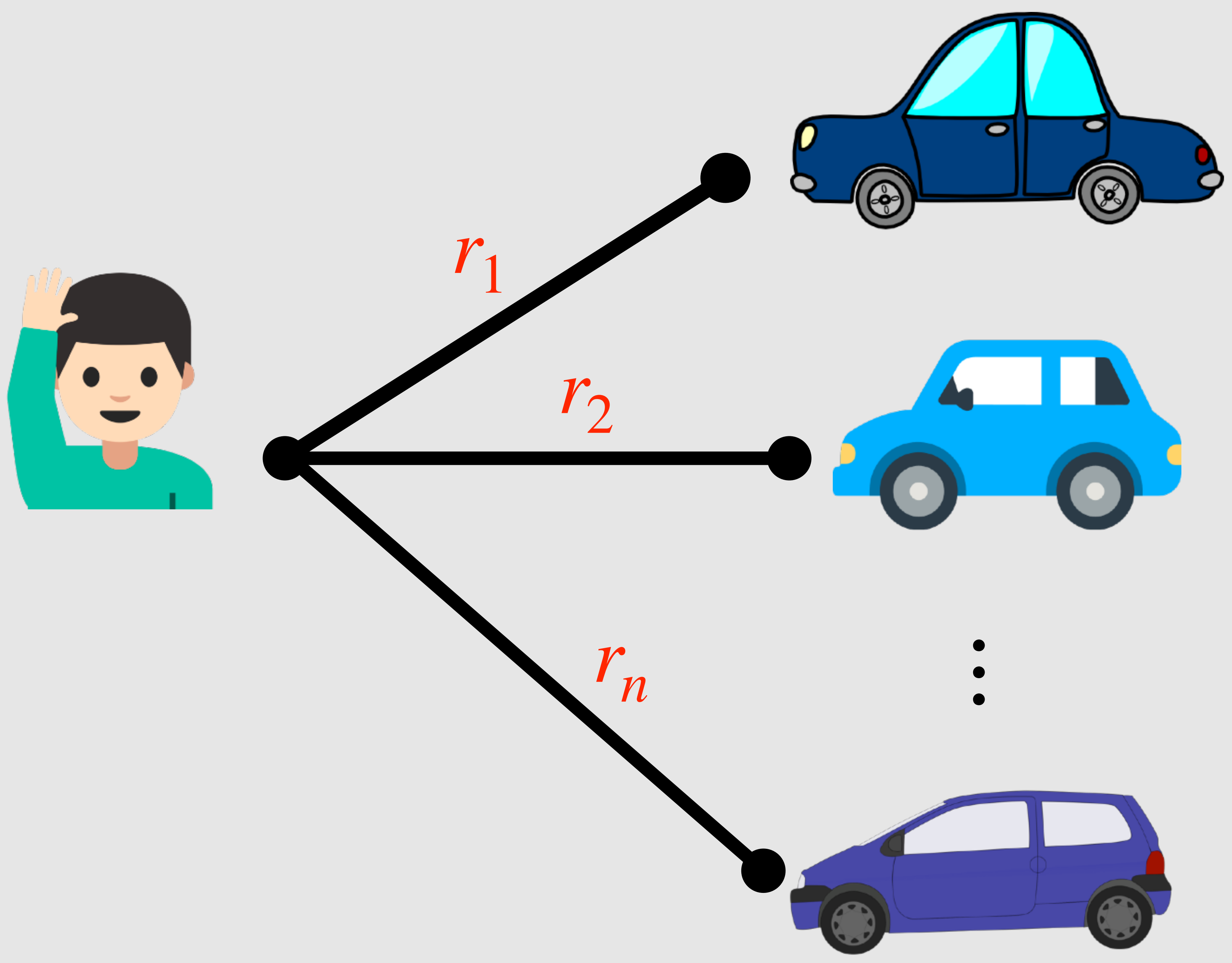
For $i \in \text{Prop}_t \setminus \{i^*\}$, discard independently w.p. p_t

Independent proposals can't beat $(1 - 1/e)$

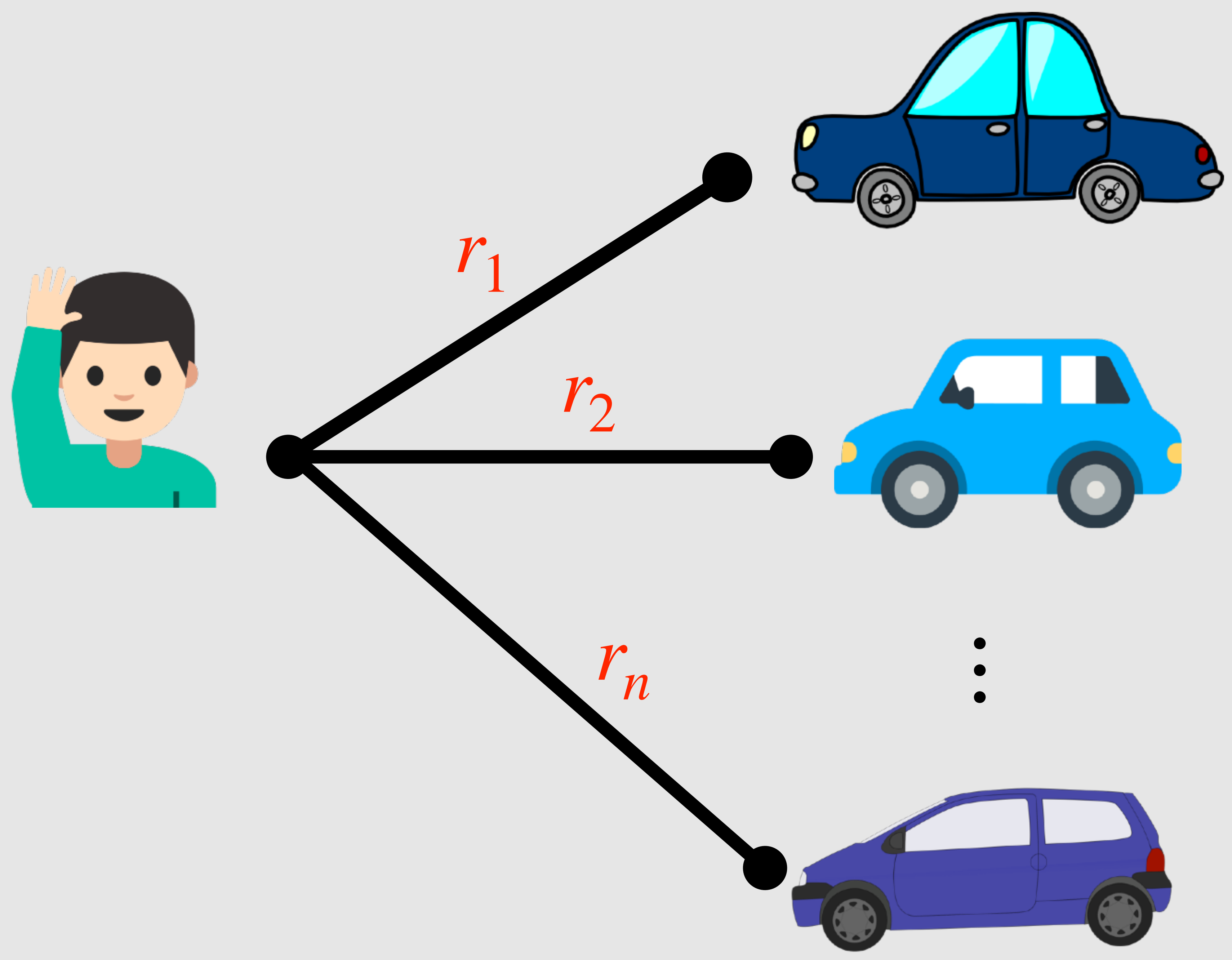


$$\Pr[\geq 1 \text{ proposal}] = 1 - (1 - 1/n)^n$$
$$\to 1 - 1/e$$

Algorithmic Contribution: Correlating Proposals



Algorithmic Contribution: Correlating Proposals



Pivotal Sampling

Input: Proposal probabilities $\{r_1, r_2, \ldots, r_n\}$

Output: Random subset S of $[n]$, satisfying

(i) $\Pr[i \in S] = r_i$

(ii) $\Pr[S$ intersects $\{1,2,\ldots,k\}] = \min\left(1, \sum_{i=1}^{k} r_i\right)$

(iii) $\{\mathbb{1}[i \in S]\}_i$ are negatively associated

Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

$r_1 = 0.3$ $r_2 = 0.6$ $r_3 = 0.7$

Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

$r_1 = 0.3$ $r_2 = 0.6$ $r_3 = 0.7$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Pivot

$r_1 = 0.3$ $r_2 = 0.6$ $r_3 = 0.7$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Pivot(r_i, r_j) outputs a random pair $(R_i, R_j) \in [0,1]^2$ such that:

- $R_i + R_j = r_i + r_j$
- (R_i, R_j) has a $\{0,1\}$ value
- $\mathbb{E}[R_i] = r_i$

Pivot

$r_1 = 0.3$ $r_2 = 0.6$ $r_3 = 0.7$

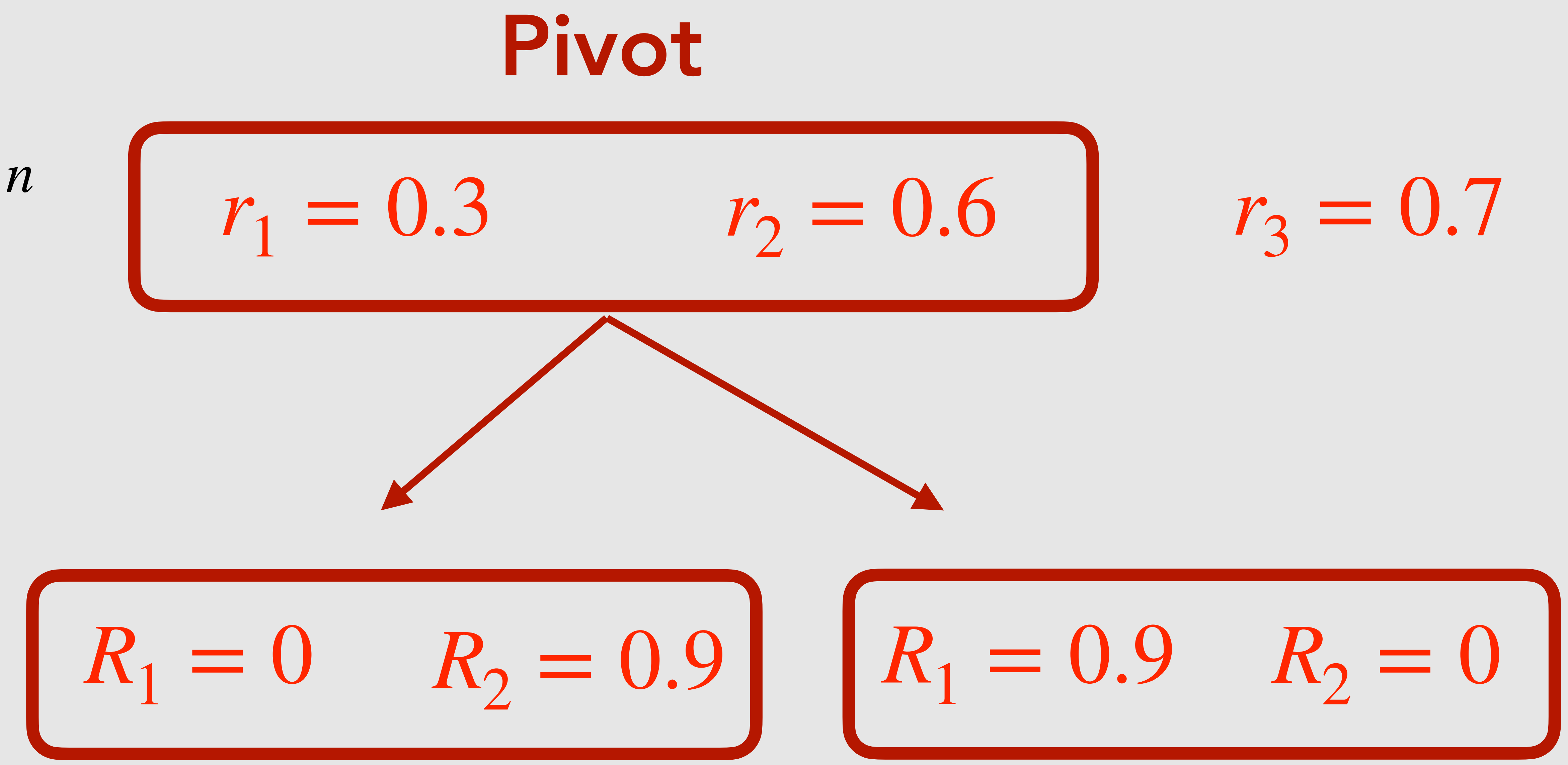
Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Pivot(r_i, r_j) outputs a random pair $(R_i, R_j) \in [0,1]^2$ such that:

- $R_i + R_j = r_i + r_j$
- (R_i, R_j) has a $\{0,1\}$ value
- $\mathbb{E}[R_i] = r_i$



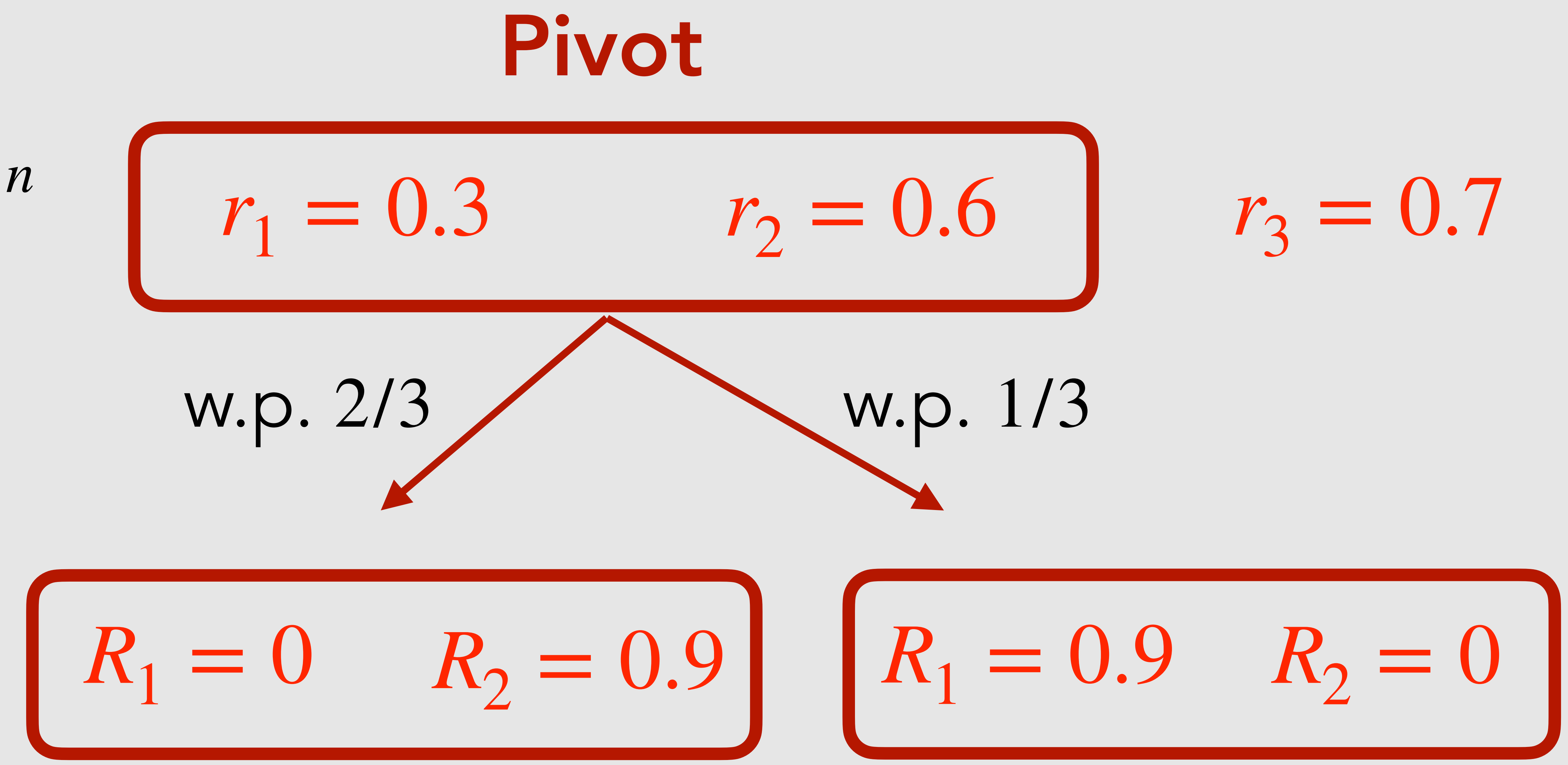
Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Pivot(r_i, r_j) outputs a random pair $(R_i, R_j) \in [0,1]^2$ such that:

- $R_i + R_j = r_i + r_j$
- (R_i, R_j) has a $\{0,1\}$ value
- $\mathbb{E}[R_i] = r_i$



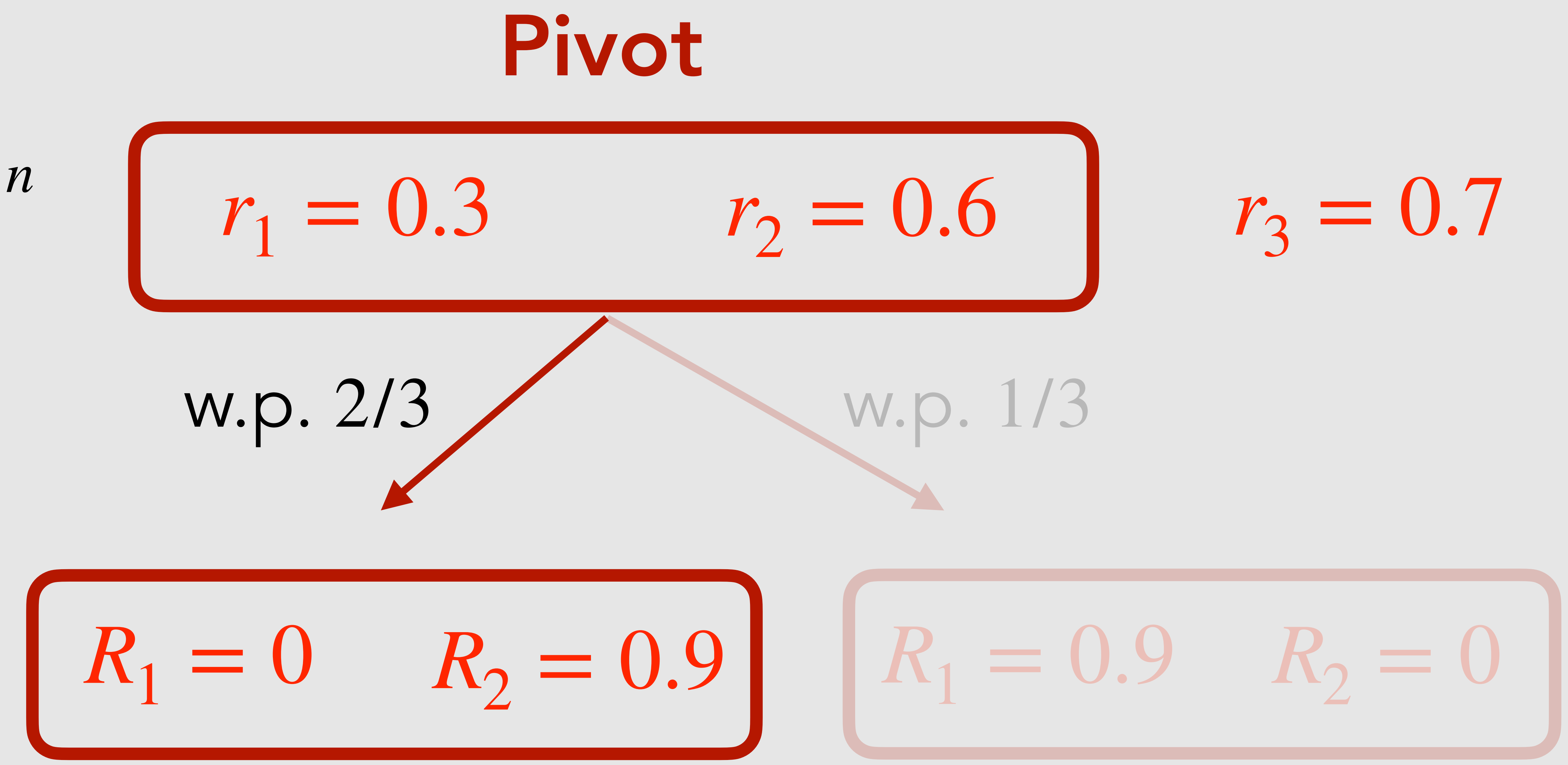
Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Pivot(r_i, r_j) outputs a random pair $(R_i, R_j) \in [0,1]^2$ such that:

- $R_i + R_j = r_i + r_j$
- (R_i, R_j) has a $\{0,1\}$ value
- $\mathbb{E}[R_i] = r_i$



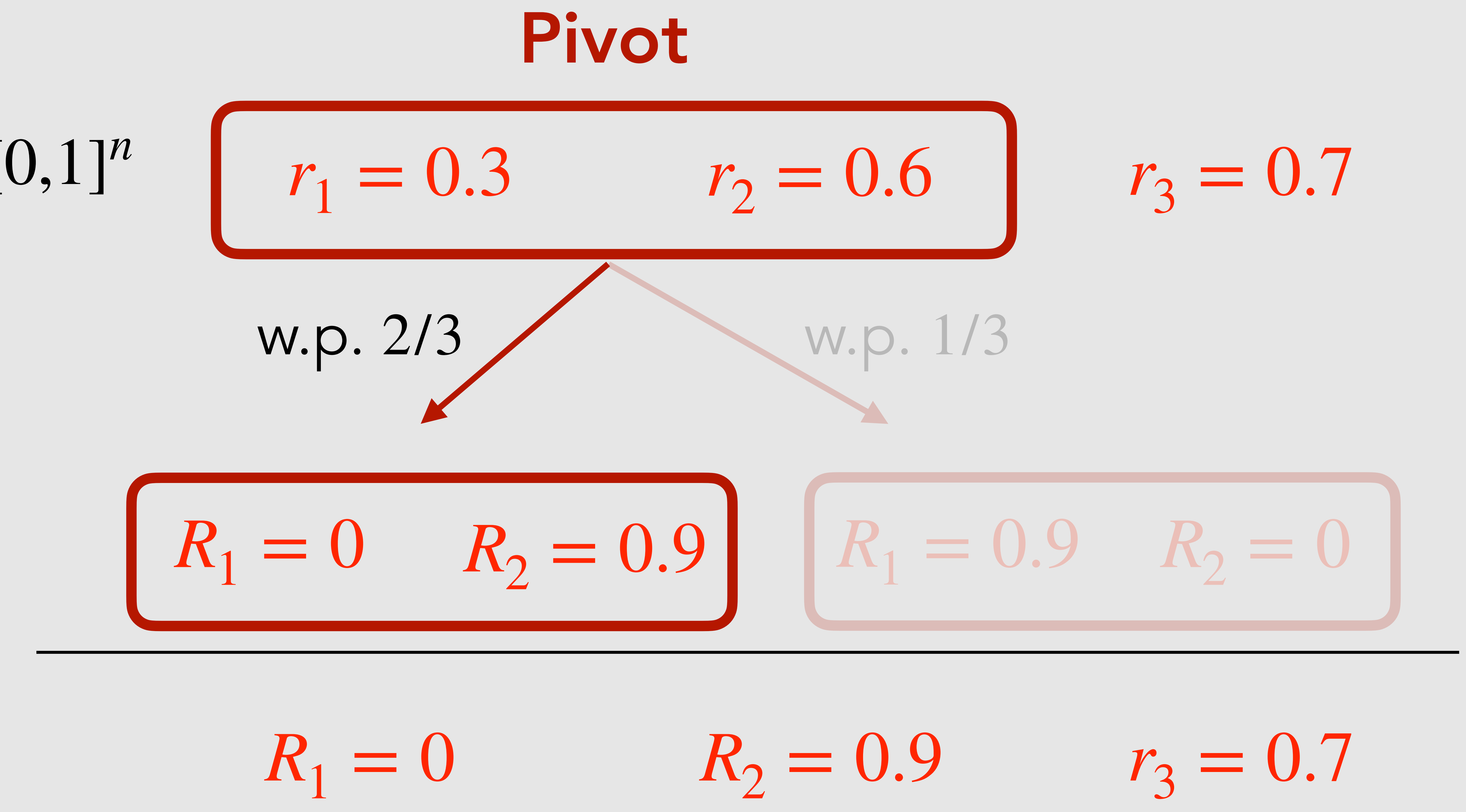
Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Pivot(r_i, r_j) outputs a random pair $(R_i, R_j) \in [0,1]^2$ such that:

- $R_i + R_j = r_i + r_j$
- (R_i, R_j) has a $\{0,1\}$ value
- $\mathbb{E}[R_i] = r_i$



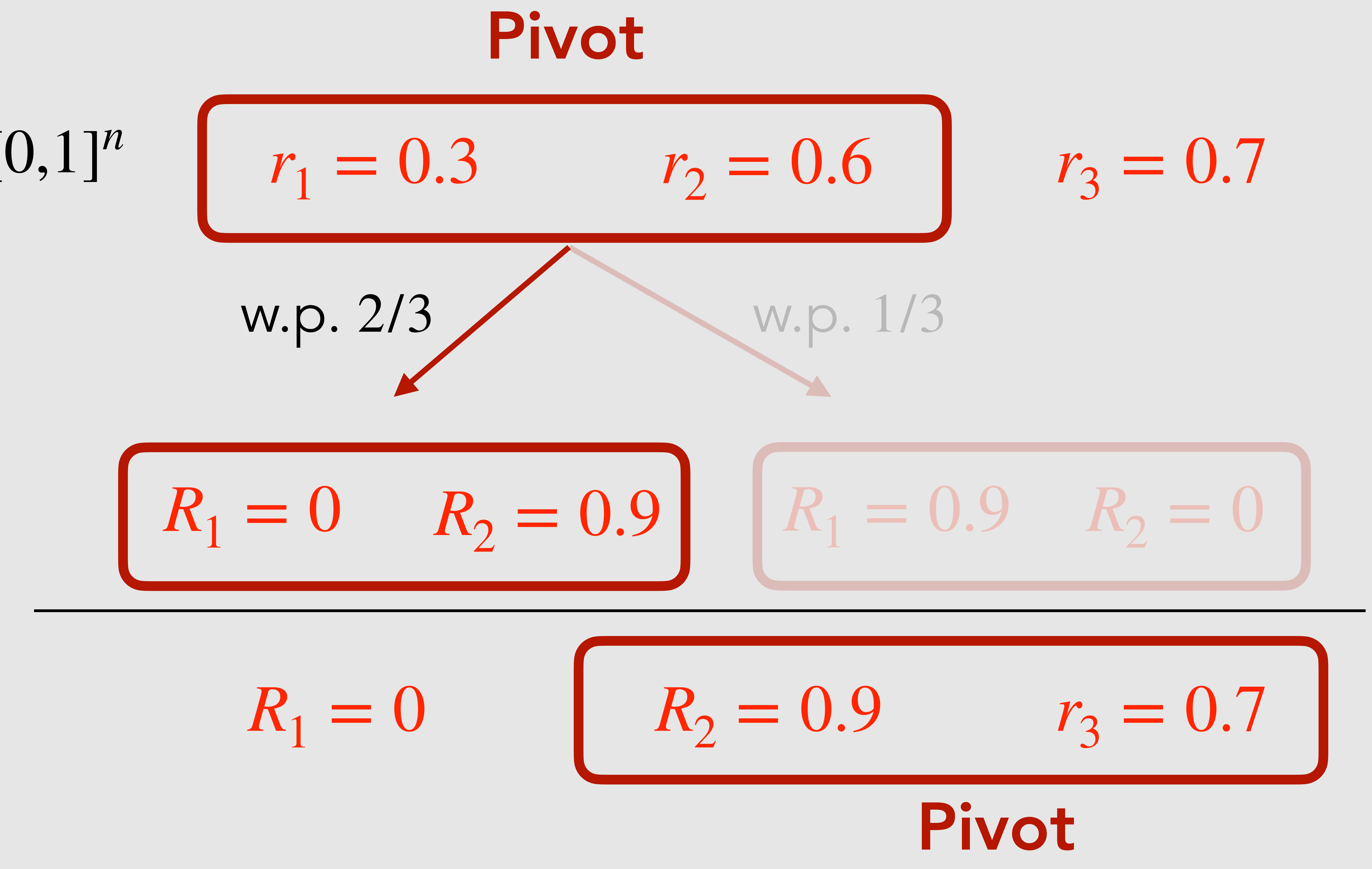
Mechanics of Pivotal Sampling [Sri'01]

Input. Proposal probabilities $(r_1, r_2, \ldots, r_n) \in [0,1]^n$

Algorithm. If r_i denotes first fractional marginal, and r_j the next, apply **Pivot**(r_i, r_j)

Pivot(r_i, r_j) outputs a random pair $(R_i, R_j) \in [0,1]^2$ such that:

- $R_i + R_j = r_i + r_j$
- (R_i, R_j) has a $\{0,1\}$ value
- $\mathbb{E}[R_i] = r_i$



Matching via Pivotal Sampling

$F_{i,t} = \mathbb{1}[i \text{ not matched/discarded at } t]$

For each t:

$$\text{Prop}_t \leftarrow \textbf{Pivotal-Sampling}\left(\left(\frac{x_{i,t}}{p_t \cdot \left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)_i\right)$$

For $i^* := \max(\text{Prop}_t)$, match to t iff t arrives

For $i \in \text{Prop}_t \setminus \{i^*\}$, discard independently w.p. p_t

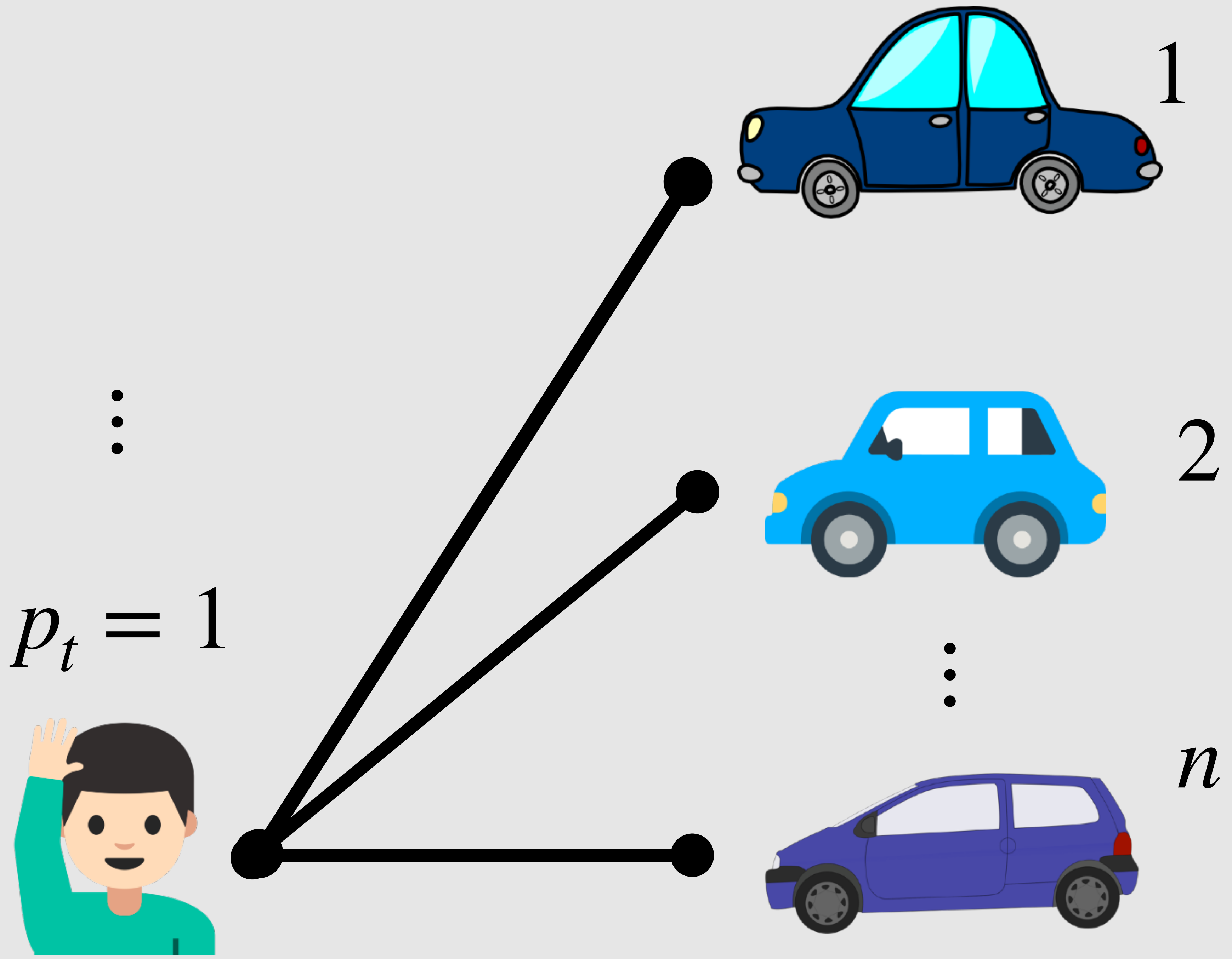
Negative Correlation of Offline Nodes

$F_{i,t} := \mathbb{1}[i$ not matched or discarded before t's arrival]

Lemma. $\{F_{i,t}\}_i$ satisfy **negative cylinder dependence (NCD)**. I.e., for any subset of offline nodes S,

$$\Pr\left[\bigwedge_{i\in S} F_{i,t}\right] \le \prod_{i\in S} \Pr[F_{i,t}] \quad \text{and} \quad \Pr\left[\bigwedge_{i\in S} \overline{F_{i,t}}\right] \le \prod_{i\in S} \Pr[\overline{F_{i,t}}].$$

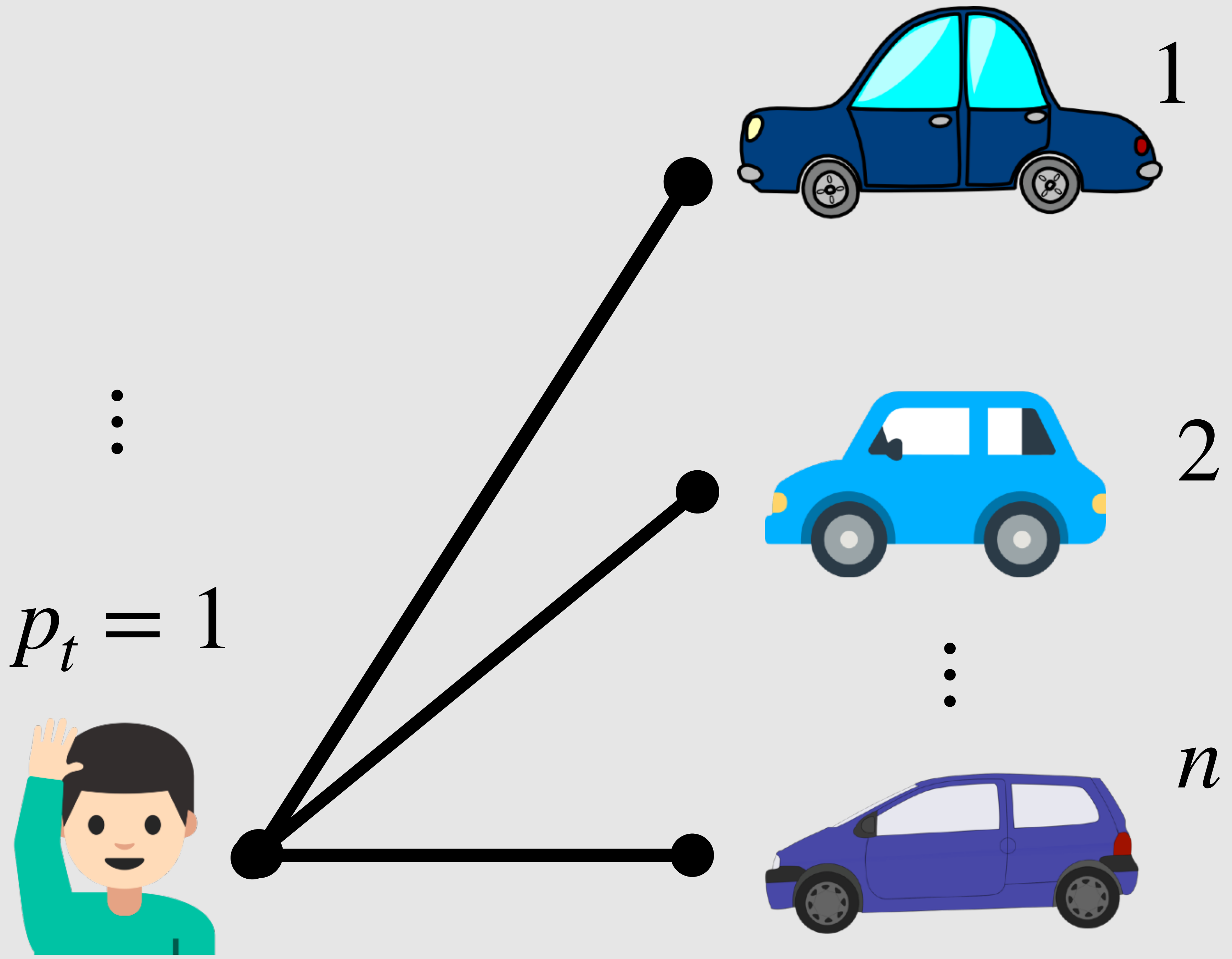
Importance of Correlation of $\{F_{i,t}\}_i$



Example. Offline nodes $i = 1, 2, \ldots, n$

- $\Pr[F_{i,t}] = 1/n$
- i's proposal probability $= 1$

Importance of Correlation of $\{F_{i,t}\}_i$

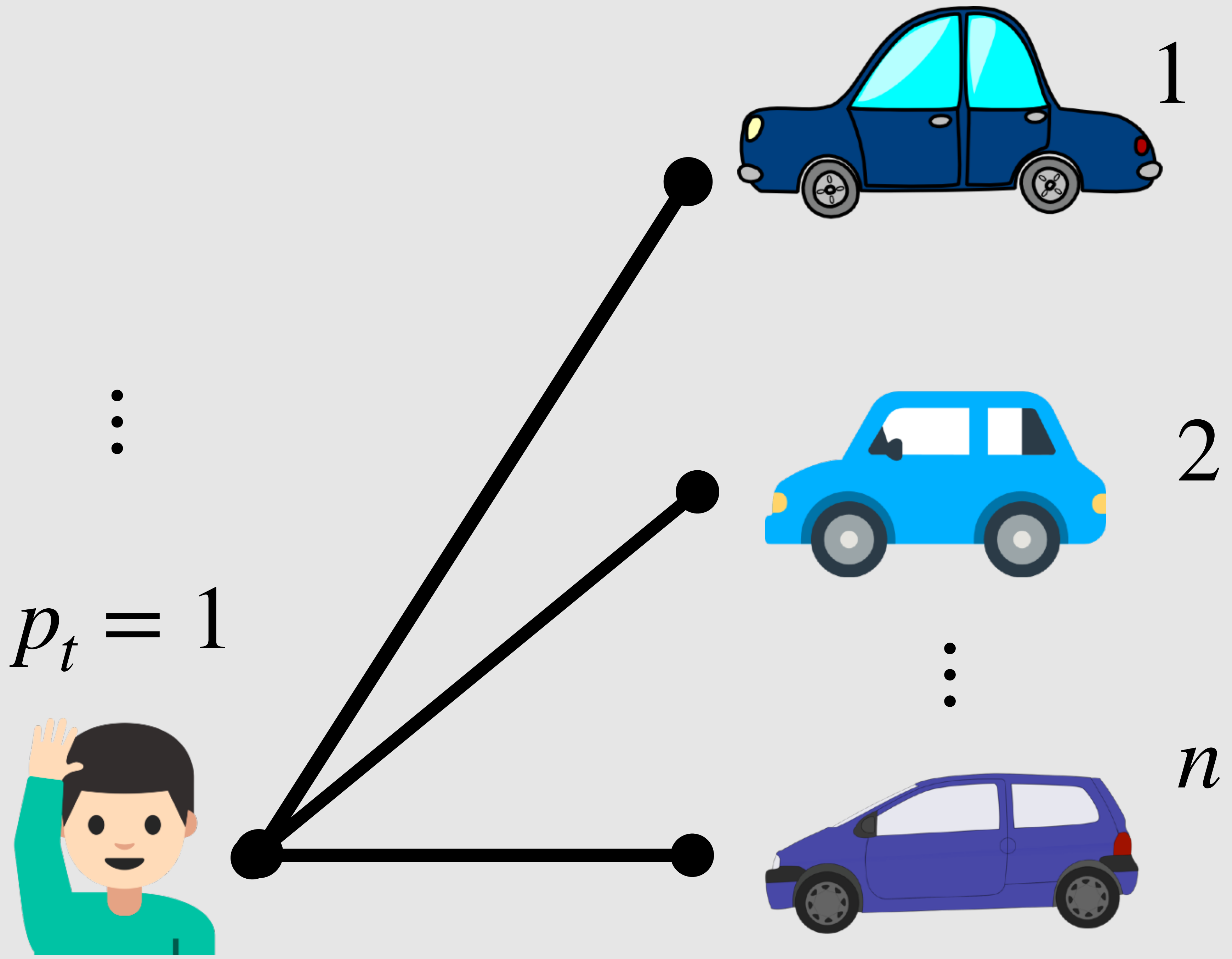


Example. Offline nodes $i = 1, 2, \ldots, n$

- $\Pr[F_{i,t}] = 1/n$
- i's proposal probability $= 1$

$\{F_{i,t}\}_i$ perfectly negatively correlated	
$\{F_{i,t}\}_i$ independent	
$\{F_{i,t}\}_i$ perfectly positively correlated	

Importance of Correlation of $\{F_{i,t}\}_i$

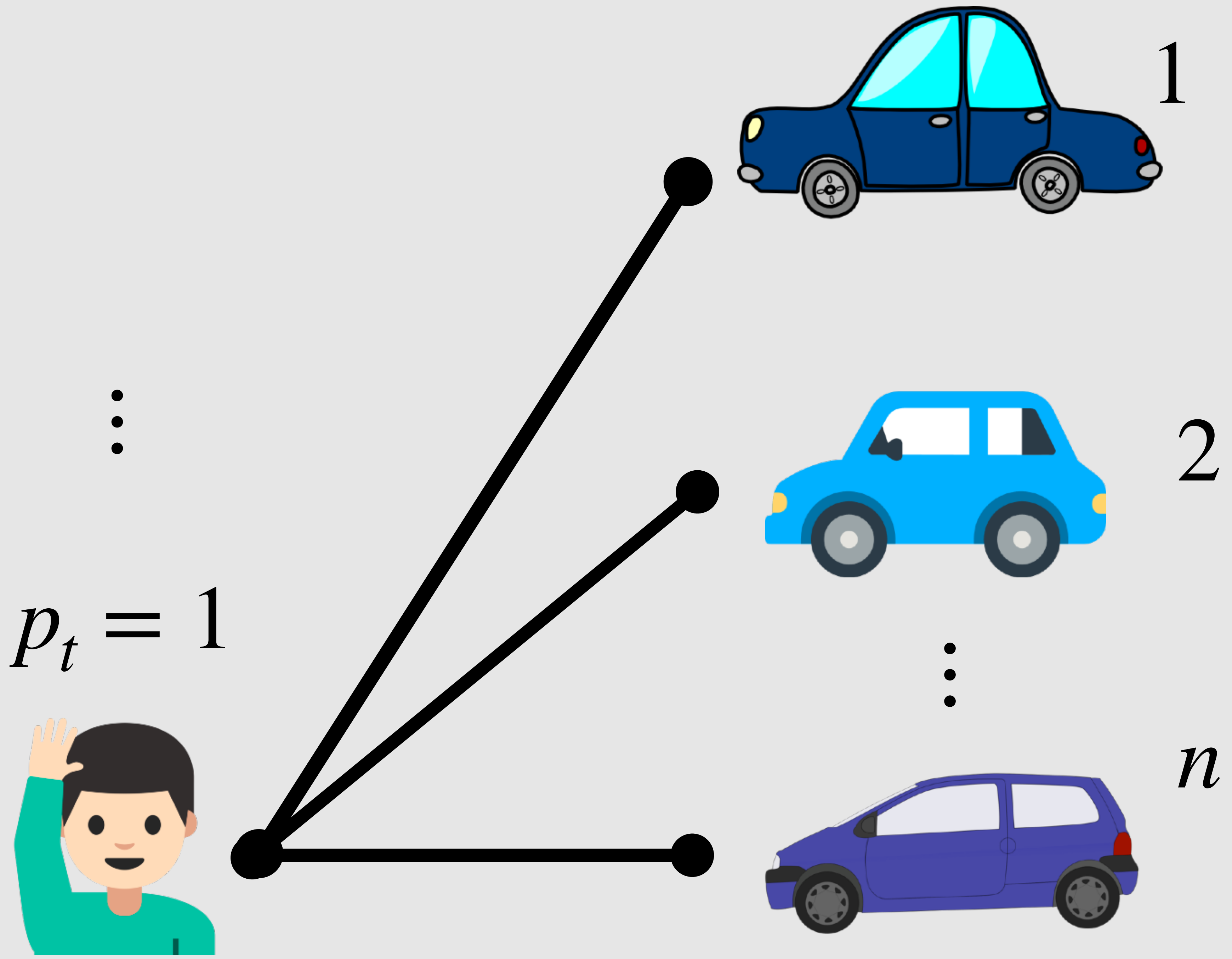


Example. Offline nodes $i = 1, 2, \ldots, n$

- $\Pr[F_{i,t}] = 1/n$
- i's proposal probability $= 1$

$\{F_{i,t}\}_i$ perfectly negatively correlated	1-approximate on t
$\{F_{i,t}\}_i$ independent	
$\{F_{i,t}\}_i$ perfectly positively correlated	

Importance of Correlation of $\{F_{i,t}\}_i$

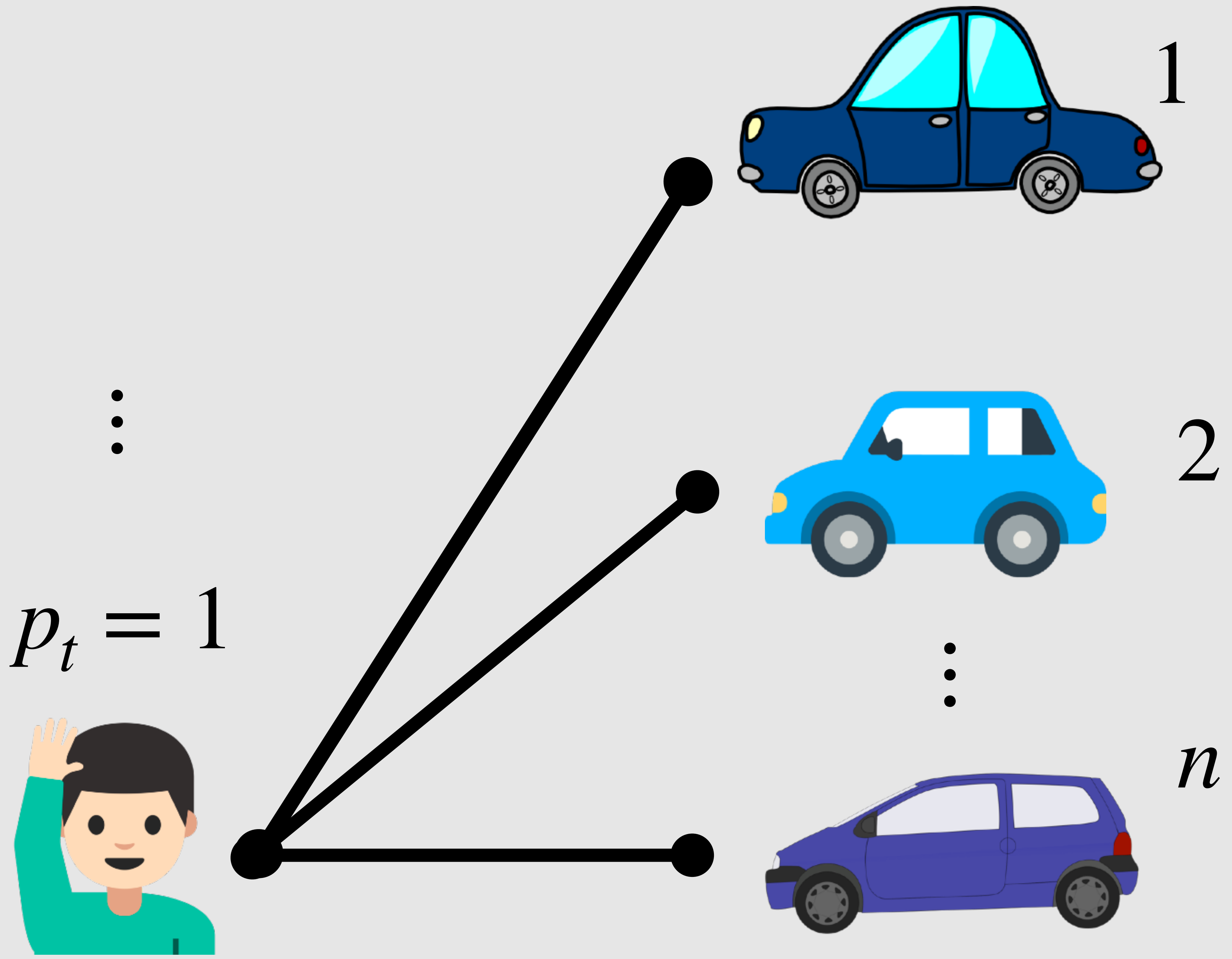


Example. Offline nodes $i = 1, 2, \ldots, n$

- $\Pr[F_{i,t}] = 1/n$
- i's proposal probability $= 1$

$\{F_{i,t}\}_i$ **perfectly negatively correlated**	1-approximate on t
$\{F_{i,t}\}_i$ **independent**	$(1 - 1/e)$-approximate on t
$\{F_{i,t}\}_i$ **perfectly positively correlated**	

Importance of Correlation of $\{F_{i,t}\}_i$



Example. Offline nodes $i = 1, 2, \ldots, n$

- $\Pr[F_{i,t}] = 1/n$
- i's proposal probability $= 1$

$\{F_{i,t}\}_i$ **perfectly negatively correlated**	1-approximate on t
$\{F_{i,t}\}_i$ **independent**	$(1 - 1/e)$-approximate on t
$\{F_{i,t}\}_i$ **perfectly positively correlated**	0-approximate on t

Proof of Approximation Ratio

$$\Pr[t \text{ matched}] = p_t \cdot \Pr[\geq 1 \text{ proposal}]$$

Proof of Approximation Ratio

$$\Pr[t \text{ matched}] = p_t \cdot \Pr[\geq 1 \text{ proposal}]$$

$$= p_t \cdot \Pr\left[\text{Pivotal-Sampling}\left(\left(\frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)_i\right) \text{ intersects } [n]\right]$$

Proof of Approximation Ratio

$$\Pr[t \text{ matched}] = p_t \cdot \Pr[\geq 1 \text{ proposal}]$$

$$= p_t \cdot \Pr\left[\text{Pivotal-Sampling}\left(\left(\frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)_i\right) \text{ intersects } [n]\right]$$

$$= p_t \cdot \mathbb{E}\left[\min\left(1, \sum_i \frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)\right]$$

Proof of Approximation Ratio

$$\Pr[t \text{ matched}] = p_t \cdot \Pr[\geq 1 \text{ proposal}]$$

$$= p_t \cdot \Pr\left[\text{Pivotal-Sampling}\left(\left(\frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)_i\right) \text{ intersects } [n]\right]$$

$$= p_t \cdot \mathbb{E}\left[\min\left(1, \sum_i \frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)\right]$$

$$= p_t \cdot \mathbb{E}\left[\min\left(1, R_t\right)\right]$$

Proof of Approximation Ratio

$$\Pr[t \text{ matched}] = p_t \cdot \Pr[\geq 1 \text{ proposal}]$$

$$= p_t \cdot \Pr\left[\text{Pivotal-Sampling}\left(\left(\frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)_i\right) \text{ intersects } [n]\right]$$

$$= p_t \cdot \mathbb{E}\left[\min\left(1, \sum_i \frac{x_{i,t}}{p_t\left(1 - \sum_{t'<t} x_{i,t'}\right)} \cdot F_{i,t}\right)\right]$$

$$= p_t \cdot \mathbb{E}\left[\min\left(1, R_t\right)\right]$$

← R_t = sum of [0,1]-weighted NCD Bernoullis

Proof of Approximation Ratio

$$\text{APX-Ratio} \geq \frac{\sum_t \Pr[t \text{ matched}]}{\sum_{i,t} x_{i,t}}$$

$$= \frac{\sum_t p_t \cdot \mathbb{E}[\min(1, R_t)]}{\sum_{i,t} x_{i,t}}$$

Proof of Approximation Ratio

$$\text{APX-Ratio} \geq \frac{\sum_t \Pr[t \text{ matched}]}{\sum_{i,t} x_{i,t}}$$

$$= \frac{\sum_t p_t \cdot \mathbb{E}[\min(1, R_t)]}{\sum_{i,t} x_{i,t}}$$

Analysis. New lower bounds on $\mathbb{E}[\min(1,R_t)]$, when R_t is the sum of weighted **NCD** Bernoullis.

Proof of Approximation Ratio

$$\text{APX-Ratio} \geq \frac{\sum_t \Pr[t \text{ matched}]}{\sum_{i,t} x_{i,t}}$$

$$= \frac{\sum_t p_t \cdot \mathbb{E}[\min(1, R_t)]}{\sum_{i,t} x_{i,t}}$$

Analysis. New lower bounds on $\mathbb{E}[\min(1, R_t)]$, when R_t is the sum of weighted **NCD** Bernoullis.

Comment. Same algorithmic template + lower bounds apply to the edge-weighted case, with an additional pre-processing step where we ***rescale*** the LP solution.

Extensions

More General Arrival Model. Extension to when online vertices have a *general distribution* over their neighborhood.

Vertex-Weighted. Can improve analysis to a 0.685-approximation (state-of-the-art).

Conclusion and Future Directions

Pivotal Sampling for Online Matching.

- New best approximation ratios vs OPT_{on} benchmark
- Negative Cylinder Dependence (NCD) of offline nodes
- New tail expectation bounds for sums of NCD Random Variables

Conclusion and Future Directions

Pivotal Sampling for Online Matching.

- New best approximation ratios vs OPT_{on} benchmark
- Negative Cylinder Dependence (NCD) of offline nodes
- New tail expectation bounds for sums of NCD Random Variables

Questions.

1. Better analysis of our algorithms? Closing the gap?

2. Constraints beyond matching?

Conclusion and Future Directions

Pivotal Sampling for Online Matching.

- New best approximation ratios vs OPT_{on} benchmark
- Negative Cylinder Dependence (NCD) of offline nodes
- New tail expectation bounds for sums of NCD Random Variables

Questions.

1. Better analysis of our algorithms? Closing the gap?

2. Constraints beyond matching?

Thank you!