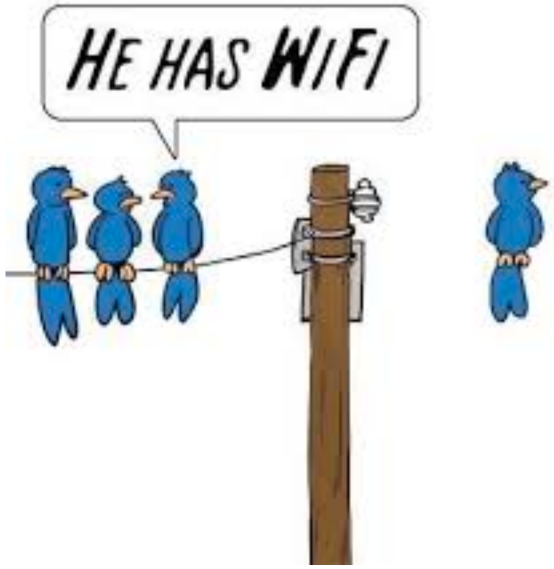
Graph Matchings and Wireless Communication

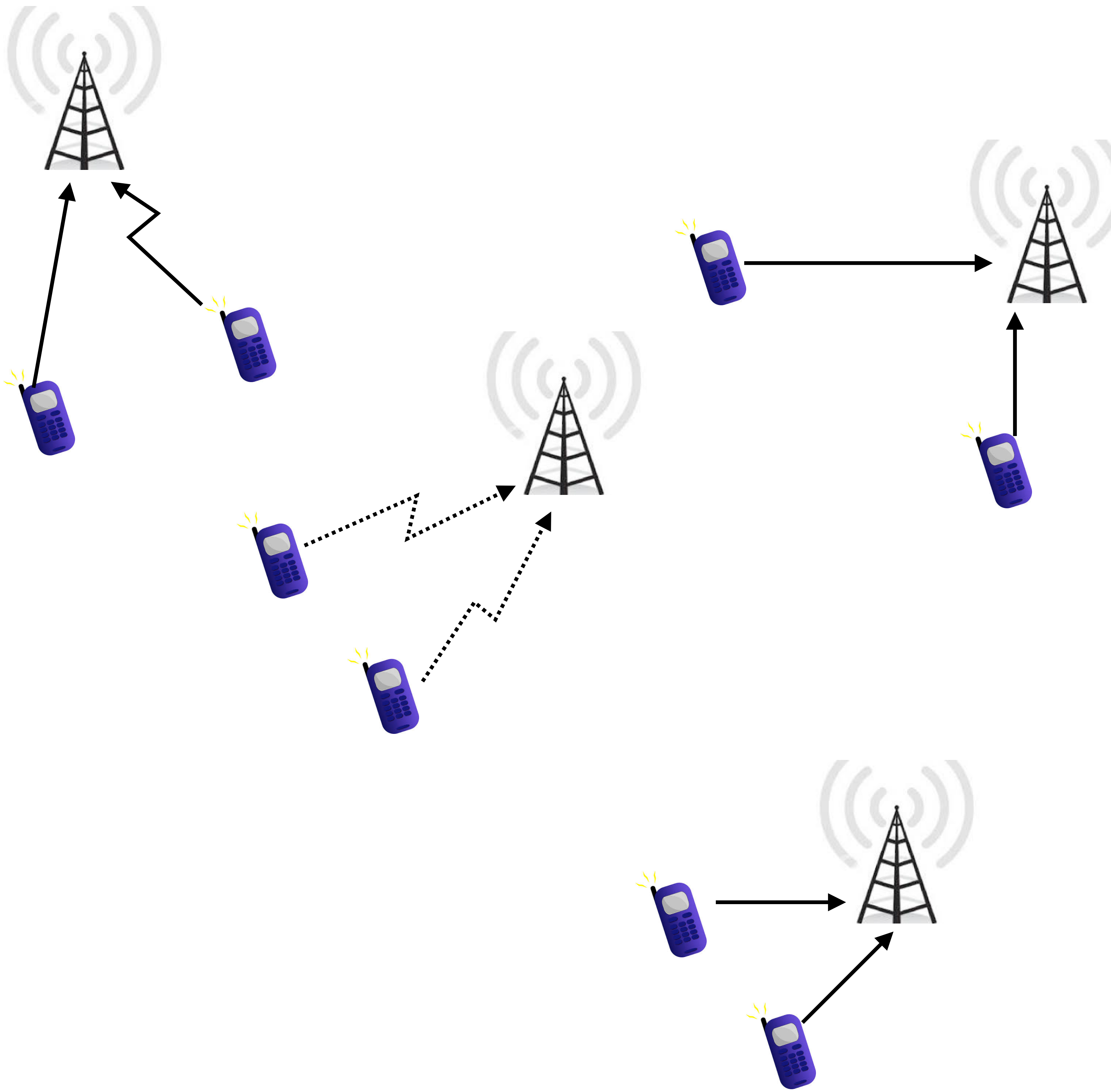
Rahul Vaze

Graph Matchings and Wireless Communication

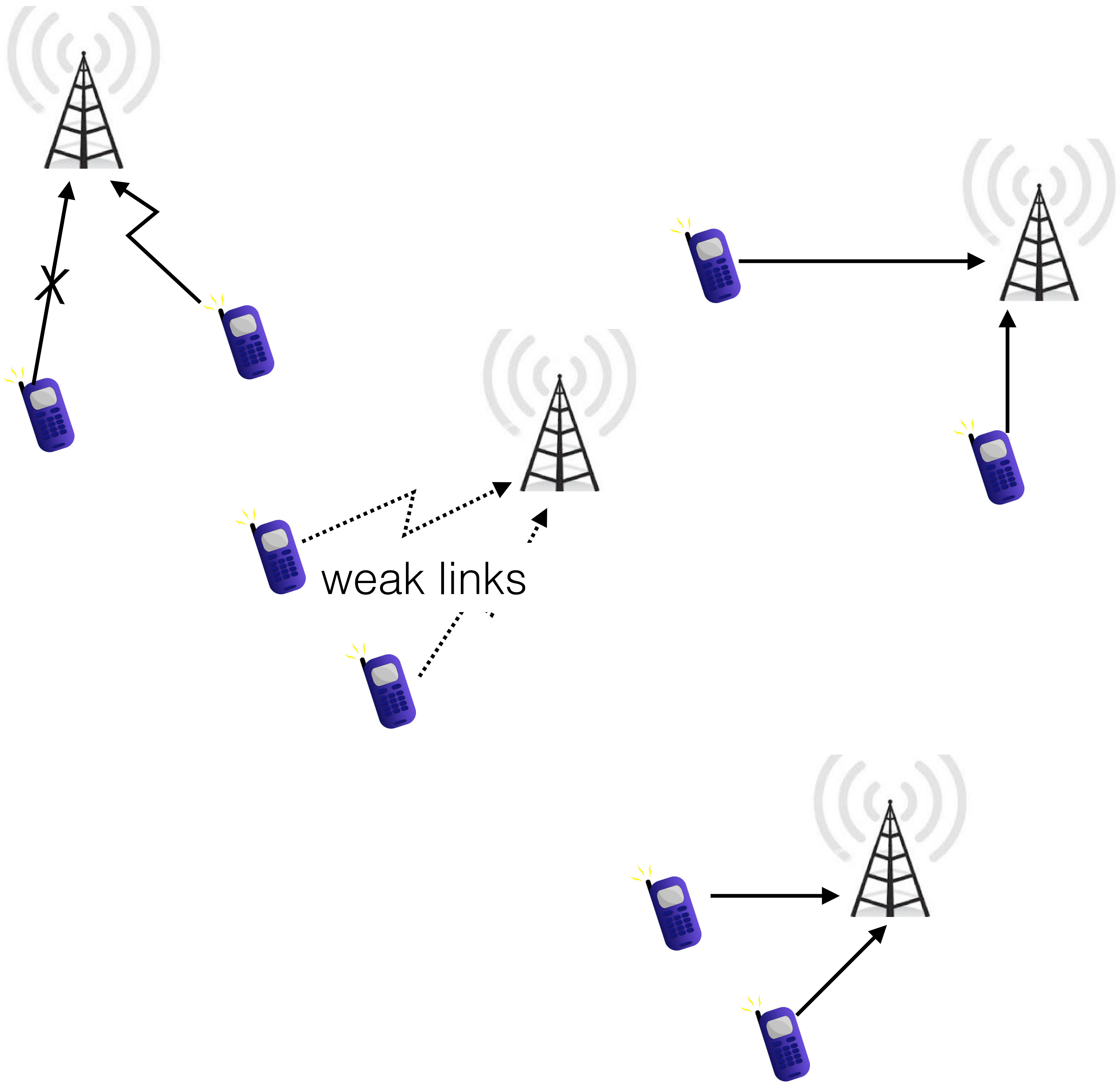


Rahul Vaze

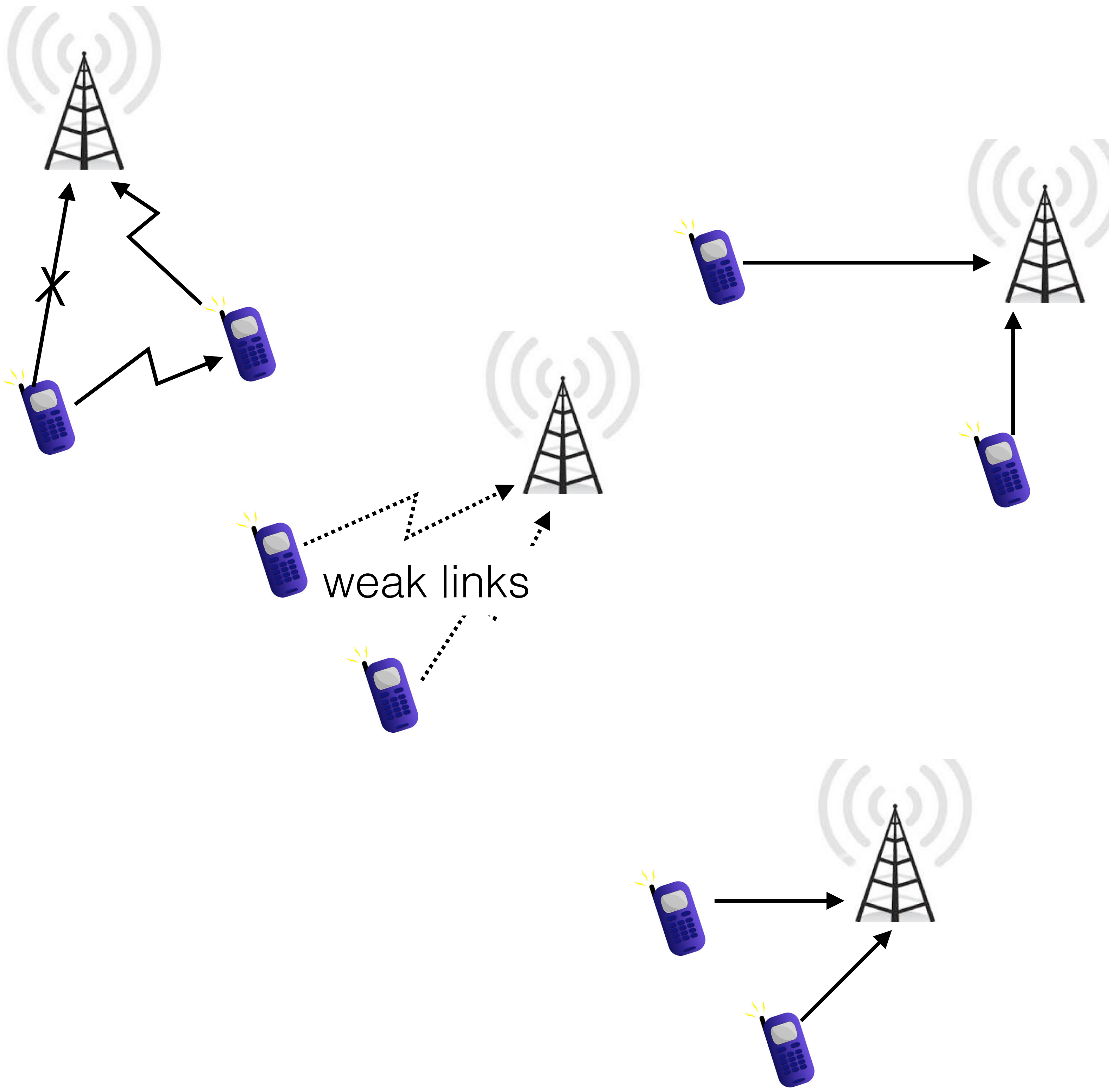
Device-2-Device Communication



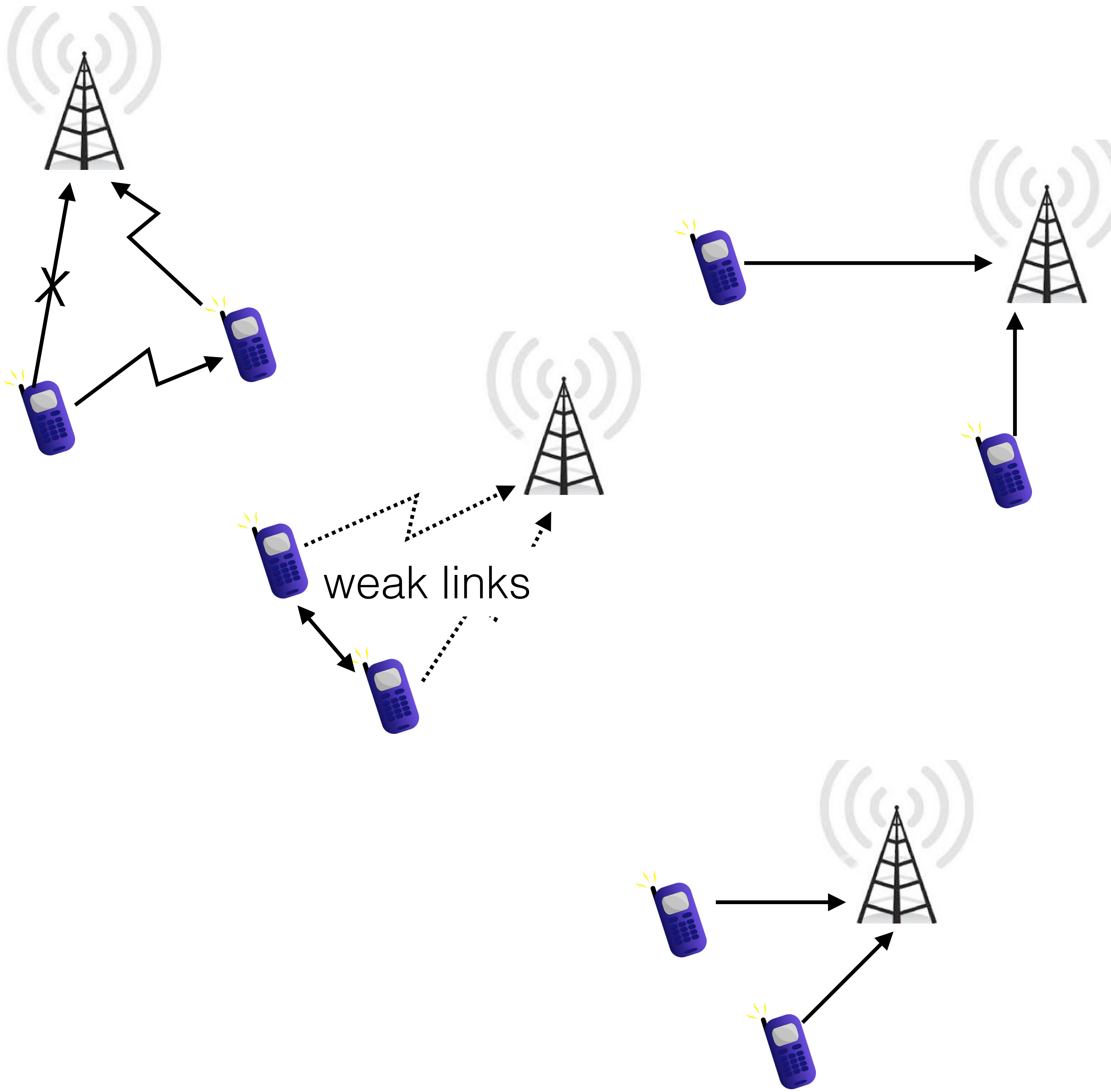
Device-2-Device Communication



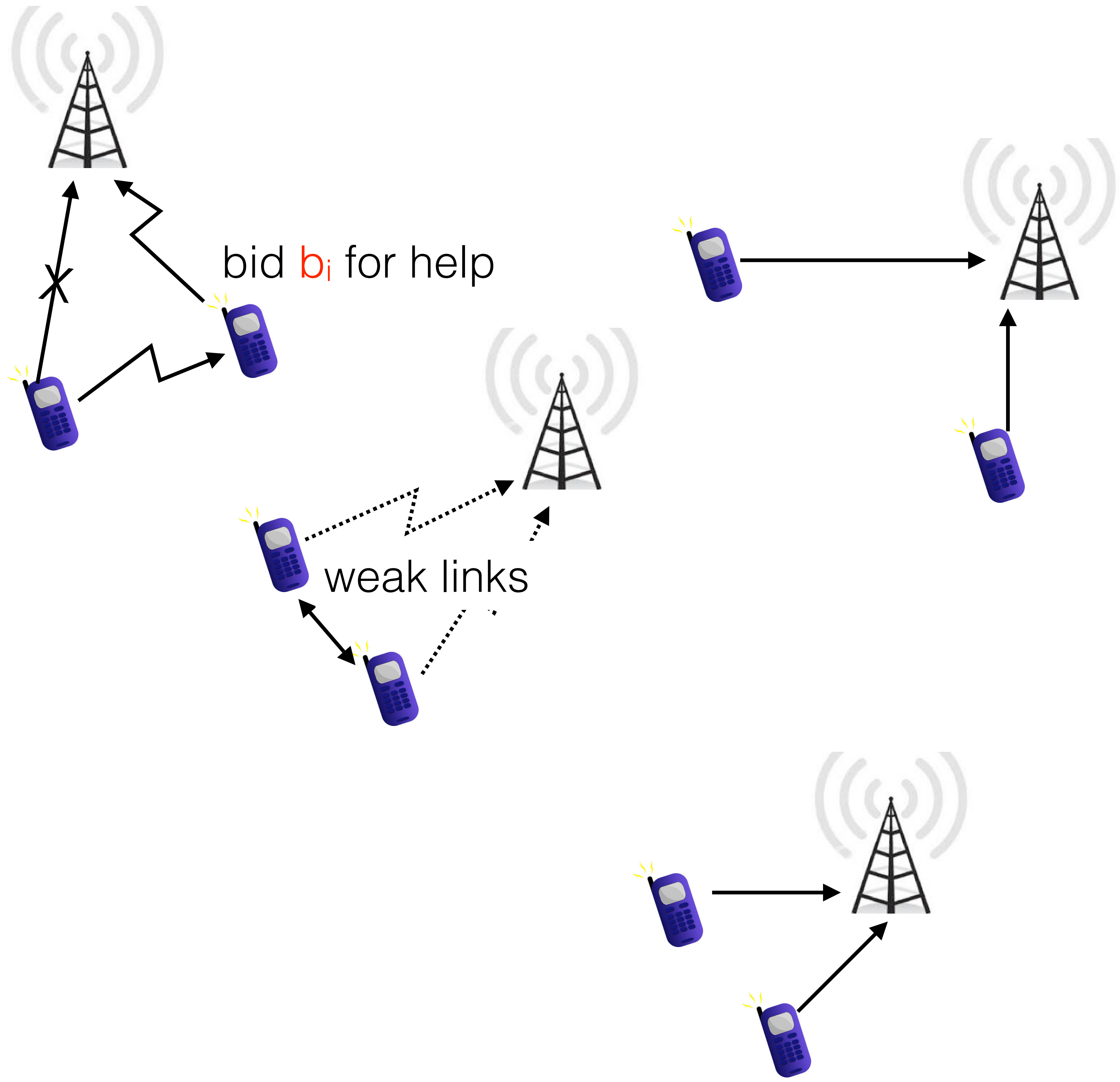
Device-2-Device Communication



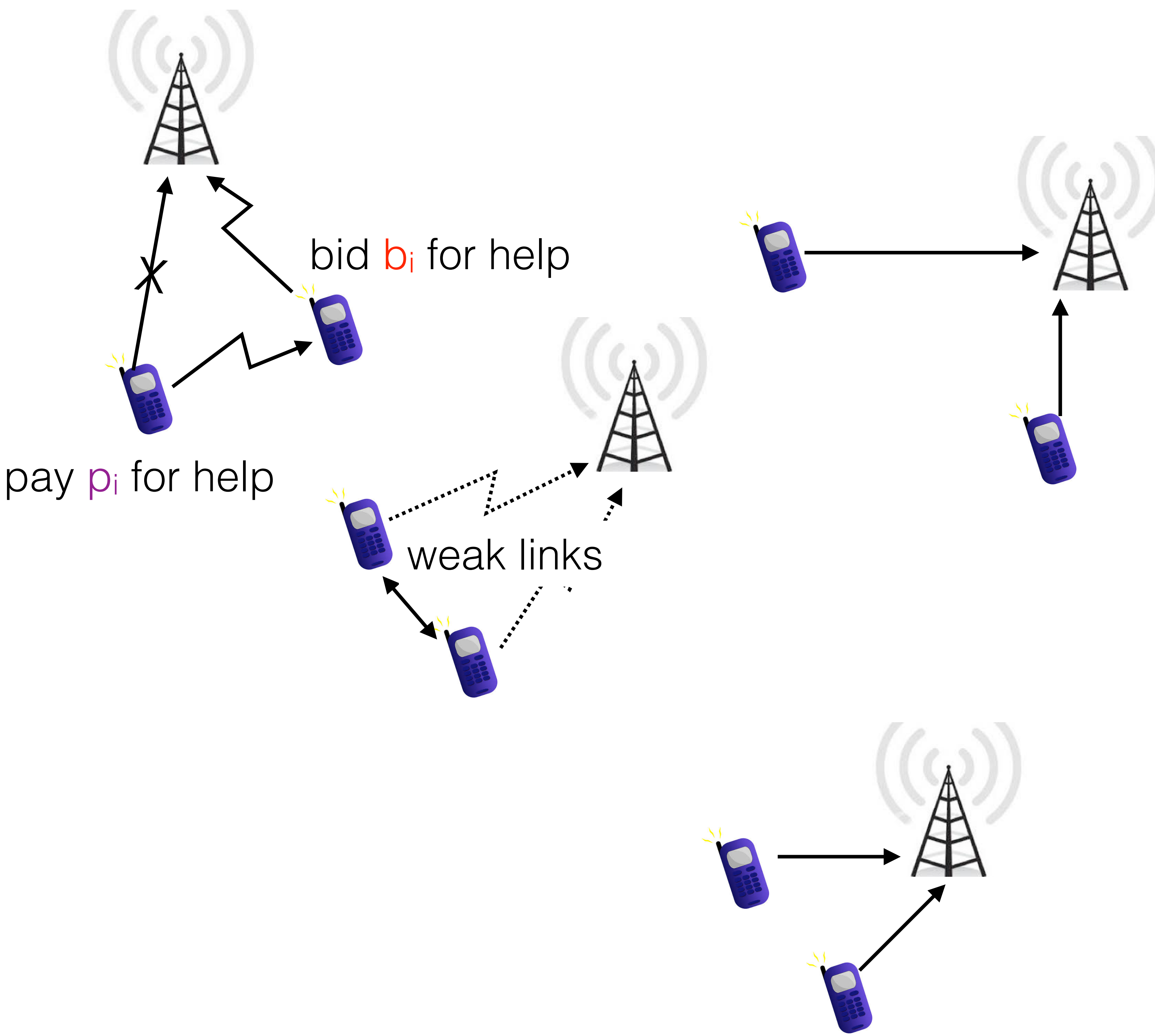
Device-2-Device Communication



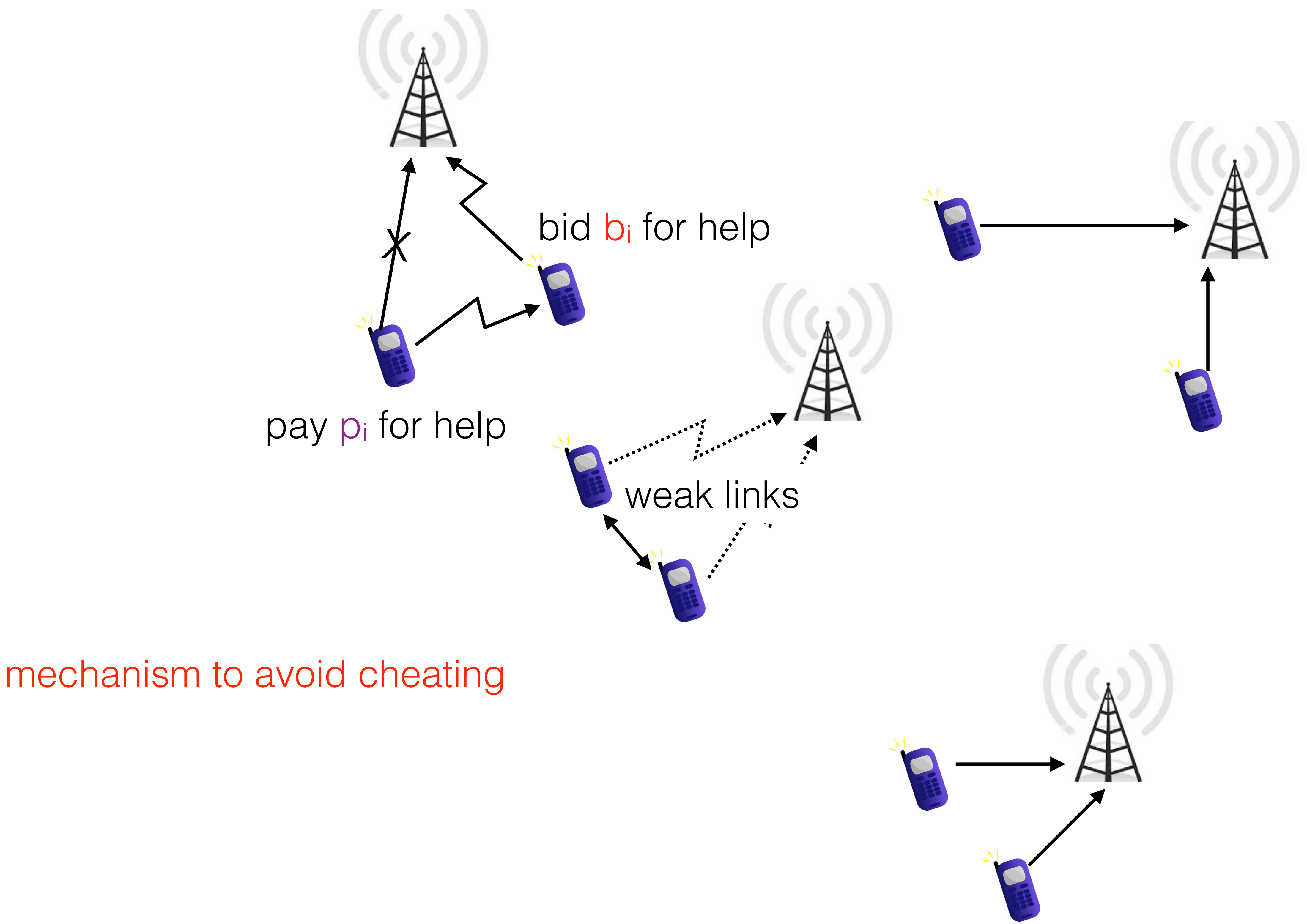
Device-2-Device Communication



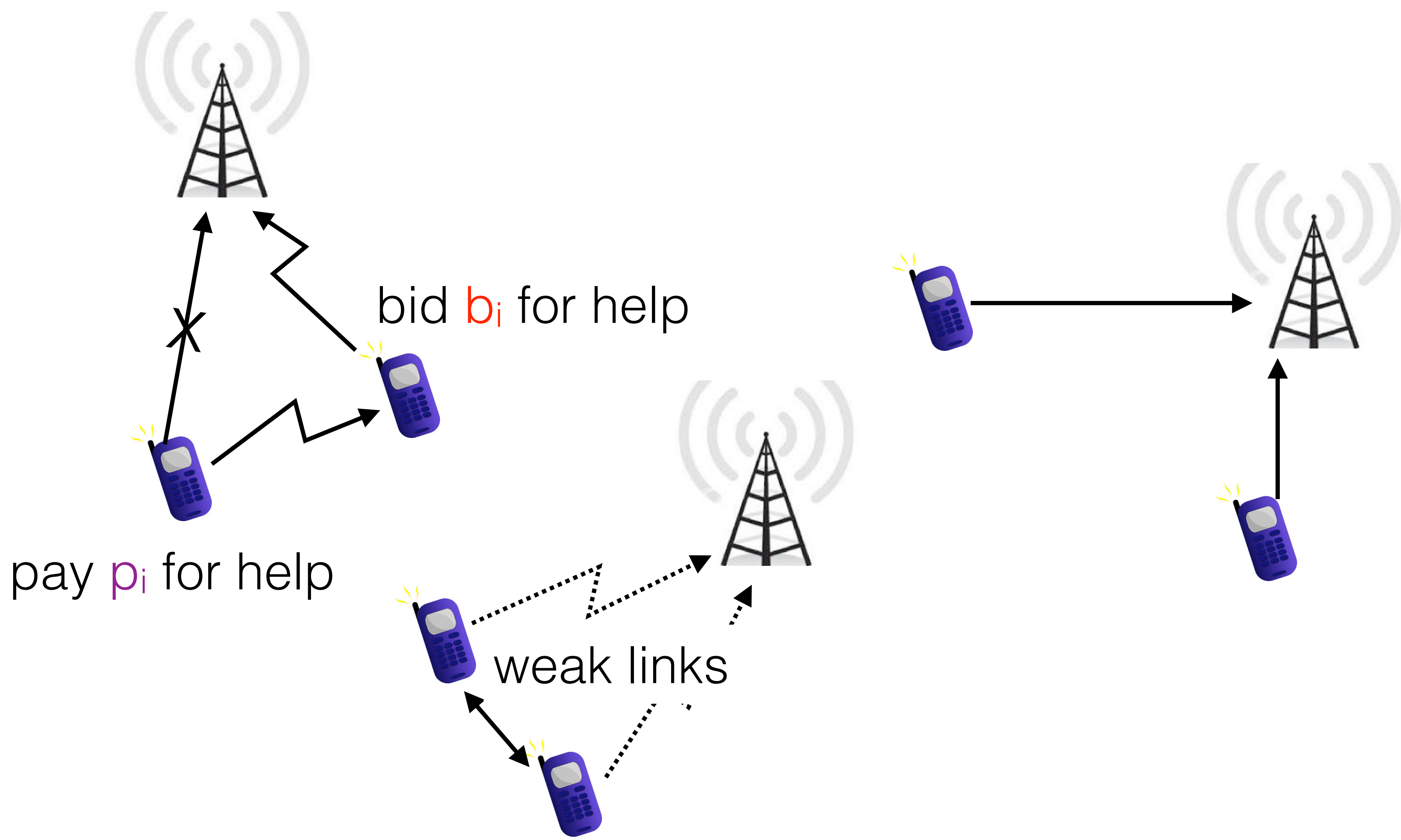
Device-2-Device Communication

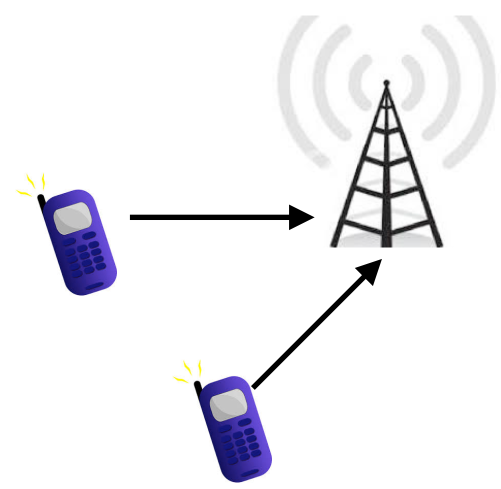


Device-2-Device Communication



Device-2-Device Communication

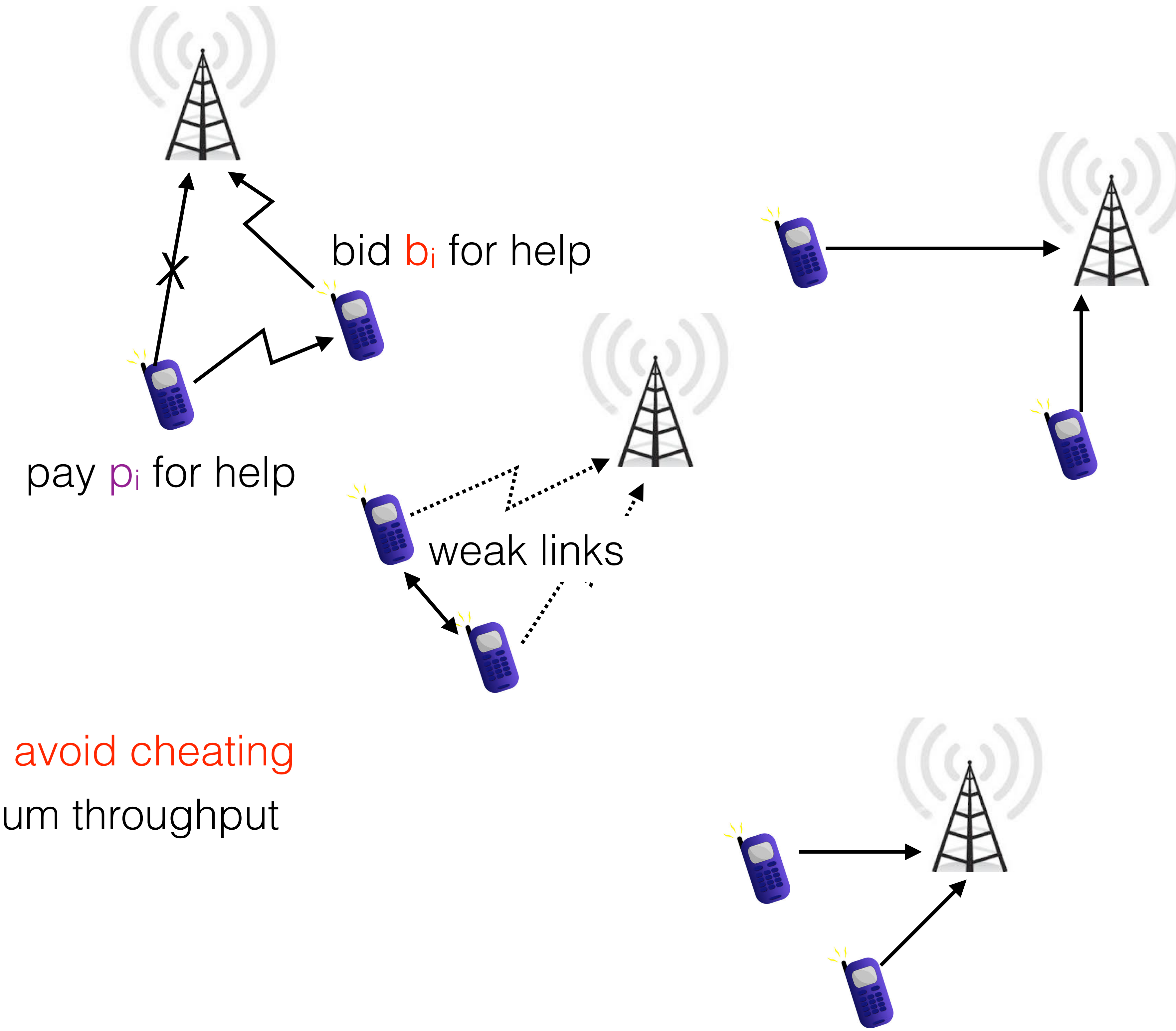




mechanism to avoid cheating

ensure maximum throughput

Device-2-Device Communication



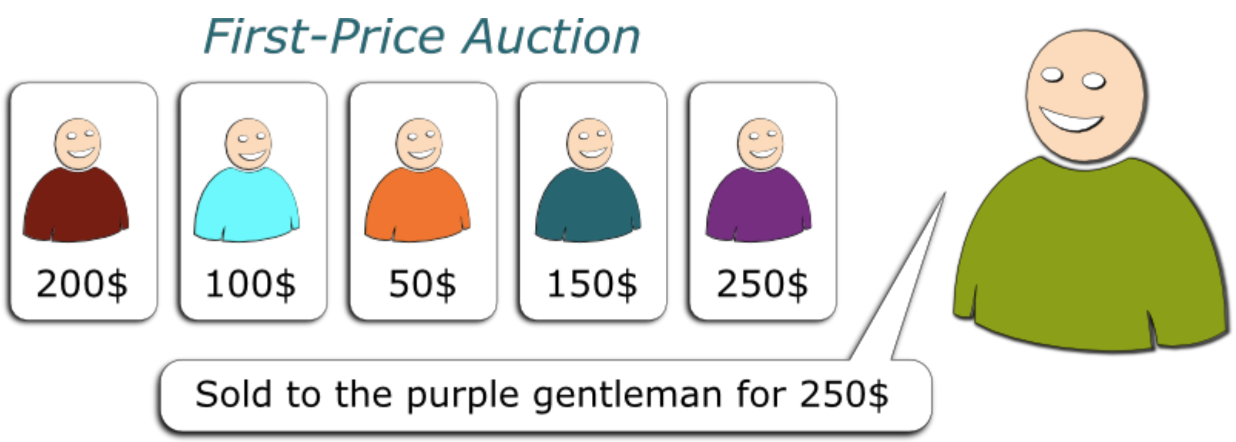
mechanism to avoid cheating

ensure maximum throughput

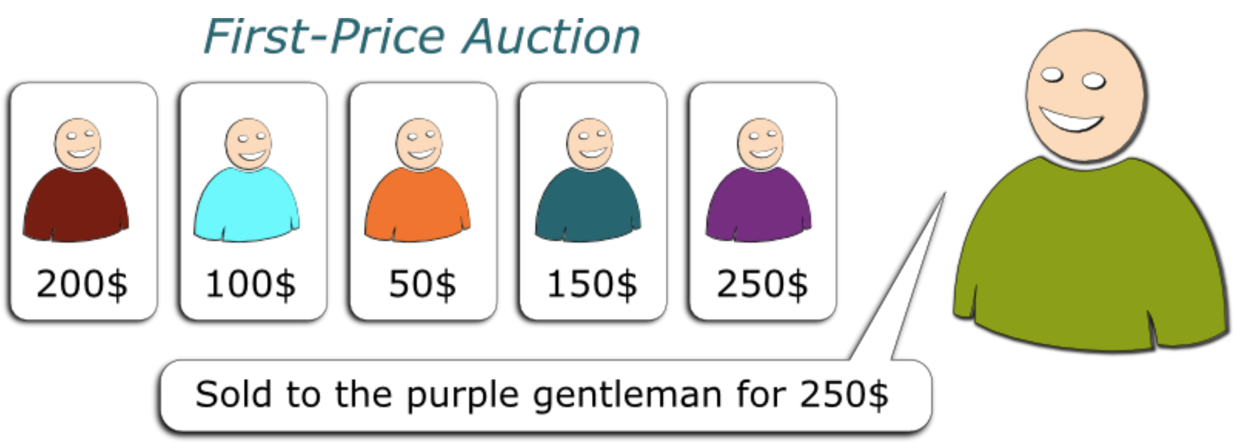
Find ONLINE optimal helper association and incentive rule that is truthful

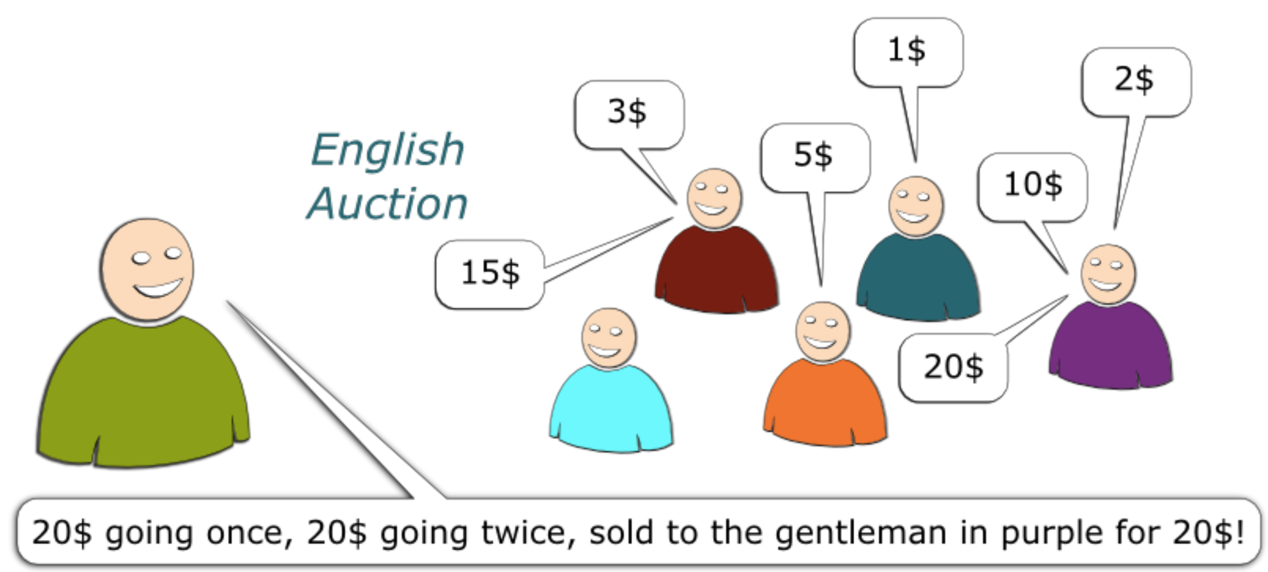
Truthful Auction

Truthful Auction



Truthful Auction



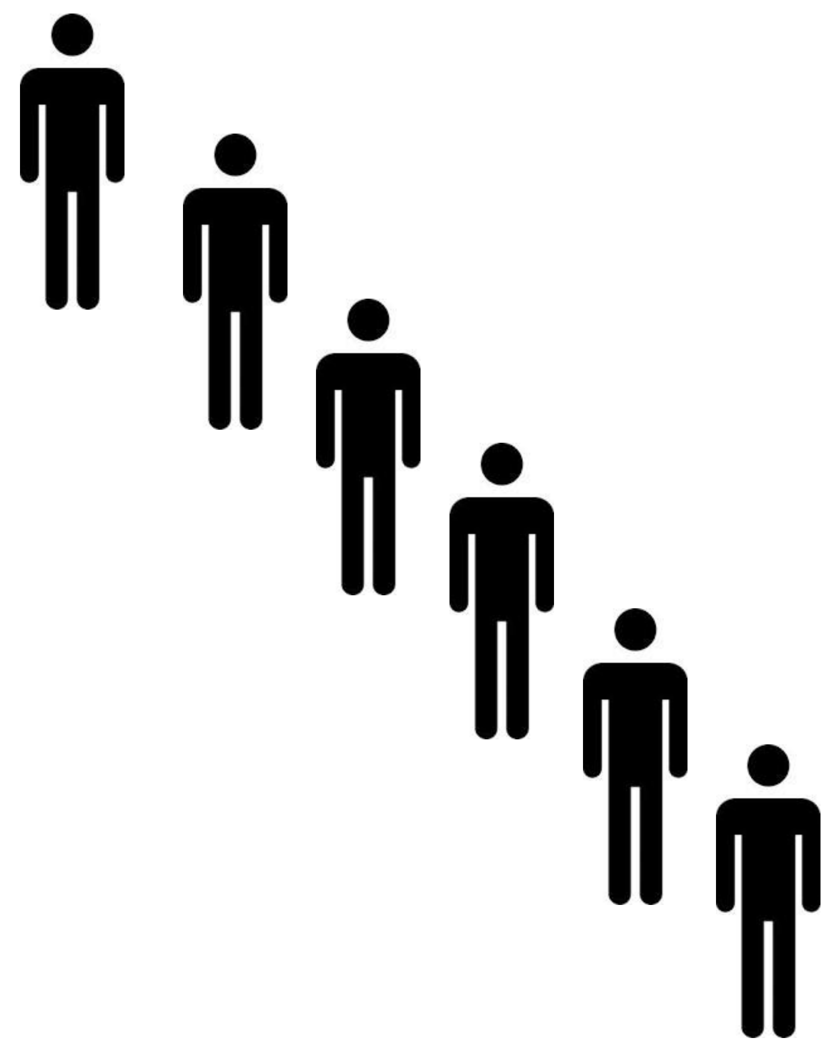


small detour

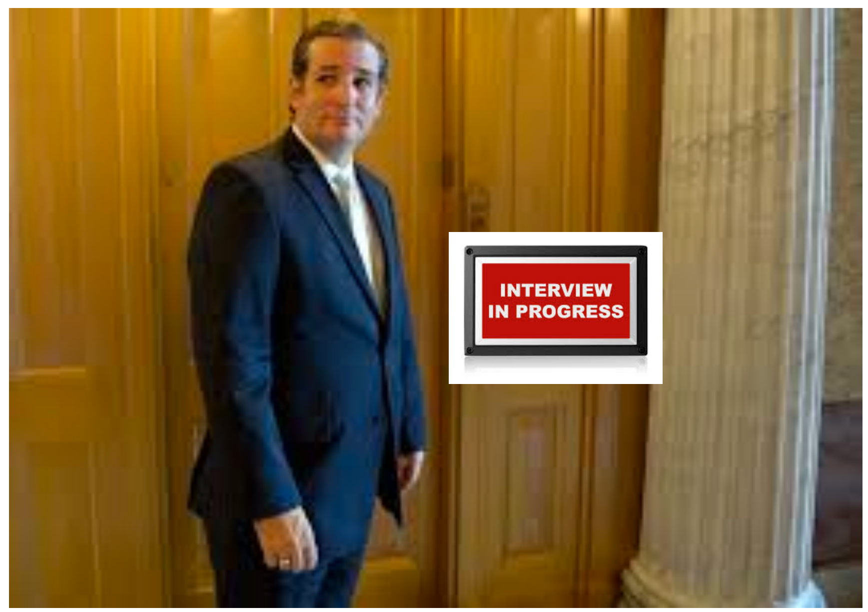

The mathematics behind a perfect match

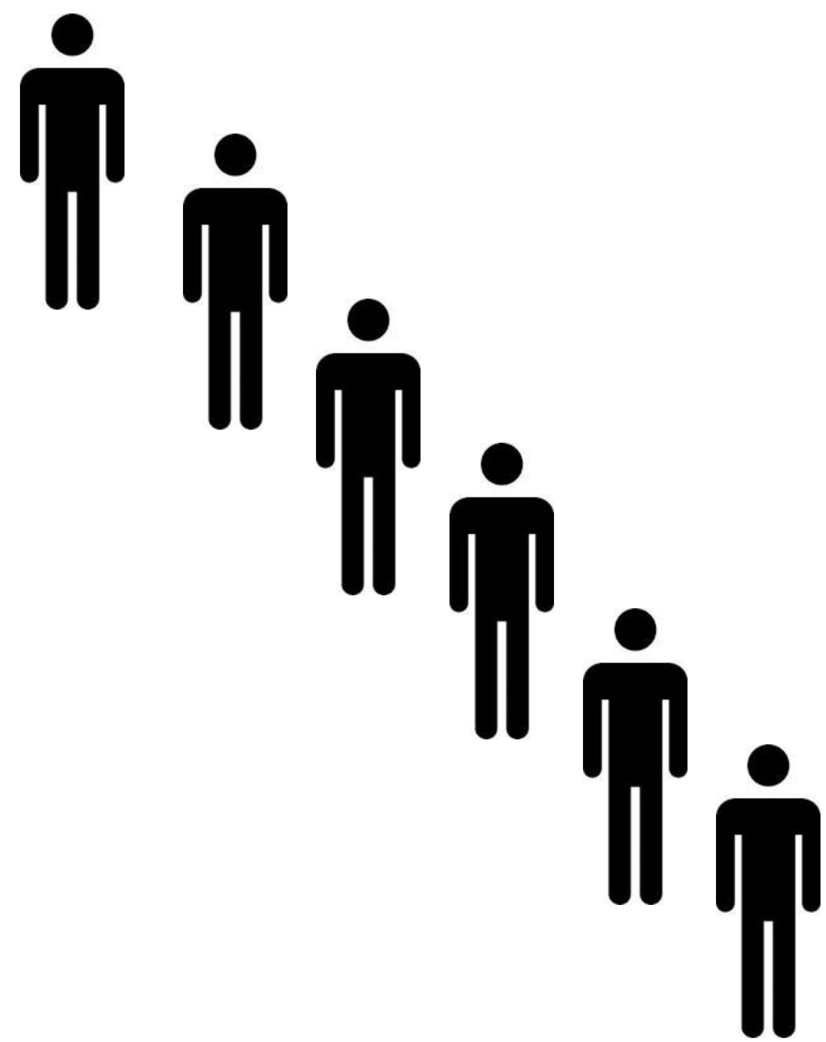
how many to date before committing !

Hiring impatient staff



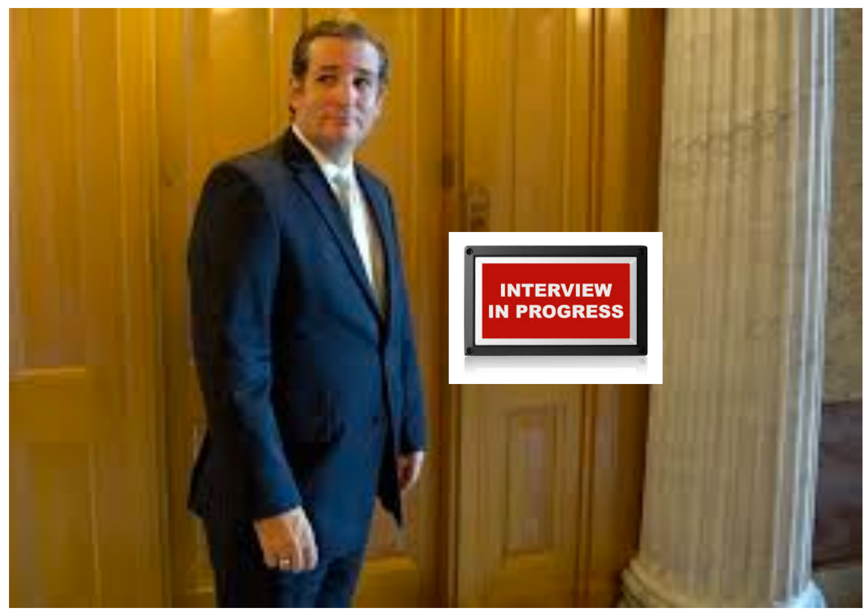
Hiring impatient staff

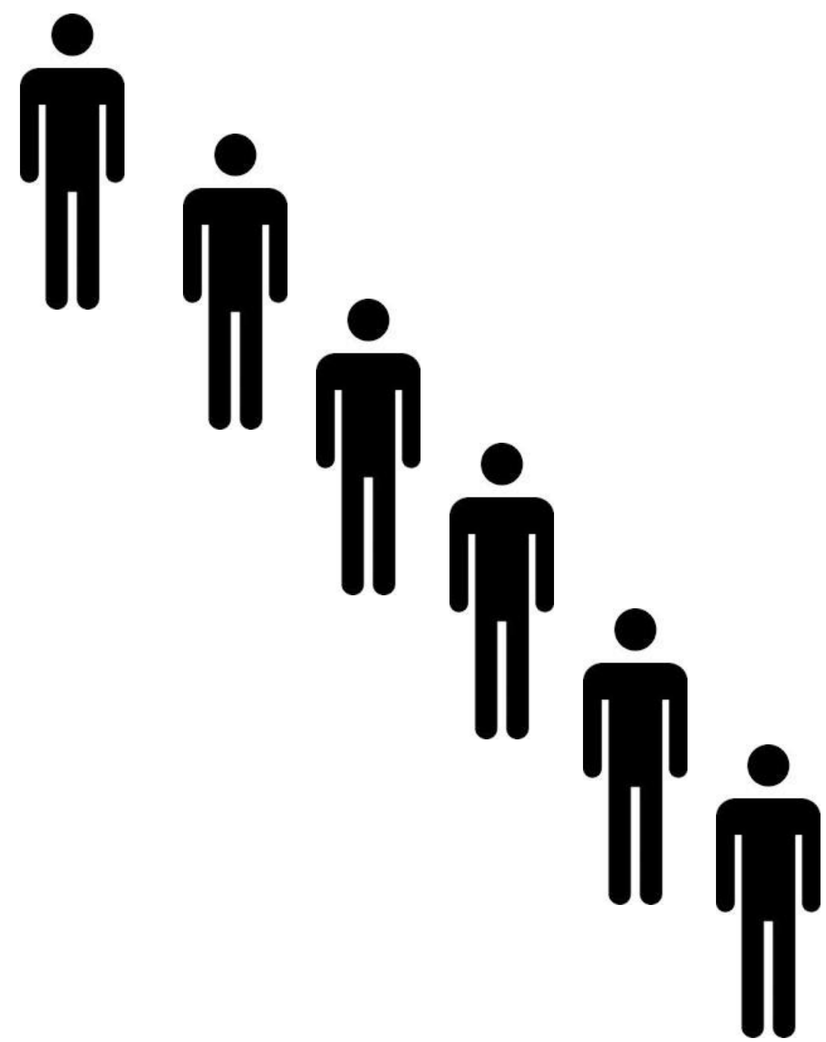




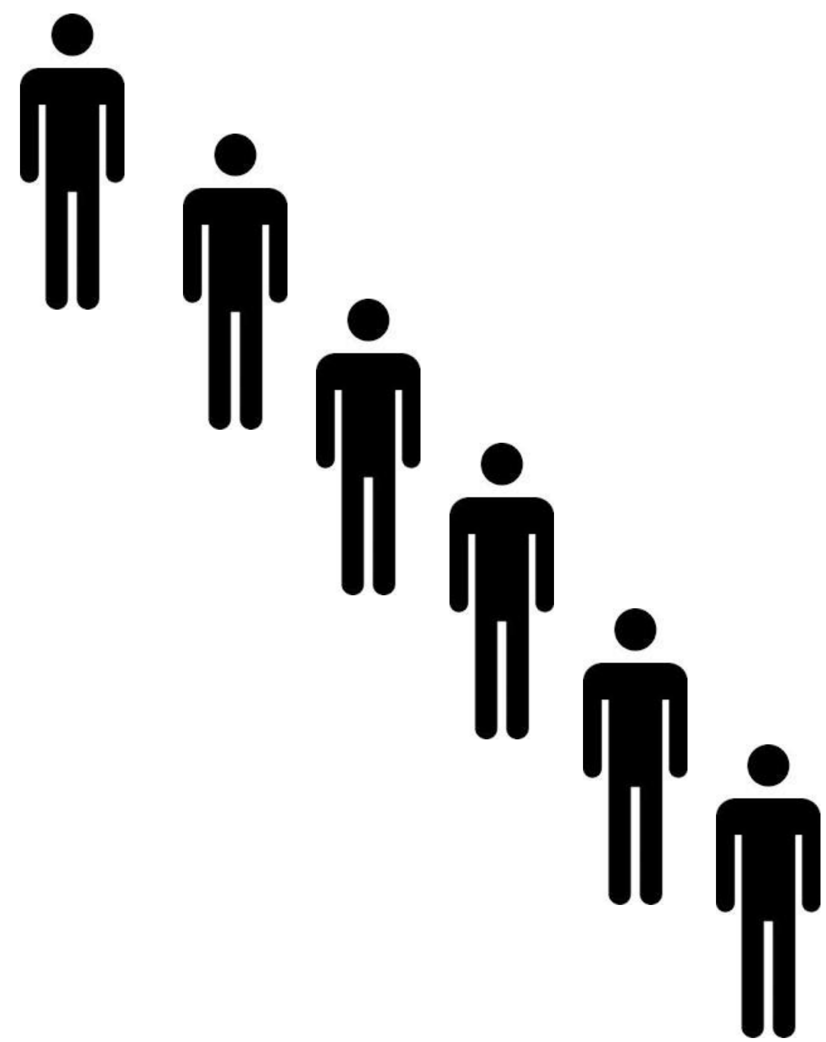
Hiring impatient staff

prob. of choosing best candidate is 1/n

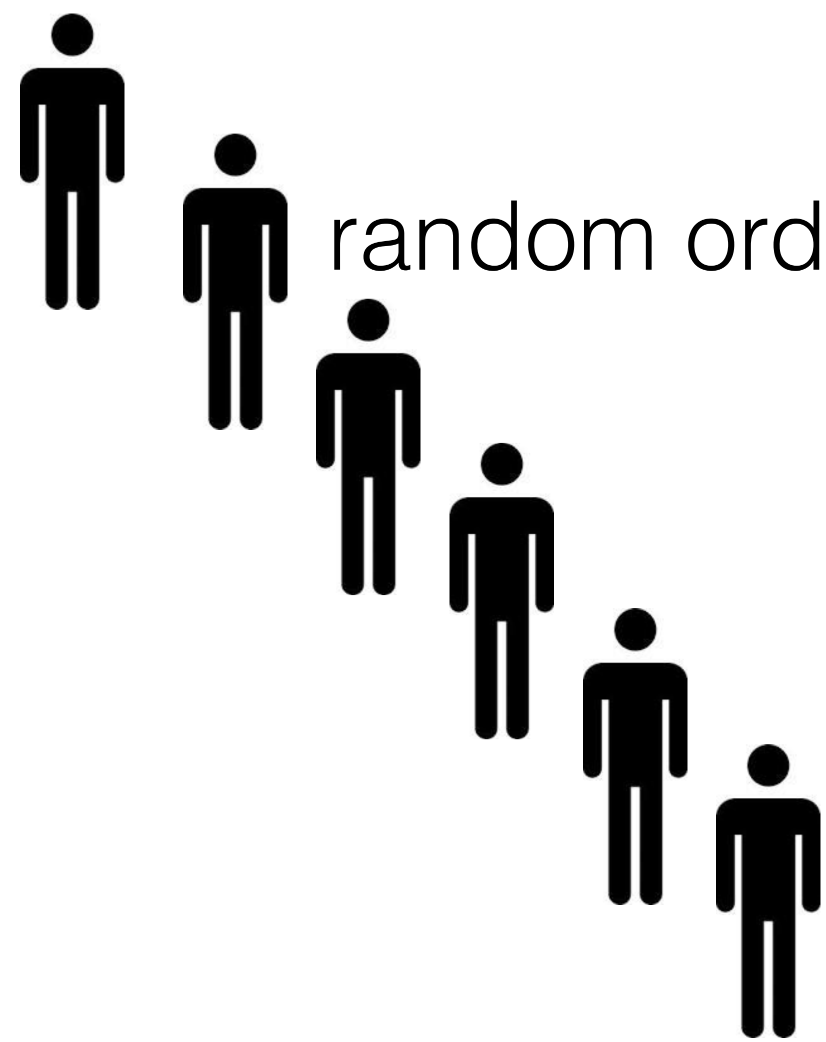




Hiring staff - not adversarial

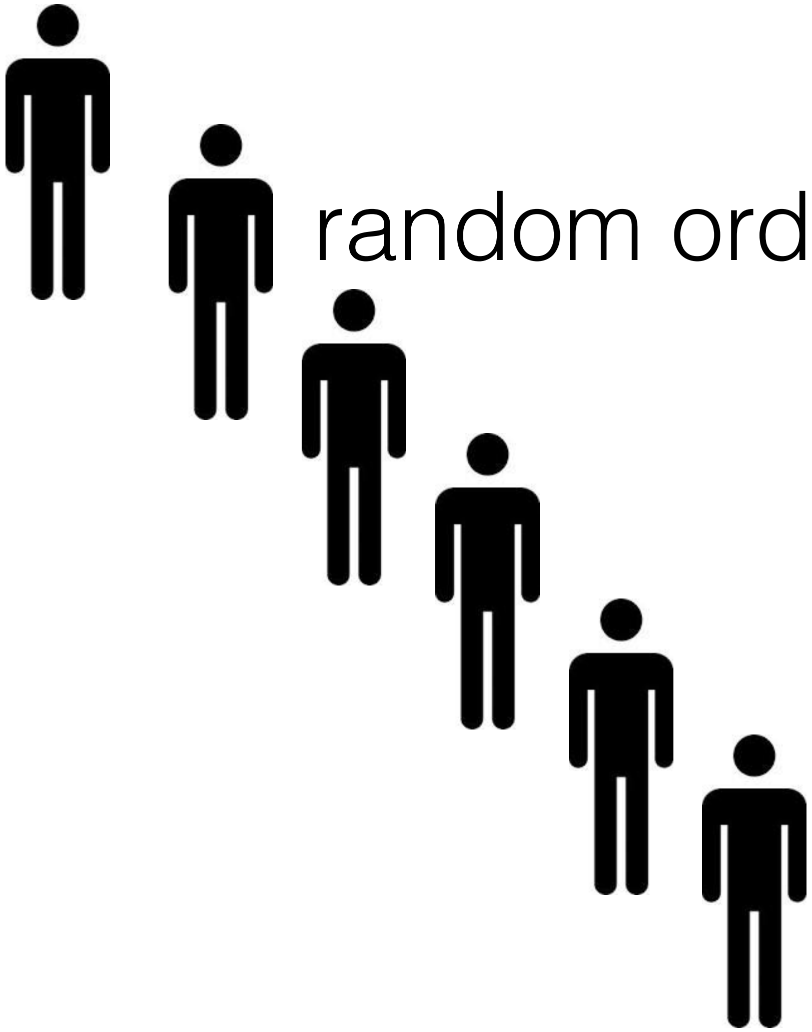


Hiring staff - not adversarial



random order

Hiring staff - not adversarial

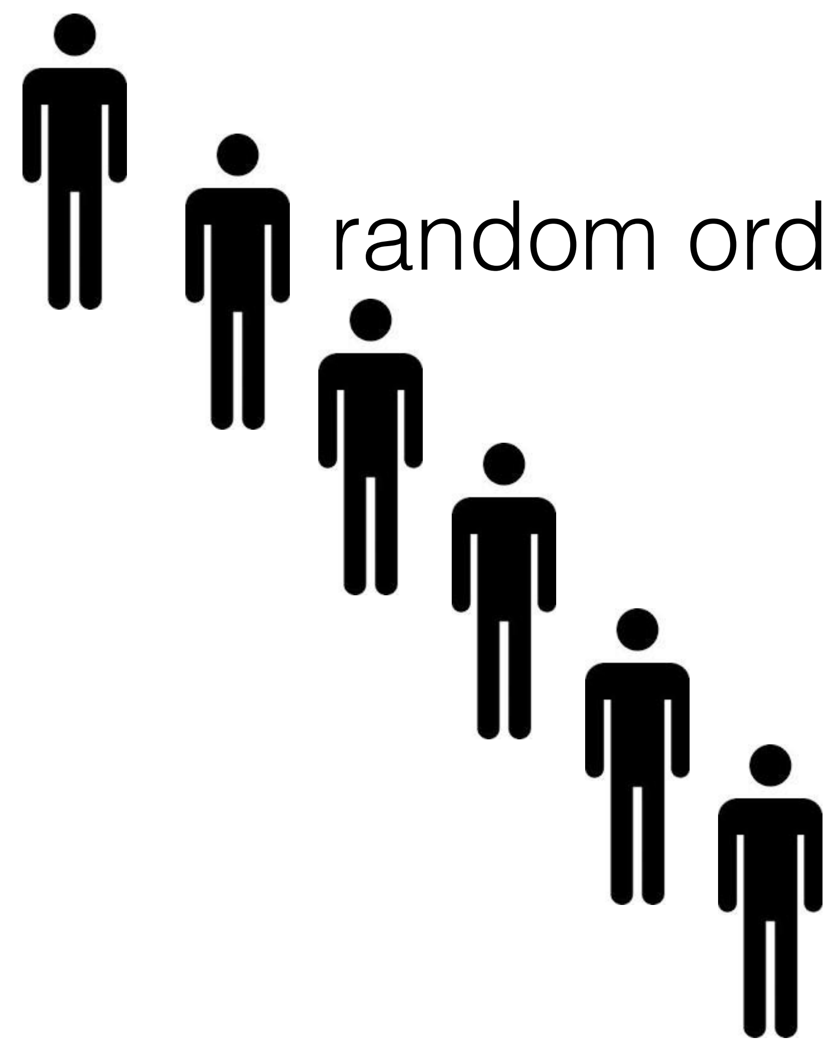


random order

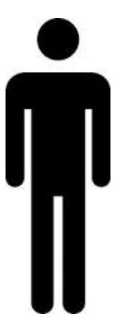
sampling phase

first half

Hiring staff - not adversarial



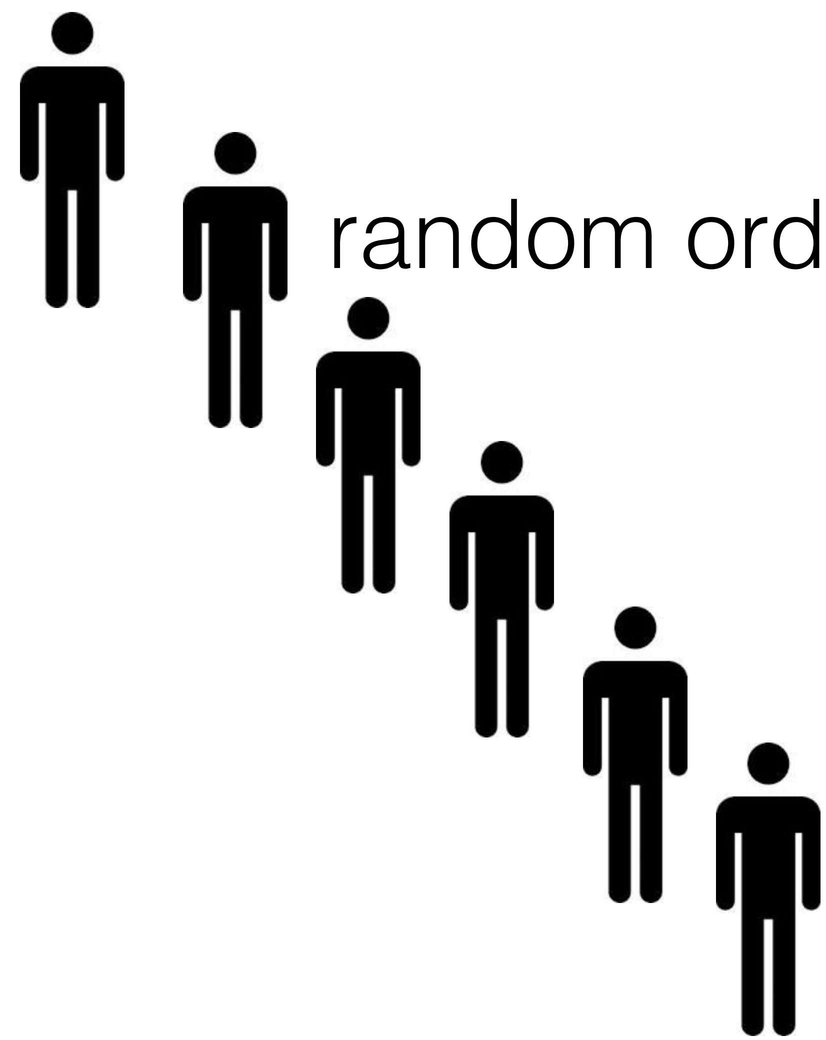
random order



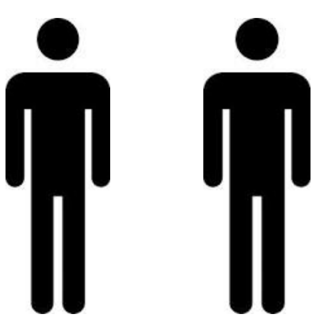
sampling phase

first half

Hiring staff - not adversarial



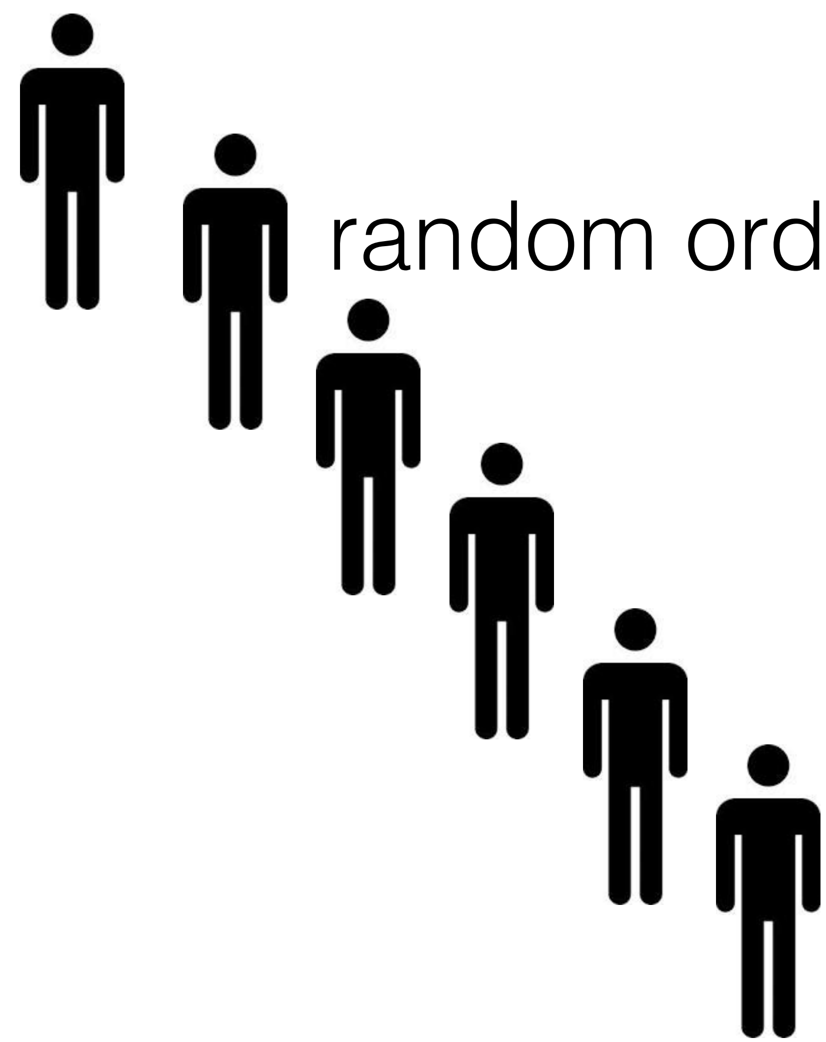
random order



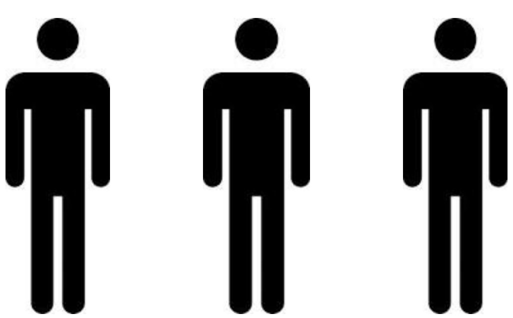
sampling phase

first half

Hiring staff - not adversarial



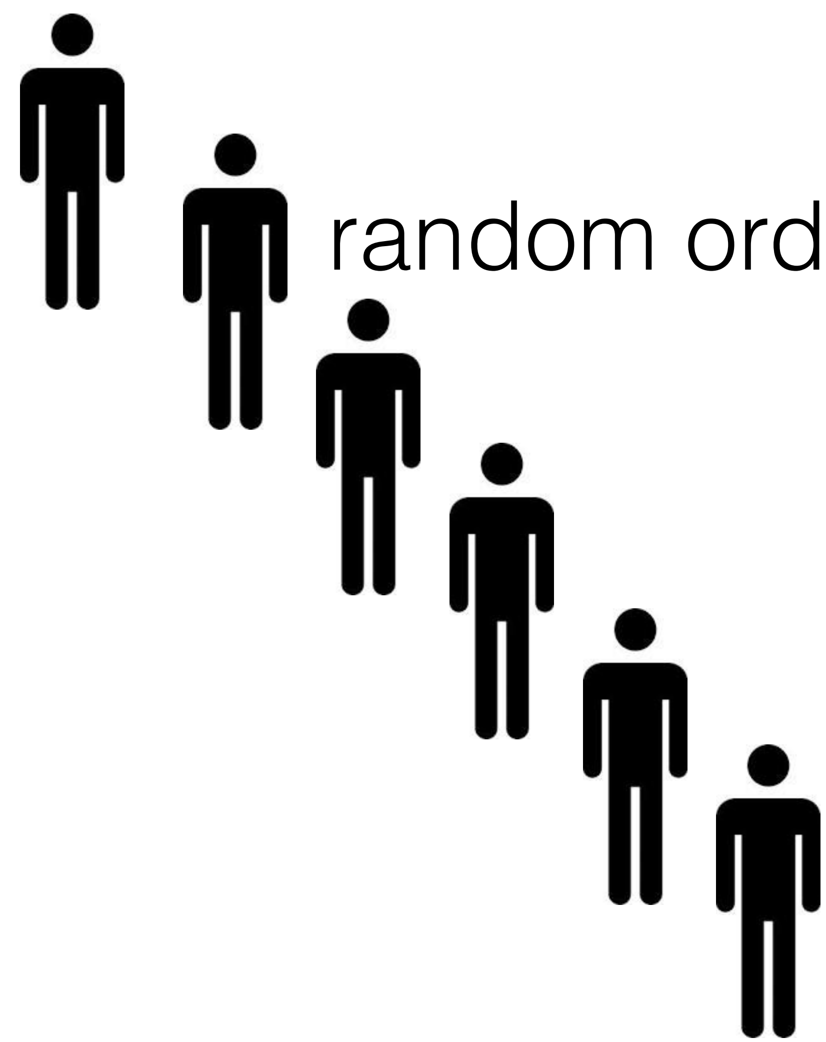
random order



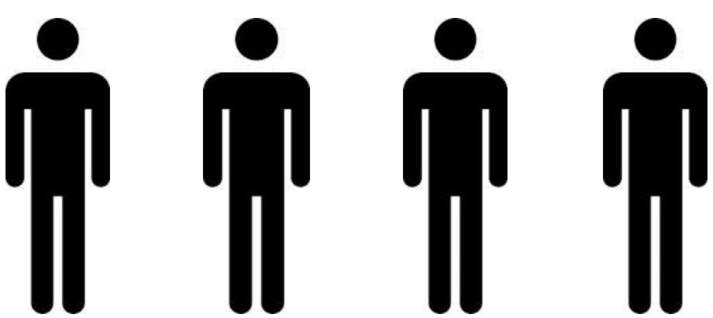
sampling phase

first half

Hiring staff - not adversarial



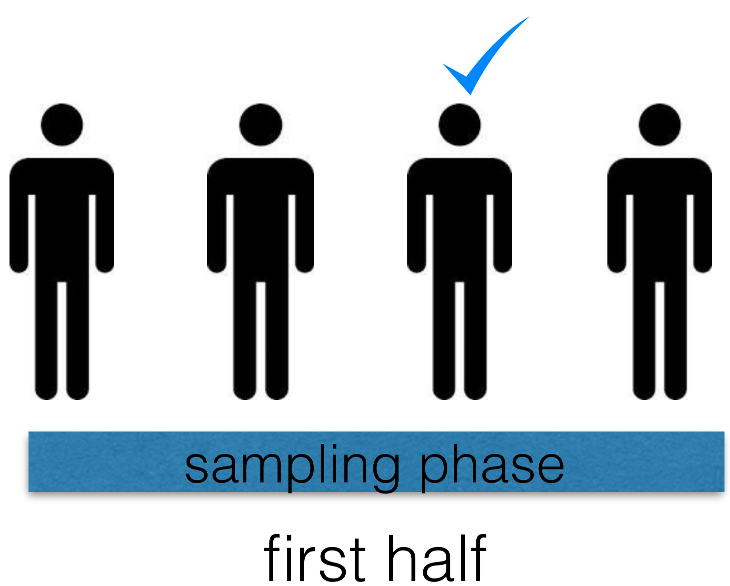
random order

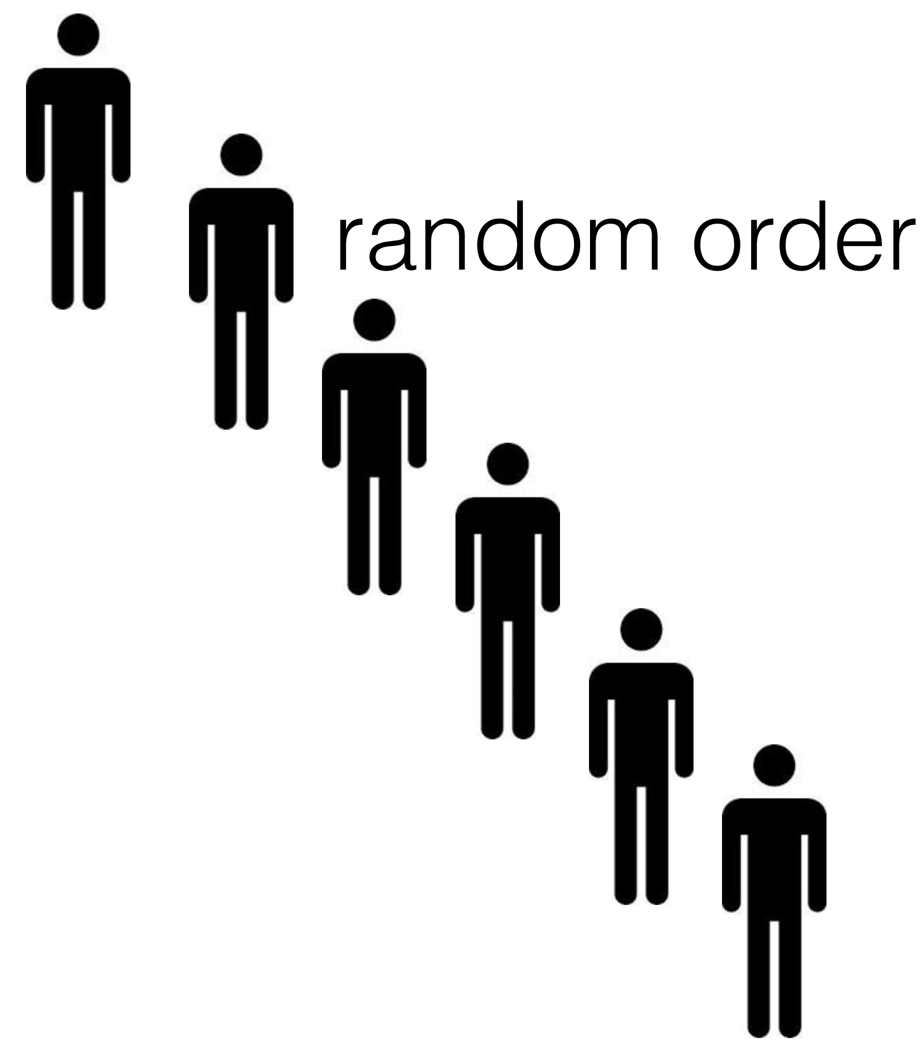


sampling phase

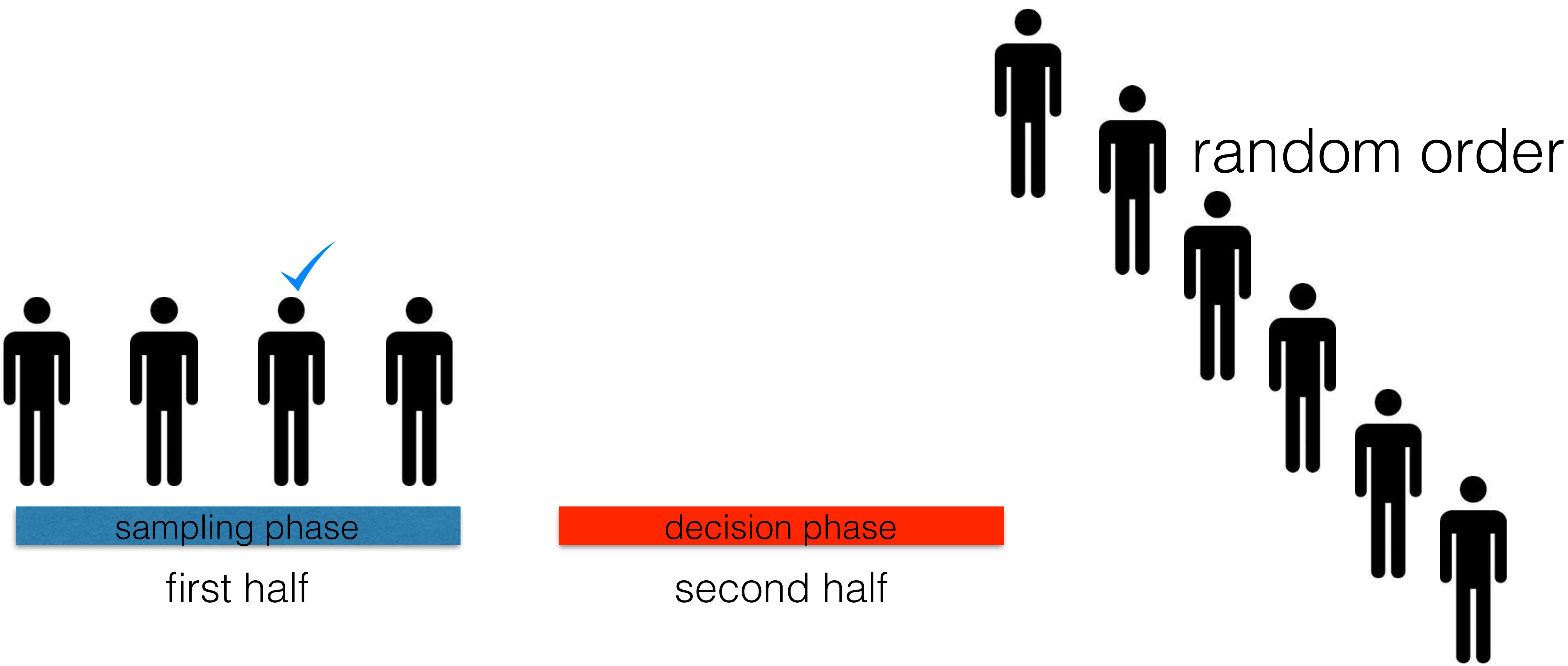
first half

Hiring staff - not adversarial





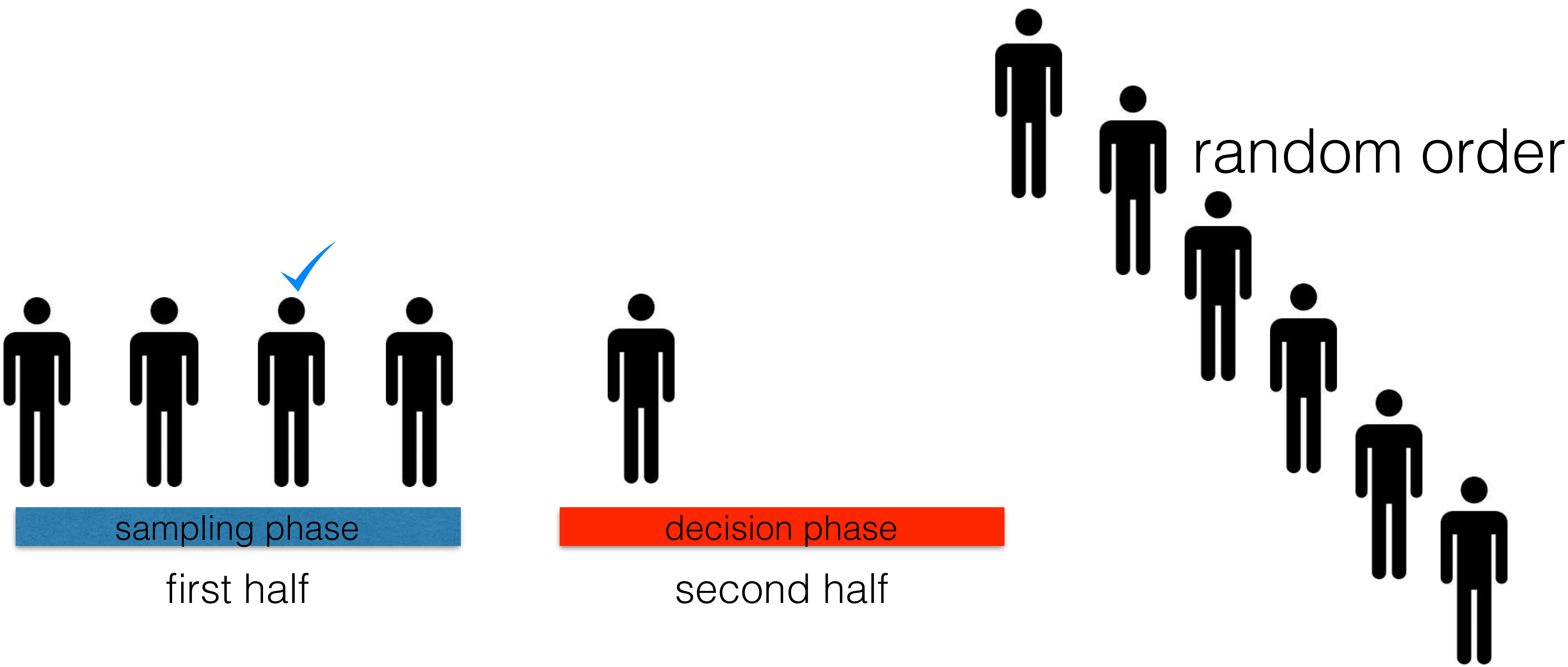
Hiring staff - not adversarial



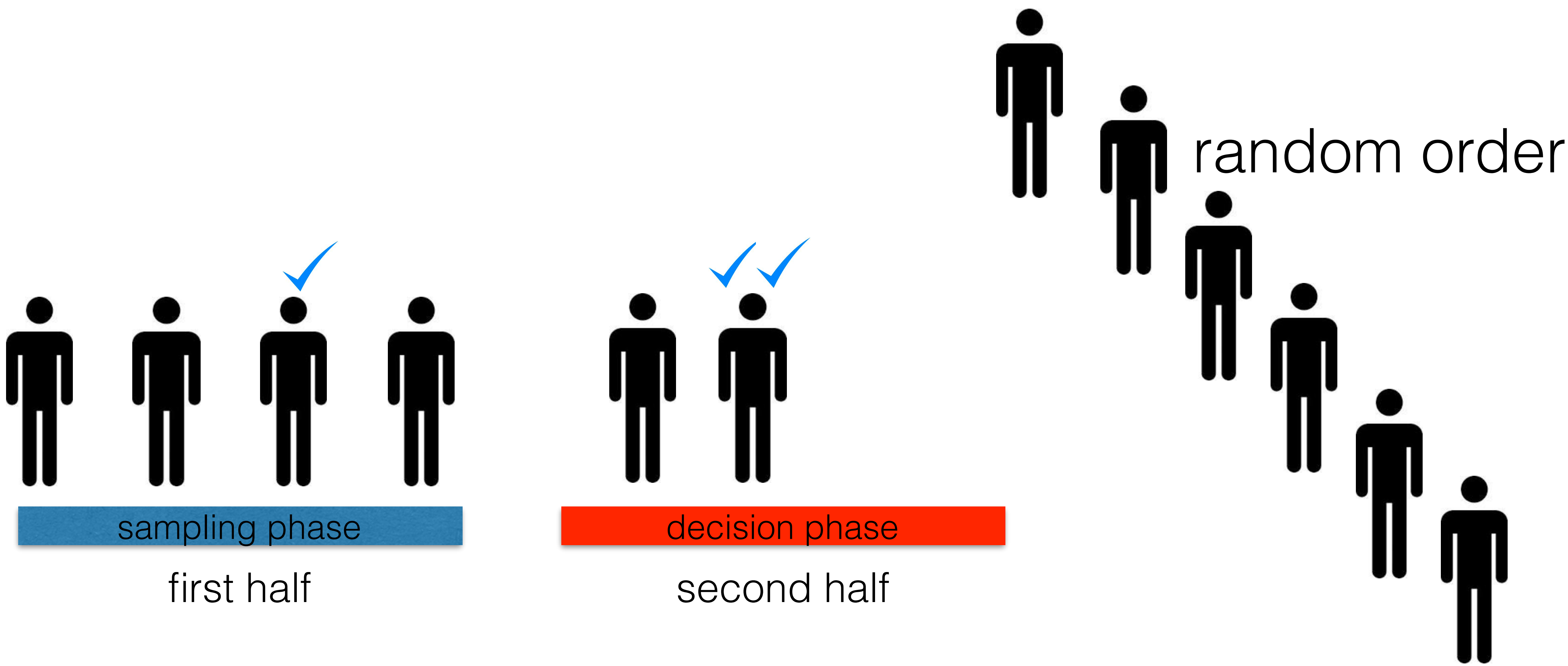
first half

second half

Hiring staff - not adversarial

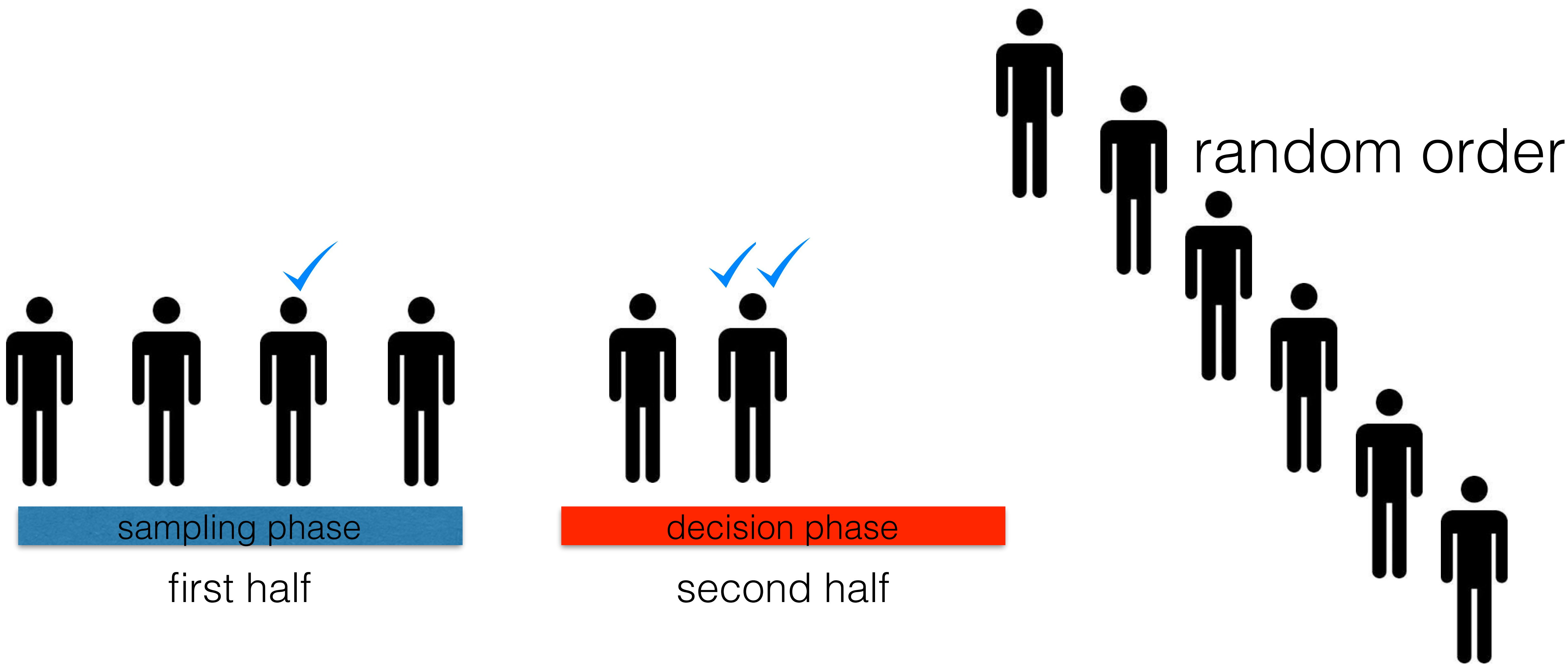


Hiring staff - not adversarial

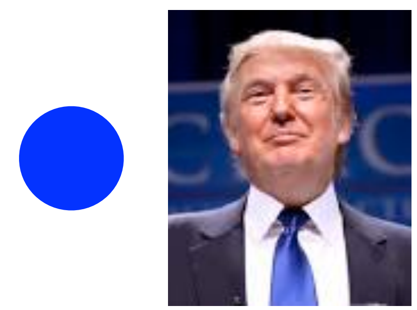


Hiring staff - not adversarial

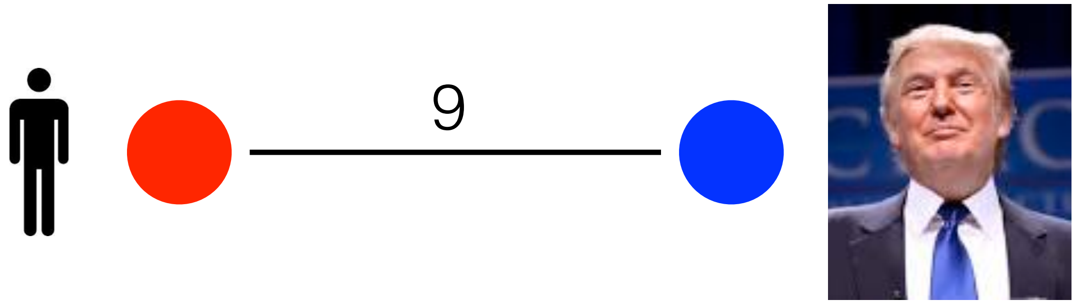
Success with prob > 1/4



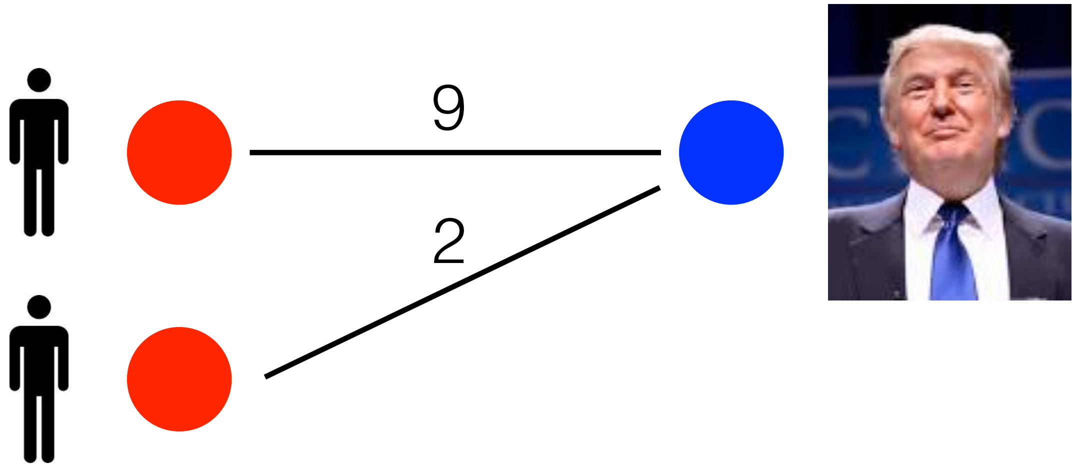
Actually Matching



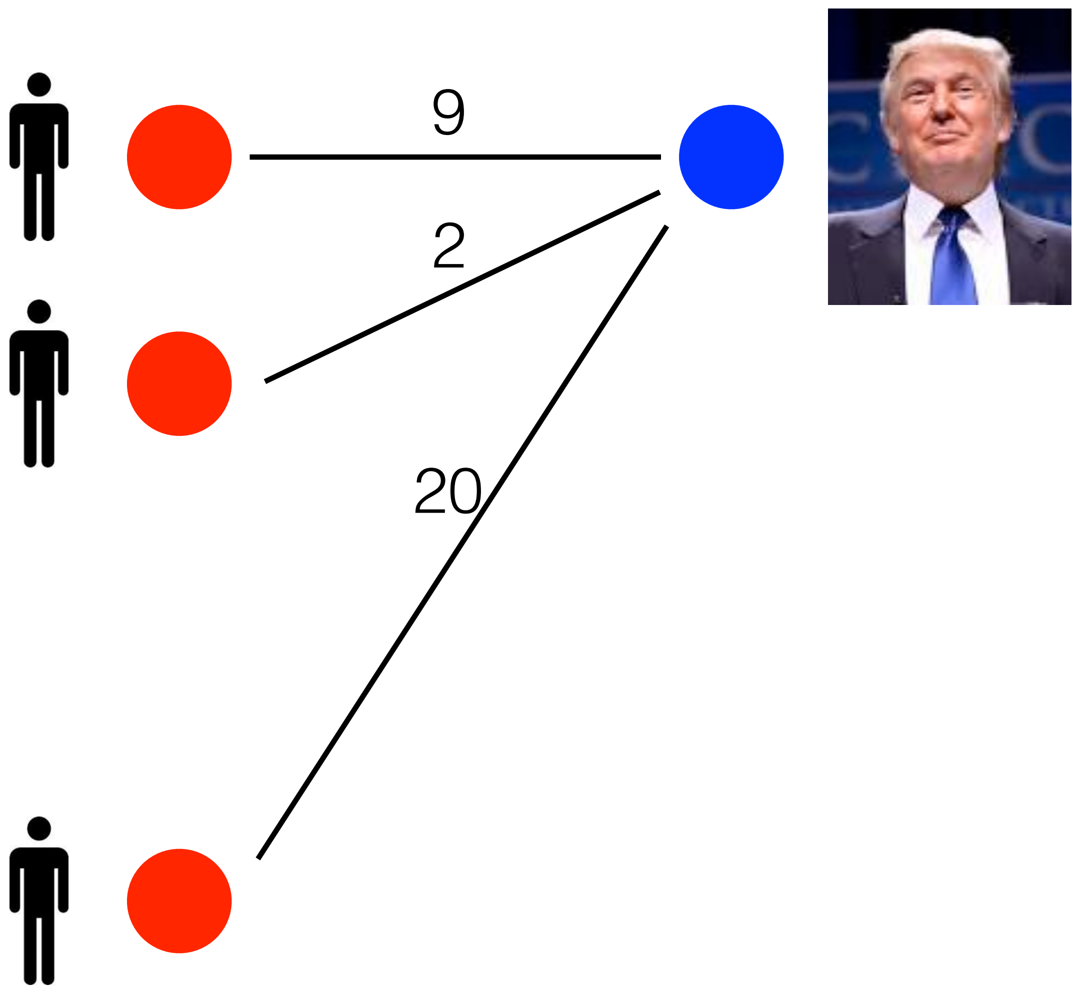
Actually Matching



Actually Matching

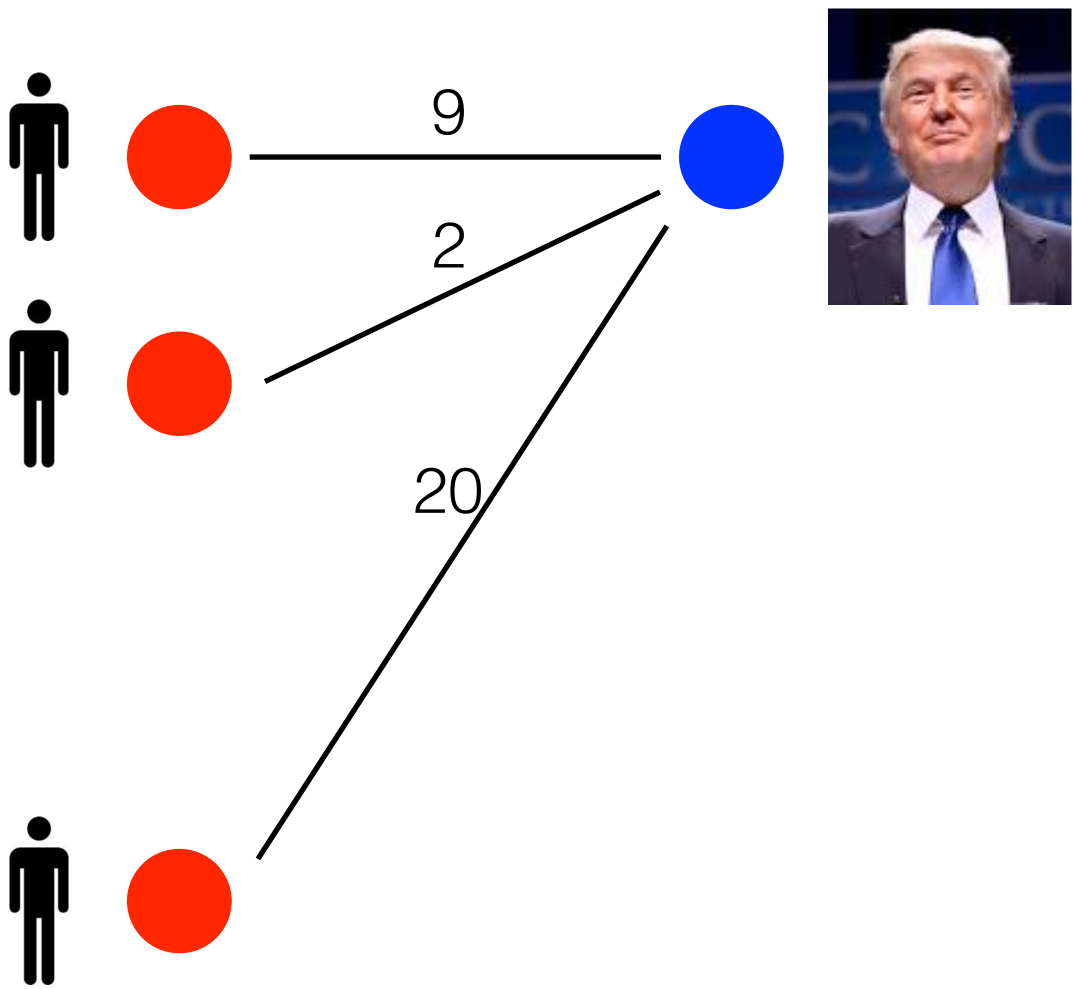


Actually Matching



Actually Matching

accept the edge with the largest weight instantaneously

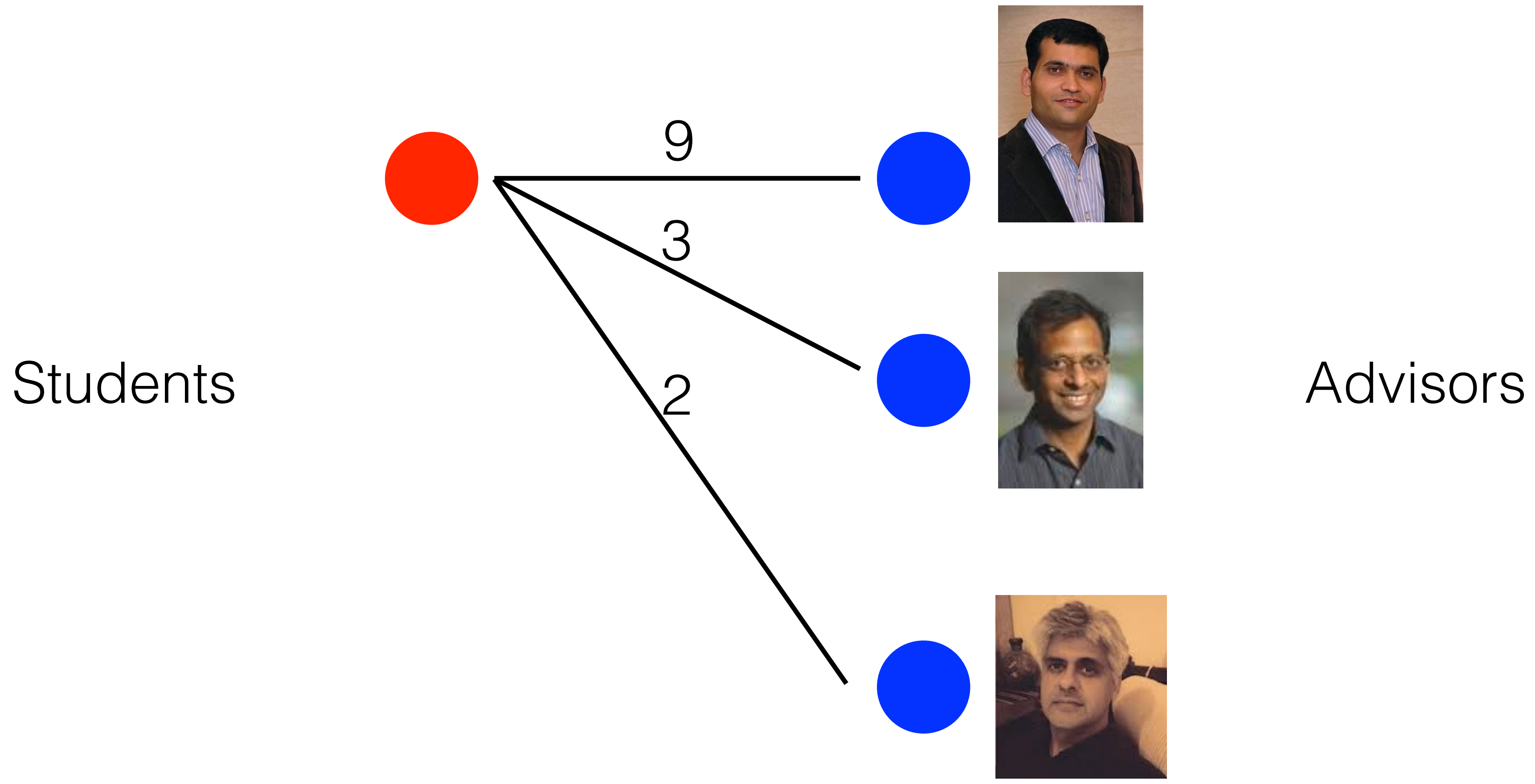


Natural Generalization

Students

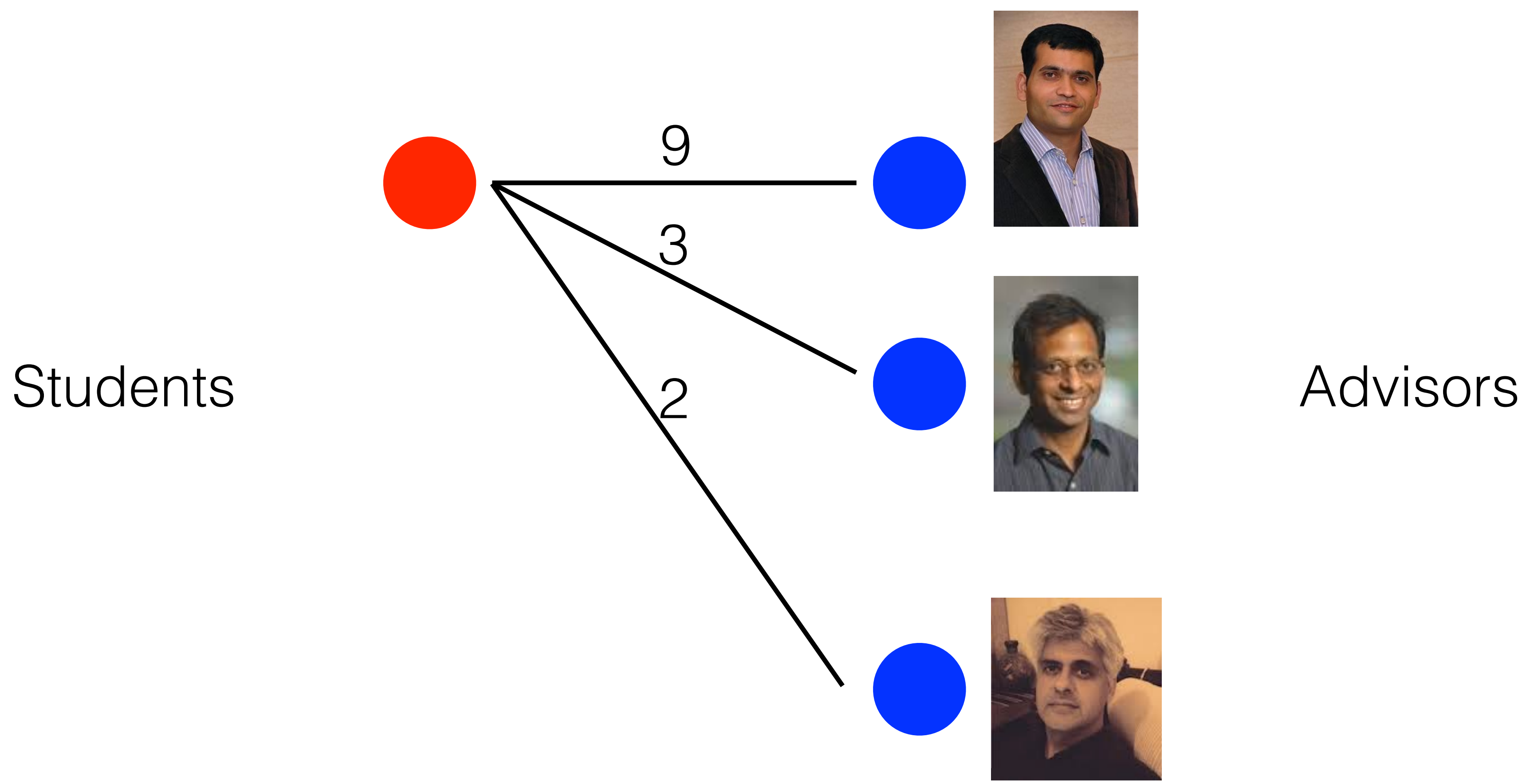
Advisors

Natural Generalization



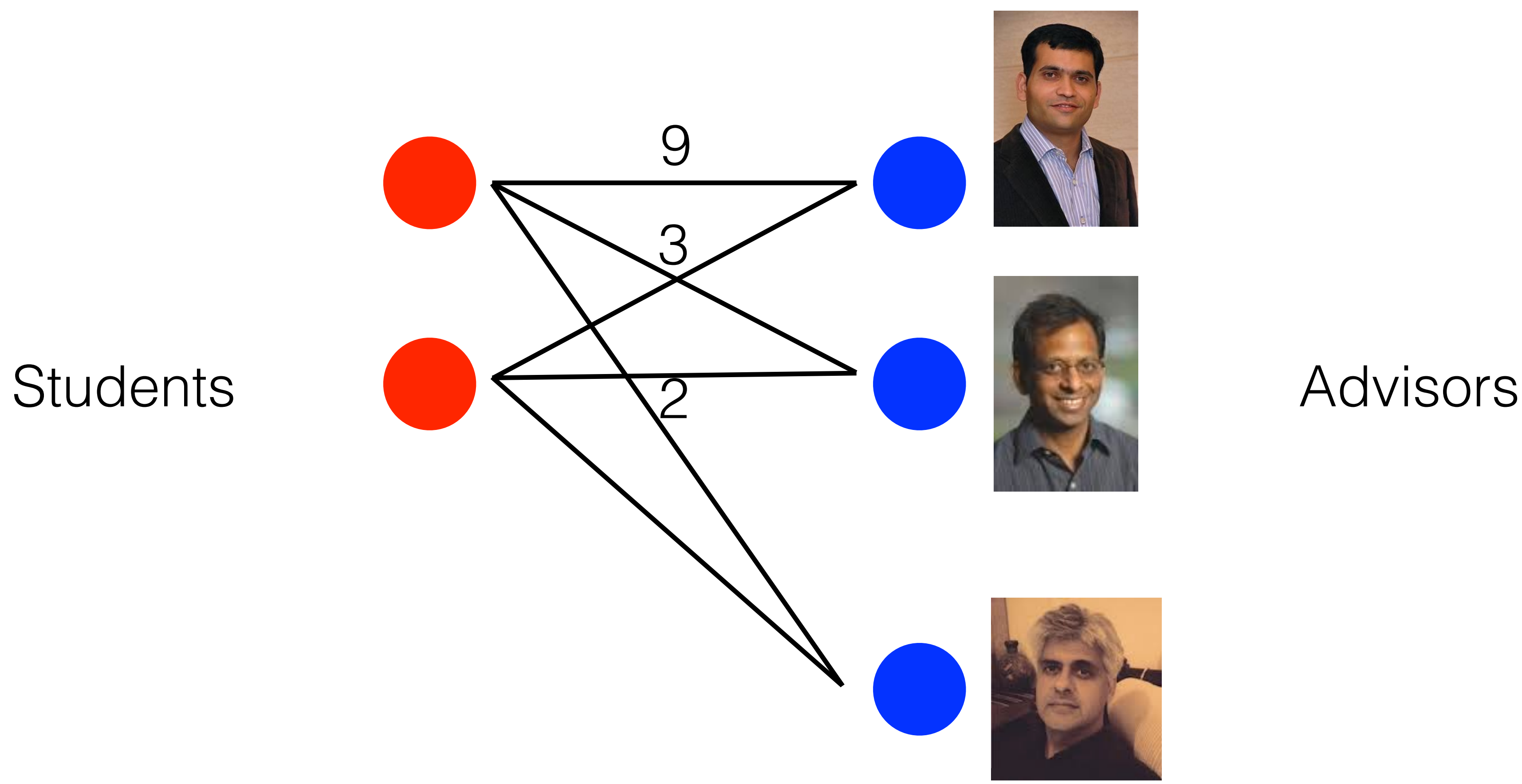
Natural Generalization

each advisor gets at most one student



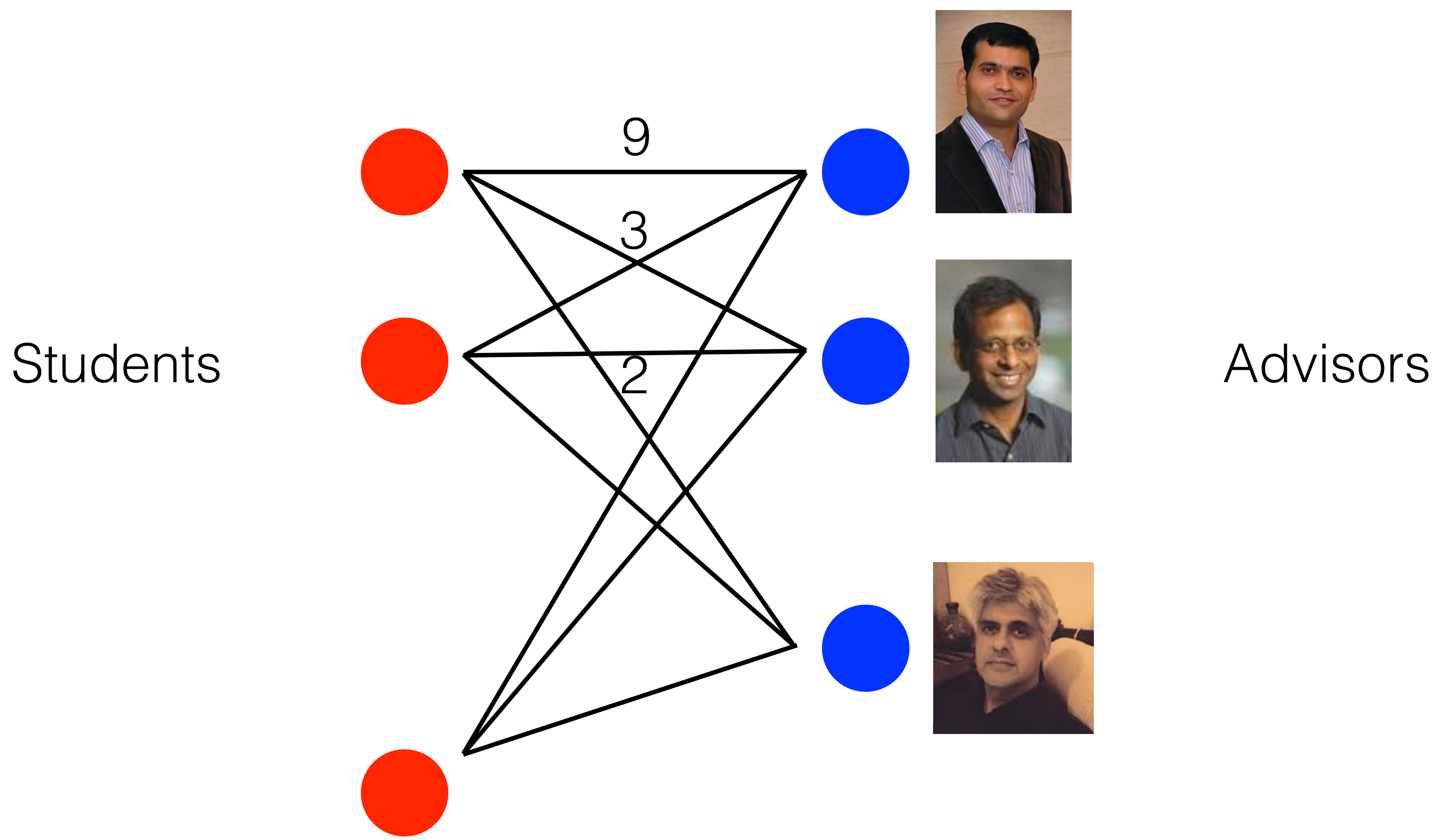
Natural Generalization

each advisor gets at most one student



Natural Generalization

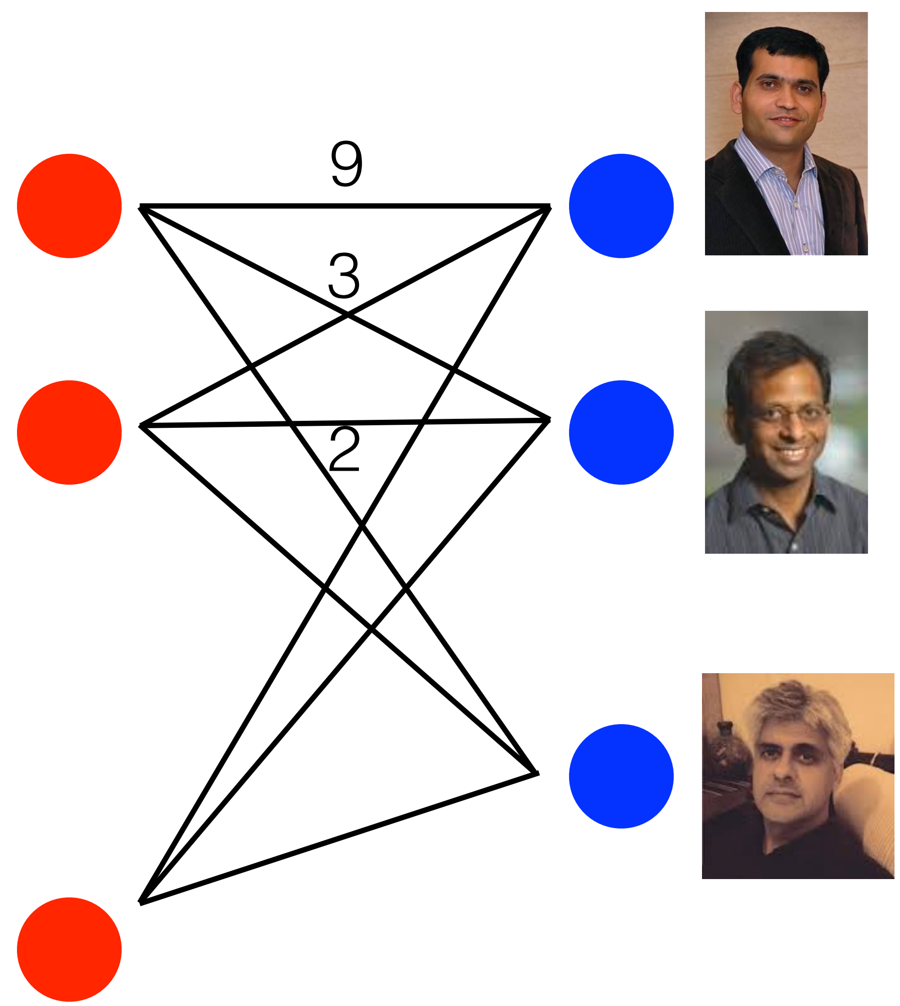
each advisor gets at most one student



Natural Generalization

each advisor gets at most one student - allocation made by **DEAN**

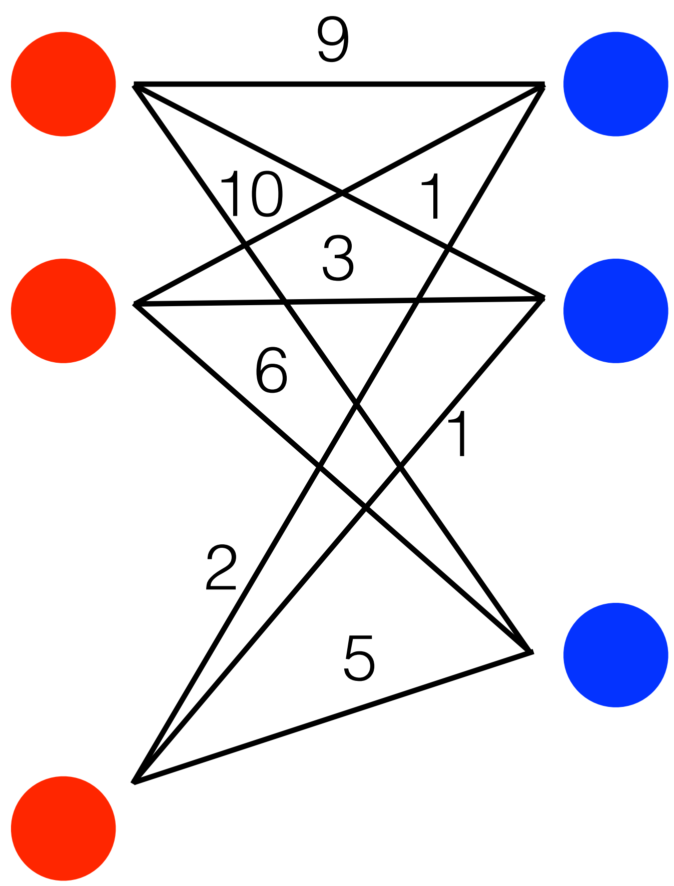
Students



Advisors

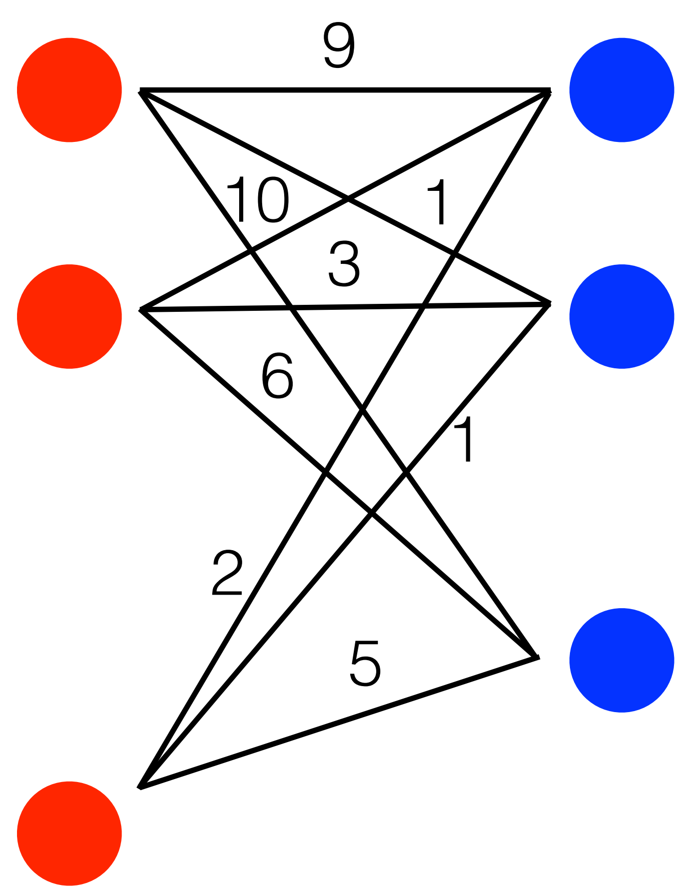
Objective: Matching with largest sum weight

Example



Example

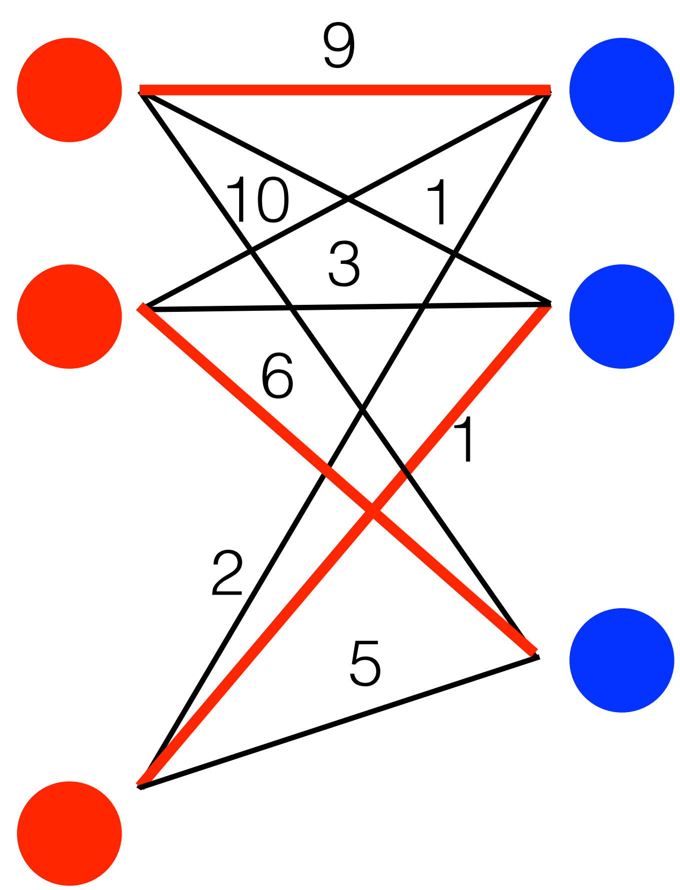
at most one accepted edge



Objective: Matching with largest sum weight

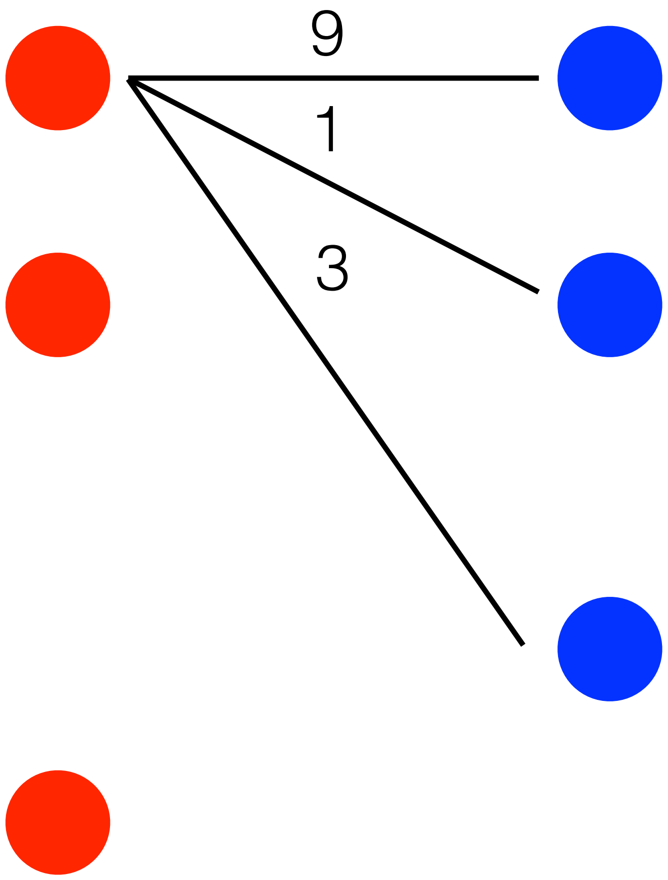
Example

at most one accepted edge

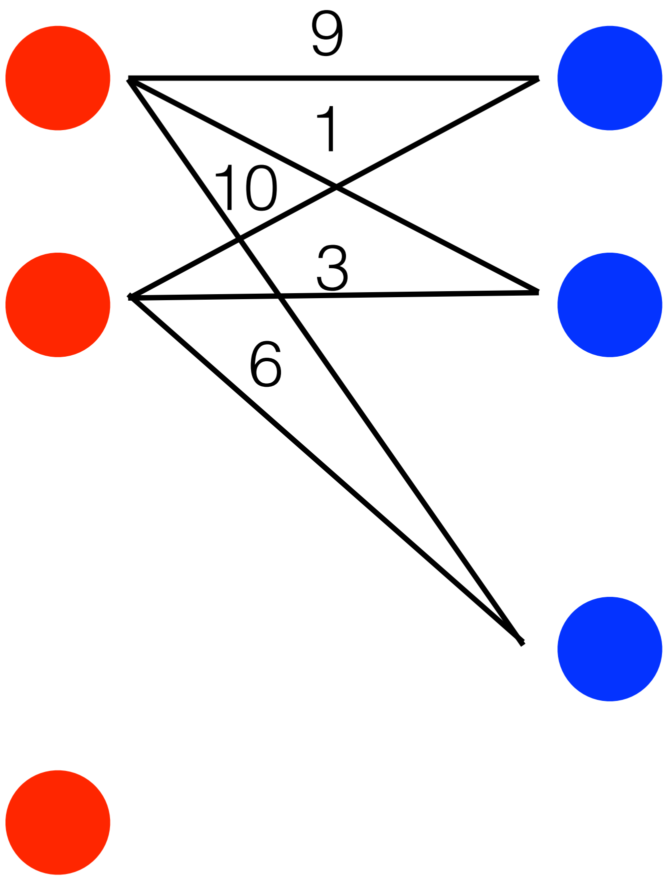


Objective: Matching with largest sum weight

How to solve this ONLINE

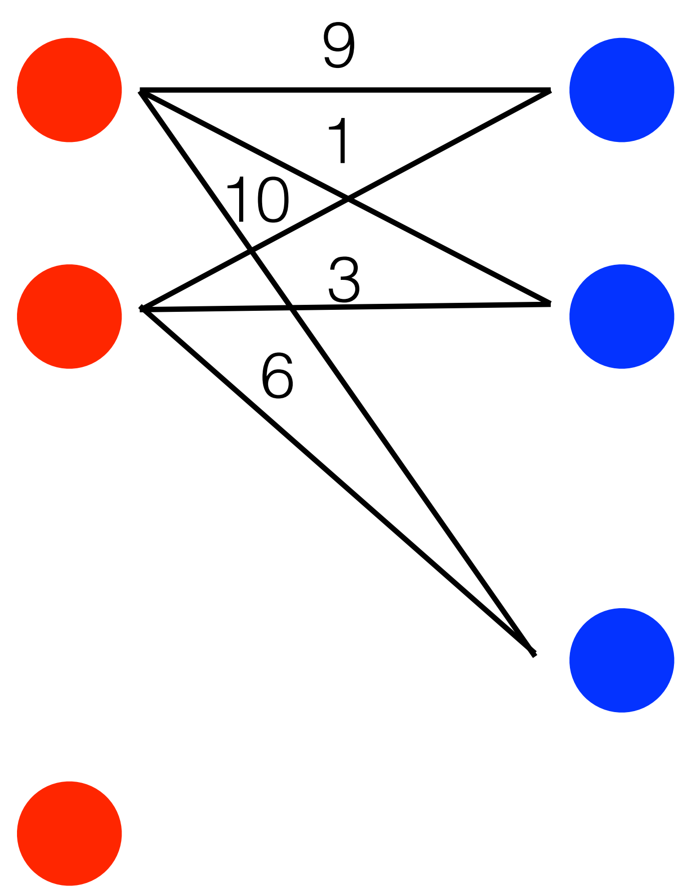


How to solve this ONLINE



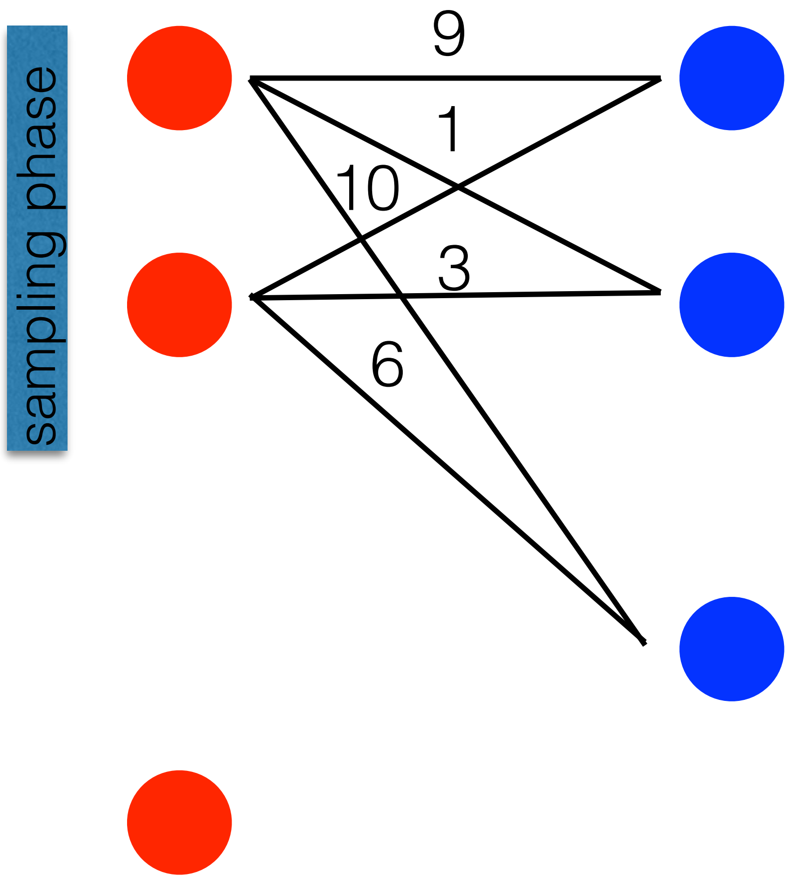
How to solve this ONLINE

Sampling idea as before



How to solve this ONLINE

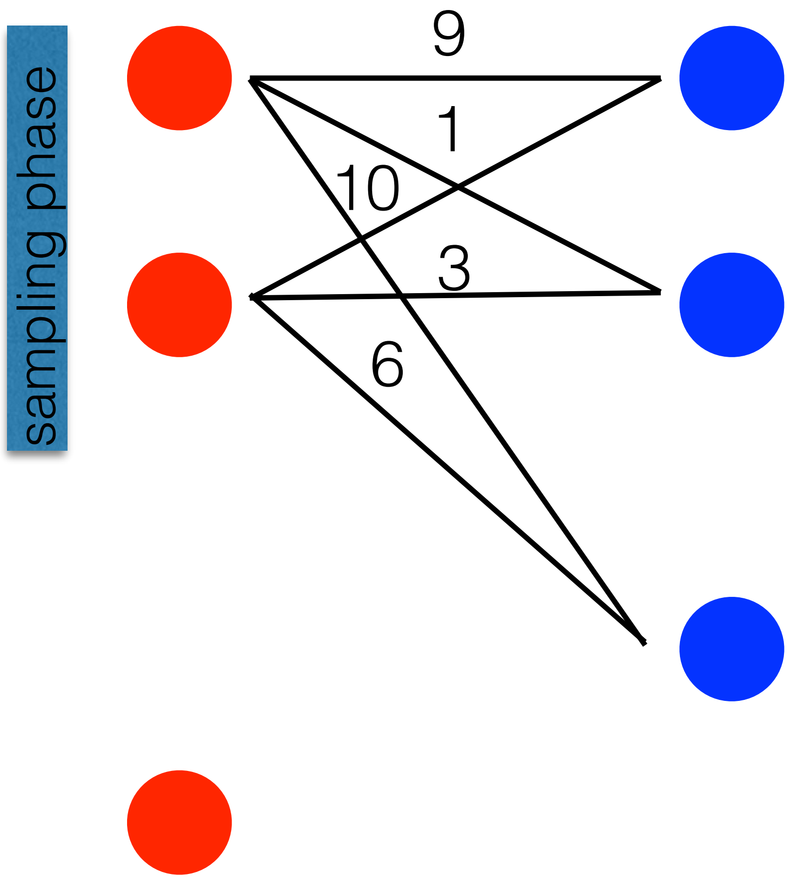
Sampling idea as before



How to solve this ONLINE

Sampling idea as before

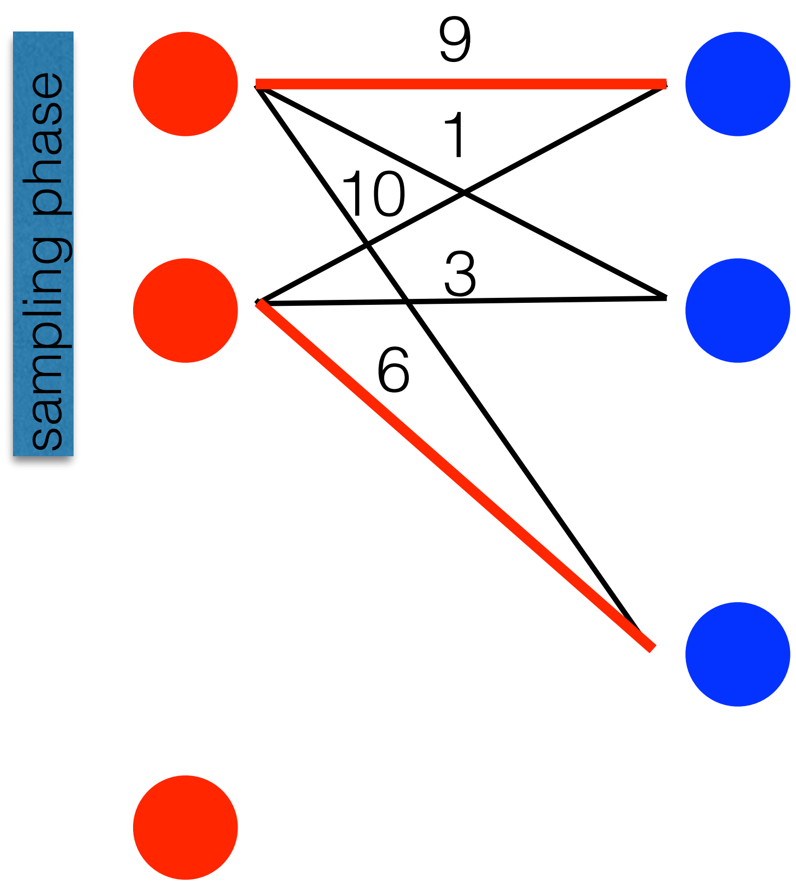
Find best matching



How to solve this ONLINE

Sampling idea as before

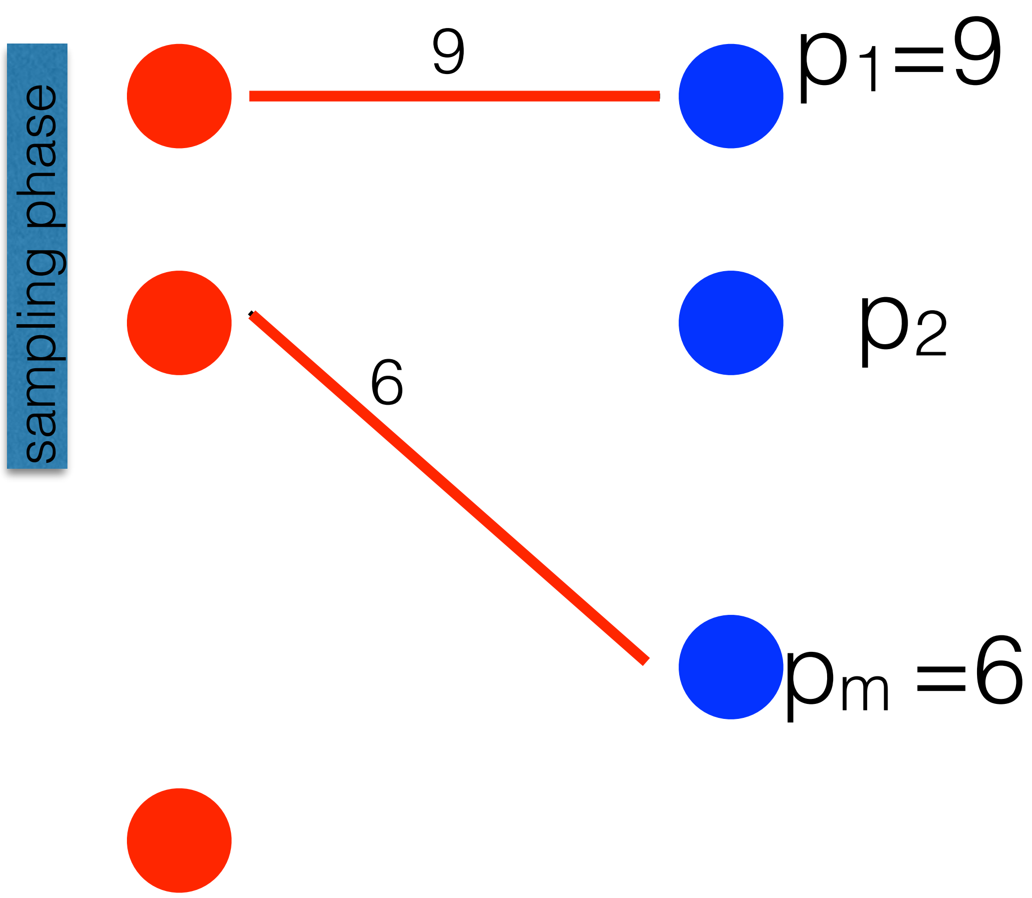
Find best matching



How to solve this ONLINE

Sampling idea as before

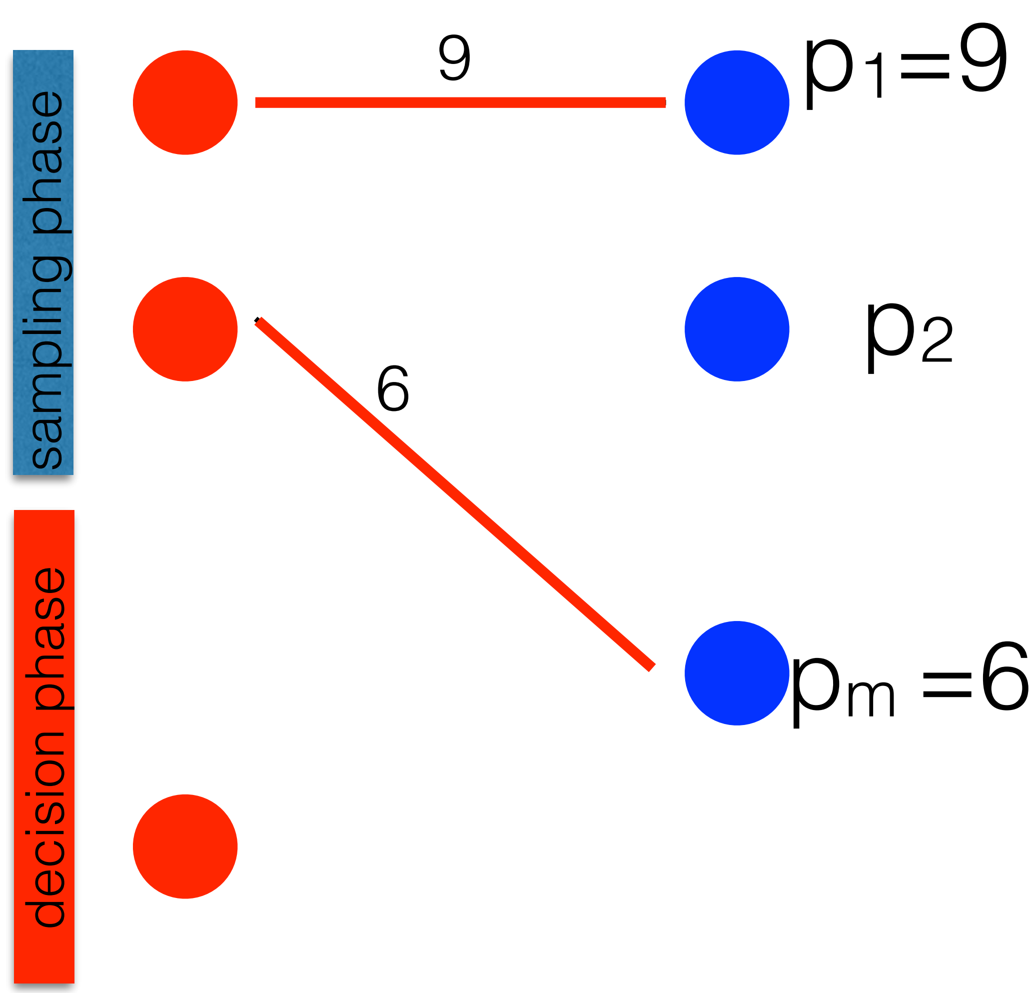
Find best matching



How to solve this ONLINE

Sampling idea as before

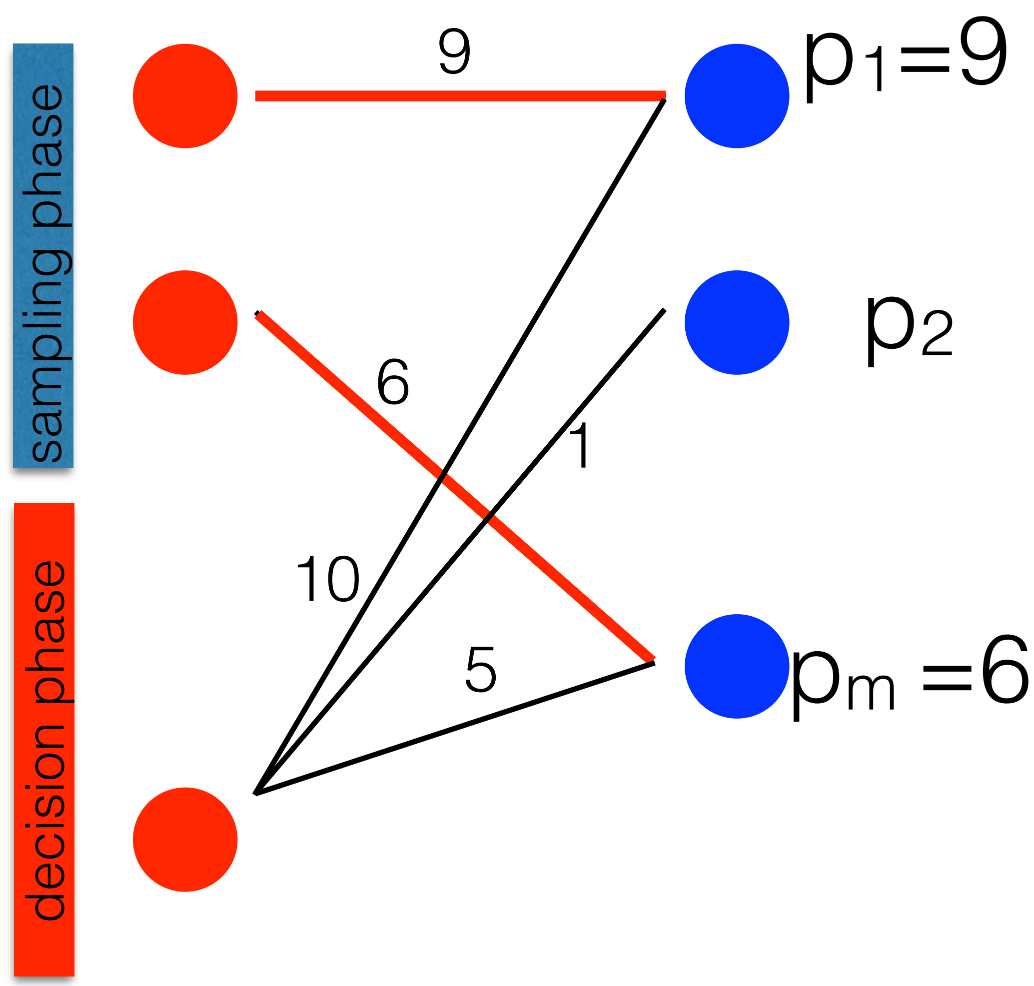
Find best matching



How to solve this ONLINE

Sampling idea as before

Find best matching

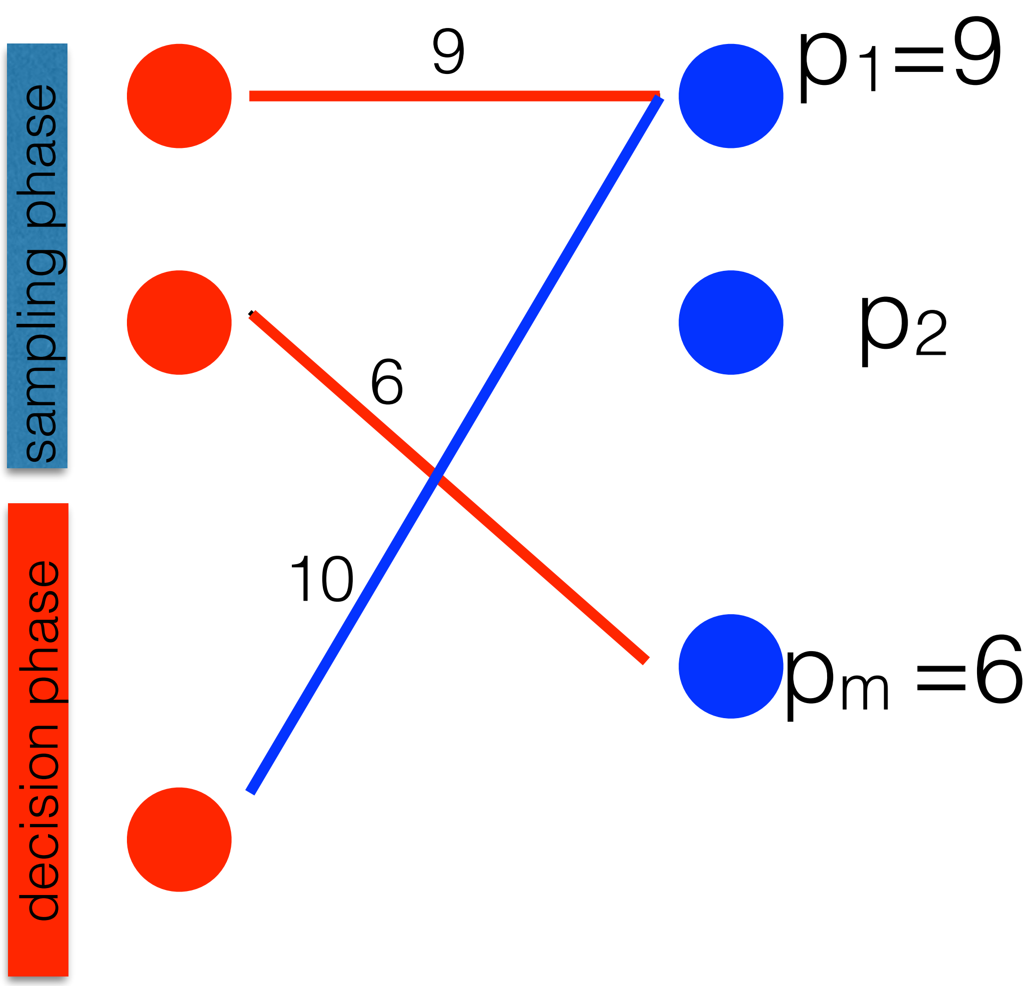


How to solve this ONLINE

Sampling idea as before

Find best matching

match the largest weight
more than price

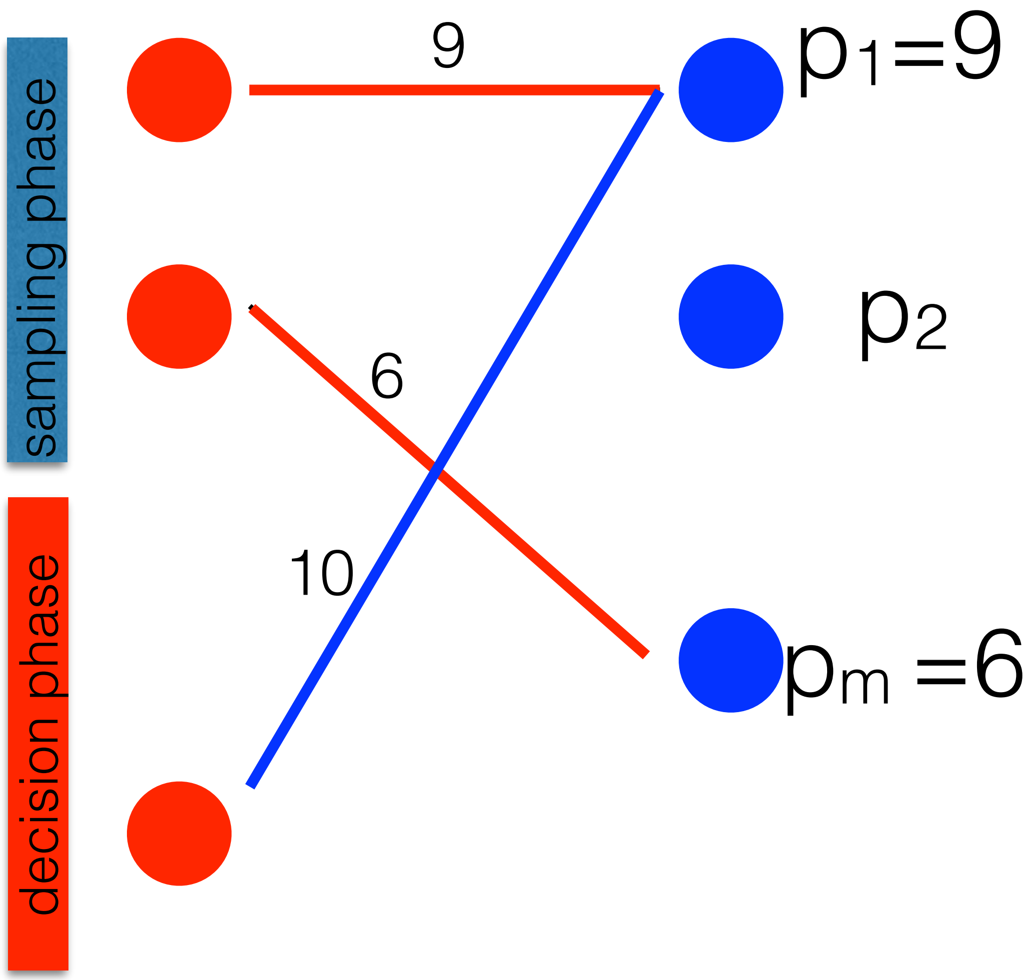


How to solve this ONLINE

Sampling idea as before

Find best matching

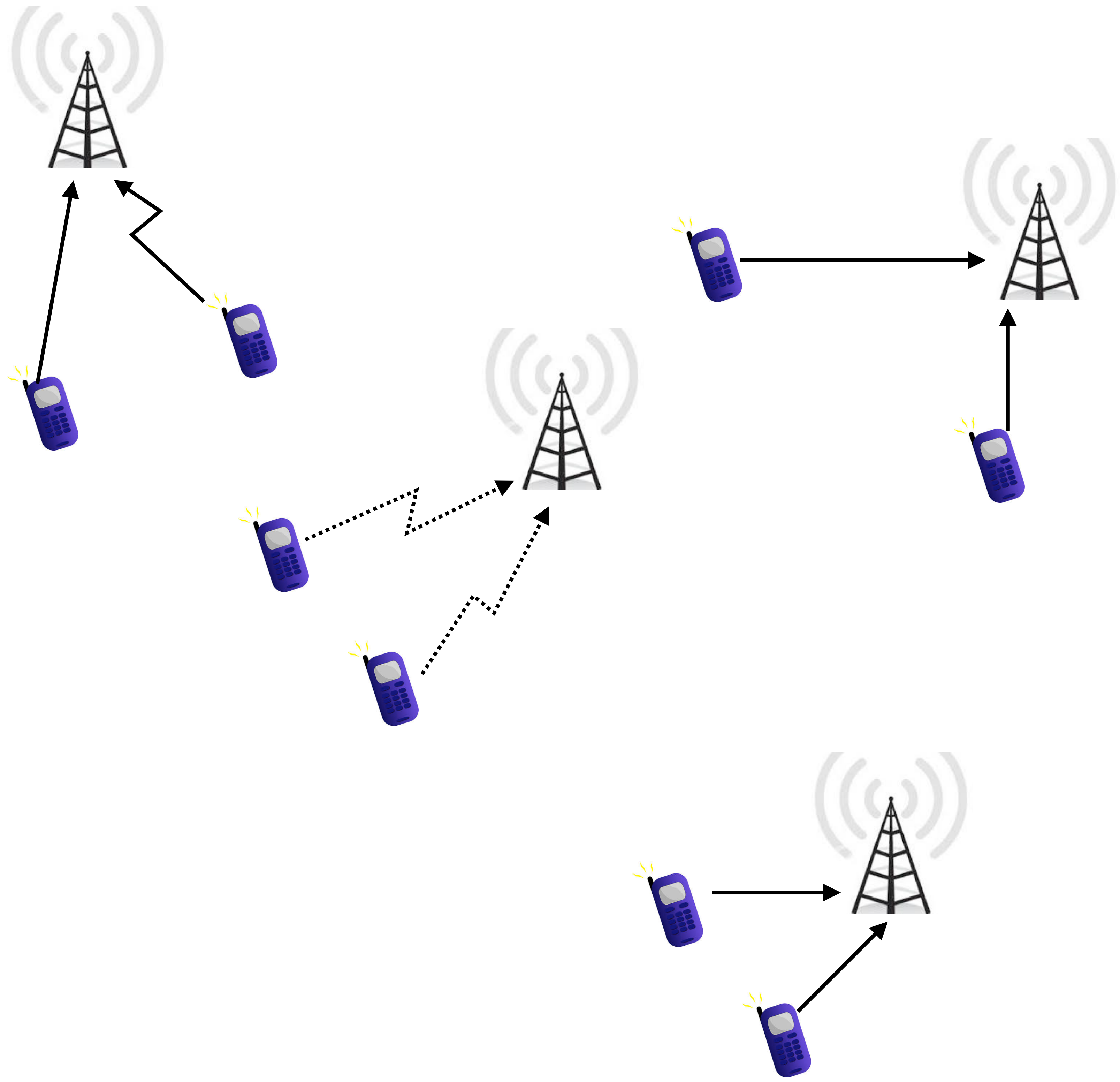
match the largest weight
more than price



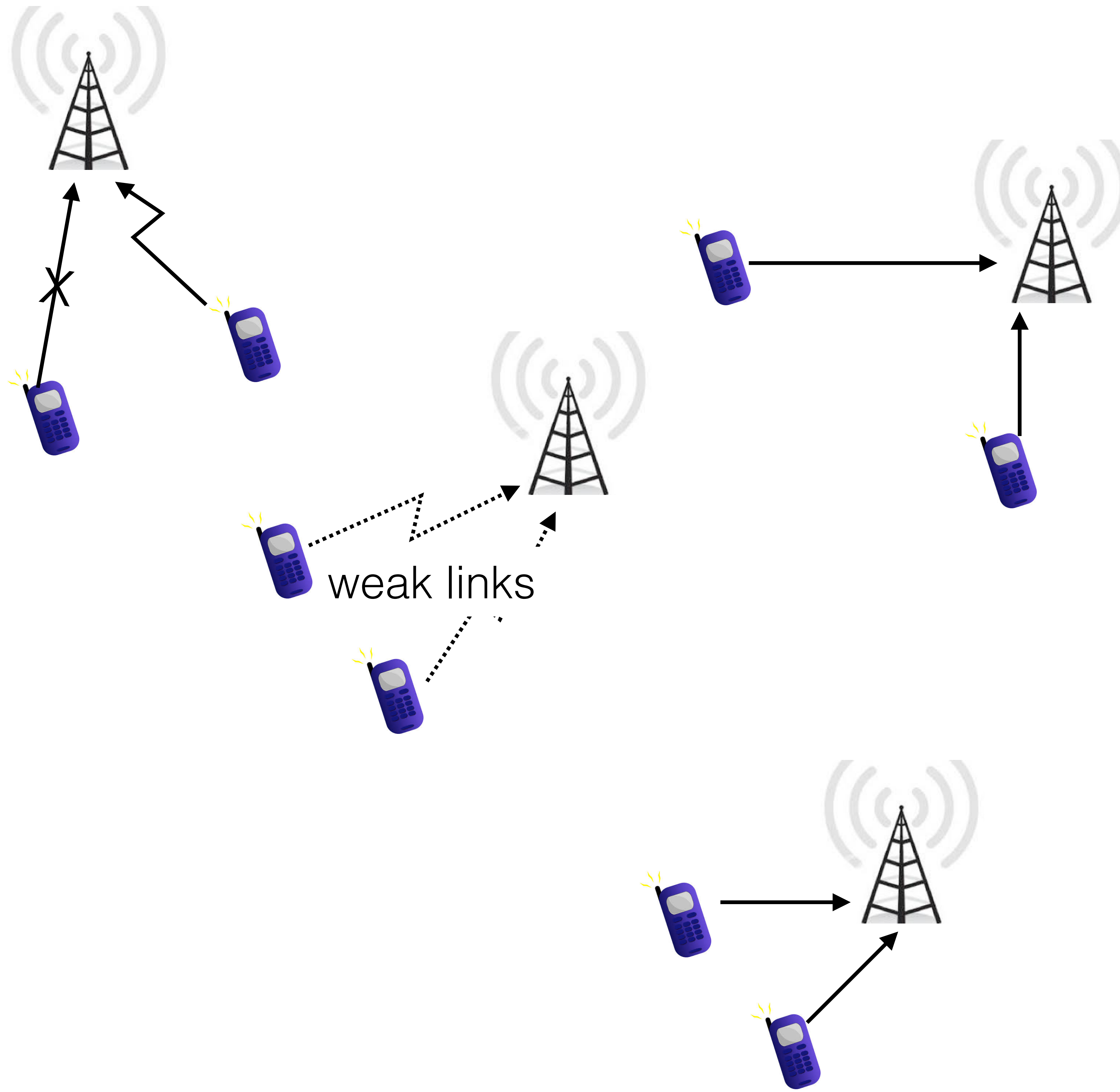
$$u(M_{\mathrm{ON}}) \geq \frac{u(M_{\mathrm{OFF}}(\mathrm{half}))}{2}$$

Result: 8-competitive/optimal [*Korula, Pal' 08***]**

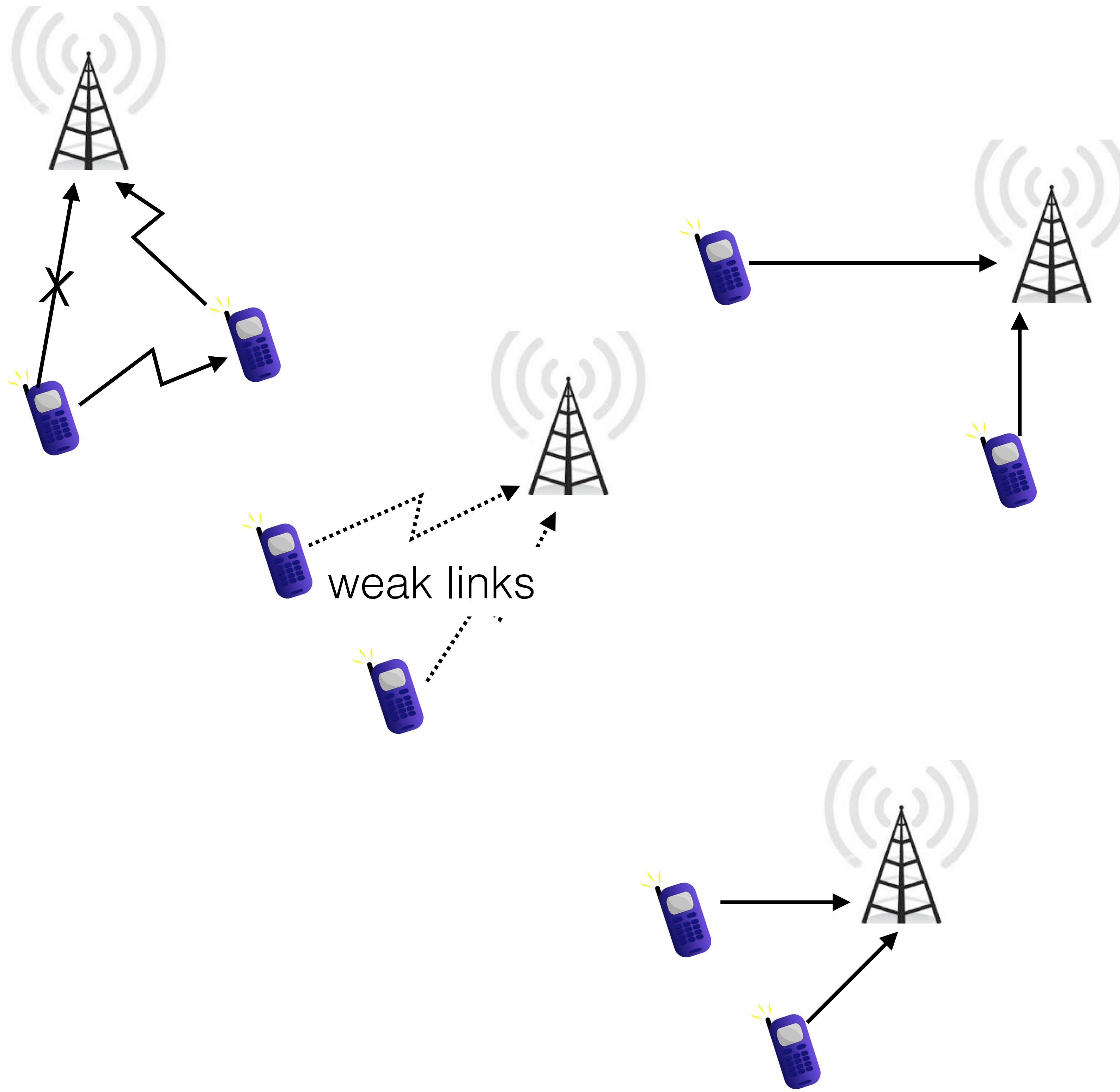
Modern Problem Device-2-Device Communication



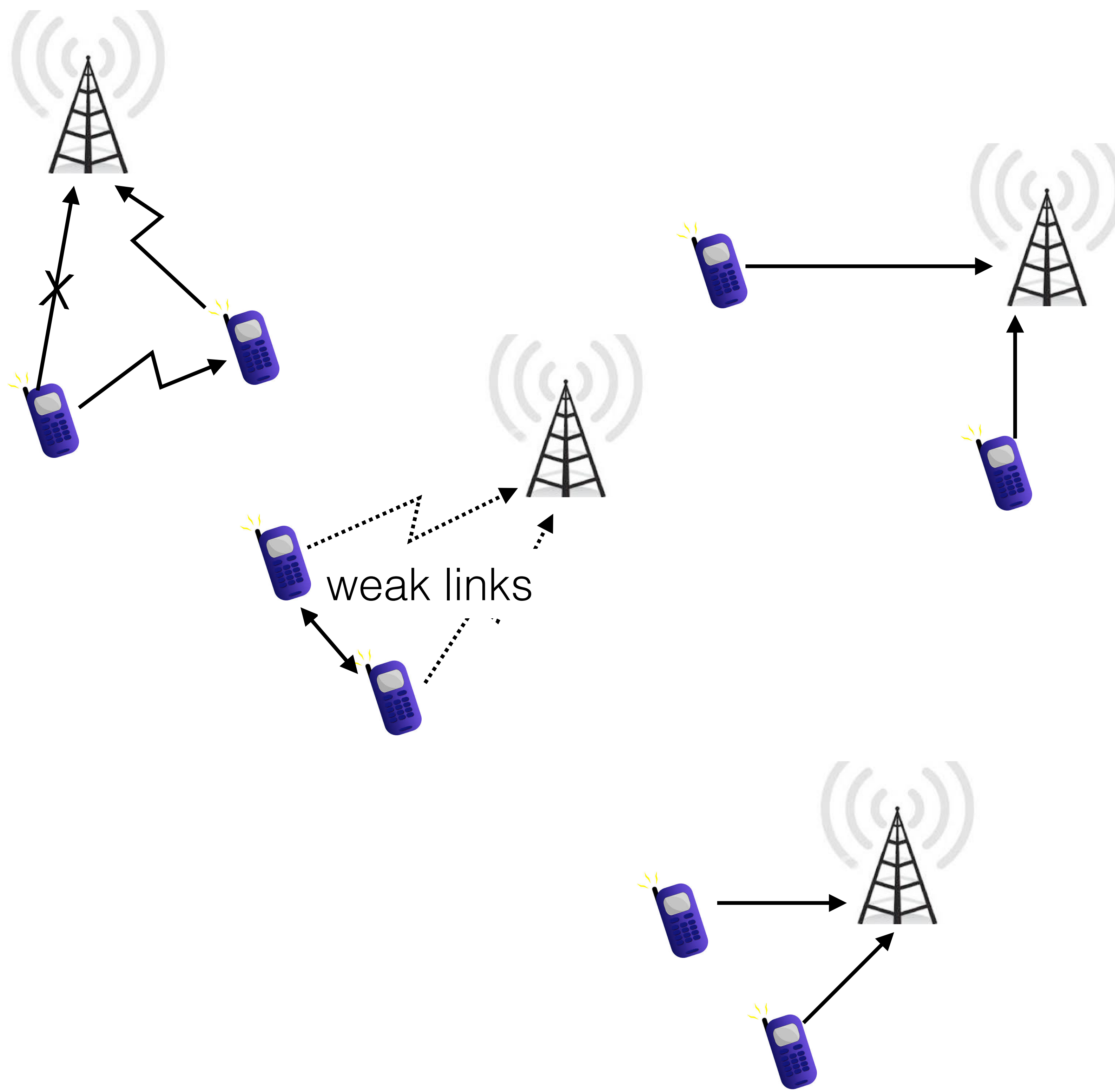
Modern Problem Device-2-Device Communication



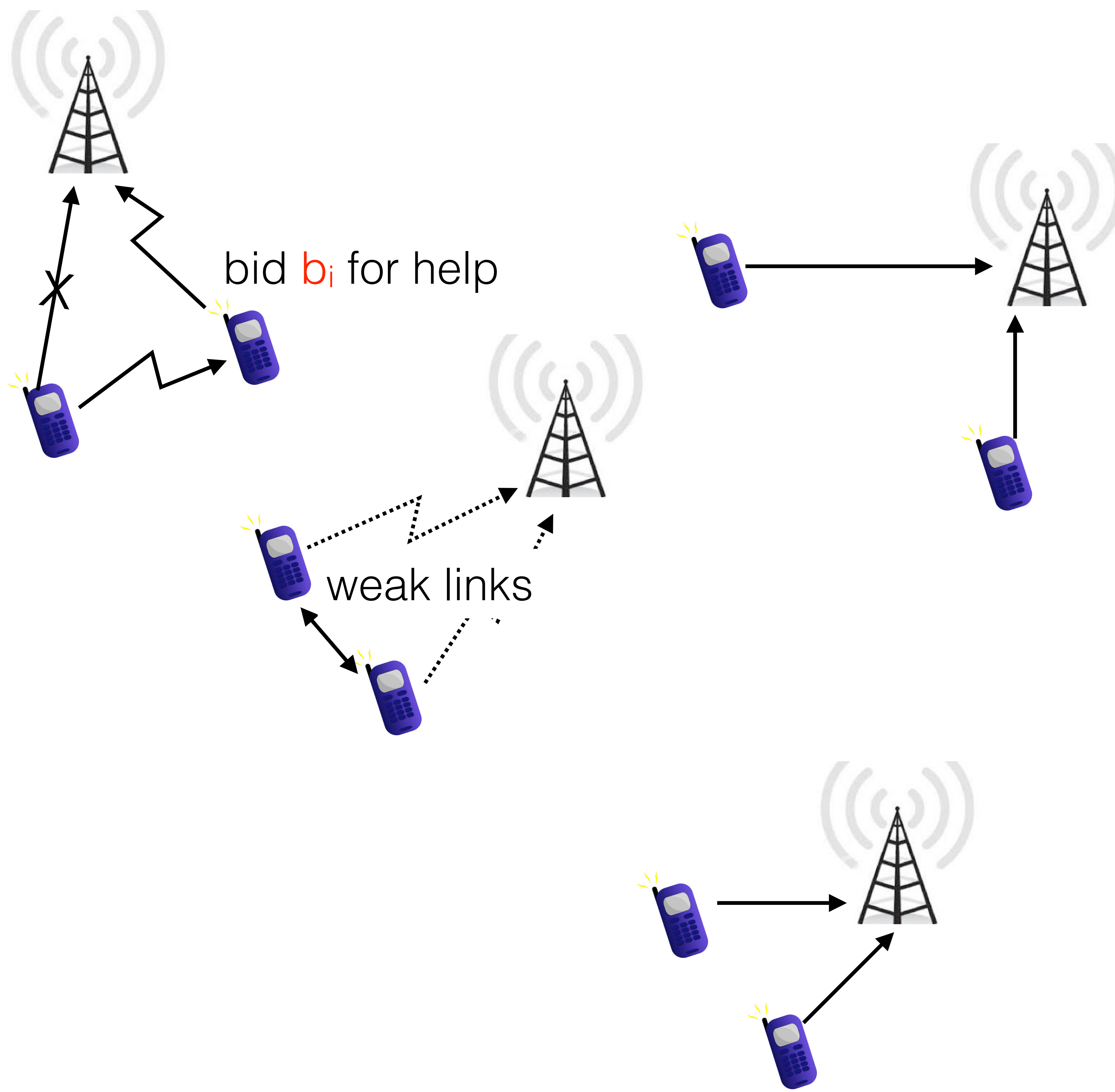
Modern Problem Device-2-Device Communication



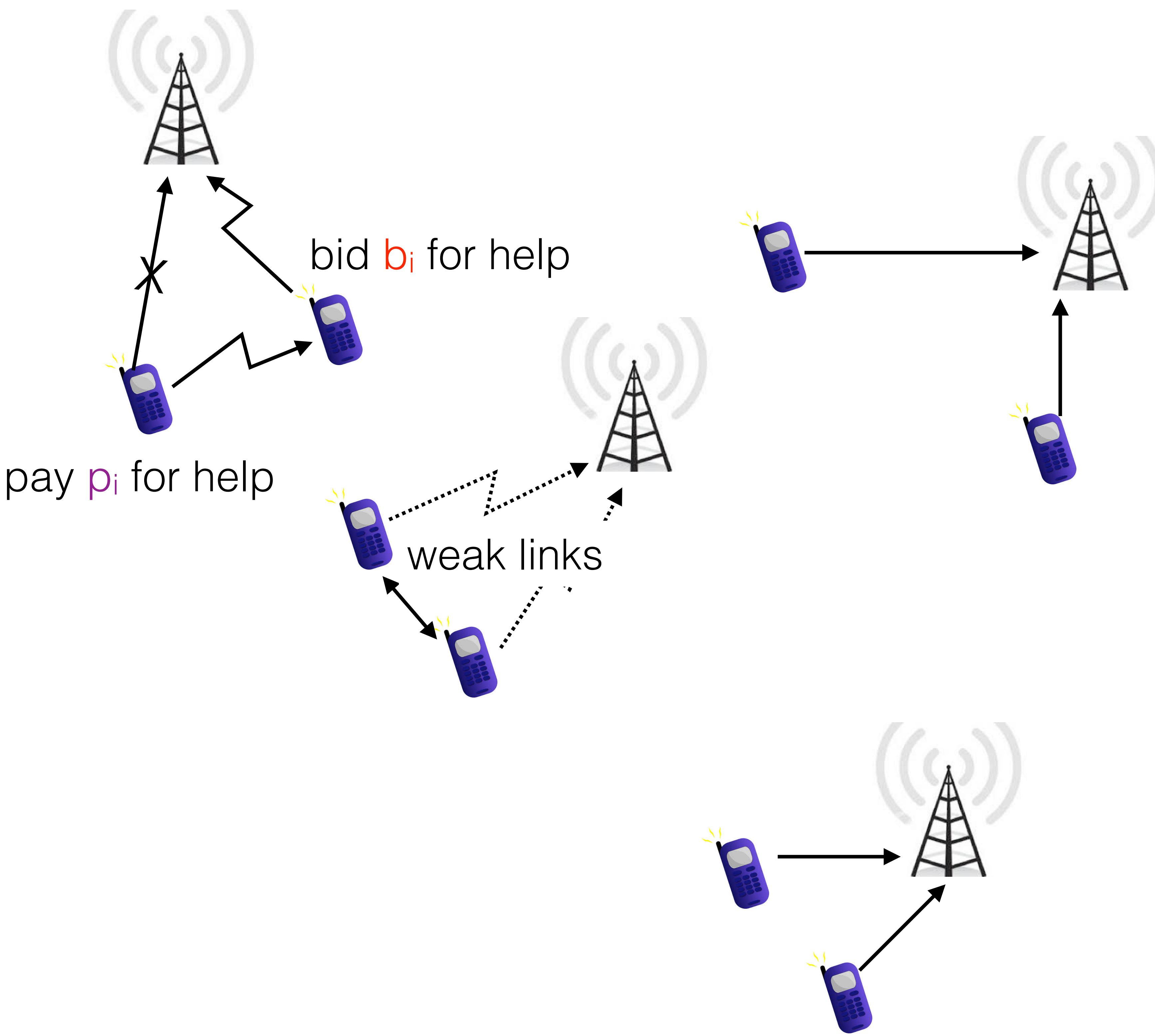
Modern Problem Device-2-Device Communication



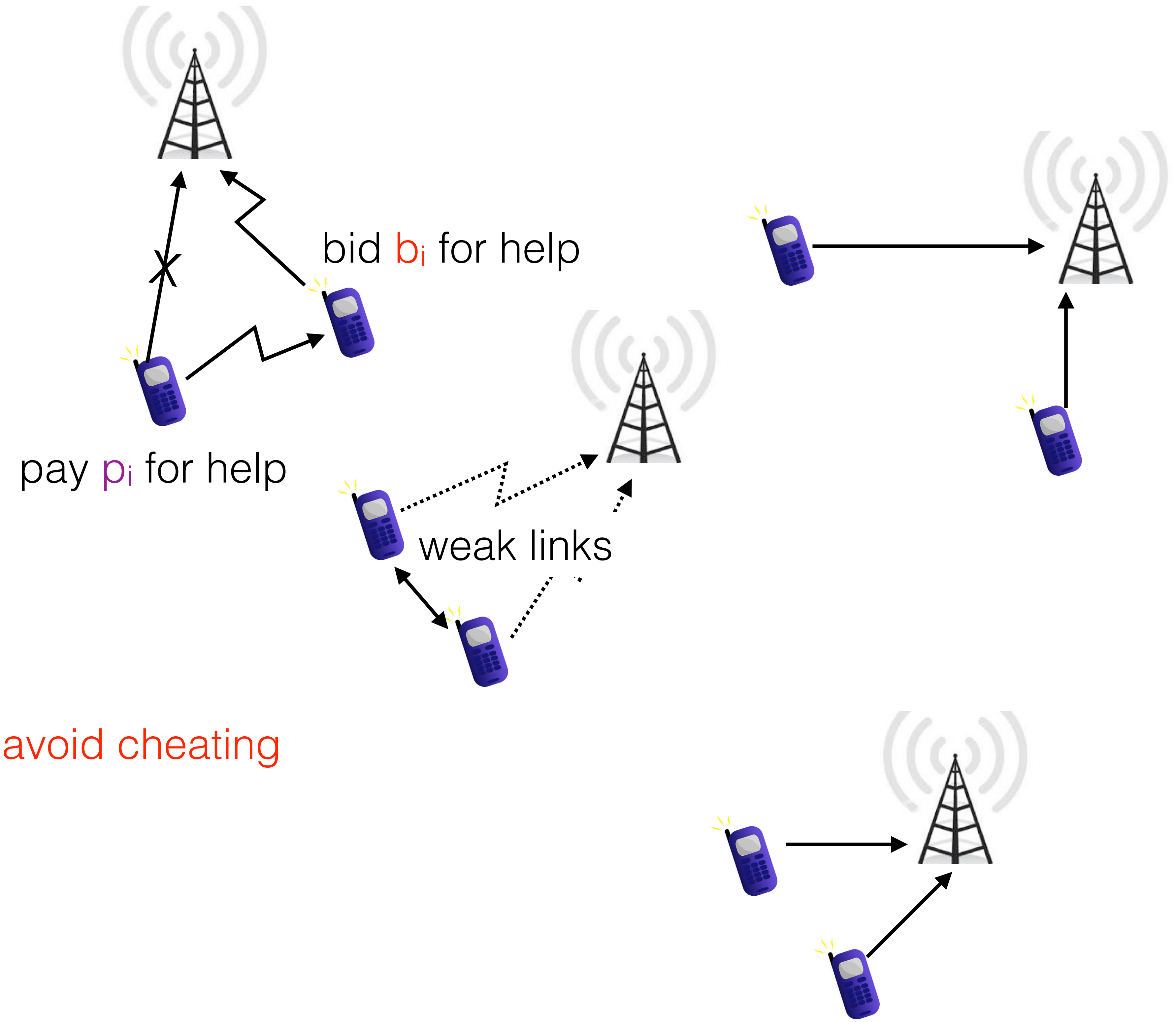
Modern Problem Device-2-Device Communication



Modern Problem Device-2-Device Communication

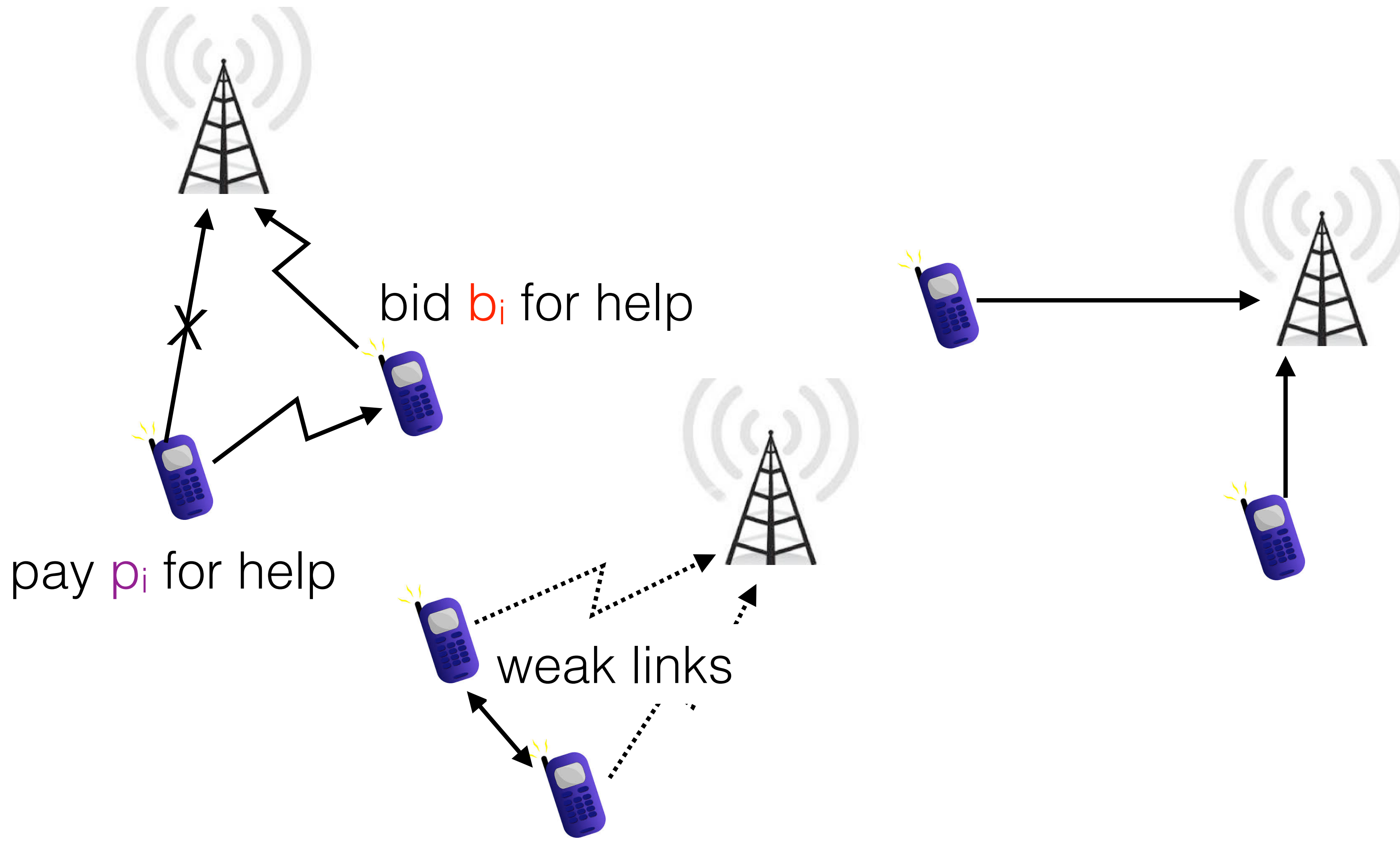


Modern Problem Device-2-Device Communication



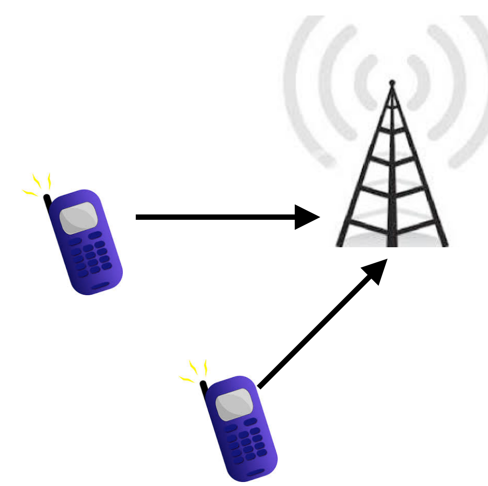
mechanism to avoid cheating

Modern Problem Device-2-Device Communication

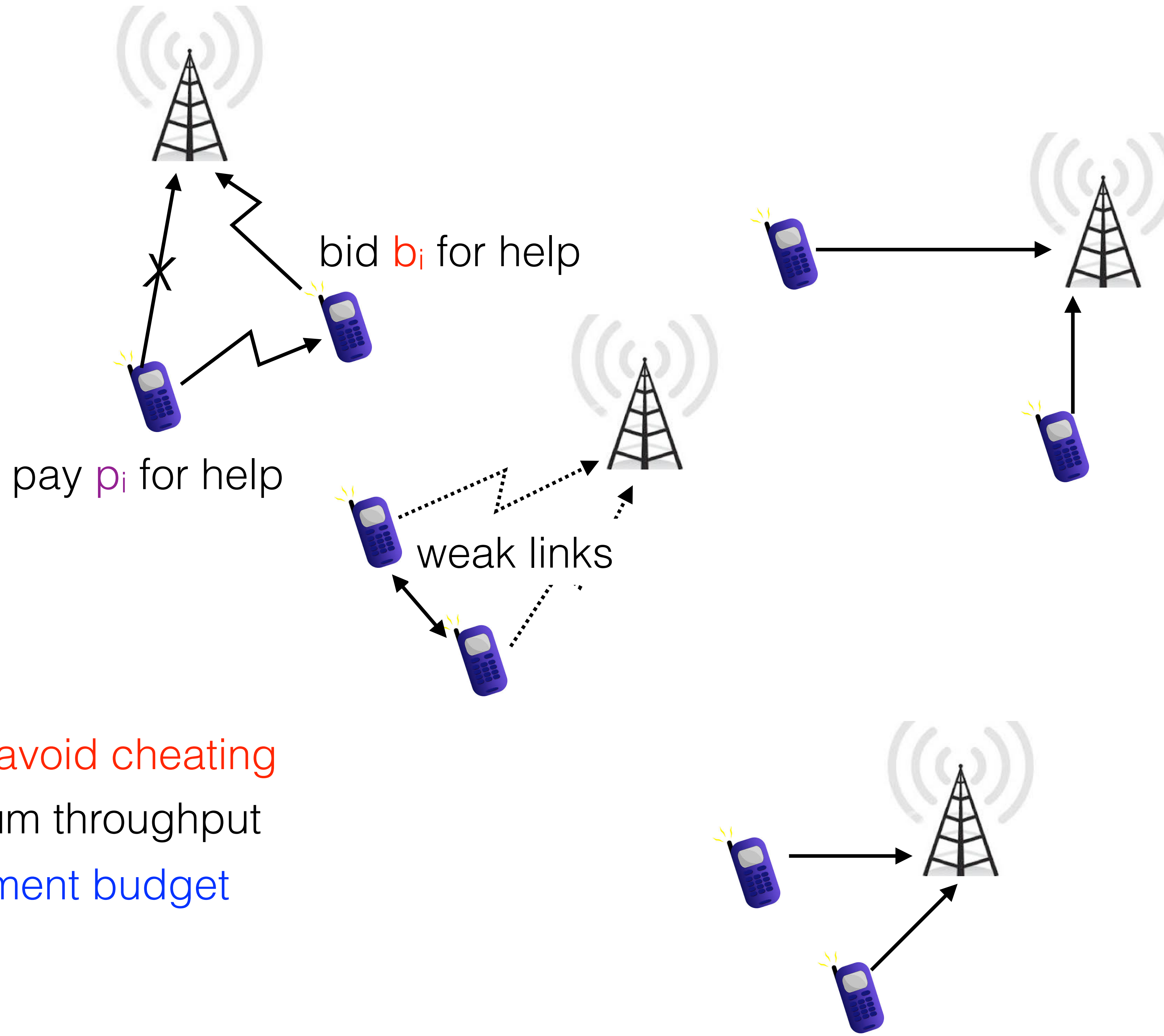


mechanism to avoid cheating

ensure maximum throughput



Modern Problem Device-2-Device Communication

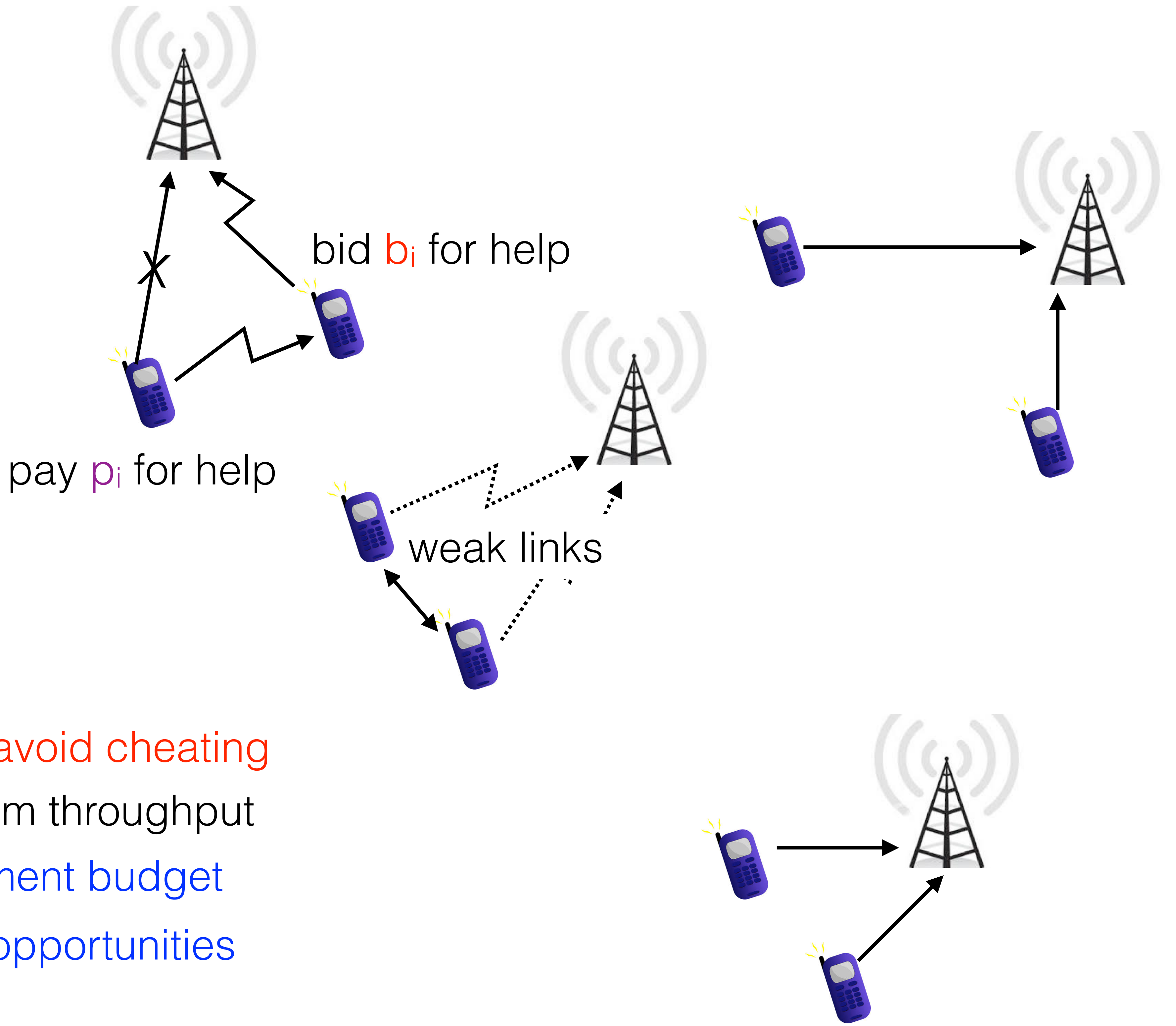


mechanism to avoid cheating

ensure maximum throughput

subject to payment budget

Modern Problem Device-2-Device Communication



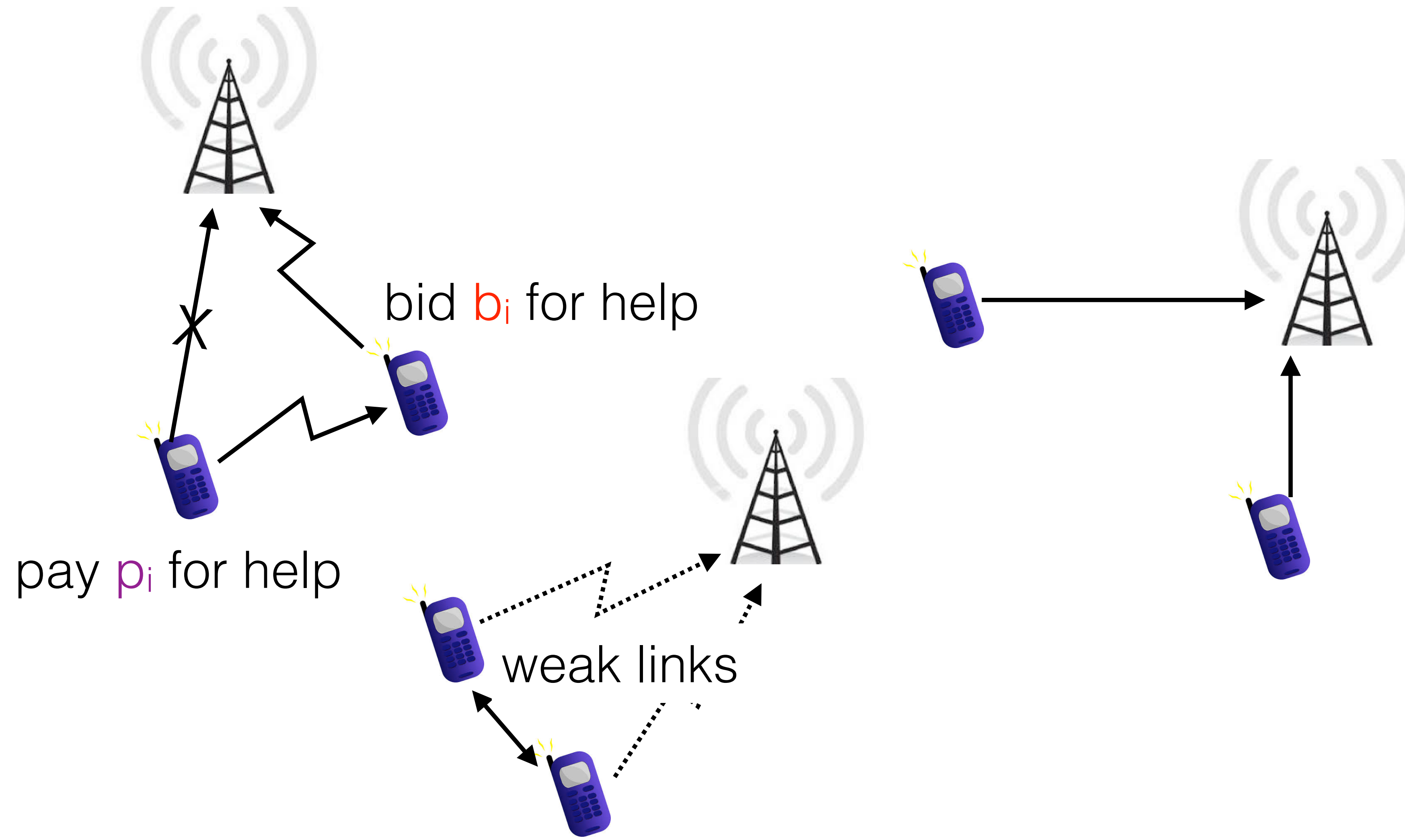
mechanism to avoid cheating

ensure maximum throughput

subject to payment budget

unknown help opportunities

Modern Problem Device-2-Device Communication

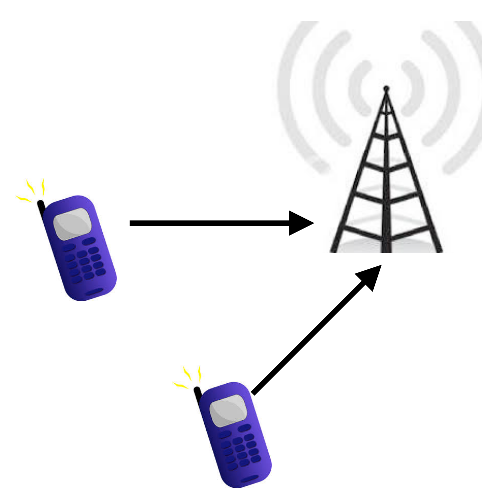


mechanism to avoid cheating

ensure maximum throughput

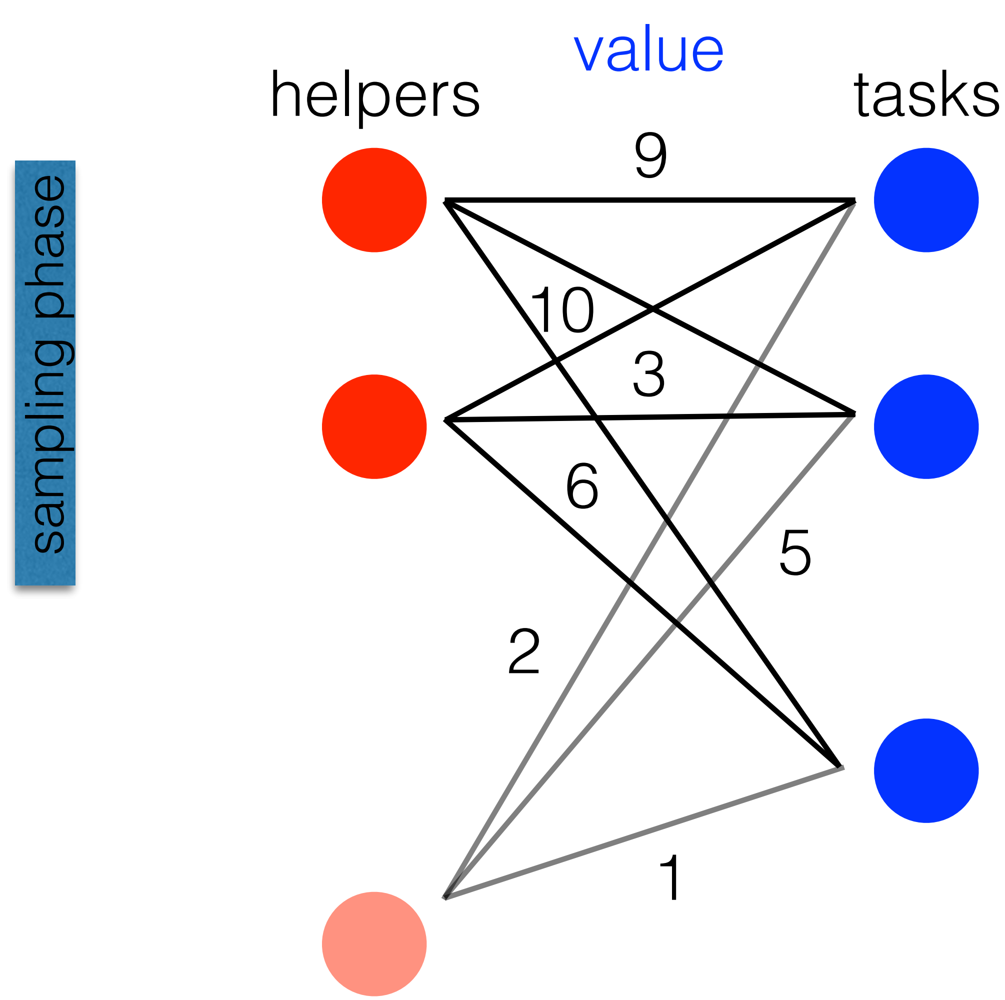
subject to payment budget

unknown help opportunities



Find optimal helper association and incentive rule that is truthful

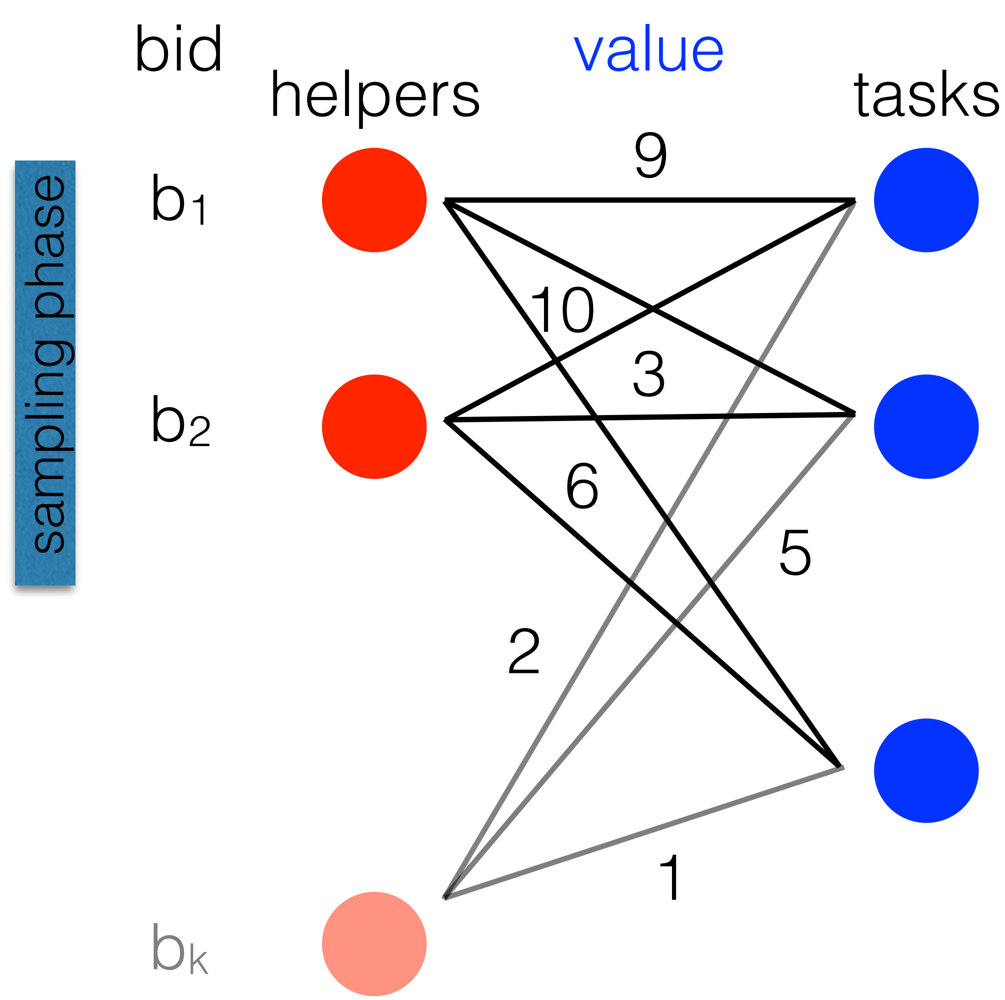
Possible Strategy for Our Problem



Objective: Truthful Matching with largest sum weight under a budget constraint

Possible Strategy for Our Problem

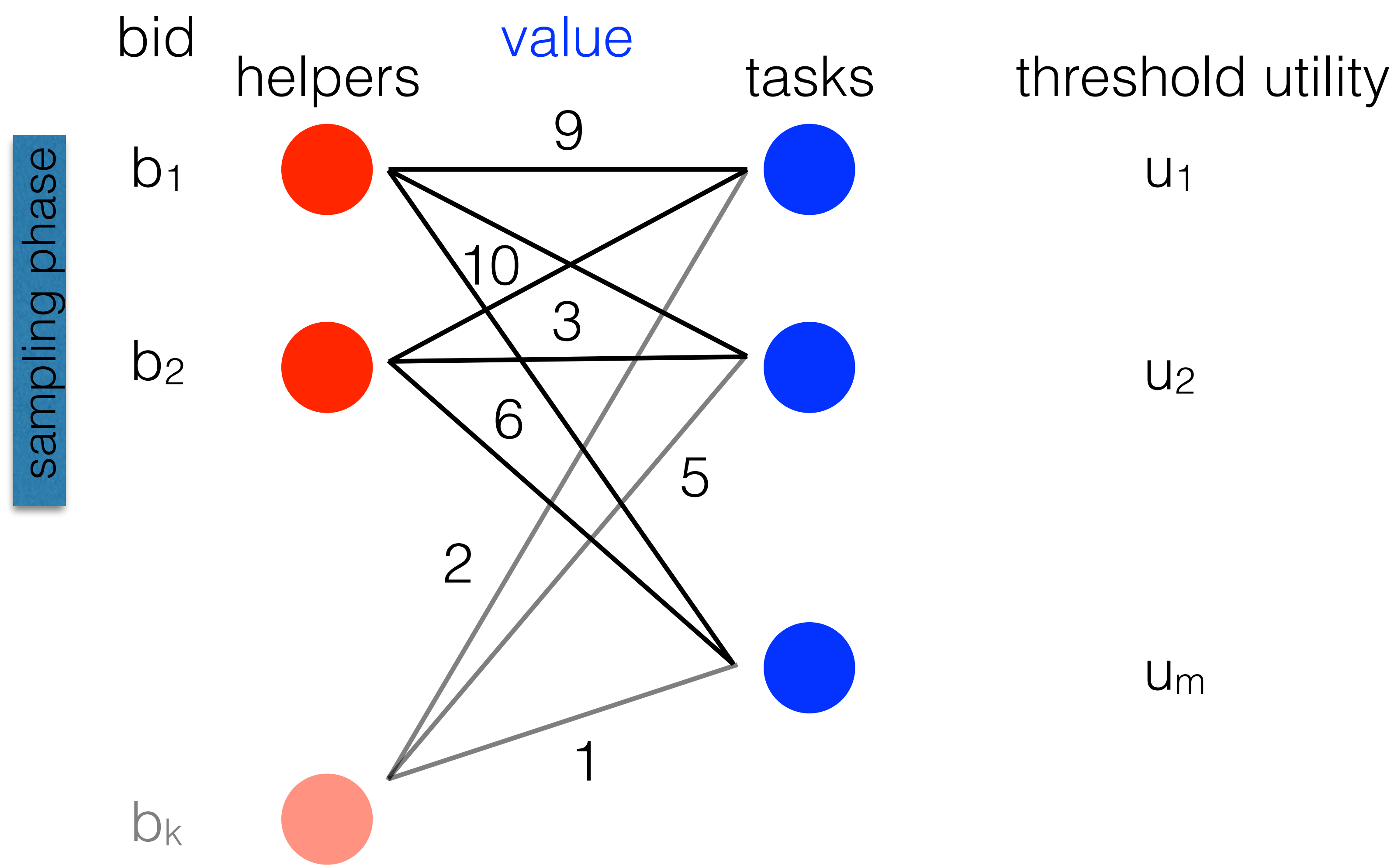
Offline Soln



Objective: Truthful Matching with largest sum weight under a budget constraint

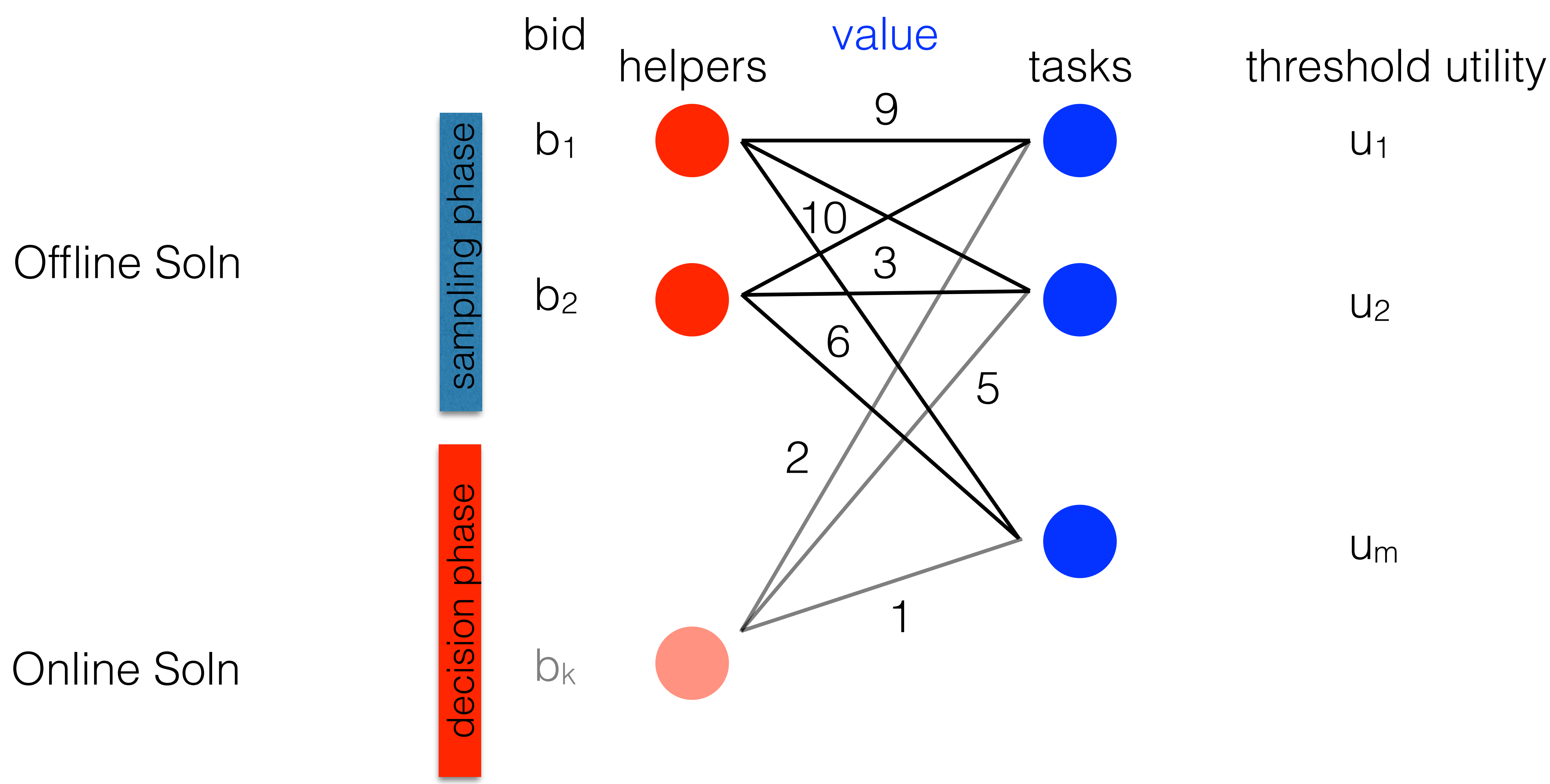
Possible Strategy for Our Problem

Offline Soln



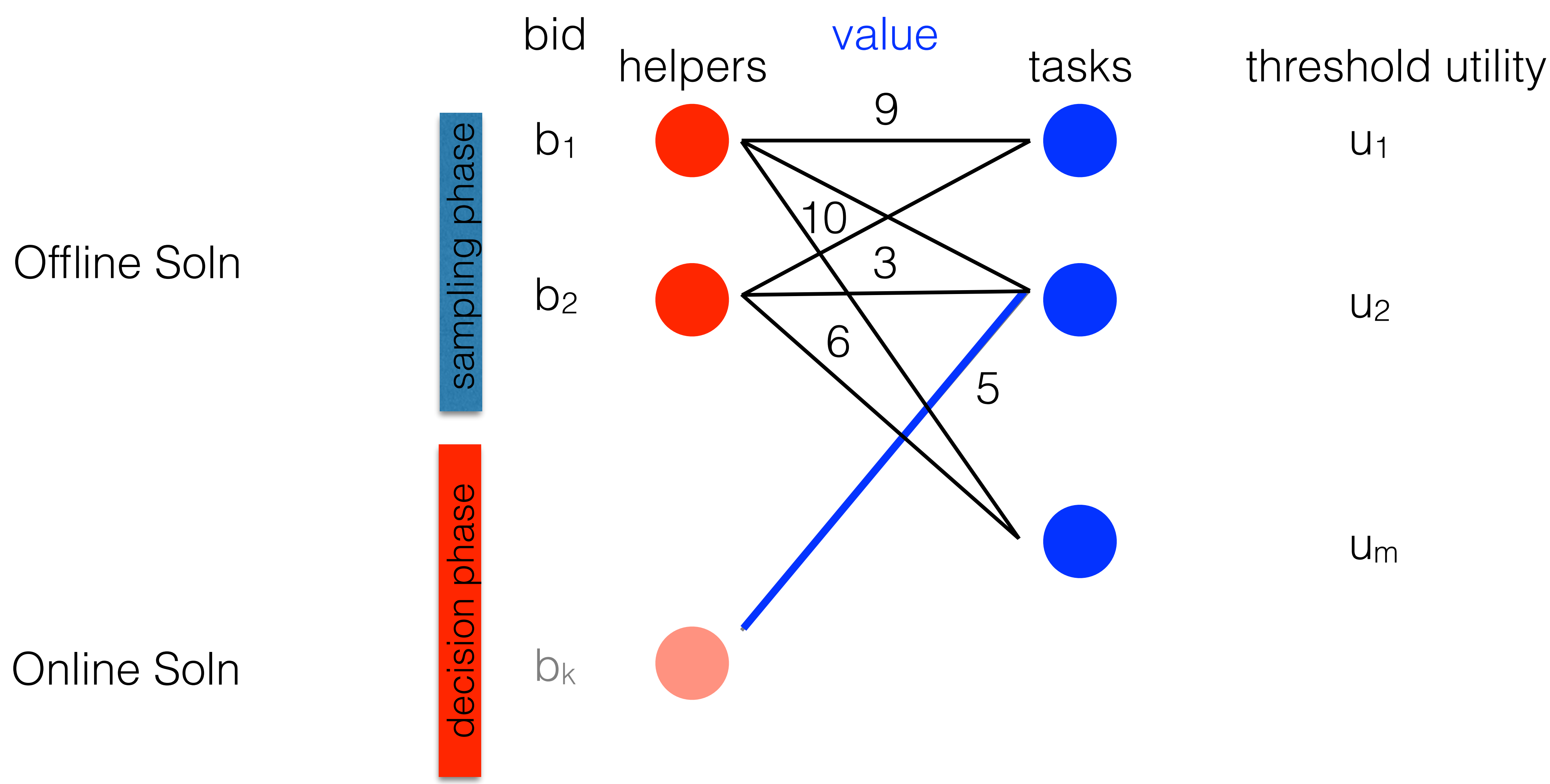
Objective: Truthful Matching with largest sum weight under a budget constraint

Possible Strategy for Our Problem



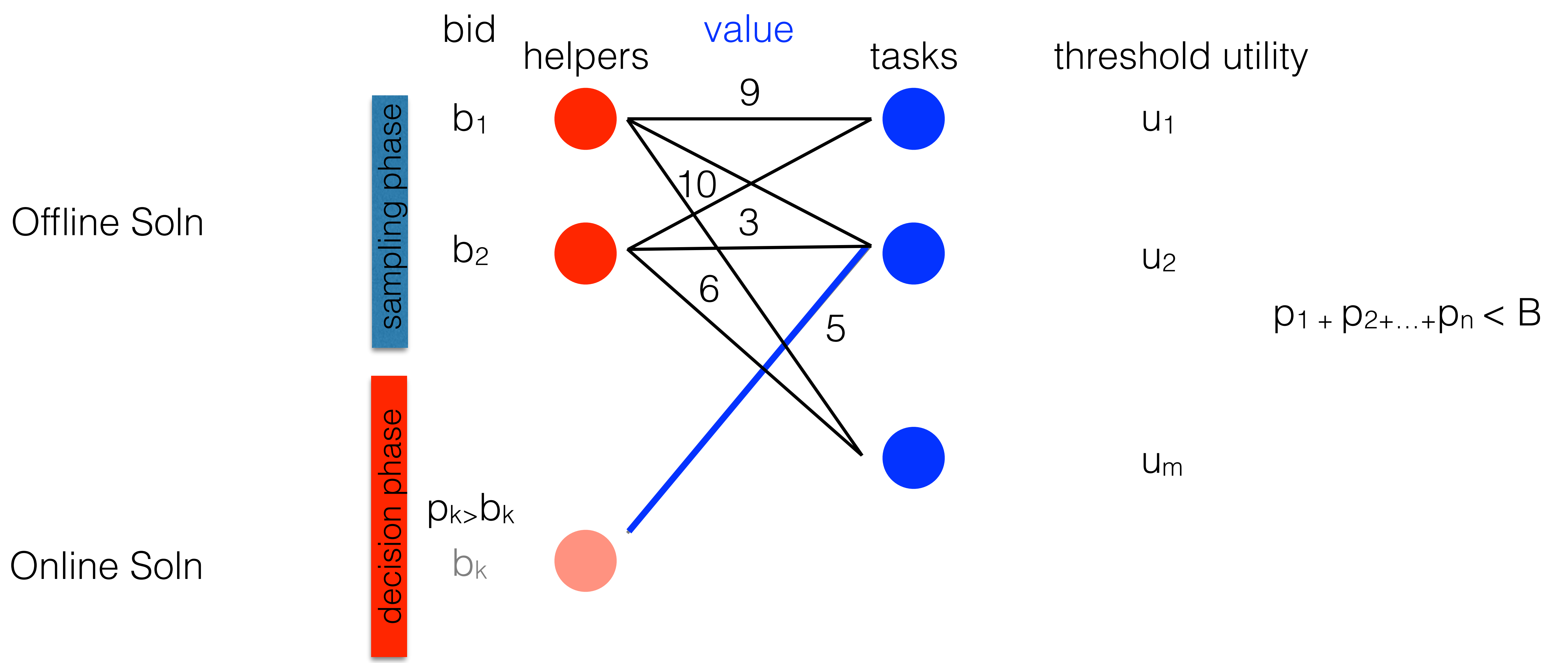
Objective: Truthful Matching with largest sum weight under a budget constraint

Possible Strategy for Our Problem



Objective: Truthful Matching with largest sum weight under a budget constraint

Possible Strategy for Our Problem



Objective: Truthful Matching with largest sum weight under a budget constraint

Known Offline Algorithm with Truthfulness

Known Offline Algorithm with Truthfulness

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Known Offline Algorithm with Truthfulness

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm *[Goel, Nikzad, Singla ‘ 14]*

M be greedy matching over G

Remove any edge e with $\frac{b(e)}{v(e)} > \frac{B}{u(M)}$

$G = G \backslash \{e\}$ **go to Step 1**

Known Offline Algorithm with Truthfulness

Problem

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

All edges part of big graph and not necessarily in small graph

Algorithm *[Goel, Nikzad, Singla ' 14]*

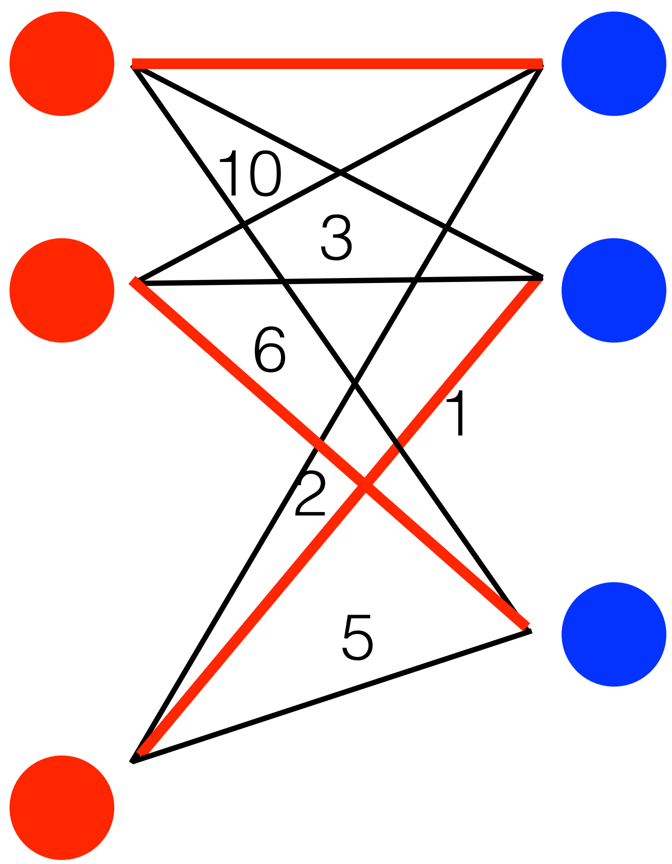
M be greedy matching over G

Remove any edge e with $\frac{b(e)}{v(e)} > \frac{B}{u(M)}$

$G = G \setminus \{e\}$ **go to Step 1**

Known Offline Algorithm with Truthfulness

Problem



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

All edges part of big graph and not necessarily in small graph

Algorithm *[Goel, Nikzad, Singla ' 14]*

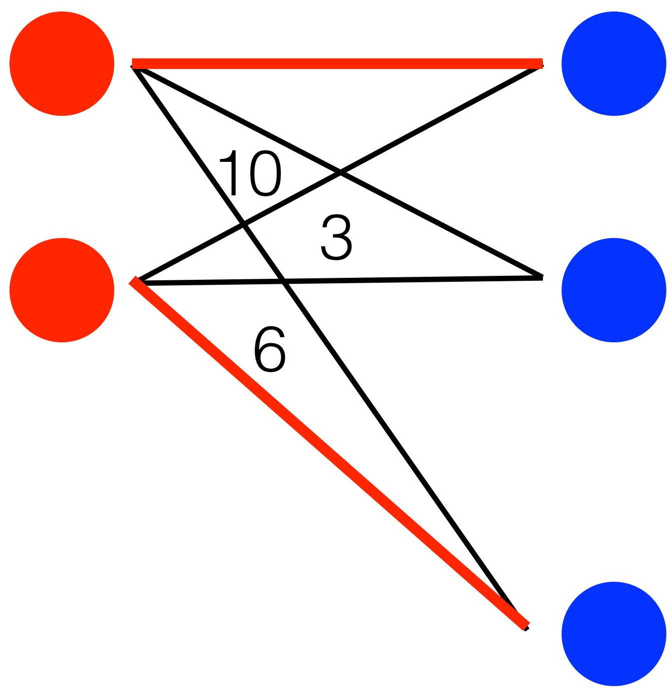
M be greedy matching over G

Remove any edge e with $\frac{b(e)}{v(e)} > \frac{B}{u(M)}$

$G = G \backslash \{e\}$ **go to Step 1**

Known Offline Algorithm with Truthfulness

Problem



All edges part of big graph and not necessarily in small graph

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm *[Goel, Nikzad, Singla ' 14]*

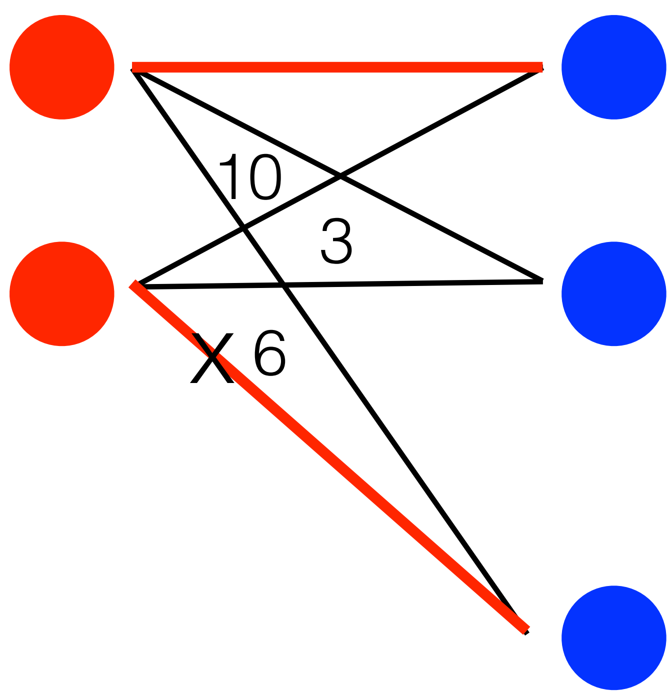
M be greedy matching over G

Remove any edge e with $\frac{b(e)}{v(e)} > \frac{B}{u(M)}$

$G = G \backslash \{e\}$ **go to Step 1**

Known Offline Algorithm with Truthfulness

Problem



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm *[Goel, Nikzad, Singla ' 14]*

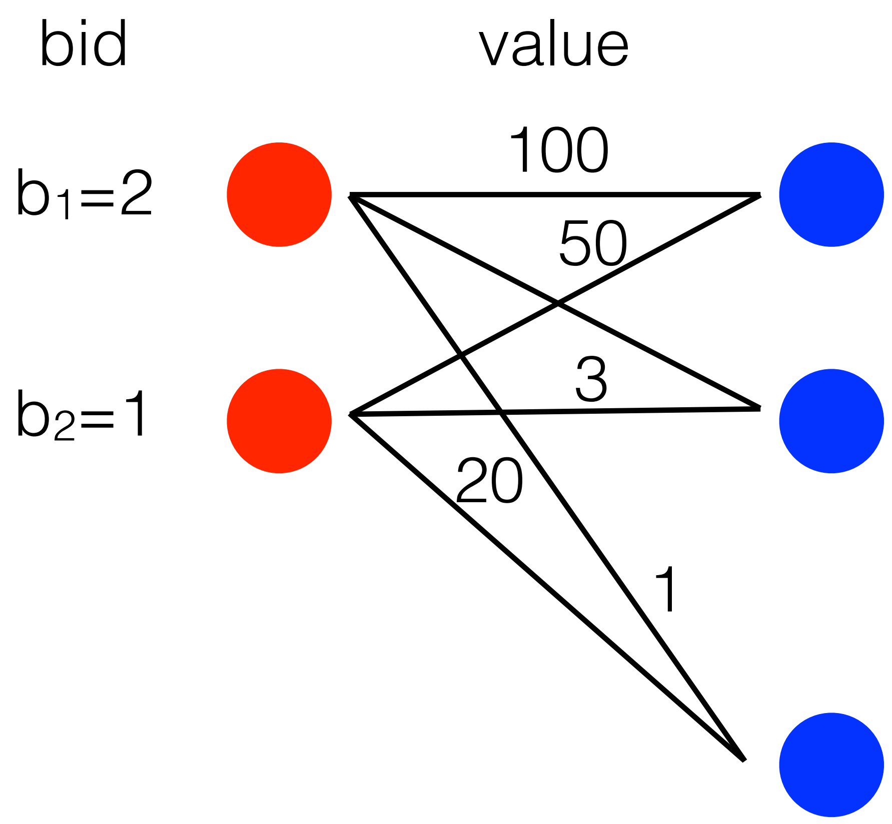
M be greedy matching over G

Remove any edge e with $\frac{b(e)}{v(e)} > \frac{B}{u(M)}$

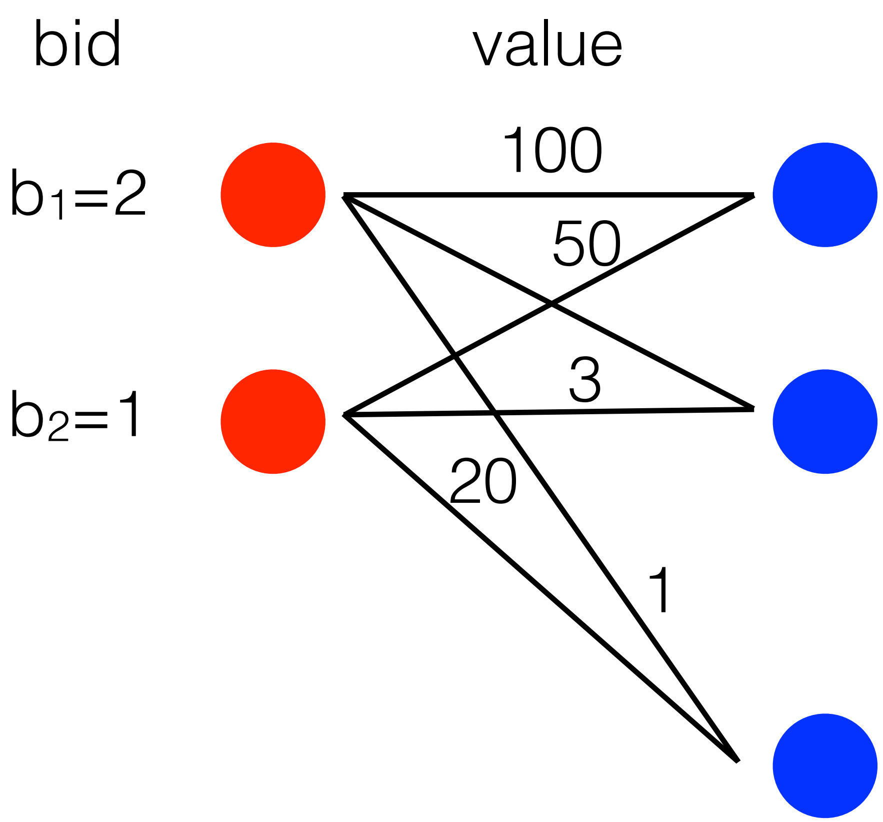
$G = G \backslash \{e\}$ **go to Step 1**

All edges part of big graph and not necessarily in small graph

Our Offline Algorithm **OFF**

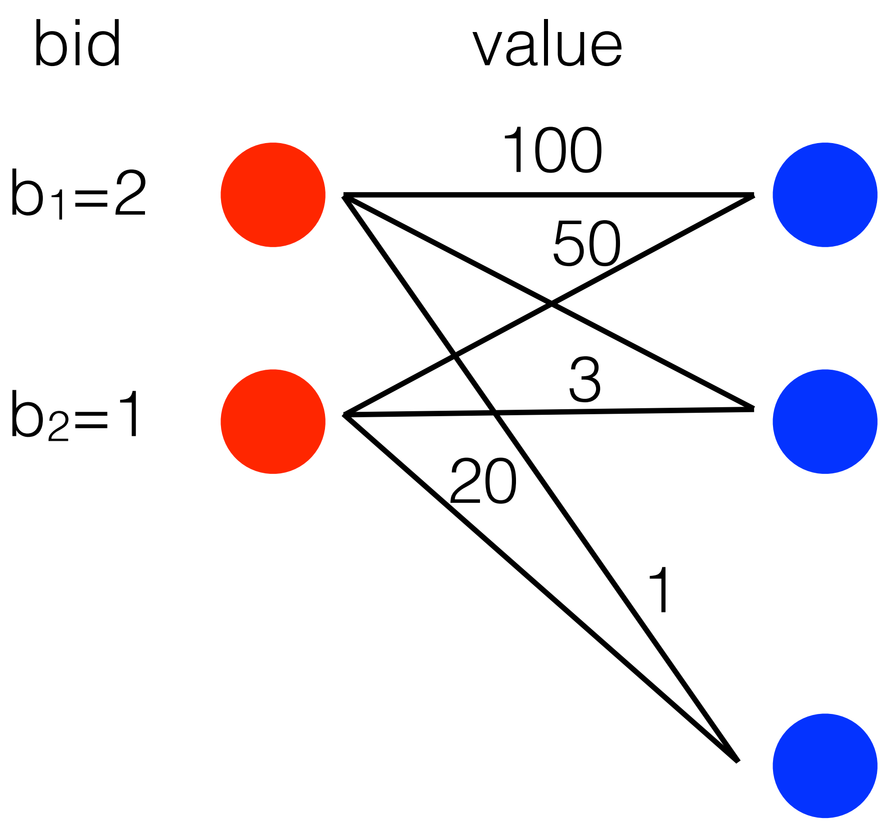


Our Offline Algorithm `OFF`



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

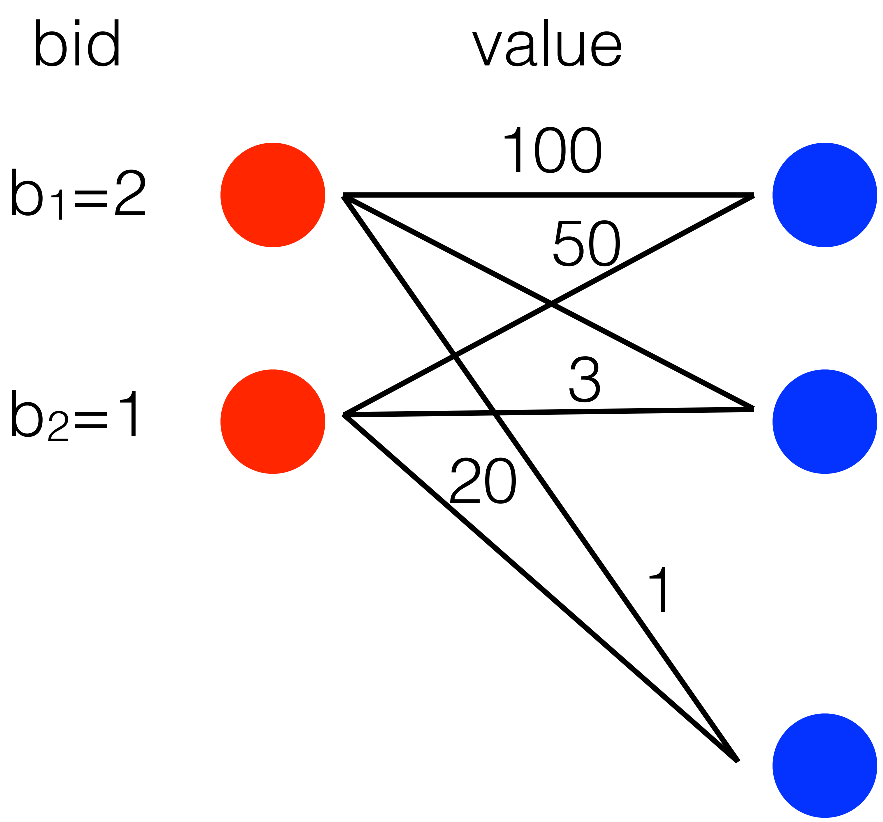
Our Offline Algorithm `OFF`



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm

Our Offline Algorithm **OFF**

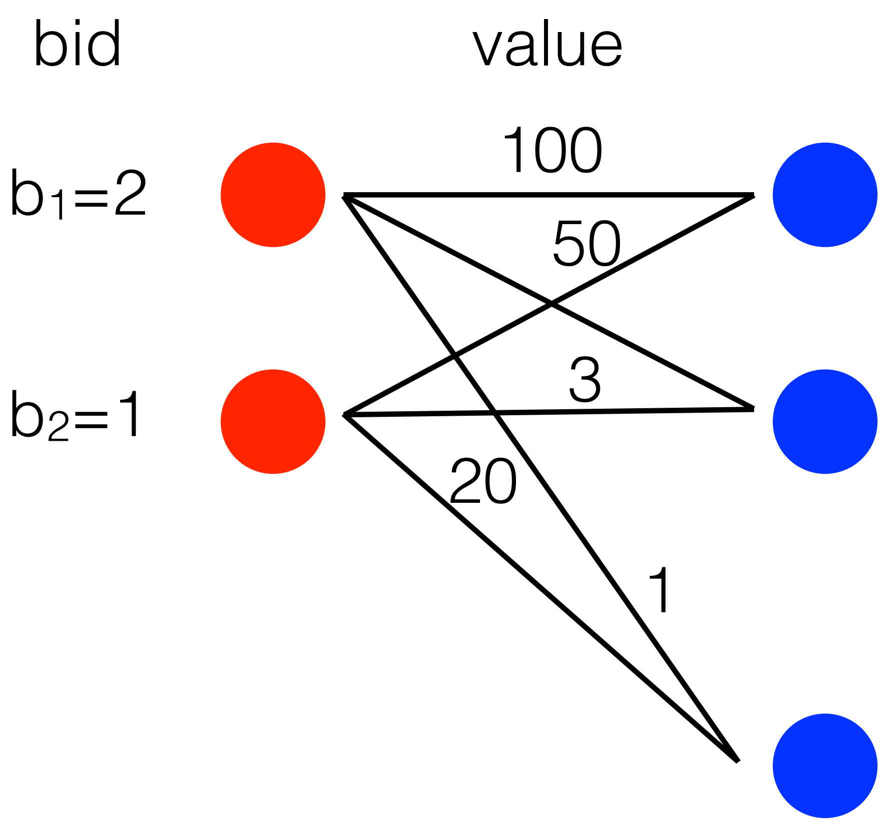


bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

Our Offline Algorithm `OFF`



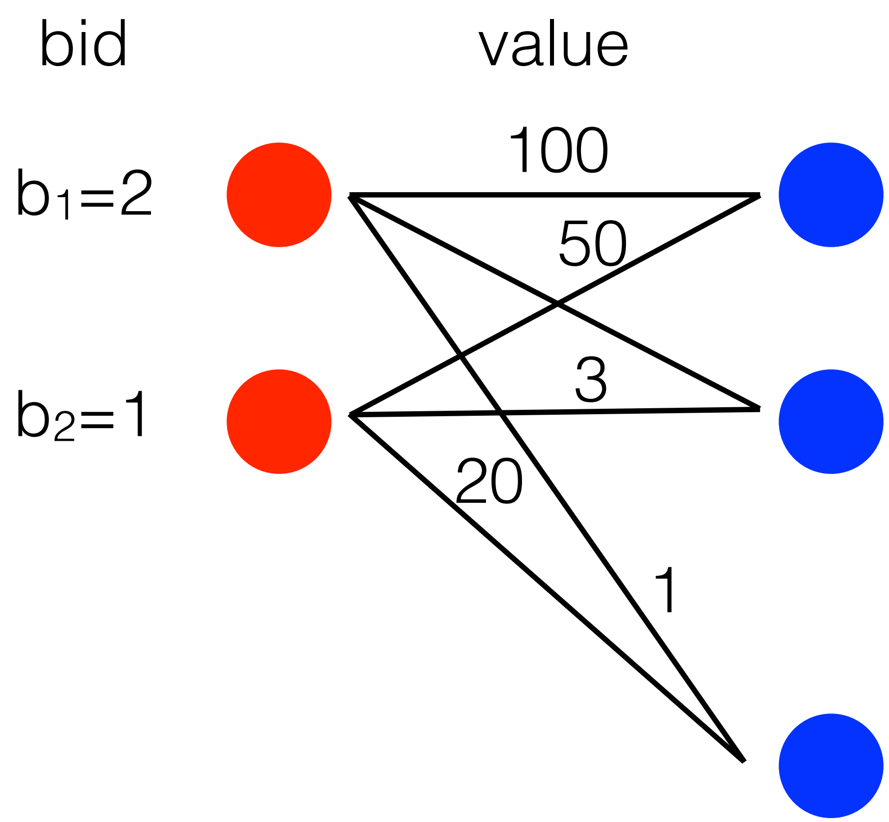
bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Our Offline Algorithm `OFF`



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

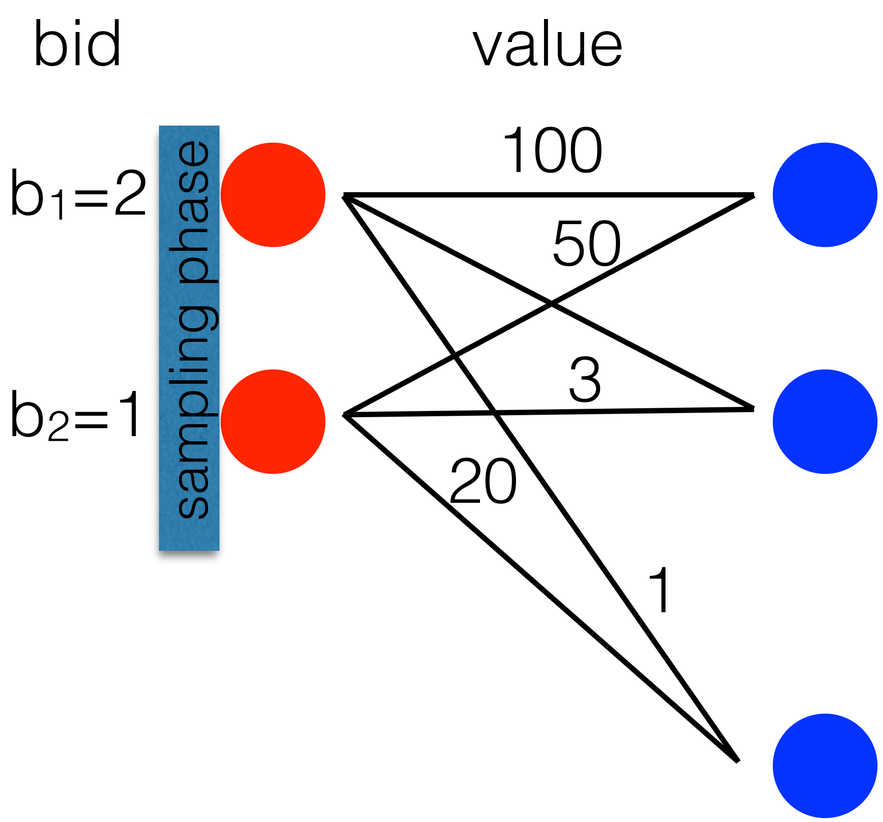
Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Our Offline Algorithm `OFF`



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

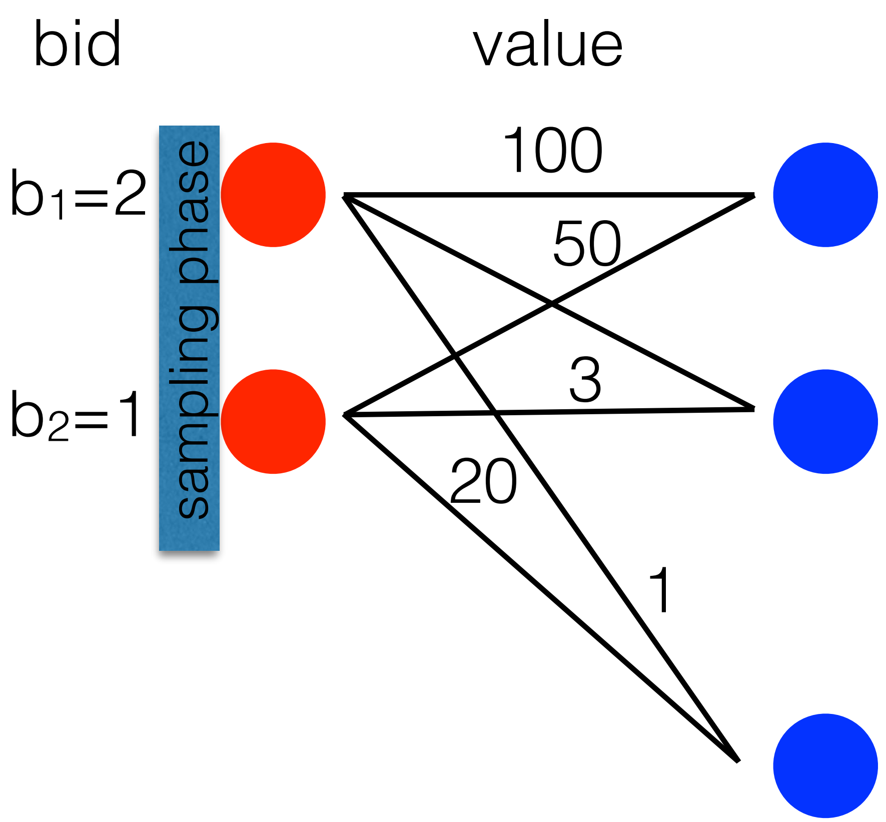
Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Our Offline Algorithm `OFF`



$\gamma = 0.1$

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

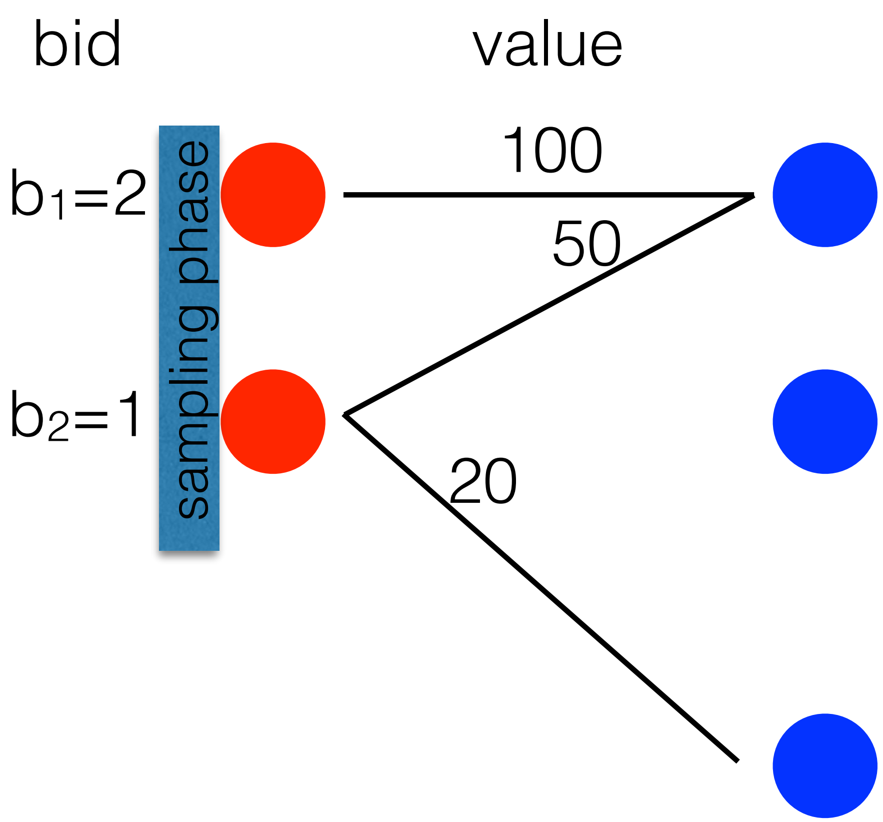
Algorithm

good Graph $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$M(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Our Offline Algorithm `OFF`



$\gamma = 0.1$

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

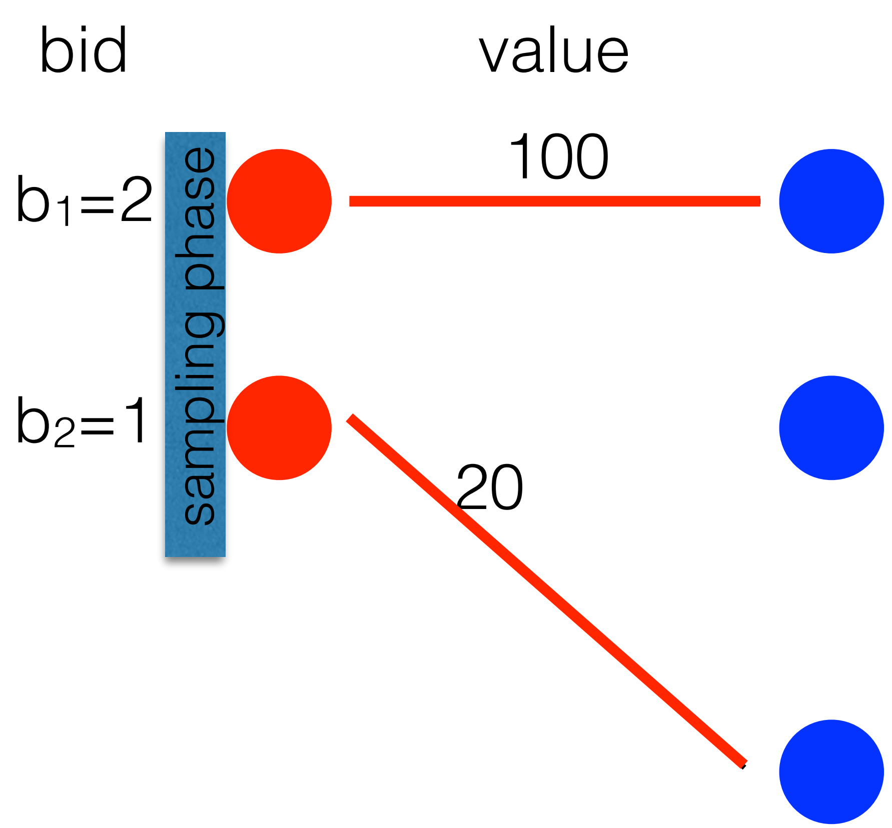
Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \le B$

Our Offline Algorithm **OFF**



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

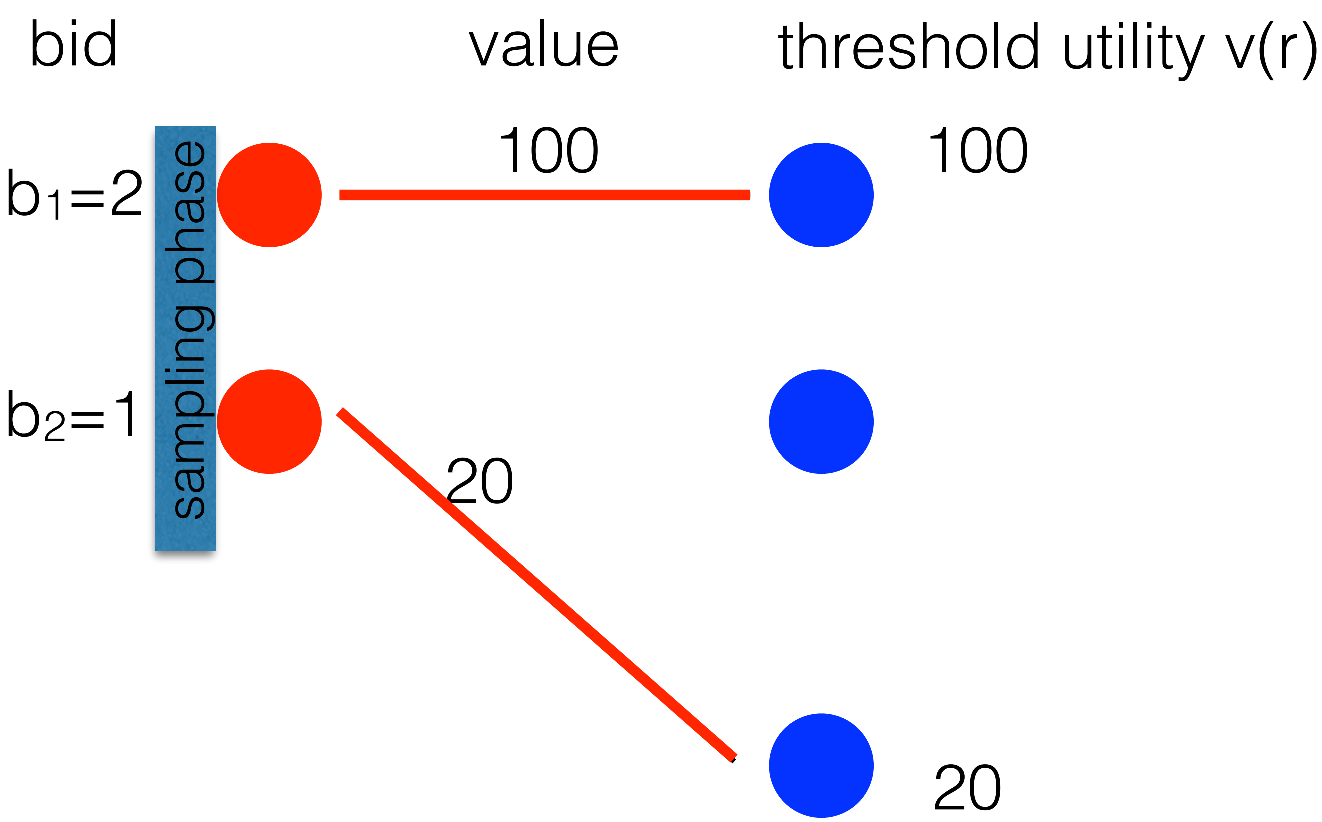
Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Our Offline Algorithm $\texttt{OFF}$



utility threshold v(r) of each blue node
to be its value in offline Matching

bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

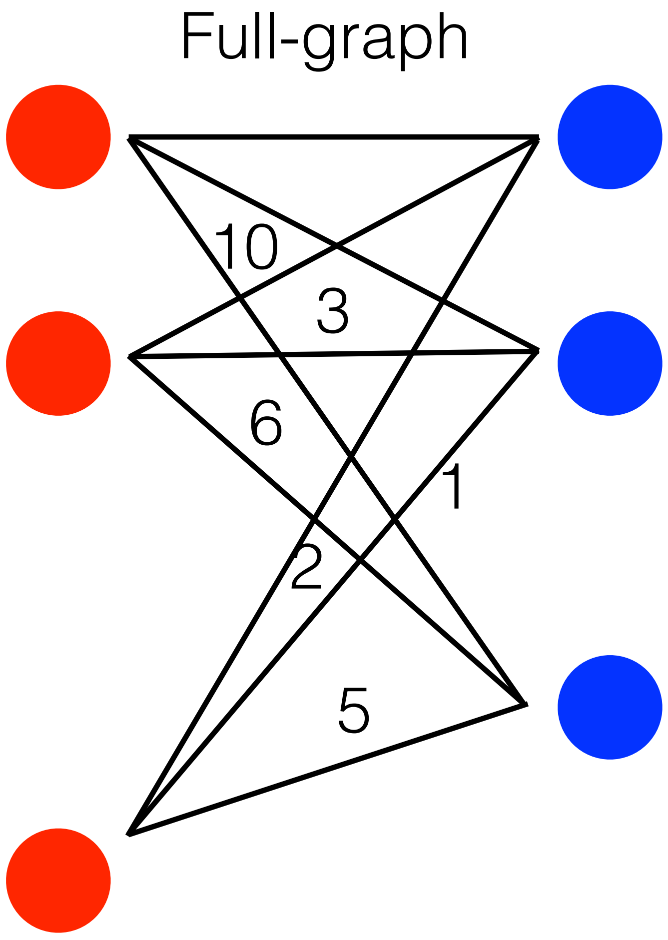
$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

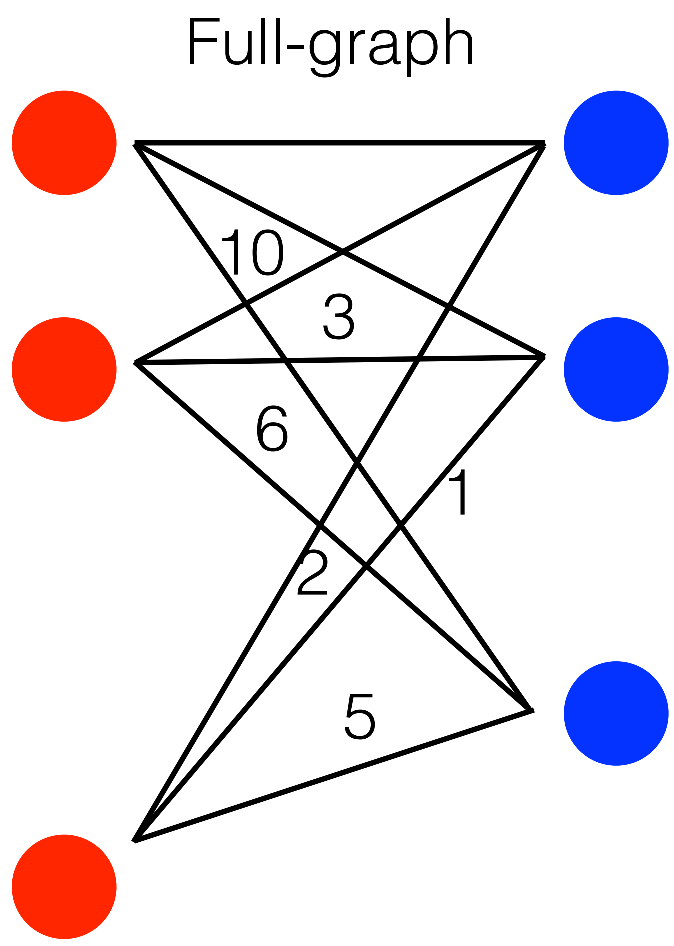
Claim : $\texttt{OFF}$ is 12 - approximate in expectation

Properties of `OFF`

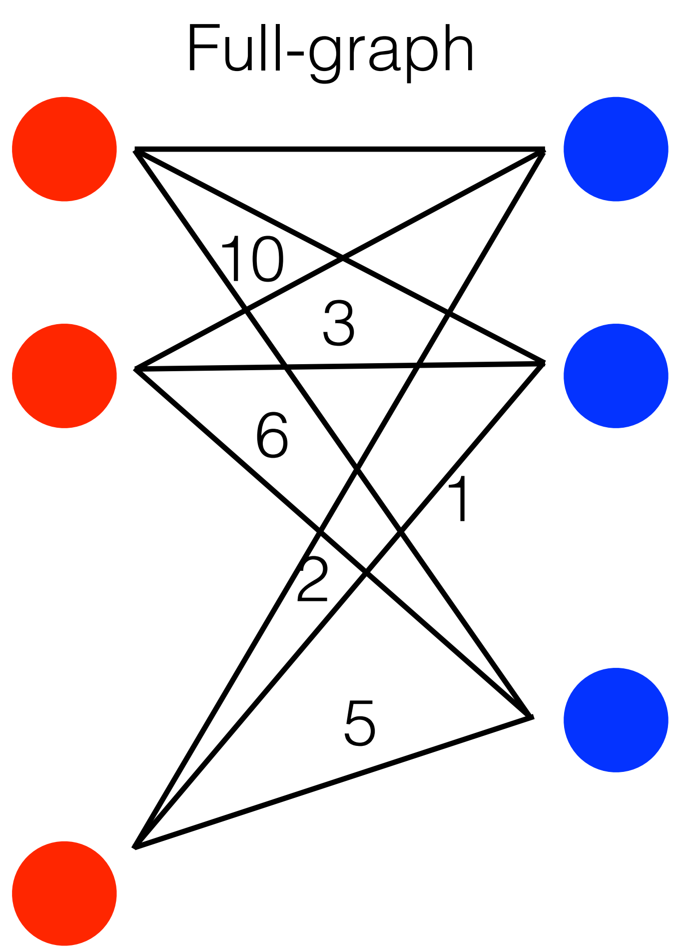
Properties of **OFF**



Properties of `OFF`



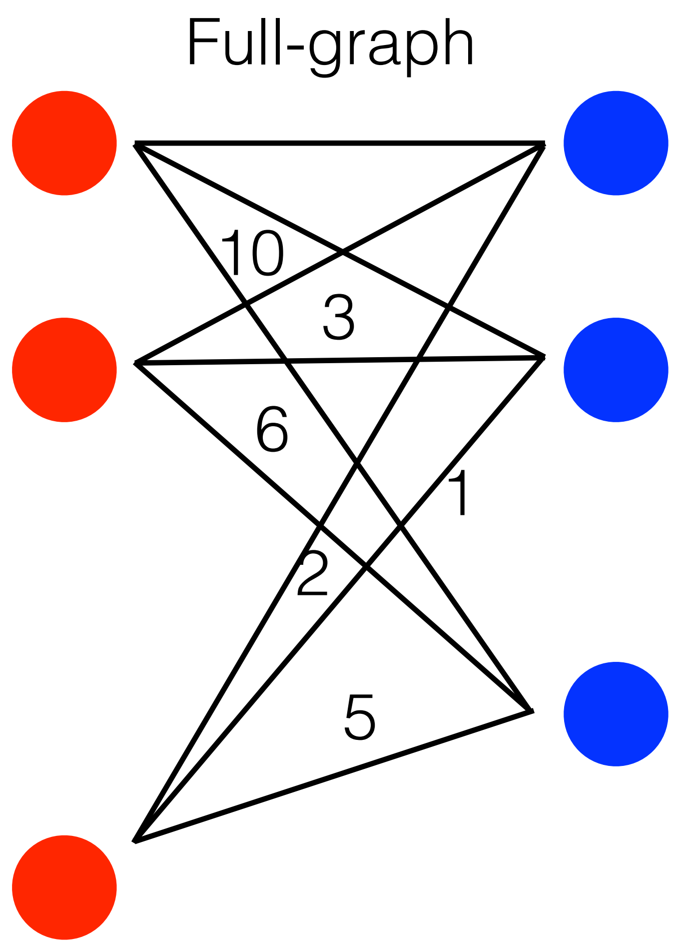
Properties of `OFF`



Greedy matching with edges

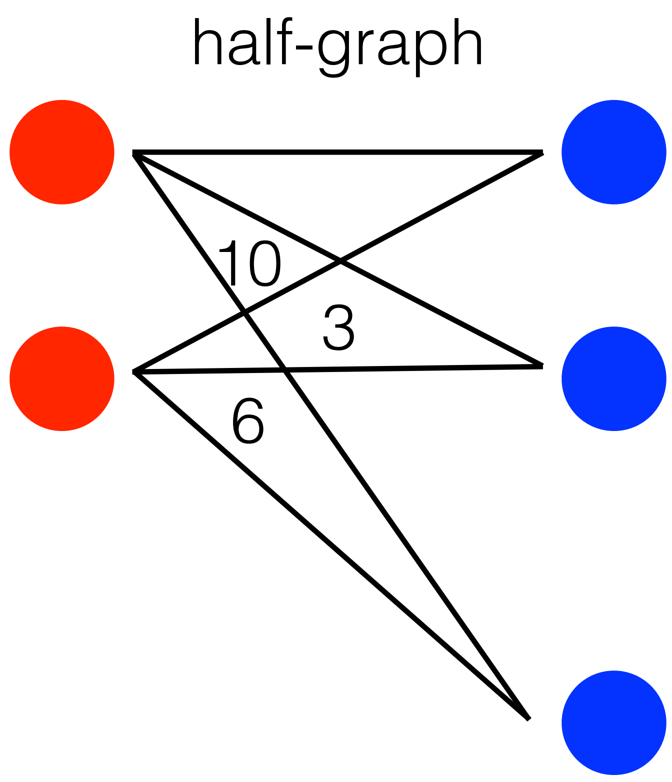
$$\boxed{\frac{b(e)}{v(e)}} < \gamma$$

Properties of `OFF`

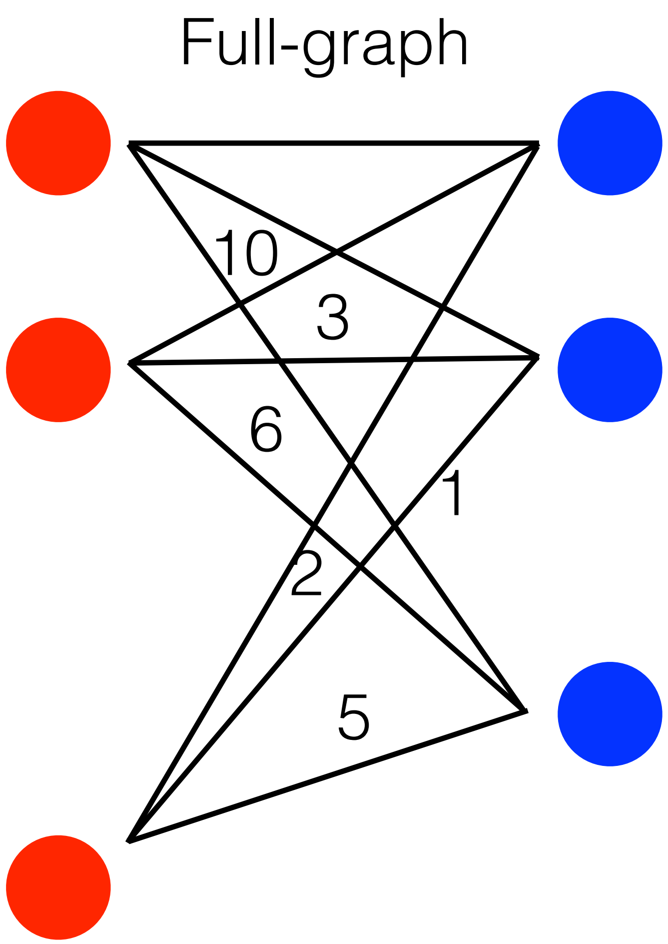


Greedy matching with edges

$$\frac{b(e)}{v(e)} < \gamma$$



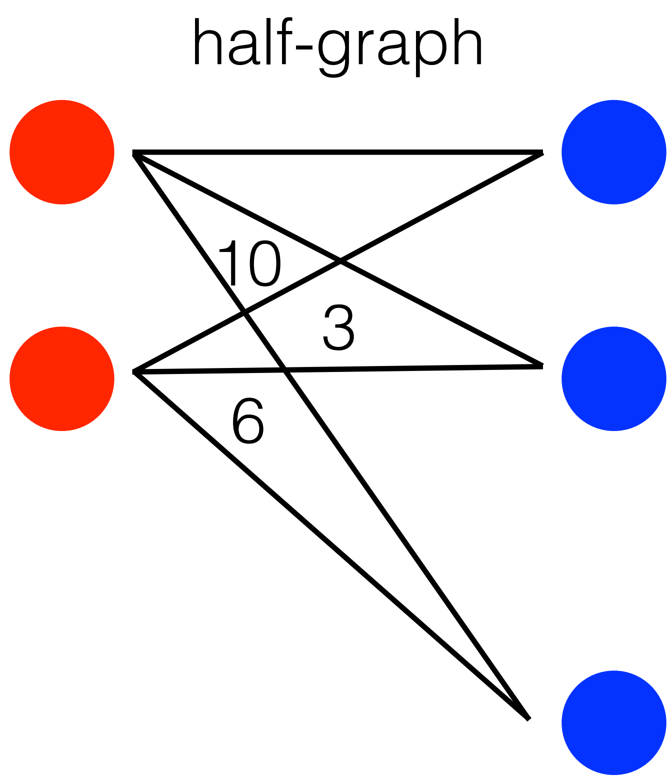
Properties of **OFF**



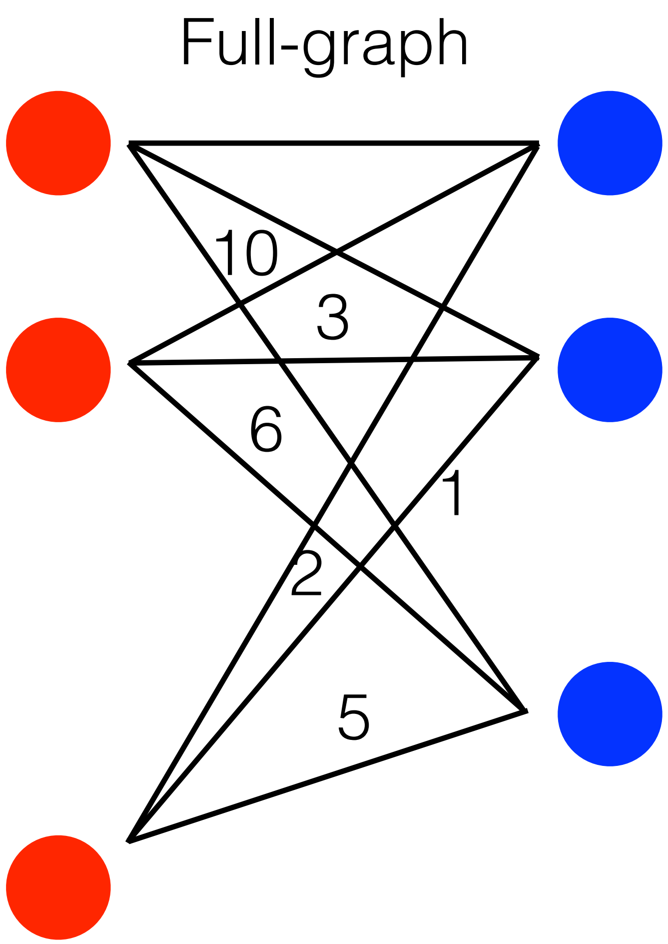
Greedy matching with edges

$$\frac{b(e)}{v(e)} < \gamma$$

$$\gamma_f \leq \gamma_h$$



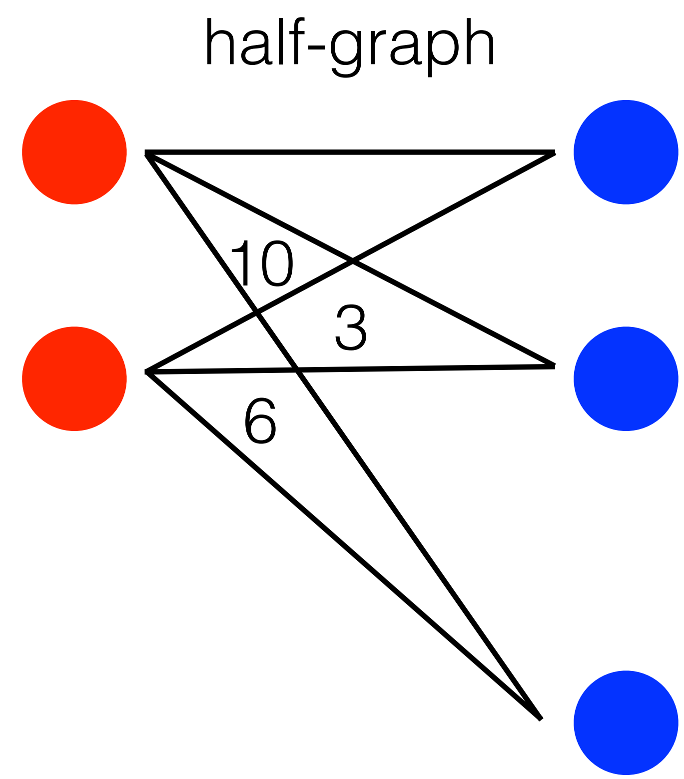
Properties of **OFF**



Greedy matching with edges

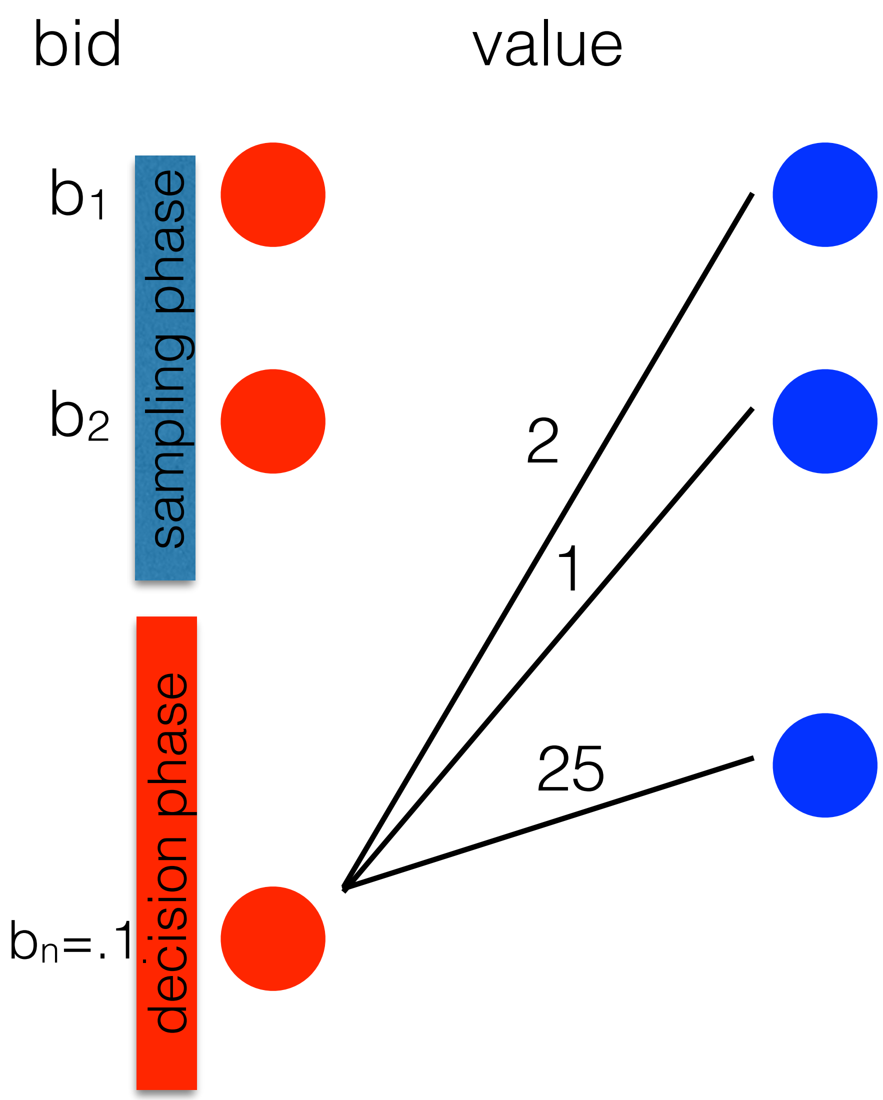
$$\frac{b(e)}{v(e)} < \gamma$$

$$\gamma_f \leq \gamma_h$$



So all eligible edges in big graph are also eligible in small graph

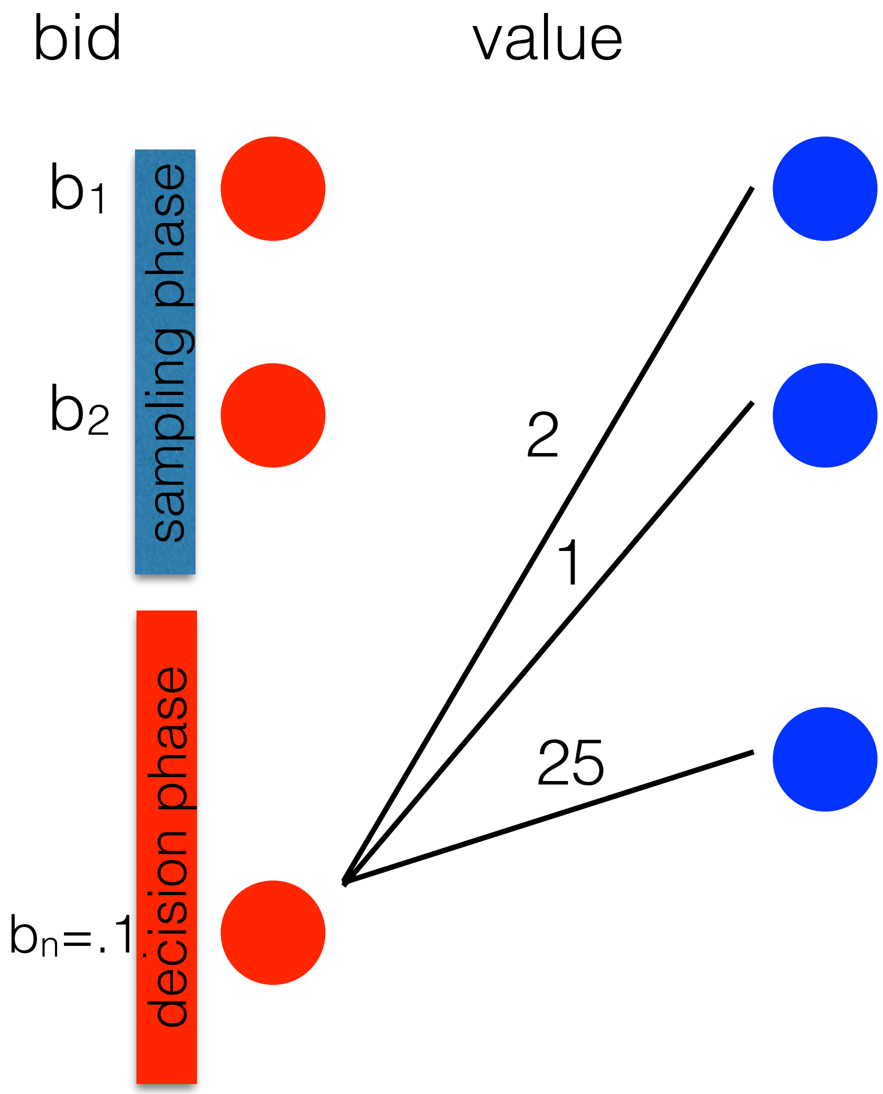
Algorithm - `ON`



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm - `ON`

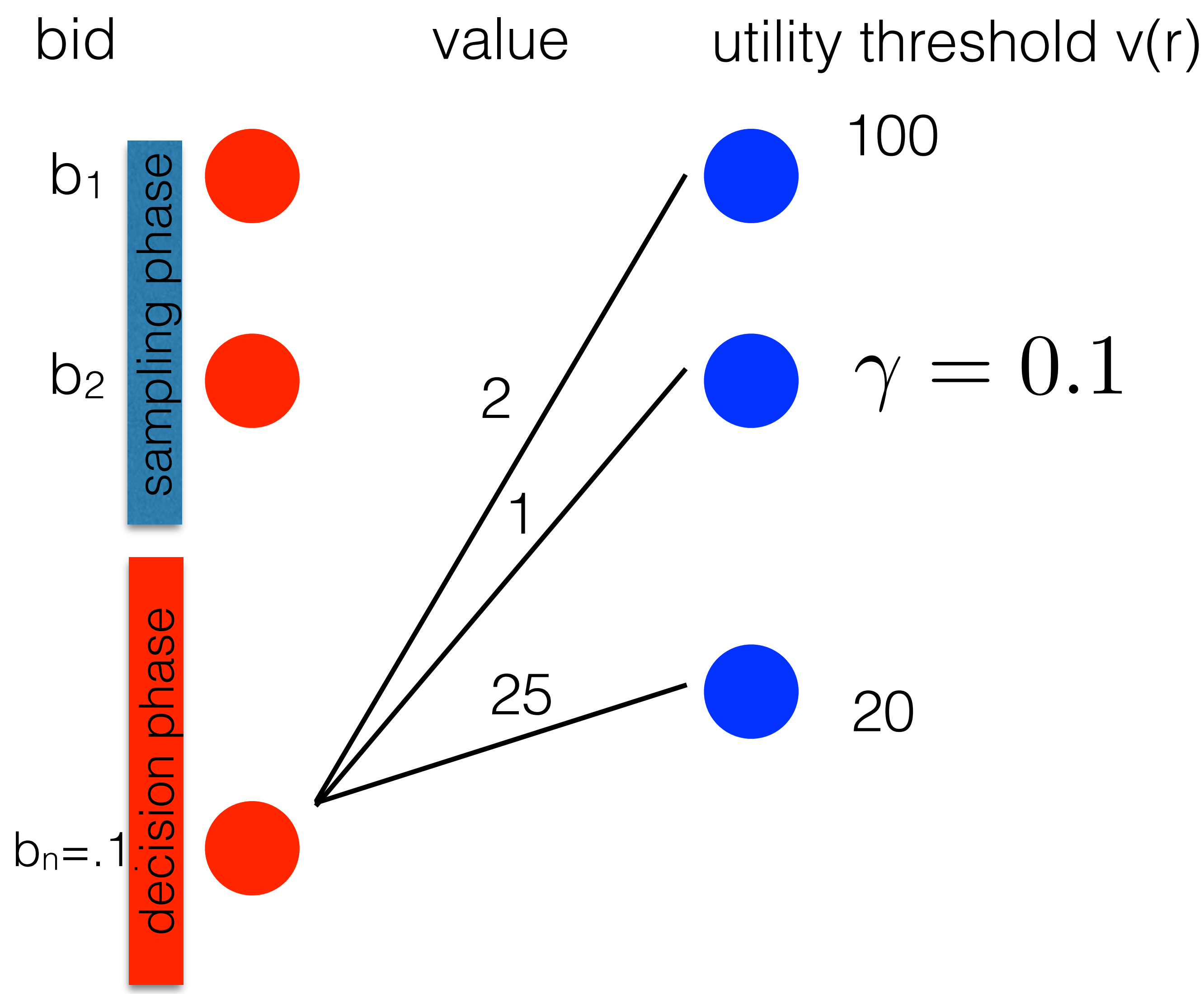
γ and threshold utility v(r) obtained from `OFF` **with budget** B/β



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm - ON

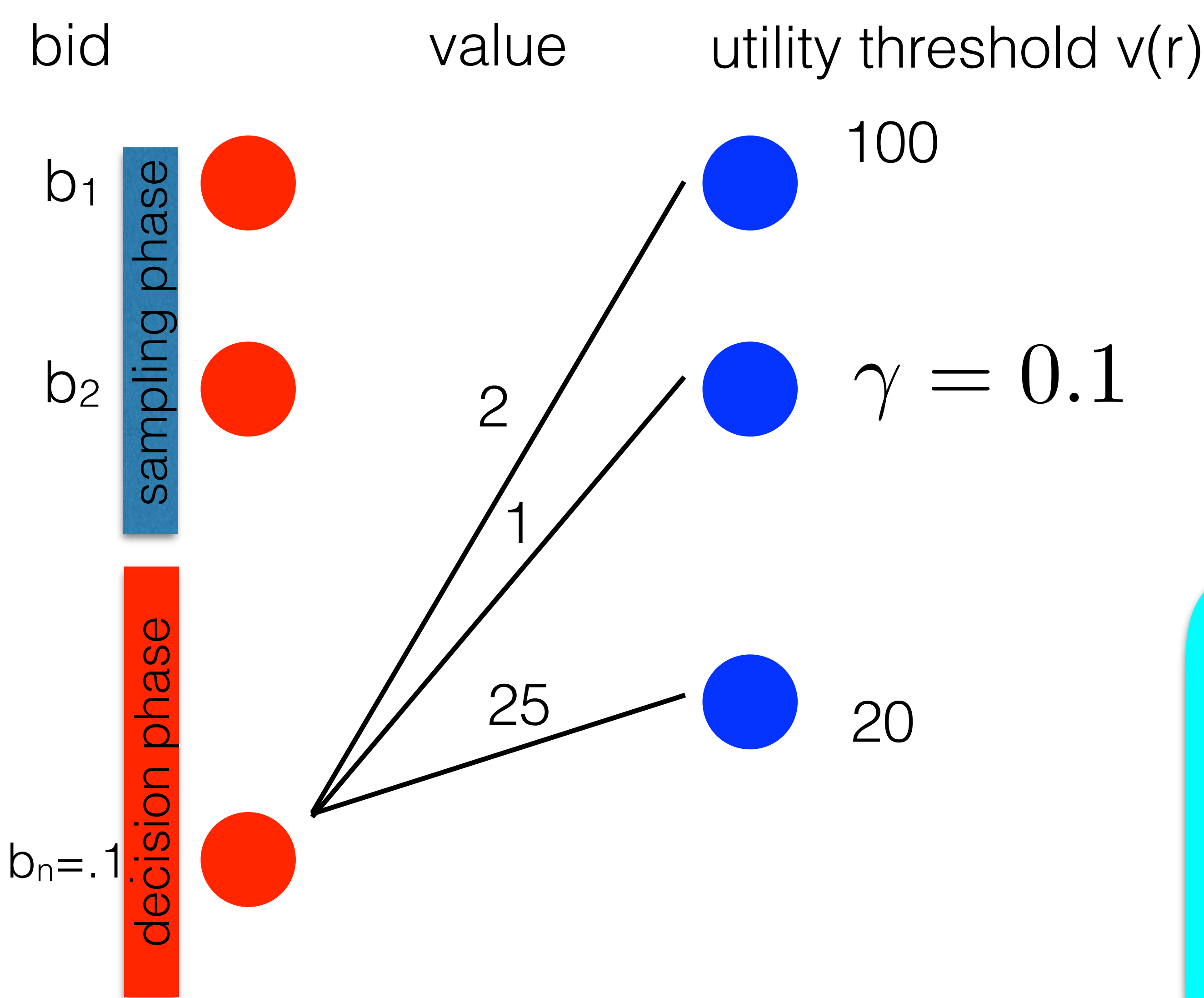
γ and threshold utility v(r) obtained from **OFF** **with budget** B/β



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

Algorithm - `ON`

γ and threshold utility v(r) obtained from `OFF` **with budget** B/β

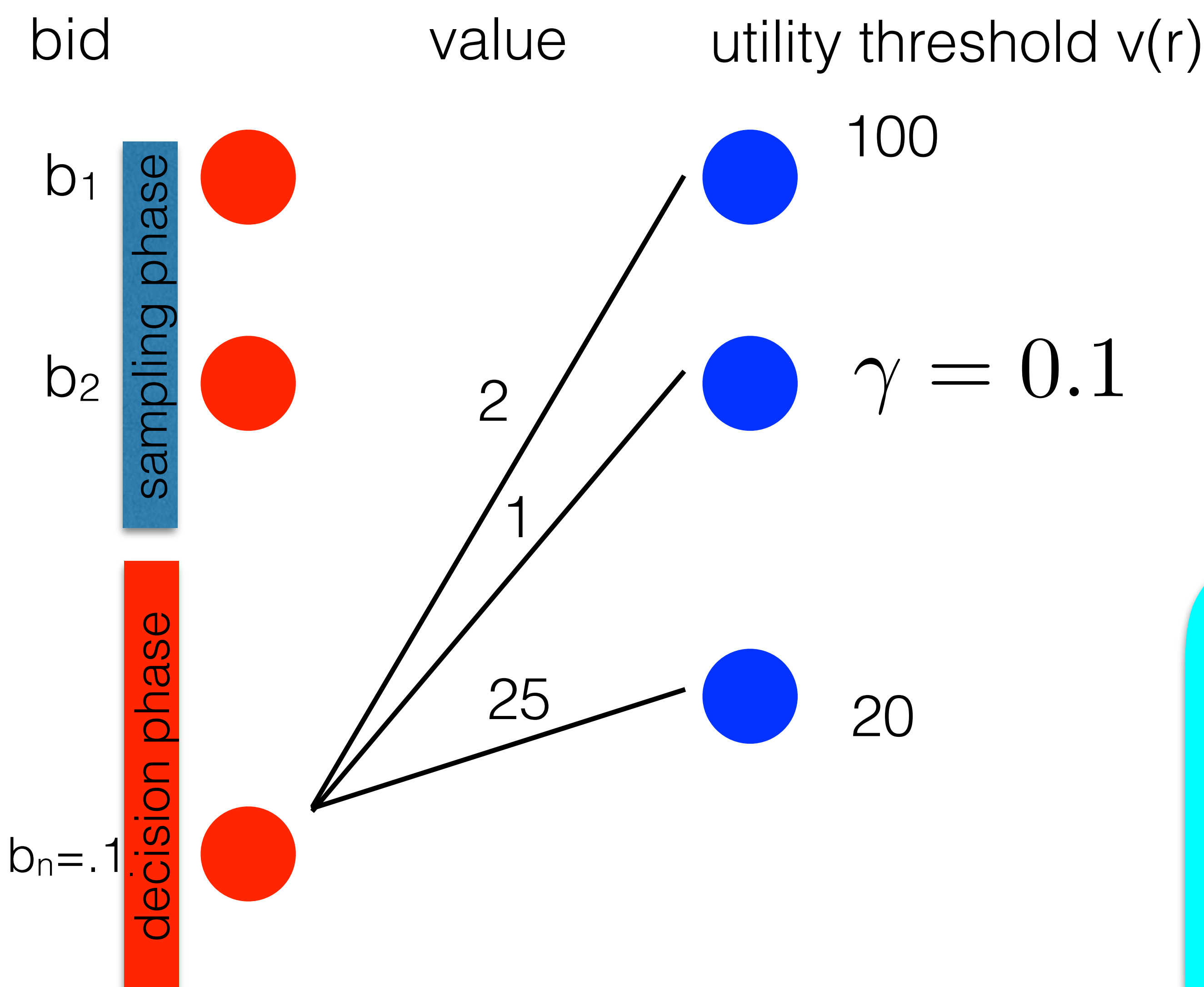


bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

on arrival of a new left vertex

Algorithm - ON

γ and threshold utility v(r) obtained from OFF **with budget** B/β



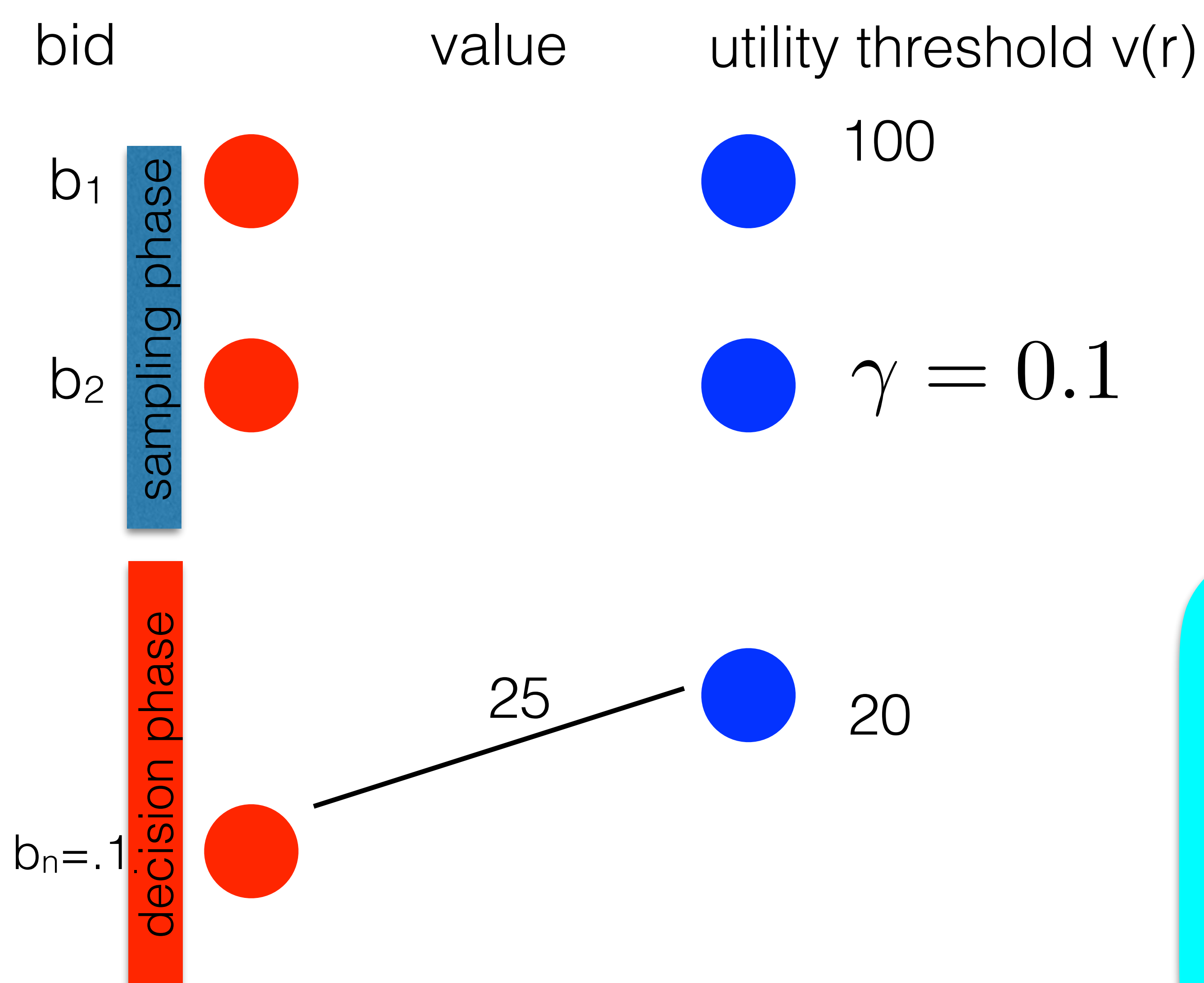
bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

Algorithm - `ON`

γ and threshold utility v(r) obtained from `OFF` **with budget** B/β



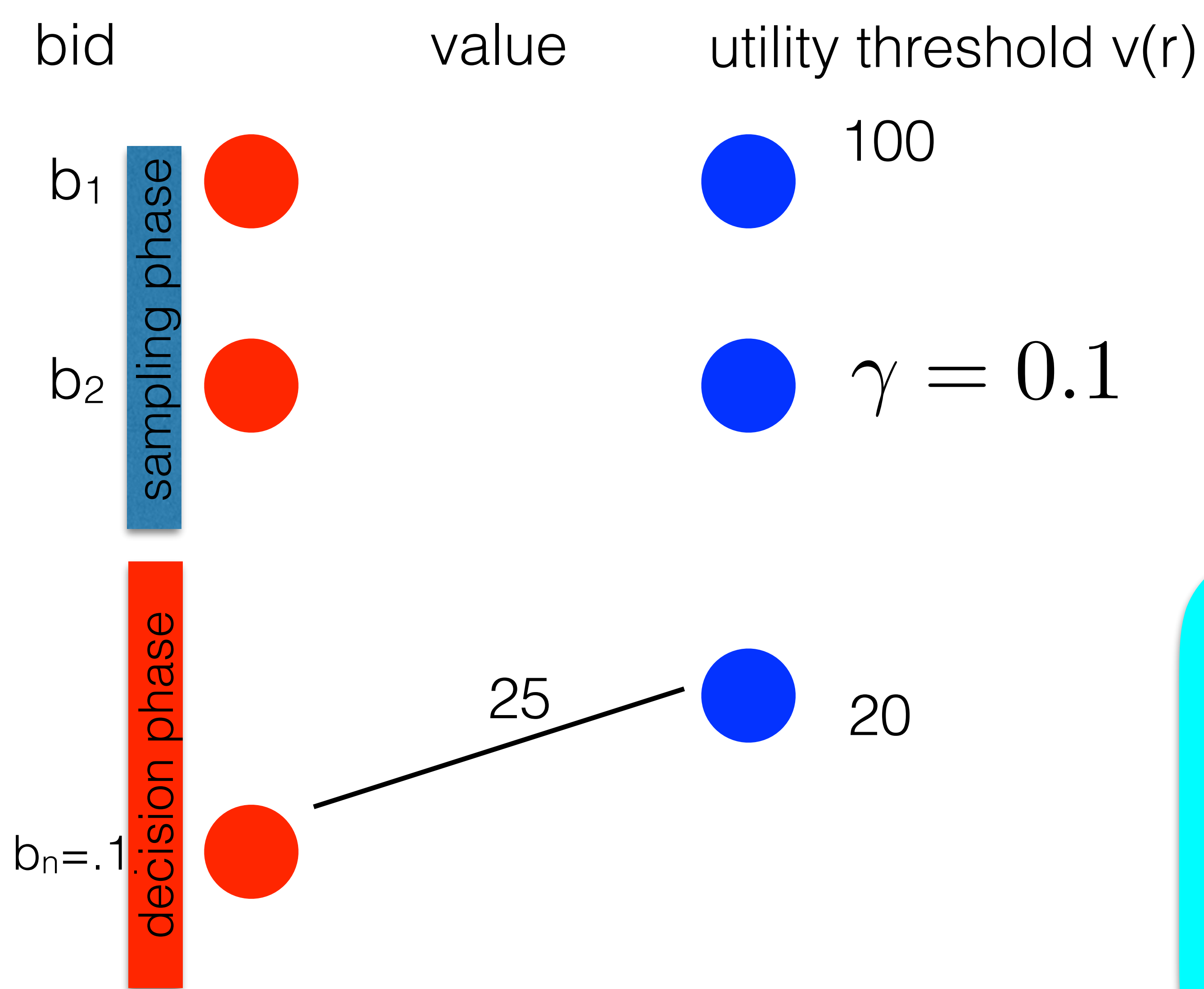
bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

Algorithm - ON

γ and threshold utility v(r) obtained from **OFF** **with budget** B/β



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

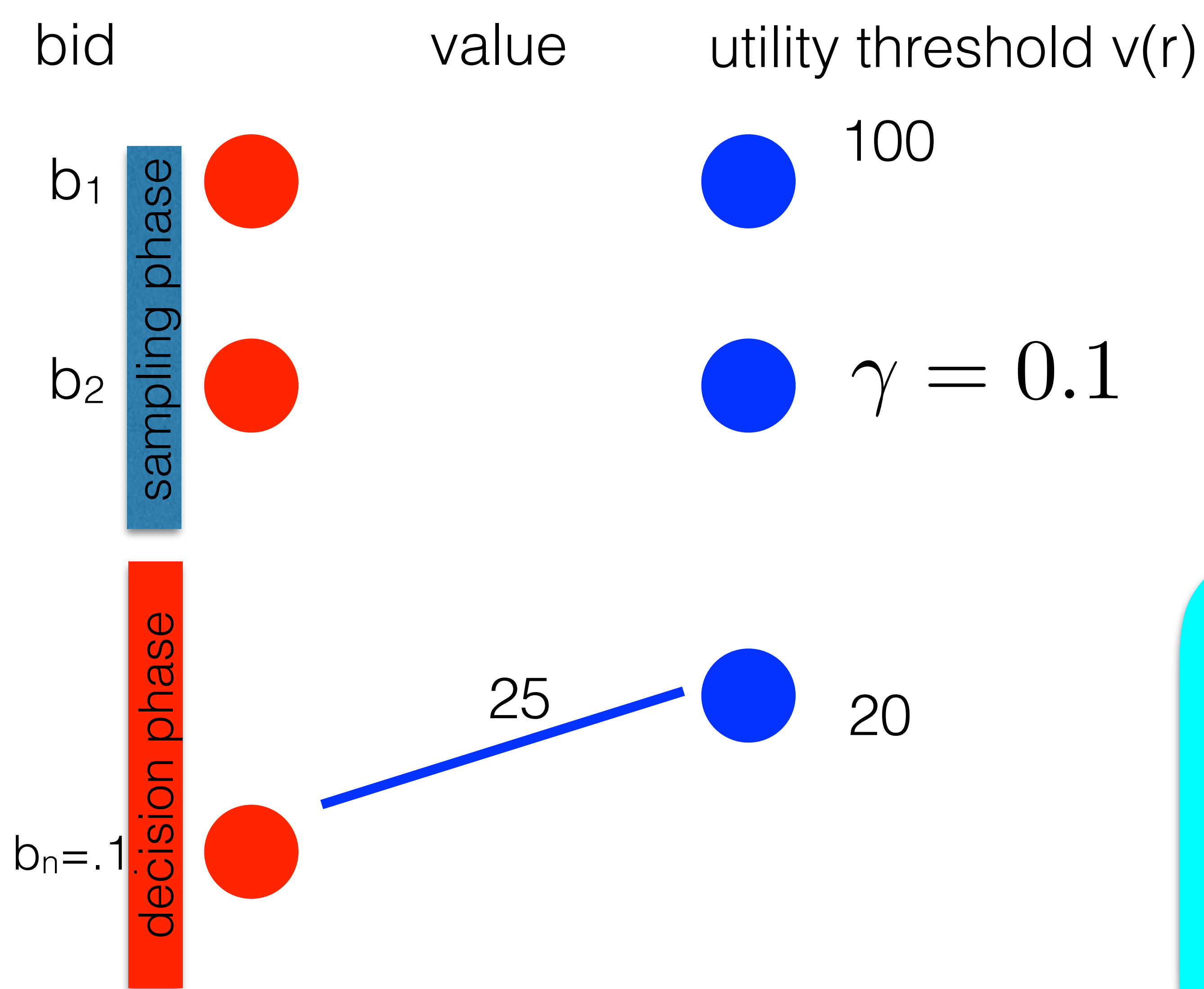
on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Algorithm - ON

γ and threshold utility v(r) obtained from **OFF** **with budget** B/β



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

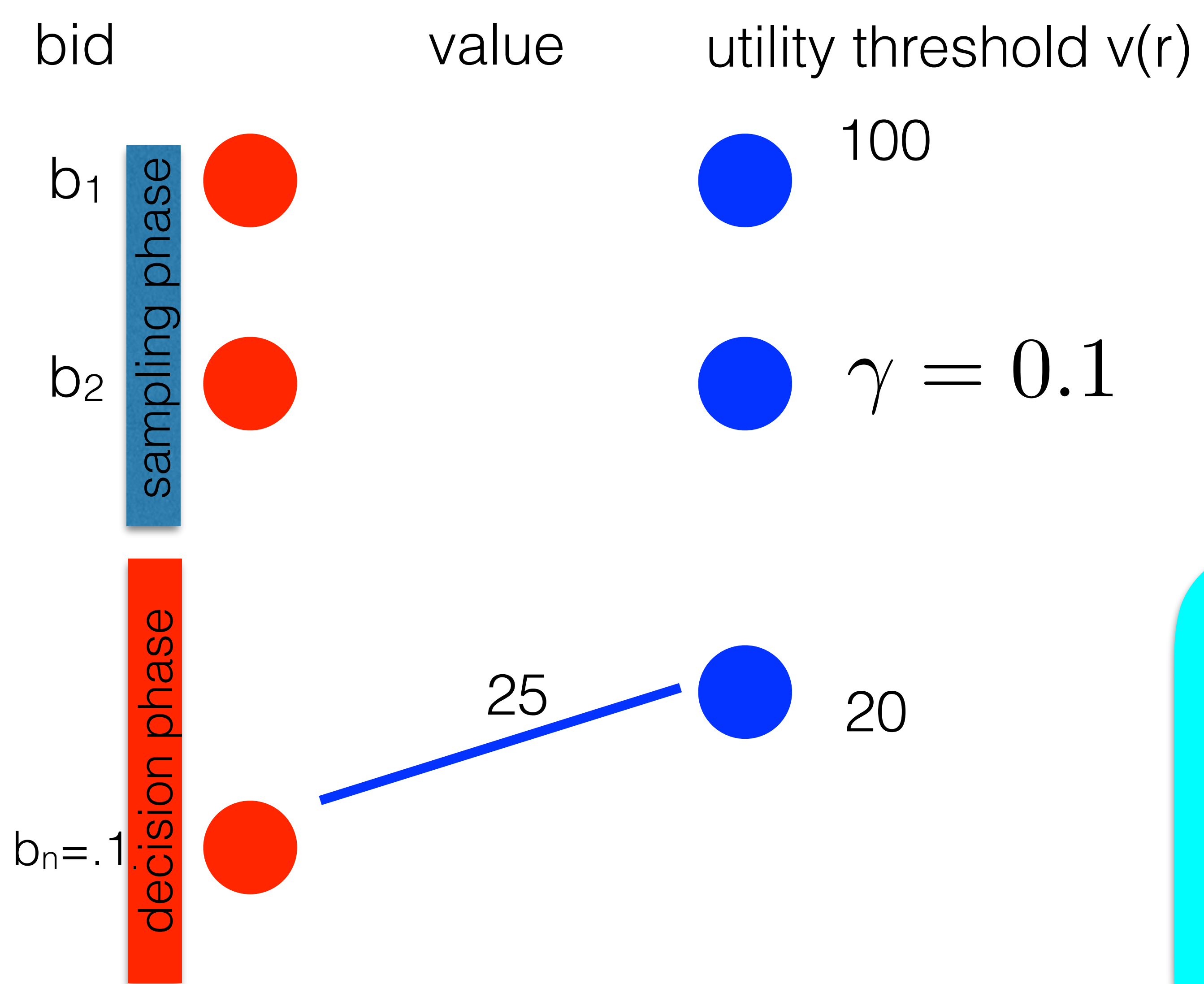
on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Algorithm - **ON**

γ and threshold utility v(r) obtained from **OFF** with budget B/β



bid to benefit ratio of an edge $\frac{b(e)}{v(e)}$

on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

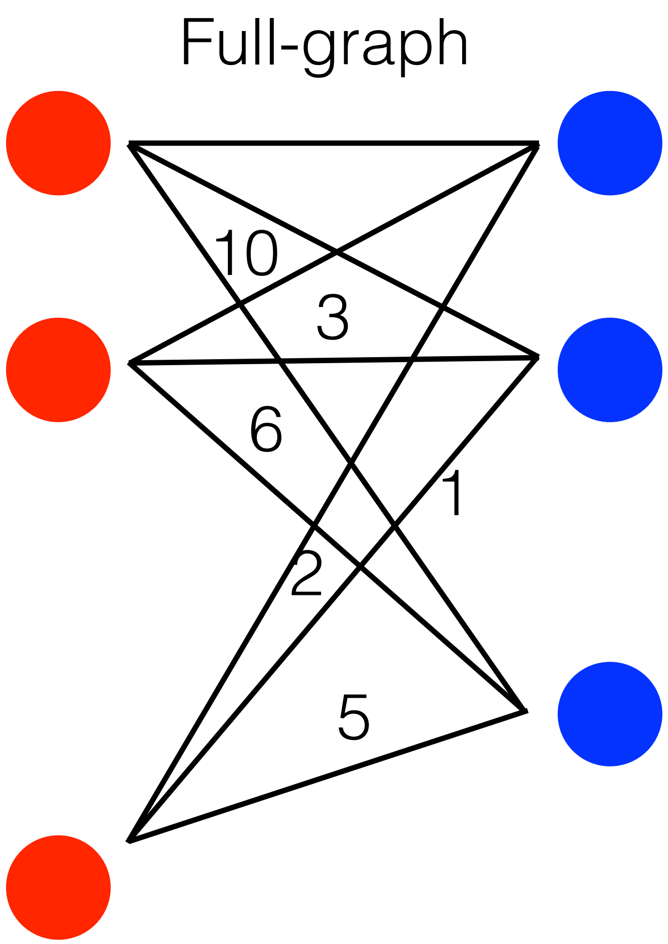
Properties of **`ON`**

24β - competitive in expectation

is truthful

satisfies the payment budget constraint

Properties of $\texttt{OFF}$



$$u(M_{\mathrm{OFF}}(\mathrm{full})) \geq \frac{u(\mathrm{OPT_{full}})}{3}$$

Algorithm

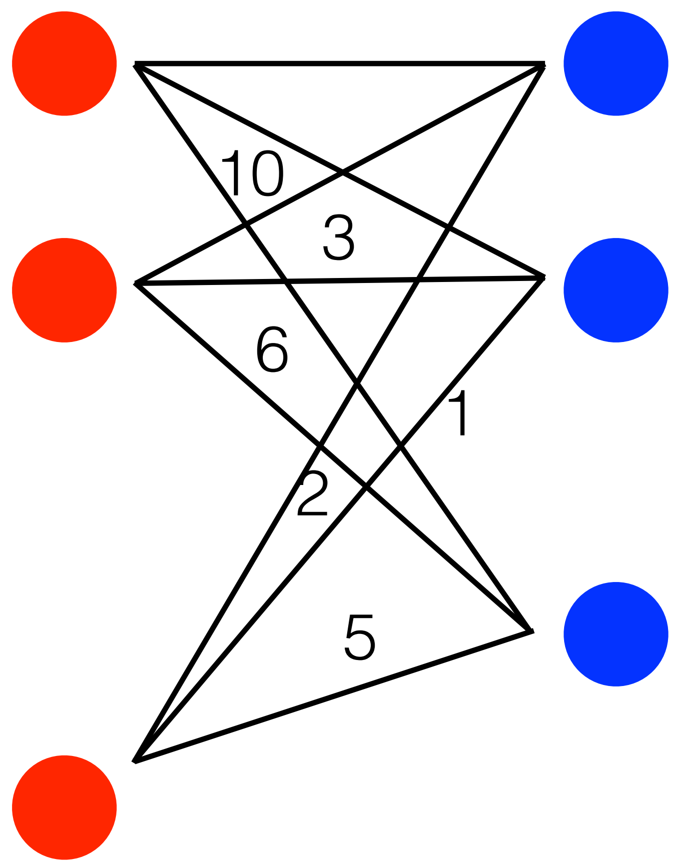
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **OFF**

Full-graph



$$u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}_{\text{full}})}{3}$$

$$\text{OPT} = \{\text{OPT}^+, \text{OPT}^-\}$$

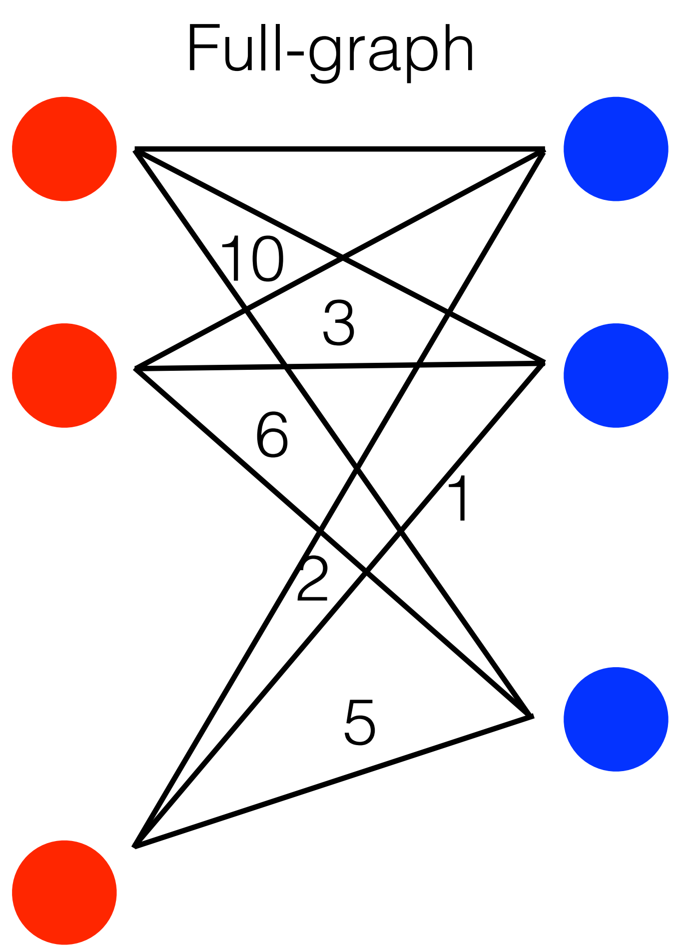
Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **OFF**



$$u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}_{\text{full}})}{3}$$

$$\text{OPT} = \{\text{OPT}^+, \text{OPT}^-\}$$

$$\text{OPT}^- = \{e \in \text{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

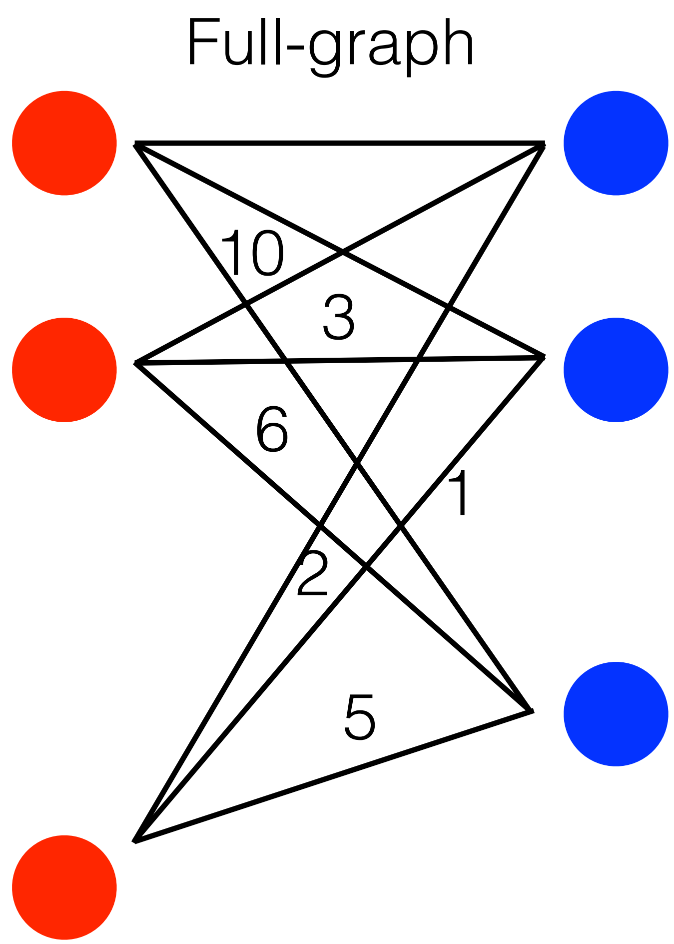
Algorithm

good Graph $\text{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\text{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **OFF**



$$u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}_{\text{full}})}{3}$$

$$\text{OPT} = \{\text{OPT}^+, \text{OPT}^-\}$$

$$\text{OPT}^- = \{e \in \text{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\text{OFF}}(\text{full})$ is a greedy matching over the same set

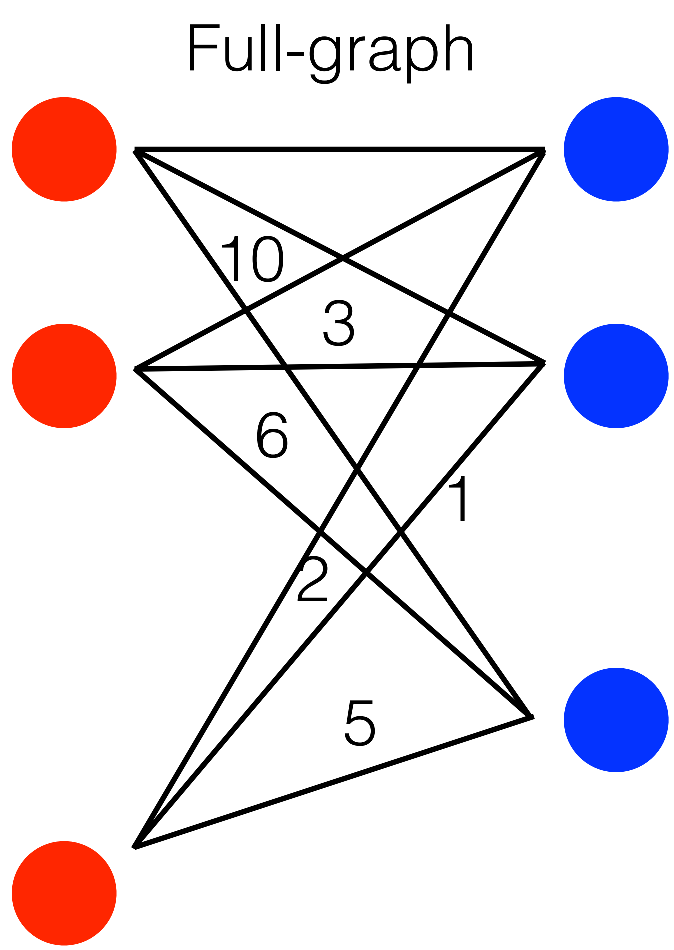
Algorithm

good Graph $\text{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\text{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of OFF



$$u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}_{\text{full}})}{3}$$

$$\text{OPT} = \{\text{OPT}^{+}, \text{OPT}^{-}\}$$

$$\text{OPT}^{-} = \{e \in \text{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\text{OFF}}(\text{full})$ is a greedy matching over the same set

$$u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}^{-})}{2}$$

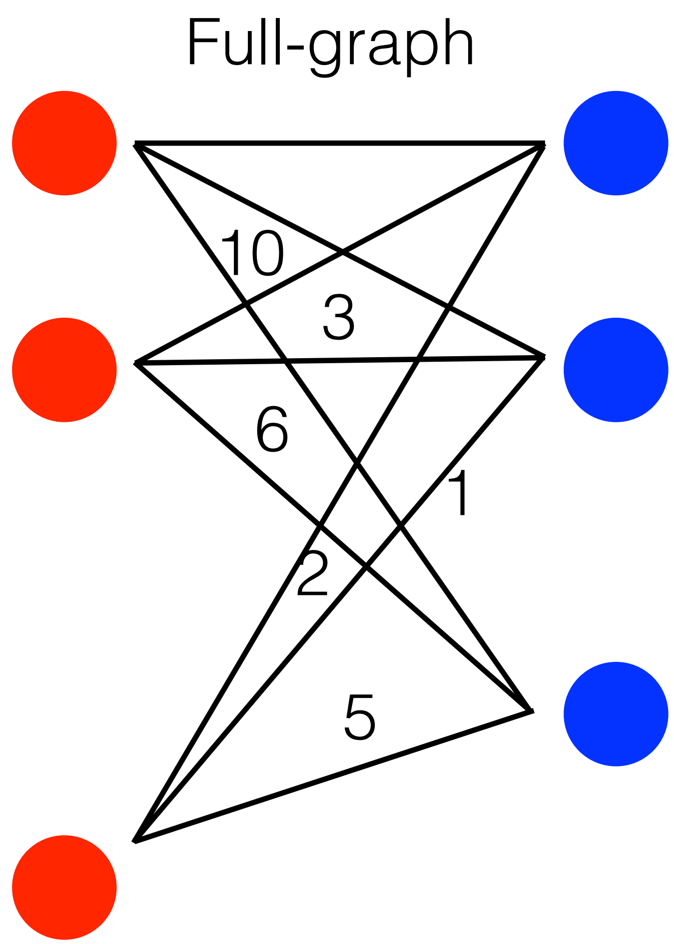
Algorithm

good Graph $\text{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\text{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of $\texttt{OFF}$



$$\color{red}{u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}_{\text{full}})}{3}}$$

$$\text{OPT} = \{\text{OPT}^{+}, \text{OPT}^{-}\}$$

$$\text{OPT}^{-} = \{e \in \text{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\text{OFF}}(\text{full})$ is a greedy matching over the same set

$$u(M_{\text{OFF}}(\text{full})) \geq \frac{u(\text{OPT}^{-})}{2}$$

For $e \in \text{OPT}^{+}, \frac{b}{u(e)} > \gamma$

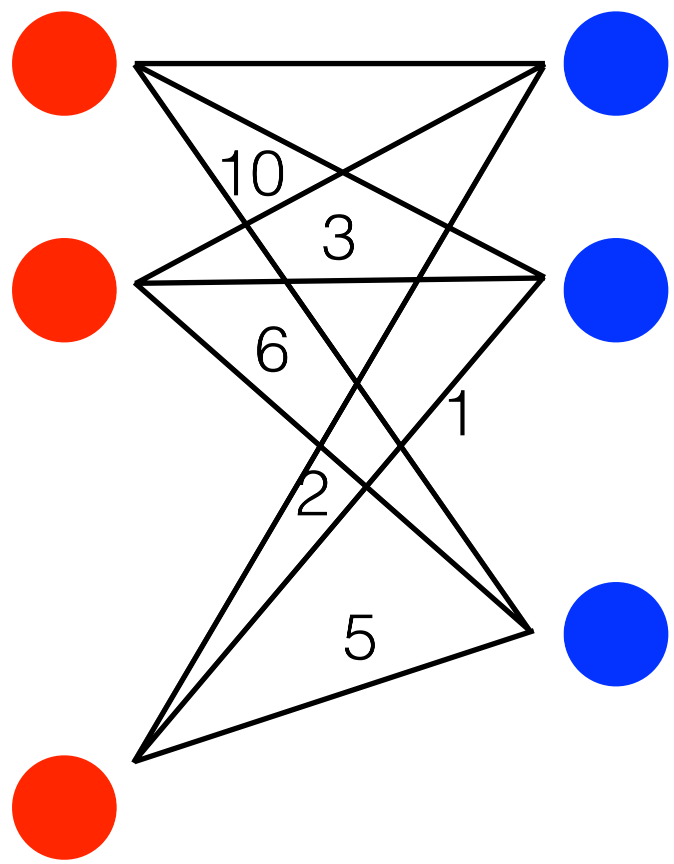
Algorithm

good Graph $\text{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\text{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of $\texttt{OFF}$



$$u(M_{\mathrm{OFF}}(\text{full})) \geq \frac{u(\mathrm{OPT}_{\text{full}})}{3}$$

$$\mathrm{OPT} = \{\mathrm{OPT}^+, \mathrm{OPT}^-\}$$

$$\mathrm{OPT}^- = \{e \in \mathrm{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\mathrm{OFF}}(\text{full})$ is a greedy matching over the same set

$$u(M_{\mathrm{OFF}}(\text{full})) \geq \frac{u(\mathrm{OPT}^-)}{2}$$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

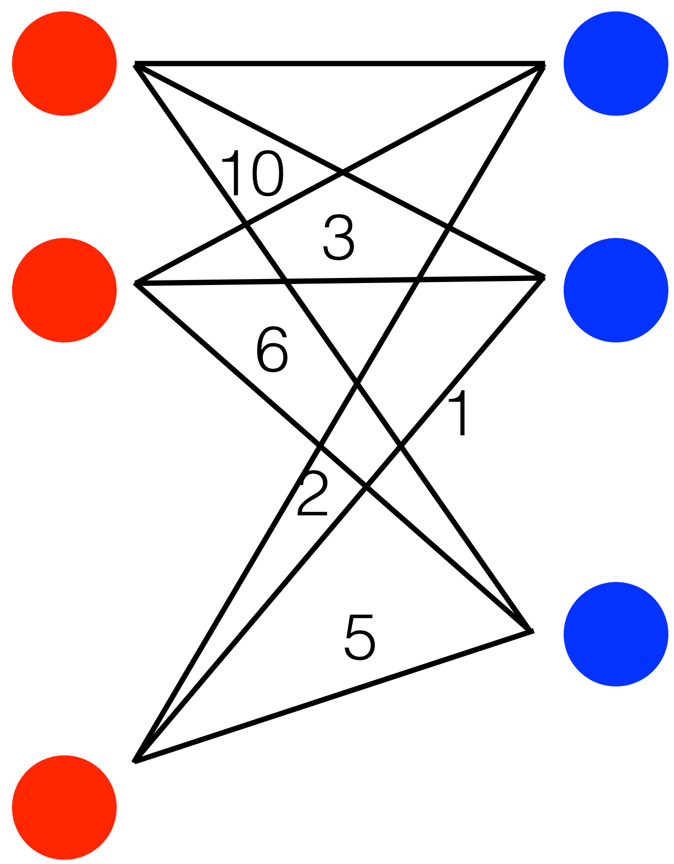
$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

For $e \in \mathrm{OPT}^+$, $\frac{b}{u(e)} > \gamma$

$$u(\mathrm{OPT}^+) \leq \sum_{e \in \mathrm{OPT}^+} u(e) \leq \sum_e \frac{c_e}{\gamma}$$

Properties of **OFF**



$$\color{red}{u(M_{\mathrm{OFF}}(\mathrm{full})) \geq \frac{u(\mathrm{OPT}_{\mathrm{full}})}{3}}$$

$$\mathrm{OPT} = \{\mathrm{OPT}^{+}, \mathrm{OPT}^{-}\}$$

$$\mathrm{OPT}^{-} = \{e \in \mathrm{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\mathrm{OFF}}(\mathrm{full})$ is a greedy matching over the same set

$$u(M_{\mathrm{OFF}}(\mathrm{full})) \geq \frac{u(\mathrm{OPT}^{-})}{2}$$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

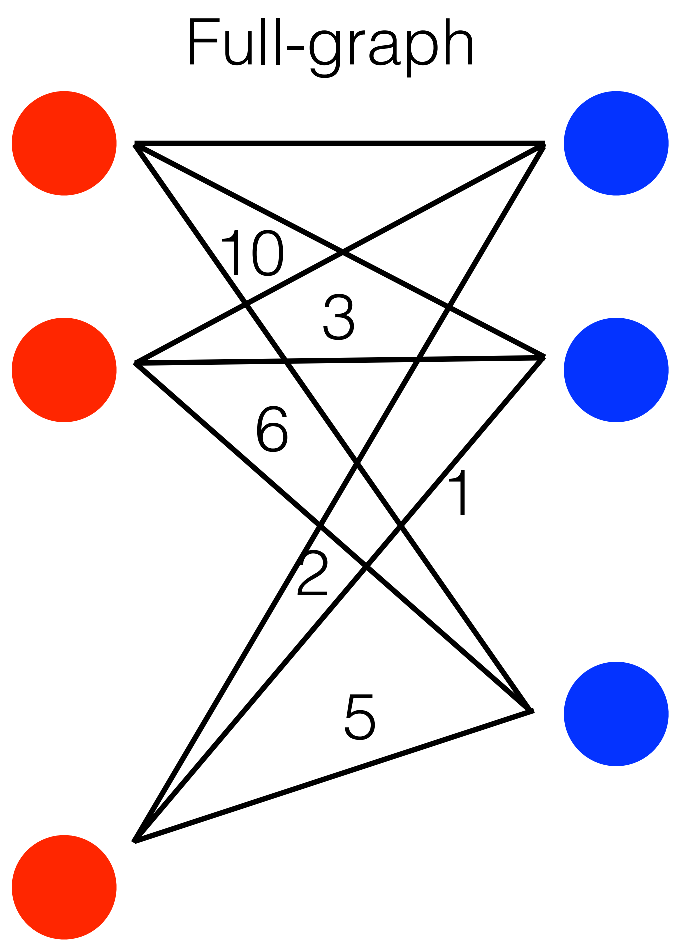
Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

For $e \in \mathrm{OPT}^{+}, \frac{b}{u(e)} > \gamma$

$$u(\mathrm{OPT}^{+}) \leq \sum_{e \in \mathrm{OPT}^{+}} u(e) \leq \sum_{e} \frac{c_e}{\gamma}$$

$$< \frac{B}{\gamma}$$

Properties of OFF



$$u(M_{\mathrm{OFF}}(\text{full})) \geq \frac{u(\mathrm{OPT}_{\text{full}})}{3}$$

$$\mathrm{OPT} = \{\mathrm{OPT}^{+}, \mathrm{OPT}^{-}\}$$

$$\mathrm{OPT}^{-} = \{e \in \mathrm{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\mathrm{OFF}}(\text{full})$ is a greedy matching over the same set

$$u(M_{\mathrm{OFF}}(\text{full})) \geq \frac{u(\mathrm{OPT}^{-})}{2}$$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

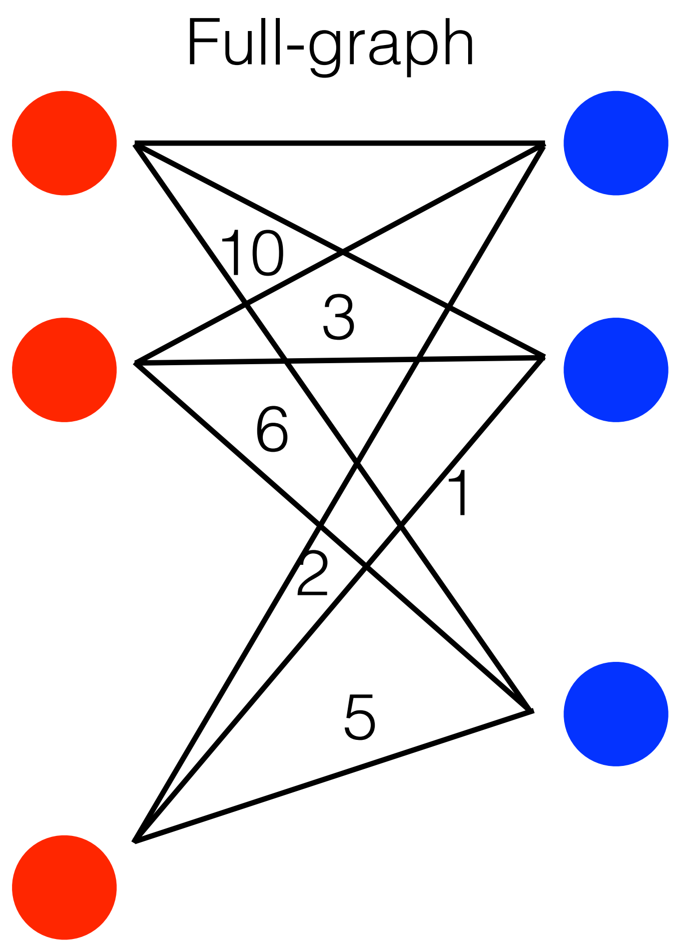
For $e \in \mathrm{OPT}^{+}$, $\frac{b}{u(e)} > \gamma$

$$u(\mathrm{OPT}^{+}) \leq \sum_{e \in \mathrm{OPT}^{+}} u(e) \leq \sum_{e} \frac{c_e}{\gamma}$$

$$< \frac{B}{\gamma}$$

$$= u(M_{\mathrm{OFF}}(\text{full}))$$

Properties of **OFF**



$$u(M_{\mathrm{OFF}}(\mathrm{full})) \geq \frac{u(\mathrm{OPT}_{\mathrm{full}})}{3}$$

$$\mathrm{OPT} = \{\mathrm{OPT}^+, \mathrm{OPT}^-\}$$

$$\mathrm{OPT}^- = \{e \in \mathrm{OPT} : \frac{b}{u(e)} \leq \gamma\}$$

$M_{\mathrm{OFF}}(\mathrm{full})$ is a greedy matching over the same set

$$u(M_{\mathrm{OFF}}(\mathrm{full})) \geq \frac{u(\mathrm{OPT}^-)}{2}$$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

For $e \in \mathrm{OPT}^+$, $\frac{b}{u(e)} > \gamma$

$$u(\mathrm{OPT}^+) \leq \sum_{e \in \mathrm{OPT}^+} u(e) \leq \sum_e \frac{c_e}{\gamma}$$

$$< \frac{B}{\gamma}$$

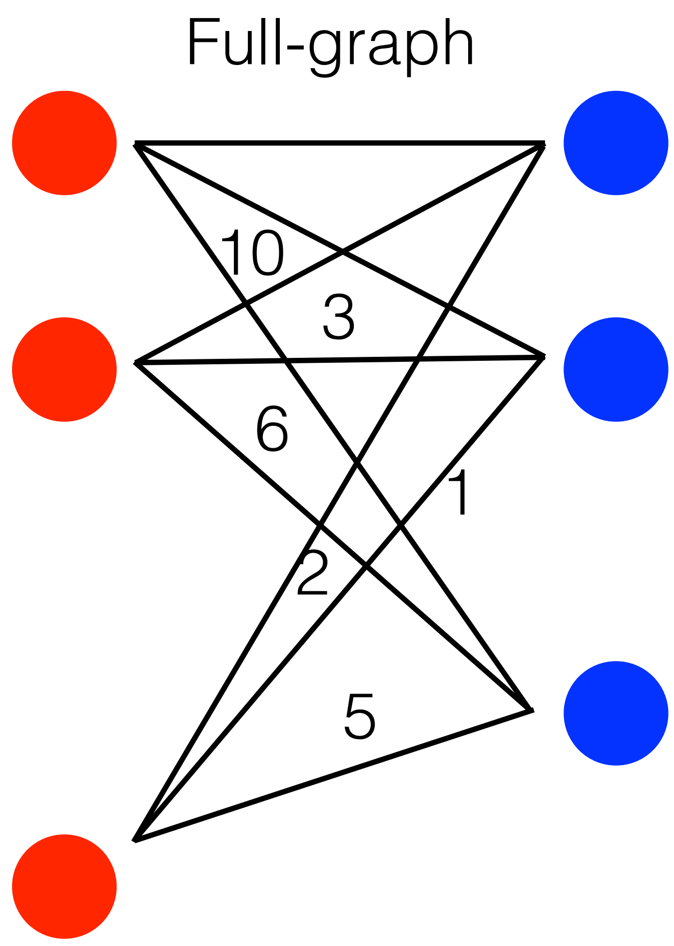
$$= u(M_{\mathrm{OFF}}(\mathrm{full}))$$

Properties of `OFF`

12 - approximate in expectation in offline phase

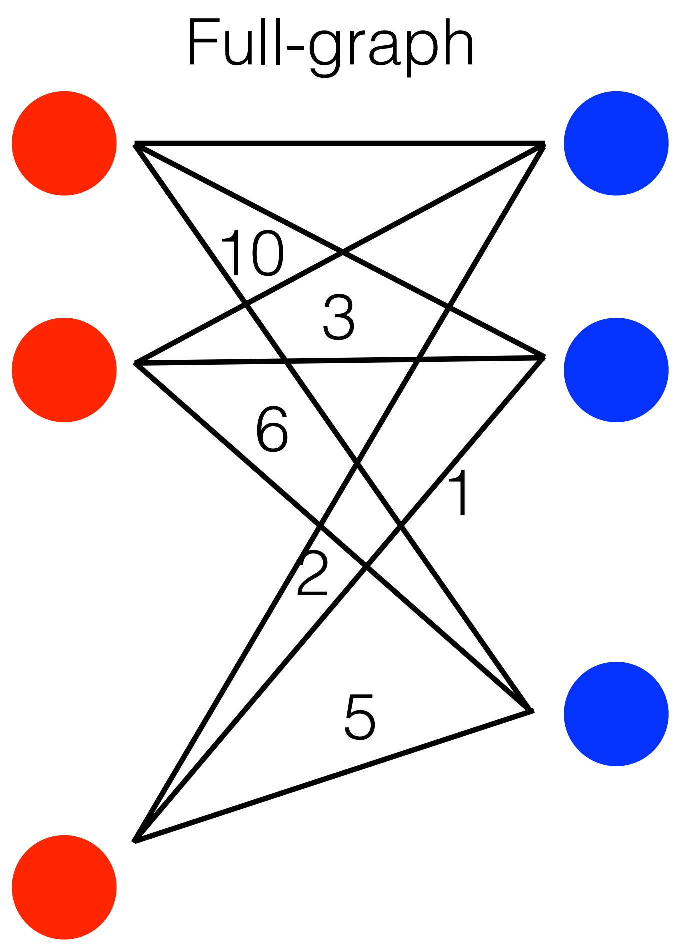
Properties of **`OFF`**

12 - approximate in expectation in offline phase



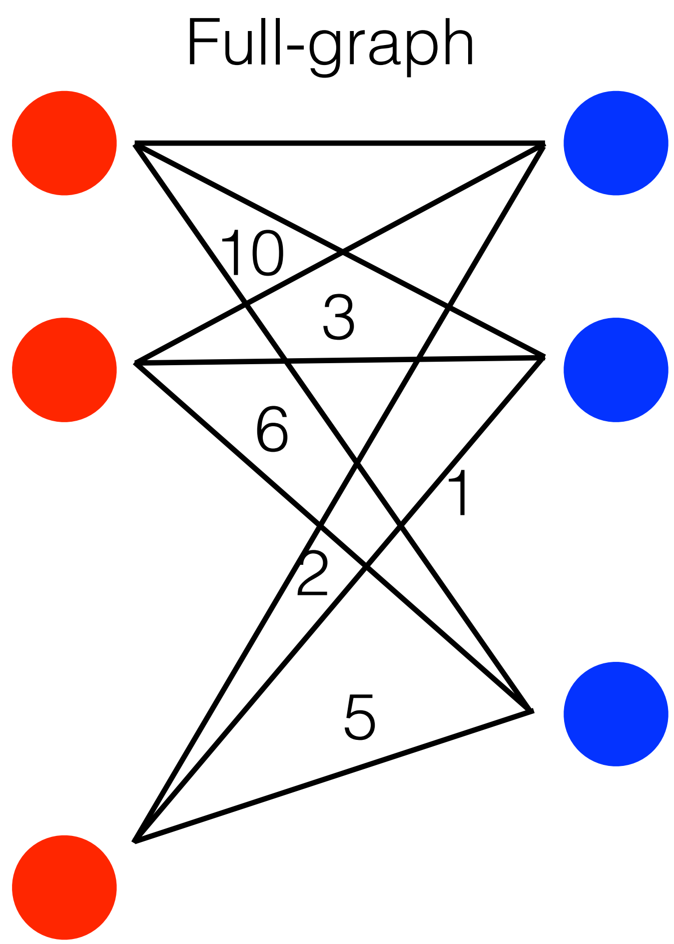
Properties of **OFF**

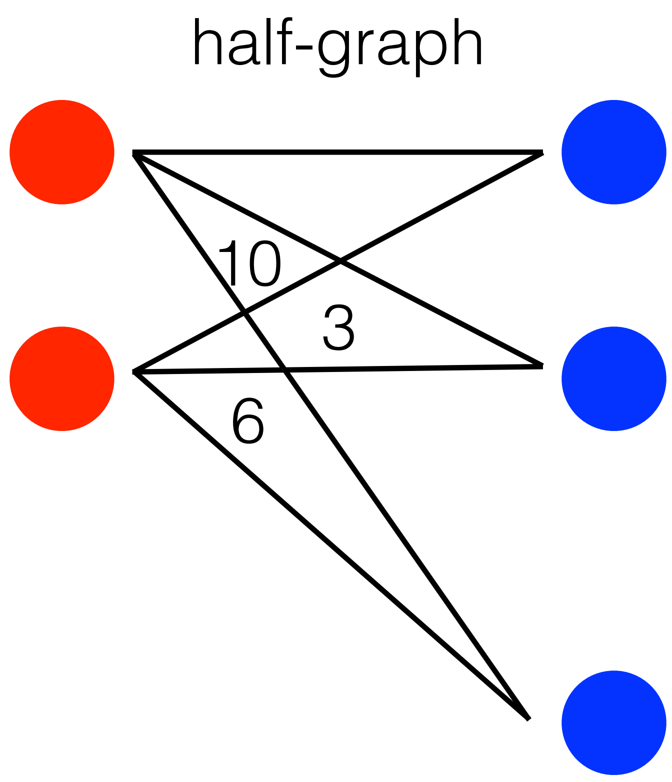
12 - approximate in expectation in offline phase



Properties of `OFF`

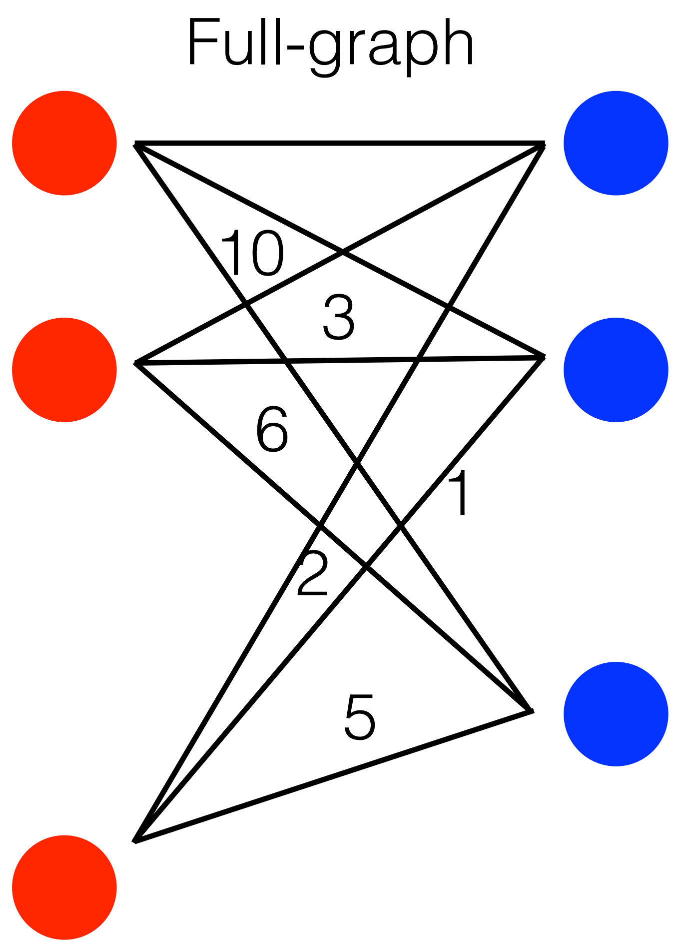
12 - approximate in expectation in offline phase

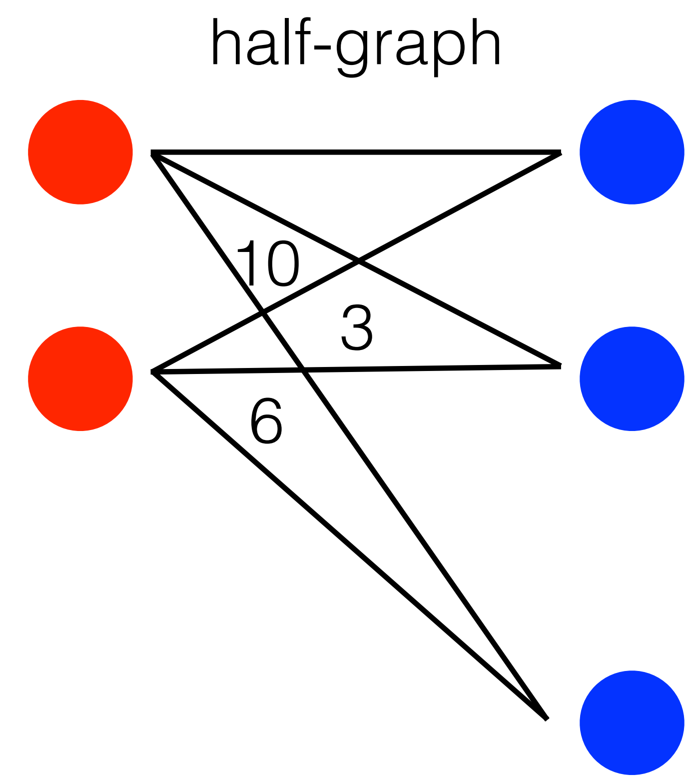




Properties of **OFF**

12 - approximate in expectation in offline phase





Algorithm

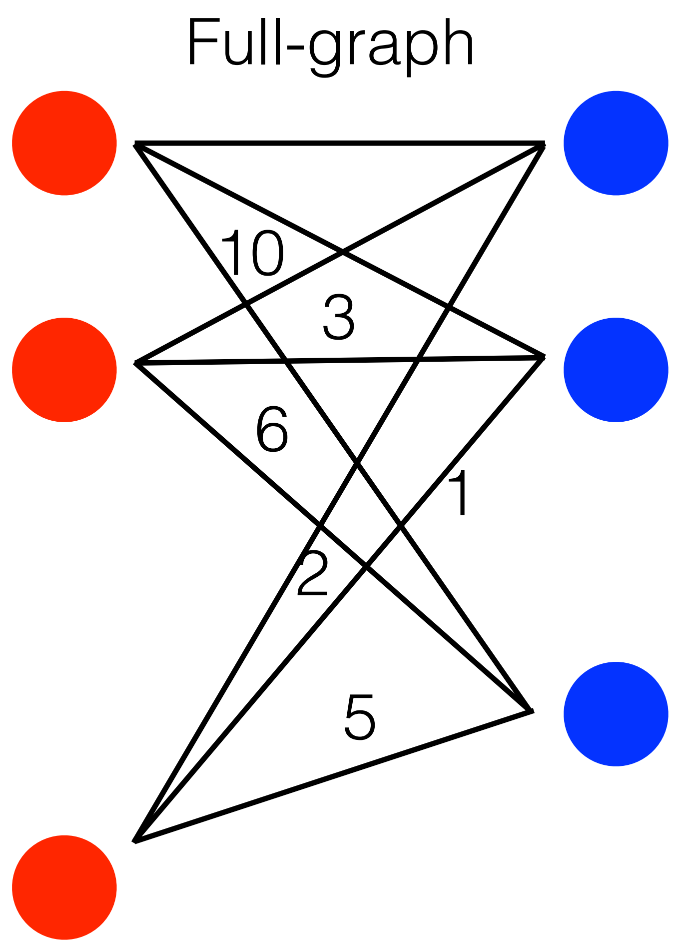
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

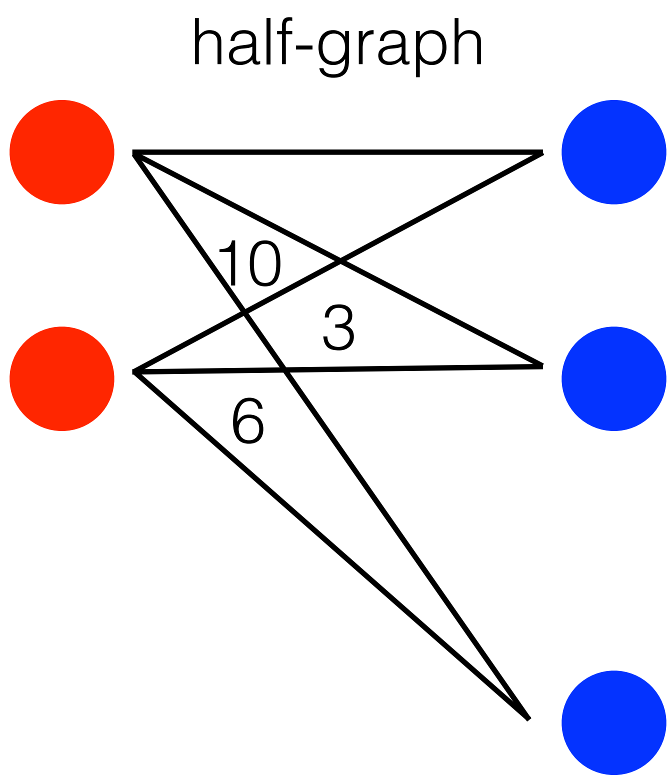
$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of `OFF`

12 - approximate in expectation in offline phase





$$u(\mathrm{OPT}_{\mathrm{half}}) \geq u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))$$

Algorithm

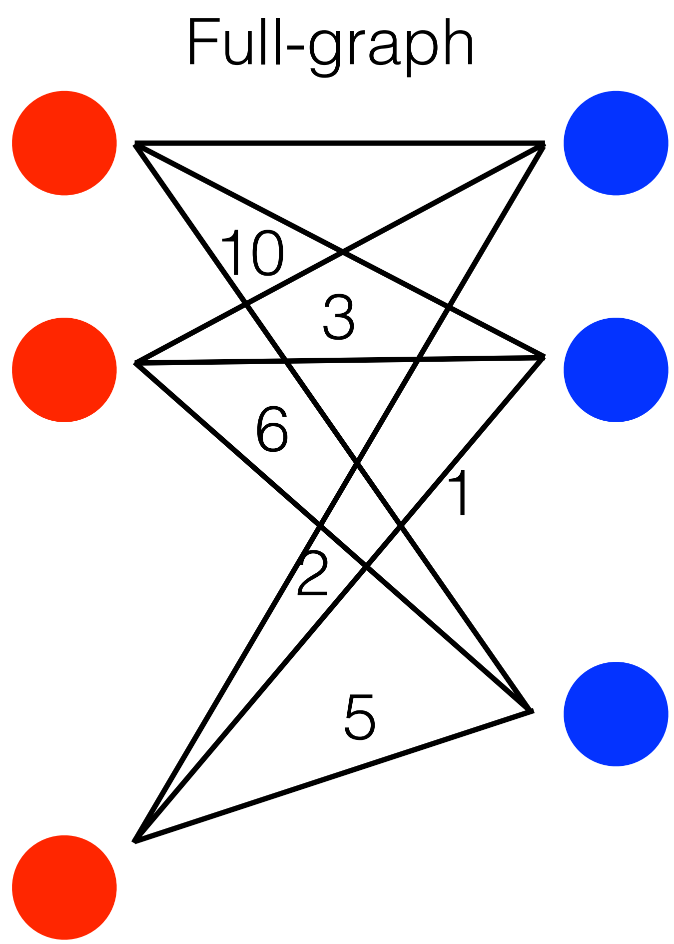
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

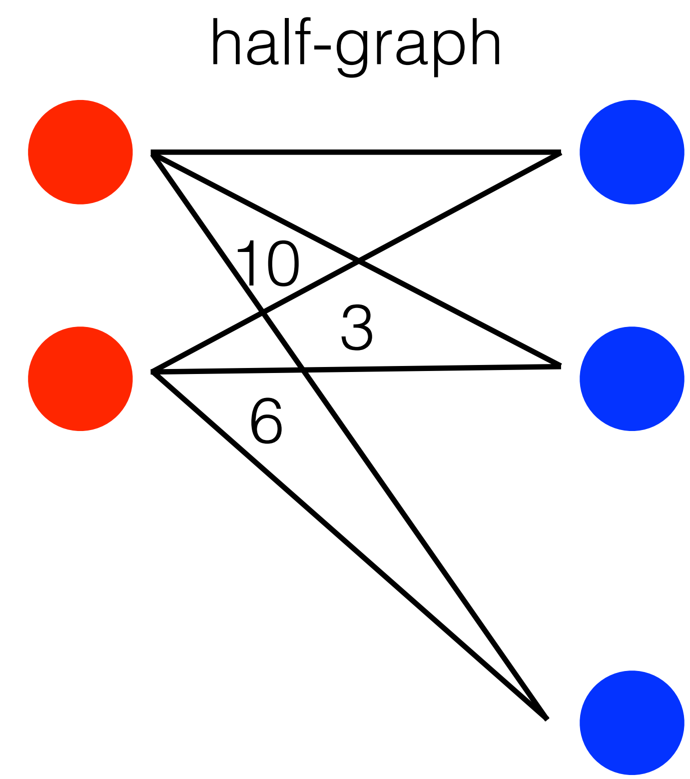
$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of `OFF`

12 - approximate in expectation in offline phase





$$u(\mathrm{OPT_{half}}) \geq u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))$$

$$\mathbf{E}\left\{u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))\right\} \geq \frac{u(M_{\mathrm{OFF}}(\mathrm{full}))}{2}$$

Algorithm

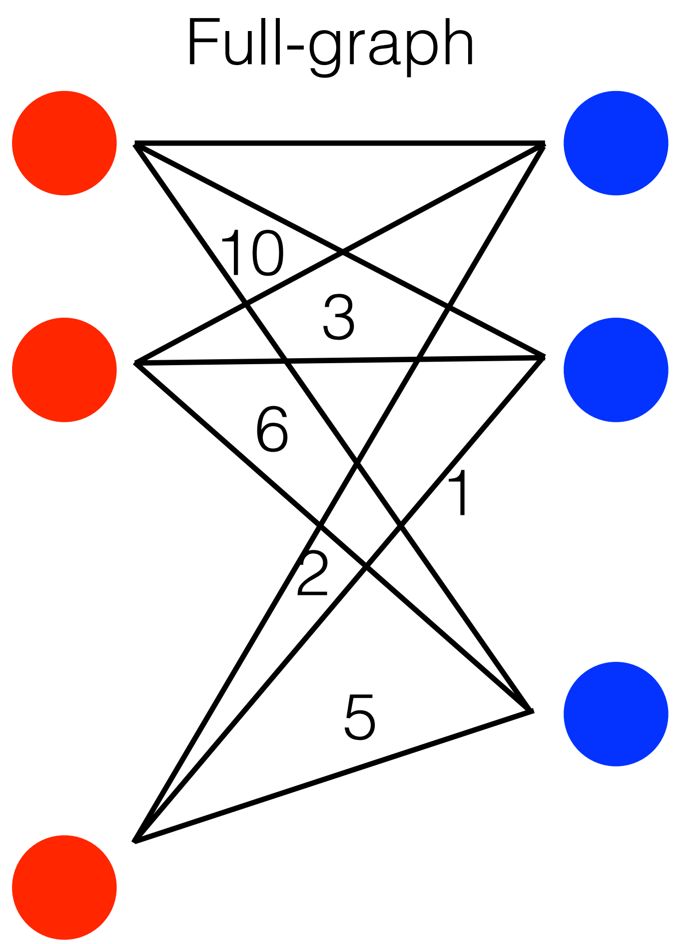
good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

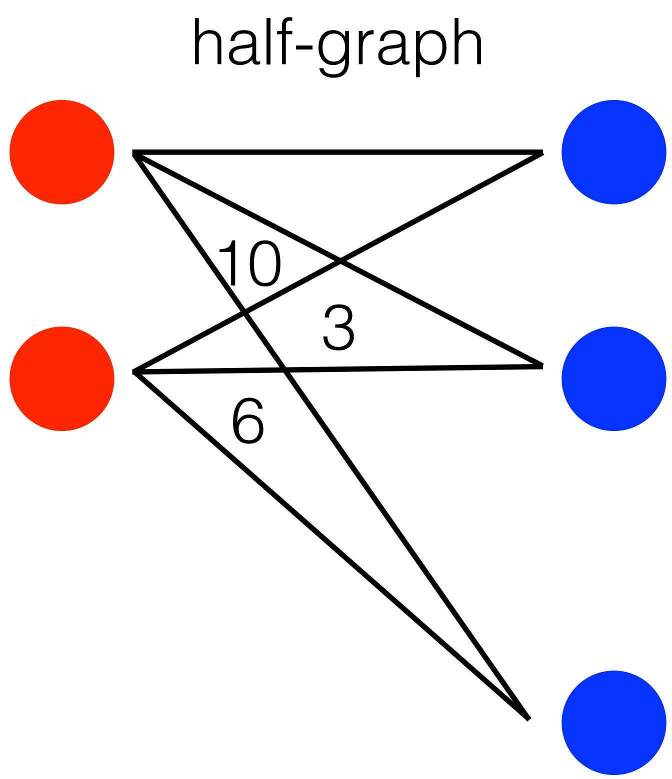
Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of OFF

12 - approximate in expectation in offline phase



$\gamma_f \leq \gamma_h$



$$u(\mathrm{OPT}_{\mathrm{half}}) \geq u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))$$

$$\mathbf{E}\left\{u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))\right\} \geq \frac{u(M_{\mathrm{OFF}}(\mathrm{full}))}{2}$$

Algorithm

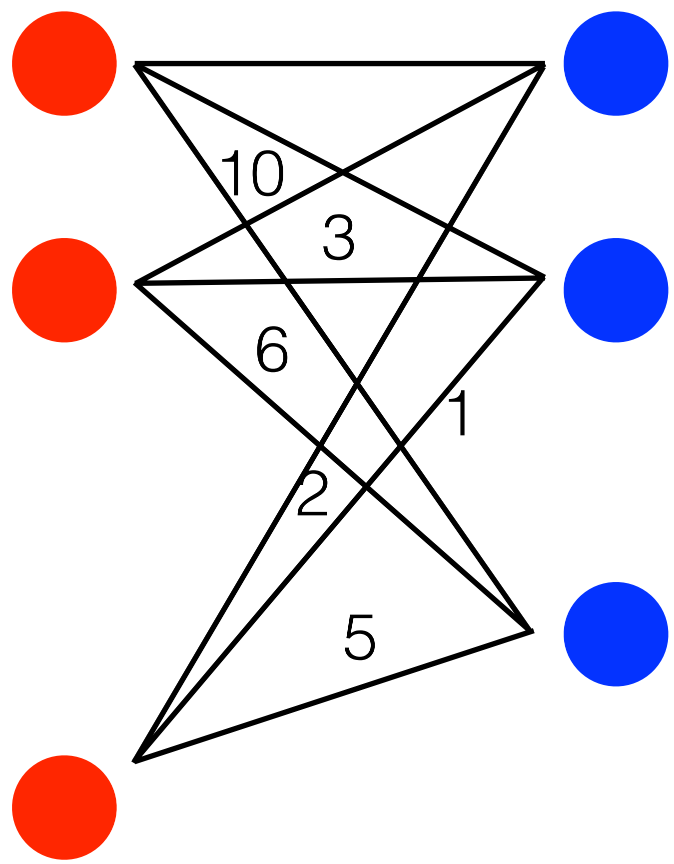
good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

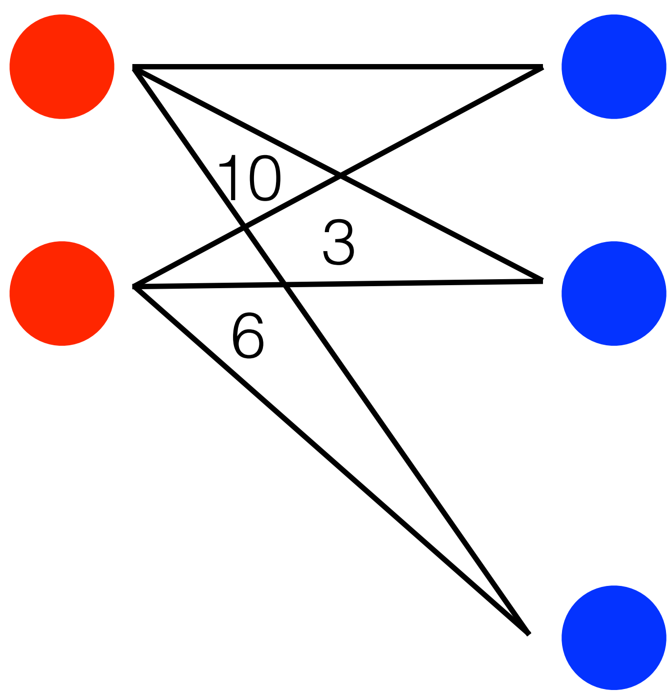
Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of `OFF`

12 - approximate in expectation in offline phase



$\gamma_f \leq \gamma_h$



$$u(\mathrm{OPT}_{\mathrm{half}}) \geq u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))$$

$$\mathbf{E}\left\{u(M_{\mathrm{OFF}}(\mathrm{full}, \mathrm{half}))\right\} \geq \frac{u(M_{\mathrm{OFF}}(\mathrm{full}))}{2}$$

$$u(M_{\mathrm{OFF}}(\mathrm{half})) \geq \frac{u(\mathrm{OPT}_{\mathrm{half}})}{2}$$

Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

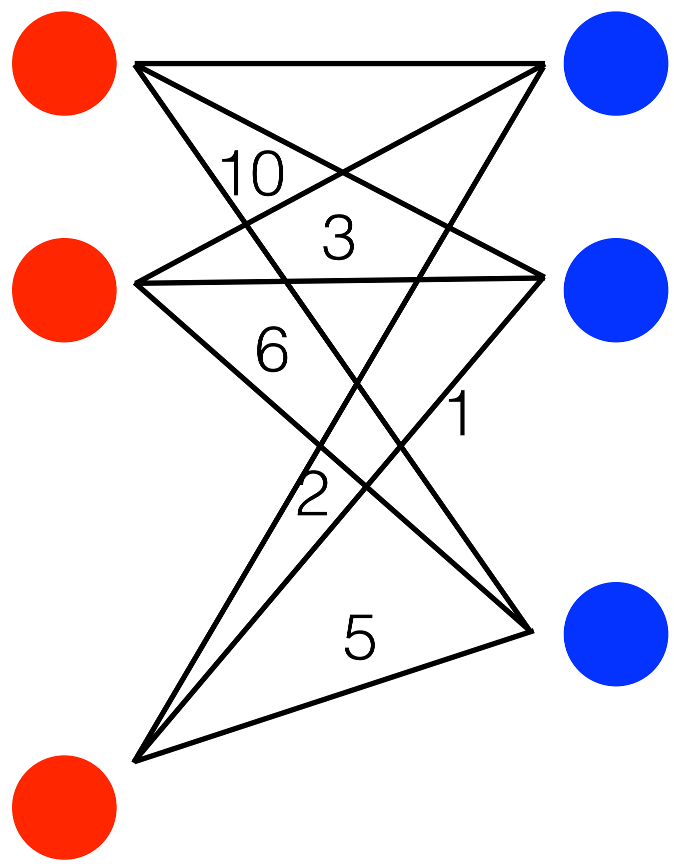
$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of `OFF`

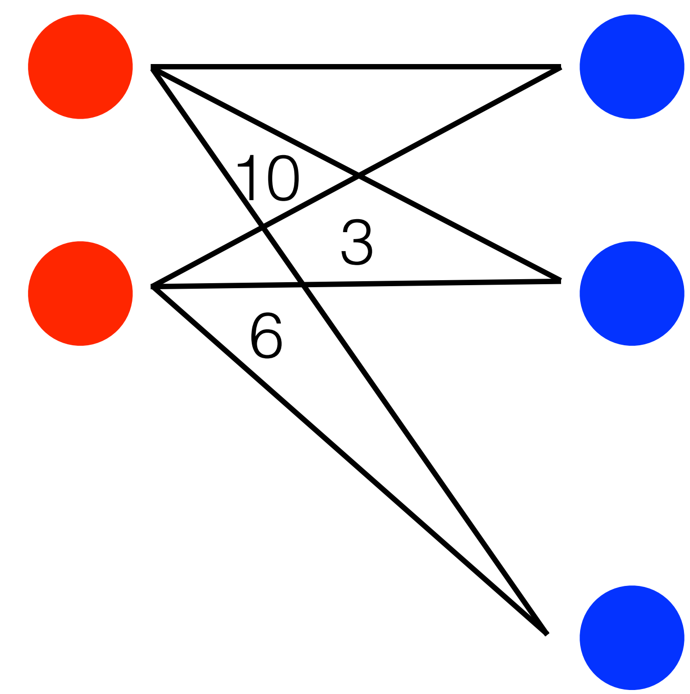
12 - approximate in expectation in offline phase

Full-graph



$\gamma_f \le \gamma_h$

half-graph



$$u(\mathrm{OPT_{half}}) \ge u(M_{\mathrm{OFF}}(\mathrm{full},\mathrm{half}))$$

$$\mathbf{E}\left\{u(M_{\mathrm{OFF}}(\mathrm{full},\mathrm{half}))\right\} \ge \frac{u(M_{\mathrm{OFF}}(\mathrm{full}))}{2}$$

$$u(M_{\mathrm{OFF}}(\mathrm{half})) \ge \frac{u(\mathrm{OPT_{half}})}{2}$$

$$u(M_{\mathrm{OFF}}(\mathrm{full})) \ge \frac{u(\mathrm{OPT_{full}})}{3}$$

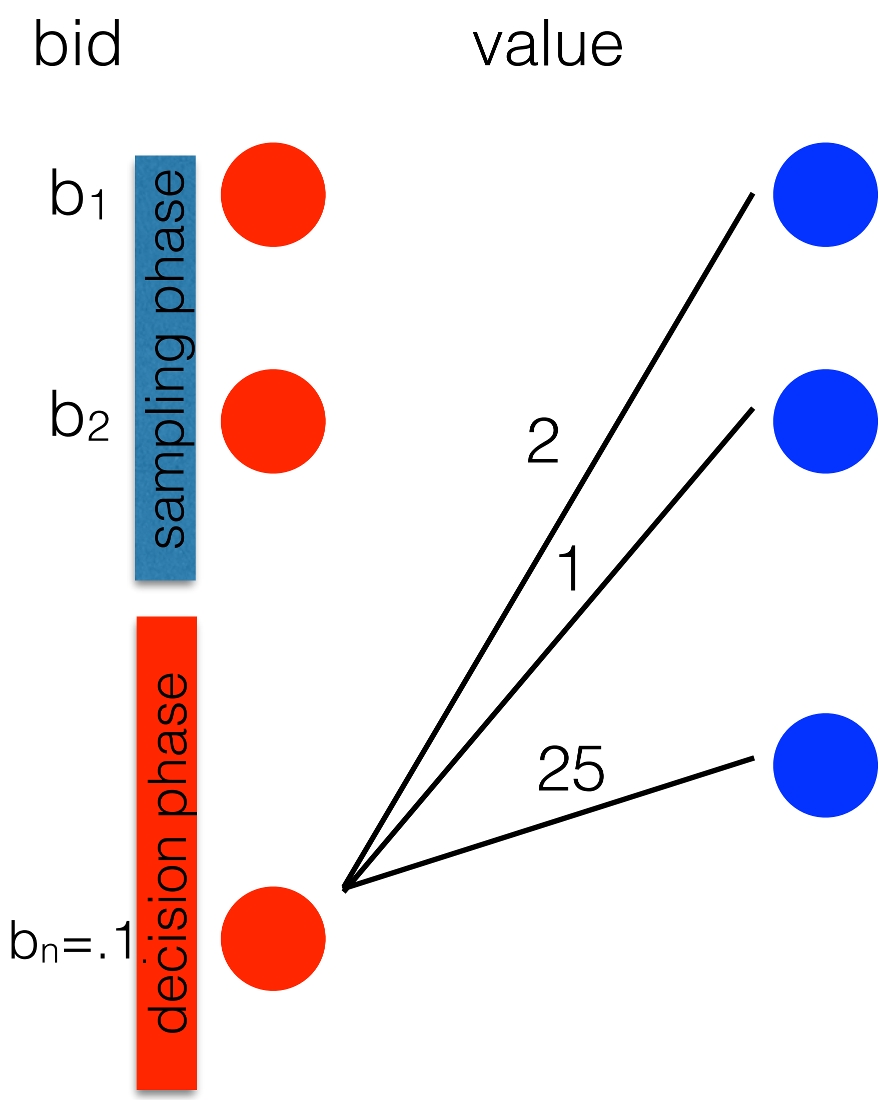
Algorithm

good Graph $\mathrm{G}(\gamma) = \left\{e \in G : \frac{b(e)}{v(e)} < \gamma\right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \le B$

Properties of ON



on arrival of a new left vertex

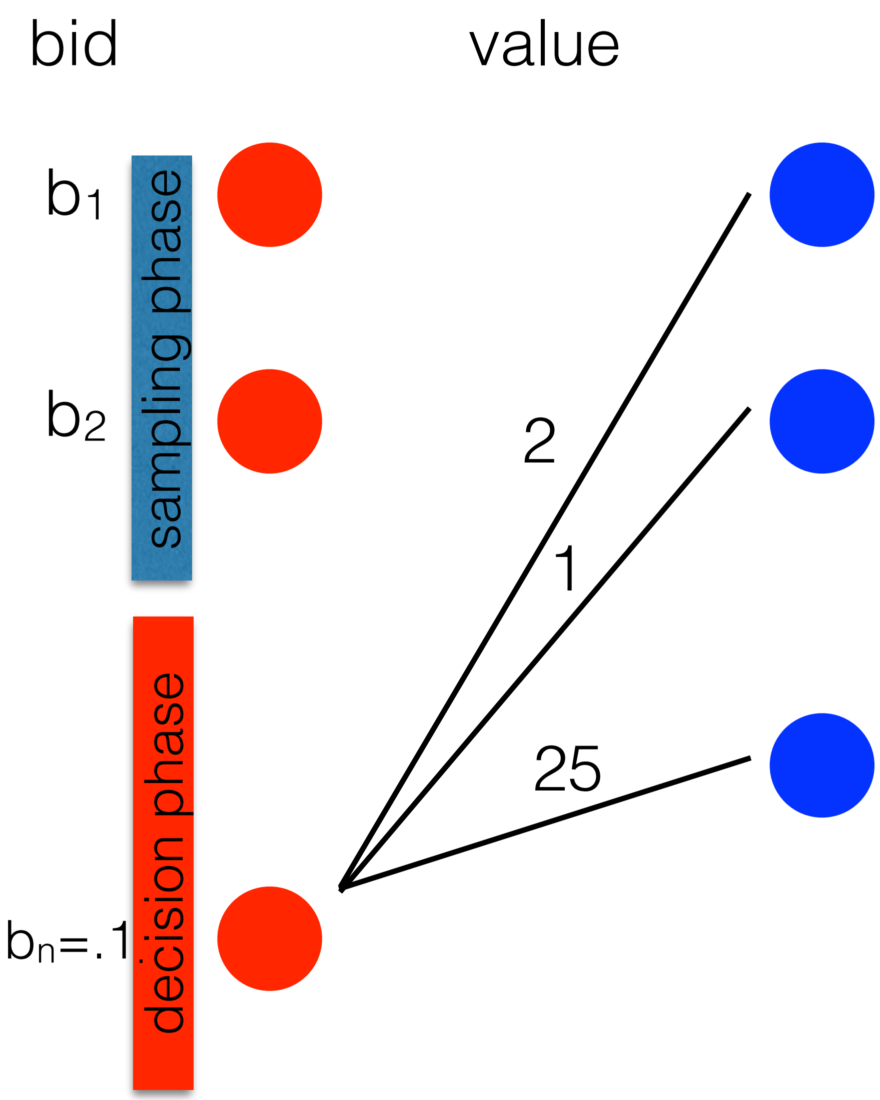
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of **ON**

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

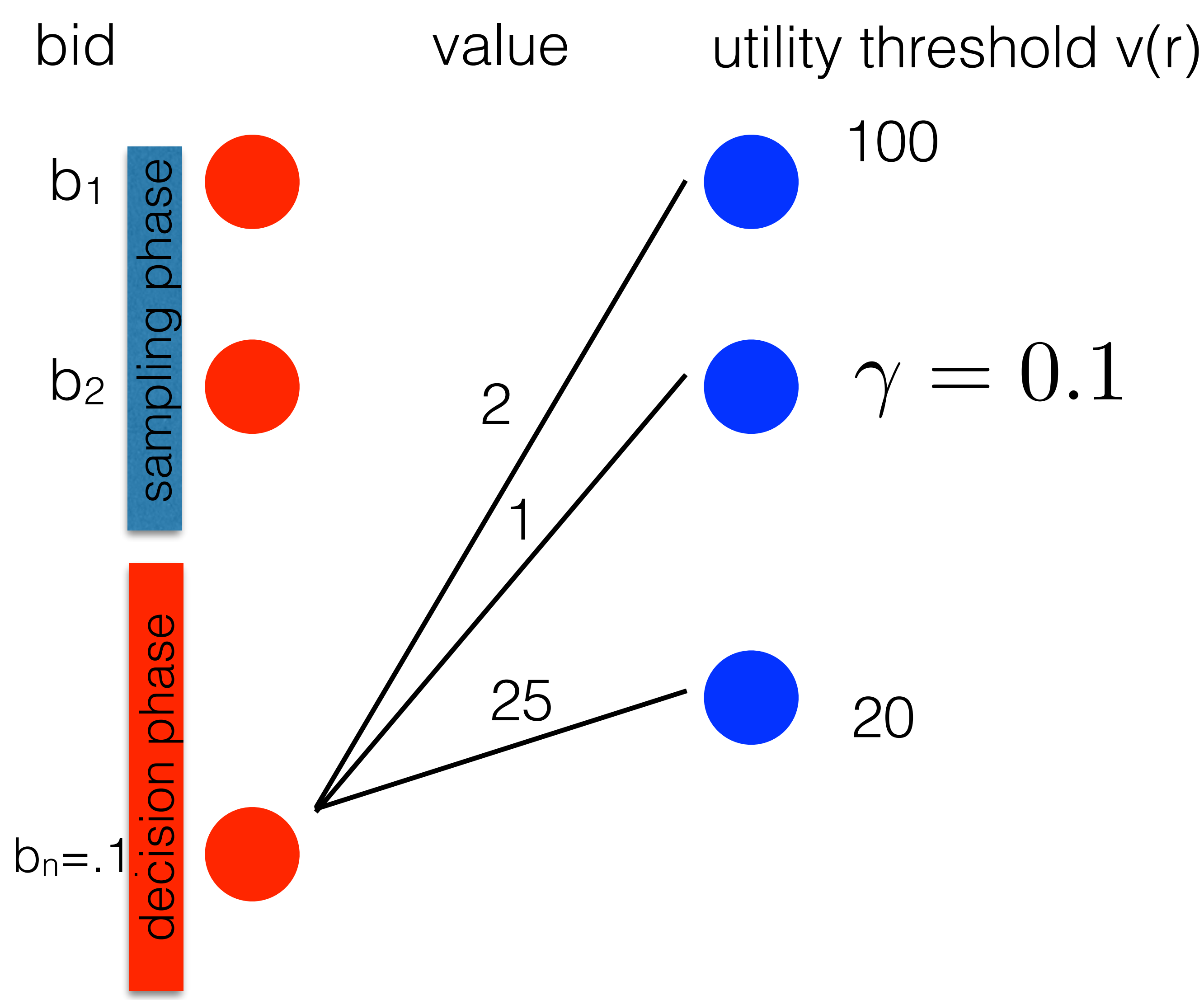
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of ON

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

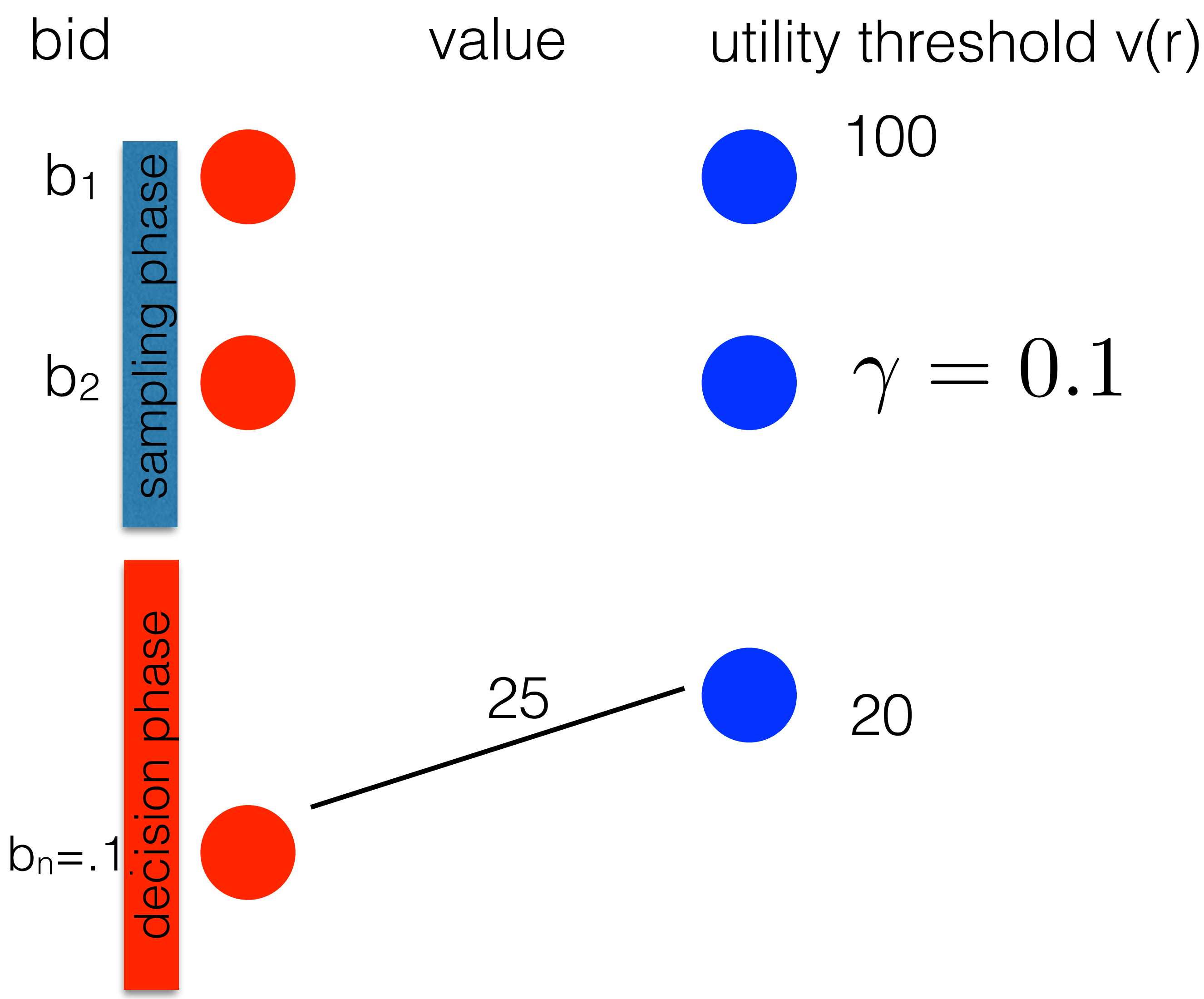
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of **ON**

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

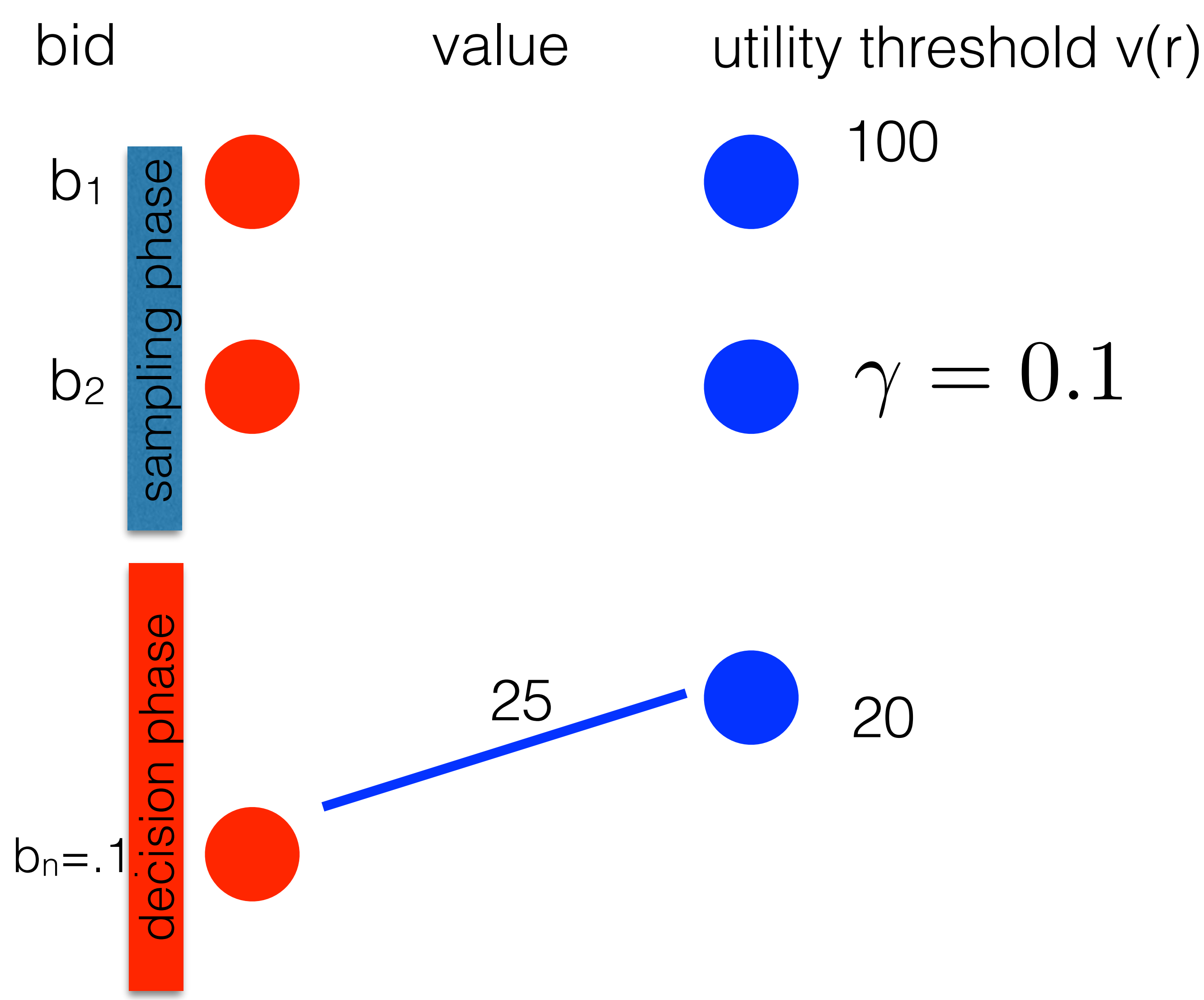
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of **ON**

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

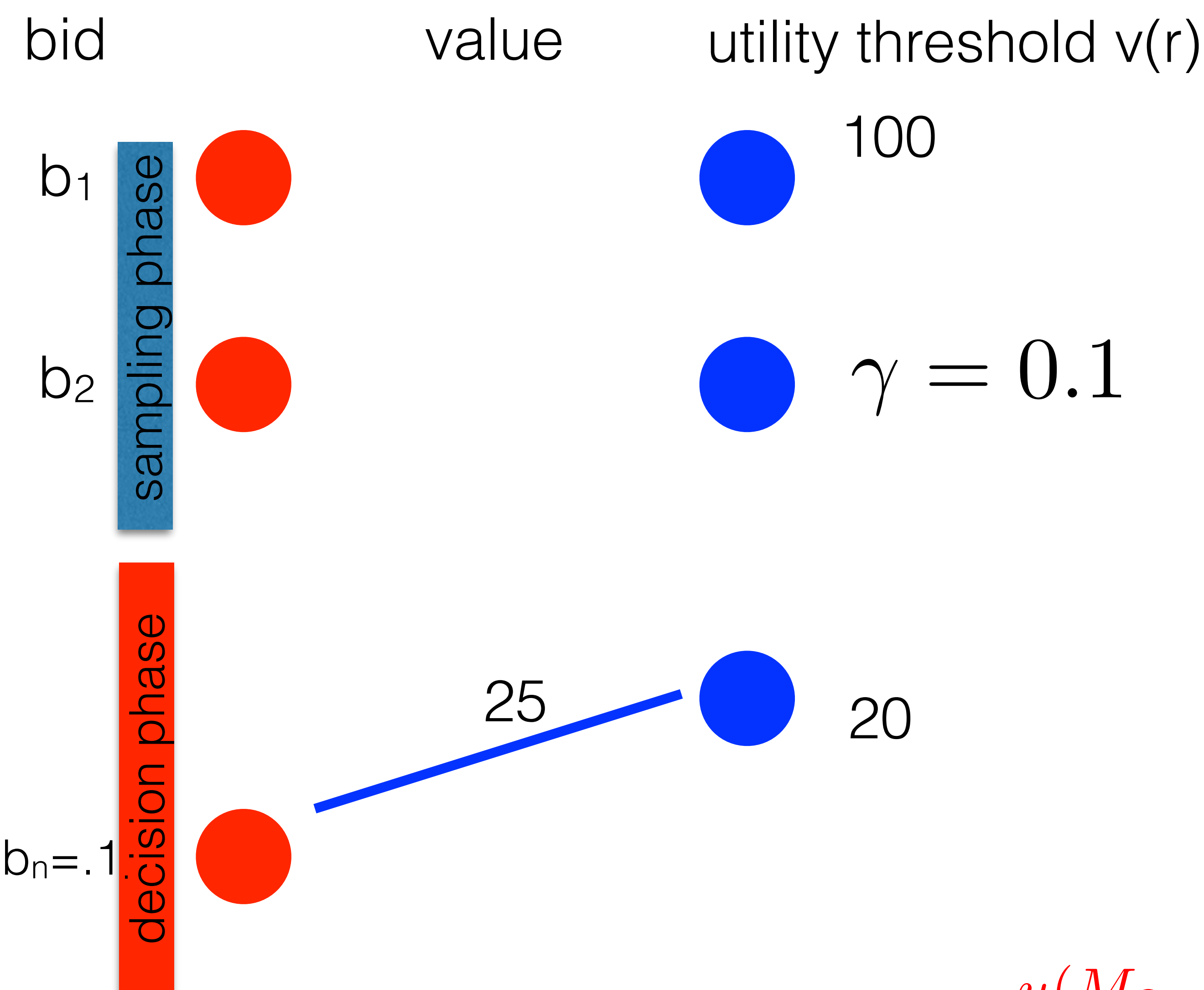
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of **ON**

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

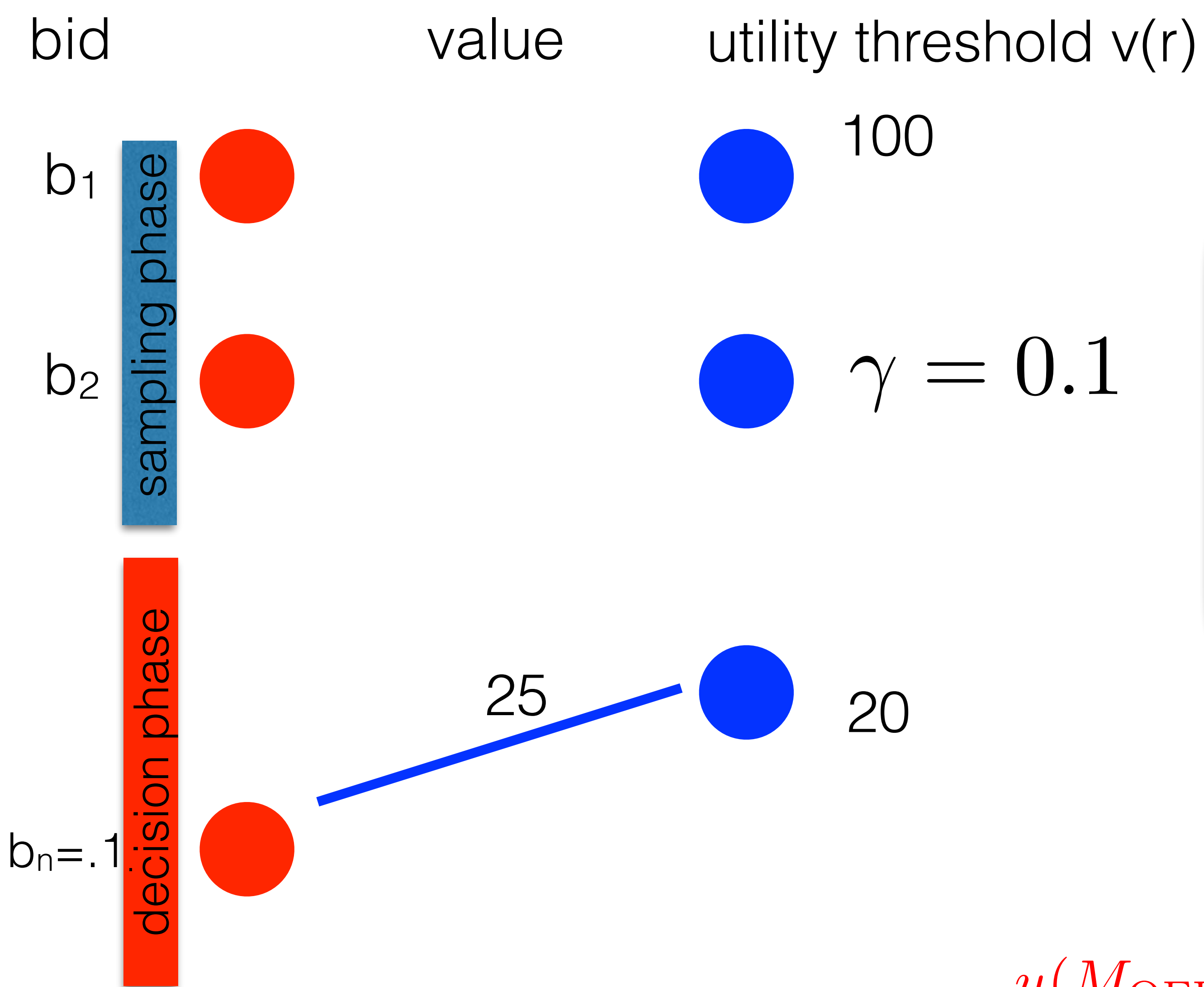
match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

$$u(M_{\mathrm{ON}}) \geq \frac{u(M_{\mathrm{OFF}}(\mathrm{half}))}{2}$$ [Korula, Pal ‘ 08]

Properties of ON

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

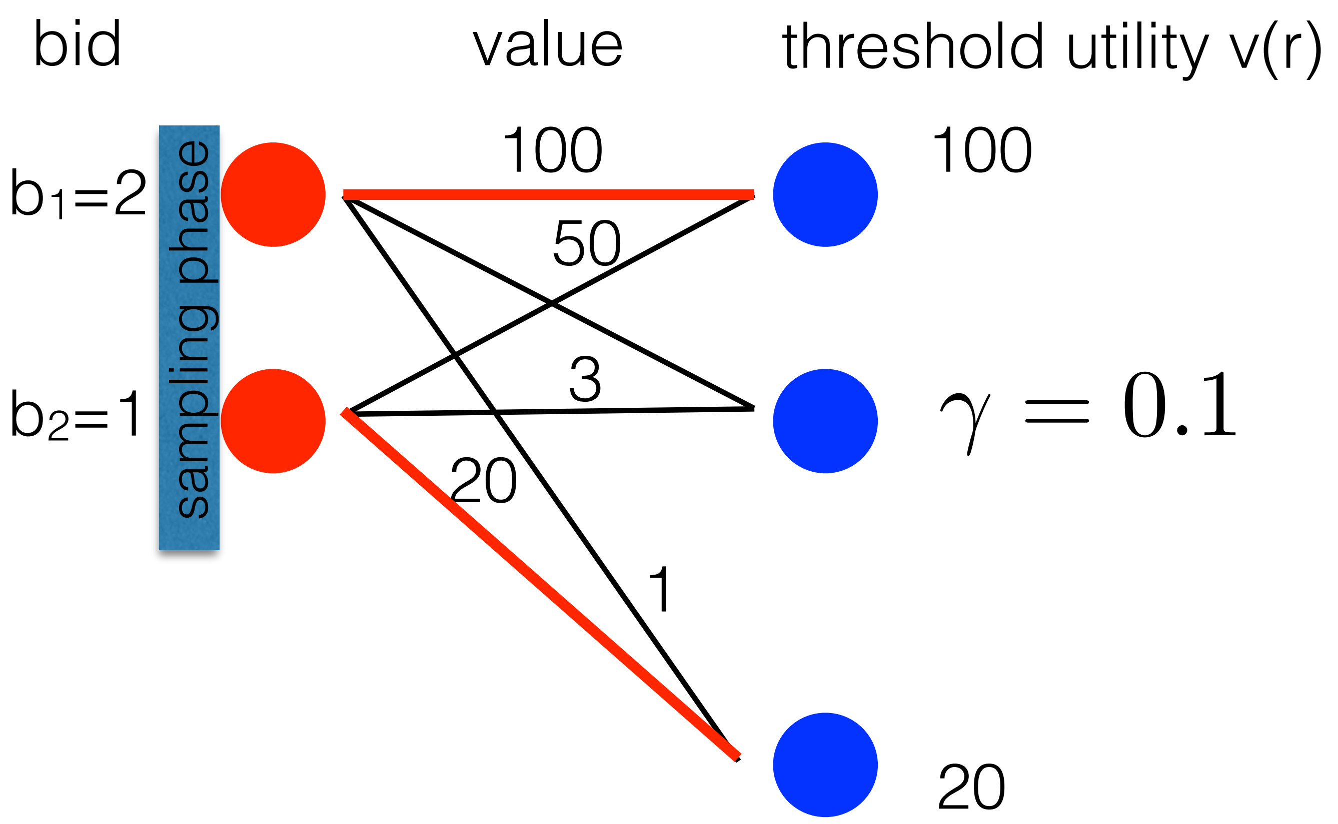
match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

$$u(M_{\mathrm{ON}}) \geq \frac{u(M_{\mathrm{OFF}}(\mathrm{half}))}{2}$$ [Korula, Pal ' 08]

Since we run OFF with budget B/β $\quad u(M_{\mathrm{ON}}) \geq \frac{\mathrm{OPT}_{\mathrm{full}}}{24\beta}$

Properties of **ON**



Algorithm - OFF

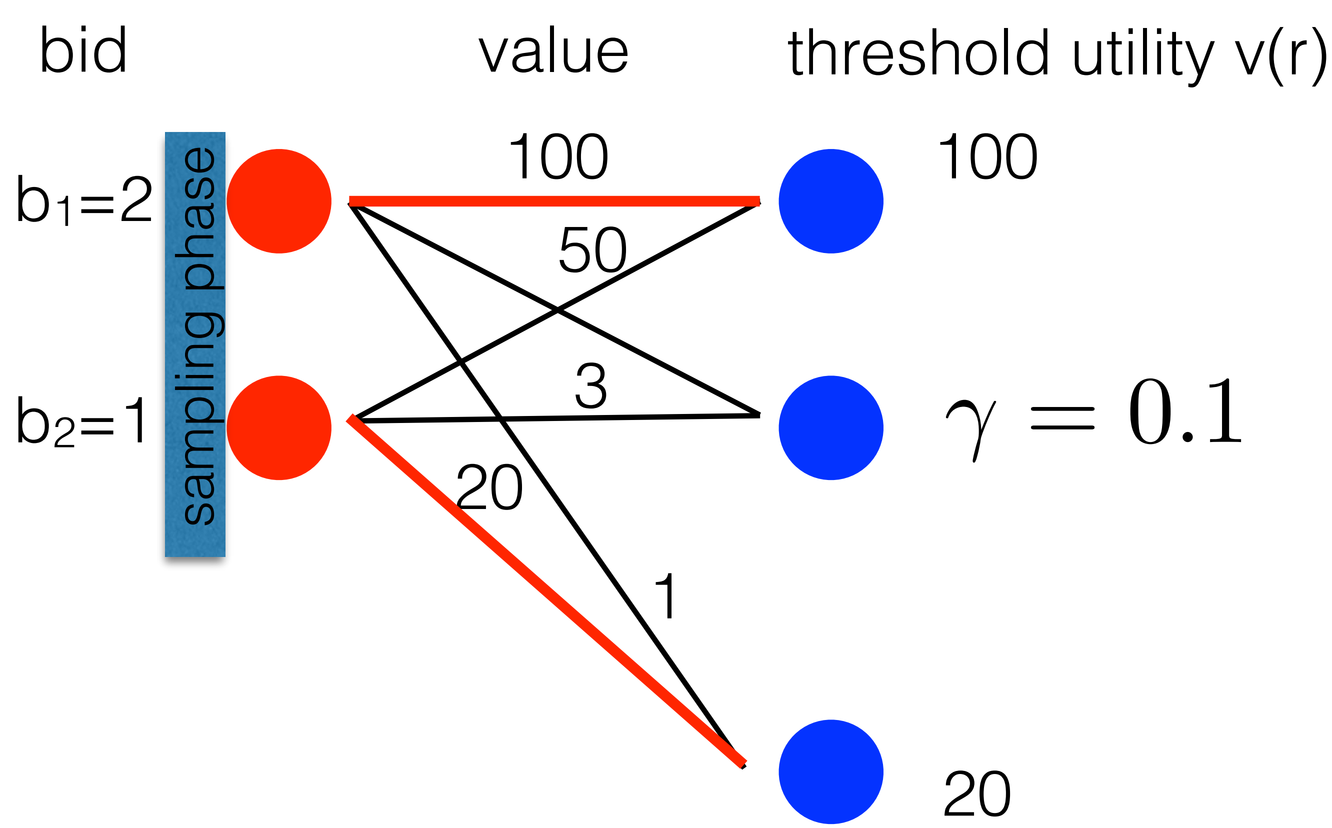
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **ON**

satisfies the payment budget constraint



Algorithm - OFF

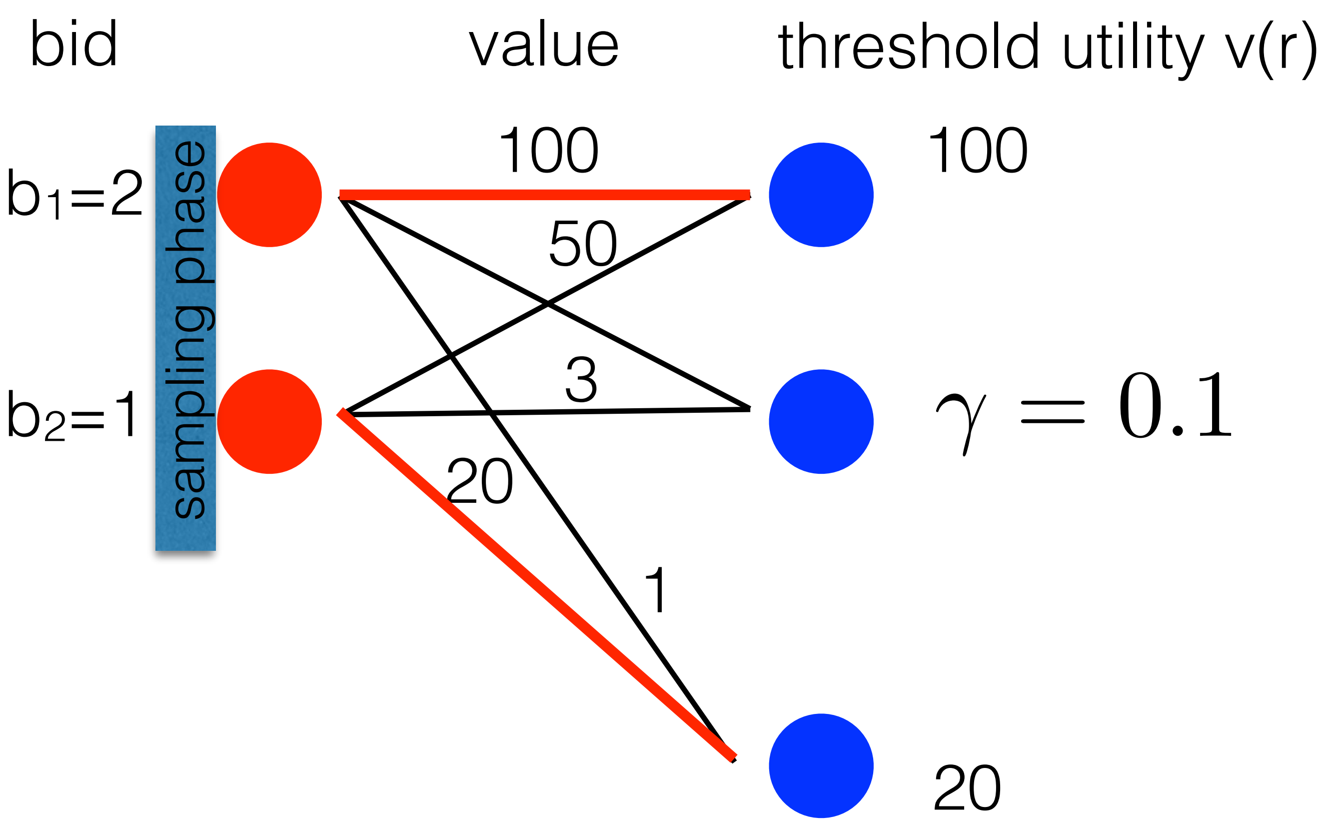
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of `ON`

satisfies the payment budget constraint



$\gamma = 0.1$

$\gamma \sum_{e \in M_{\mathrm{OFF}}} v(r) \leq B/\beta$

Algorithm - `OFF`

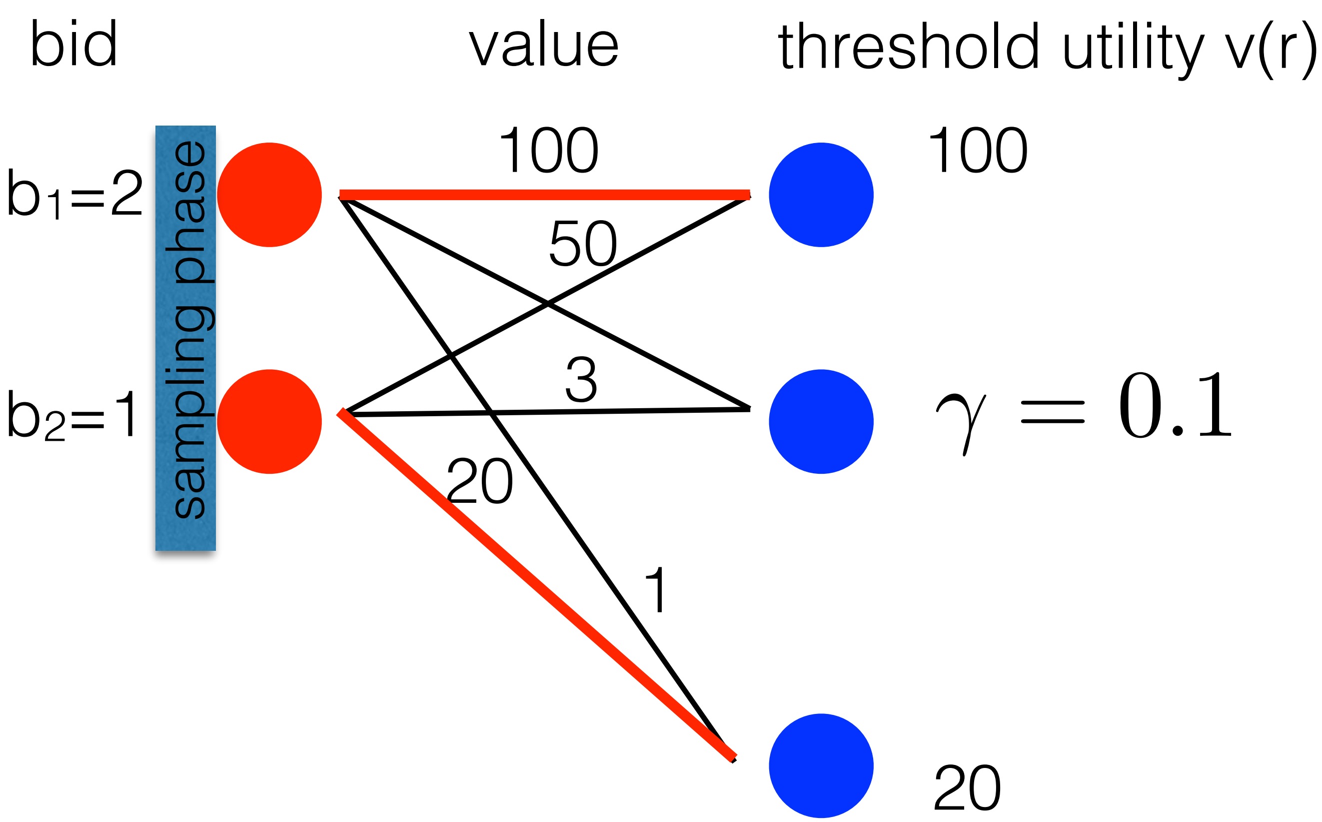
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **ON**

satisfies the payment budget constraint



$\gamma = 0.1$

$$\gamma \sum_{e \in M_{\mathrm{OFF}}} v(r) \leq B/\beta$$

$$p_\ell = \gamma \beta v(r)$$

Algorithm - OFF

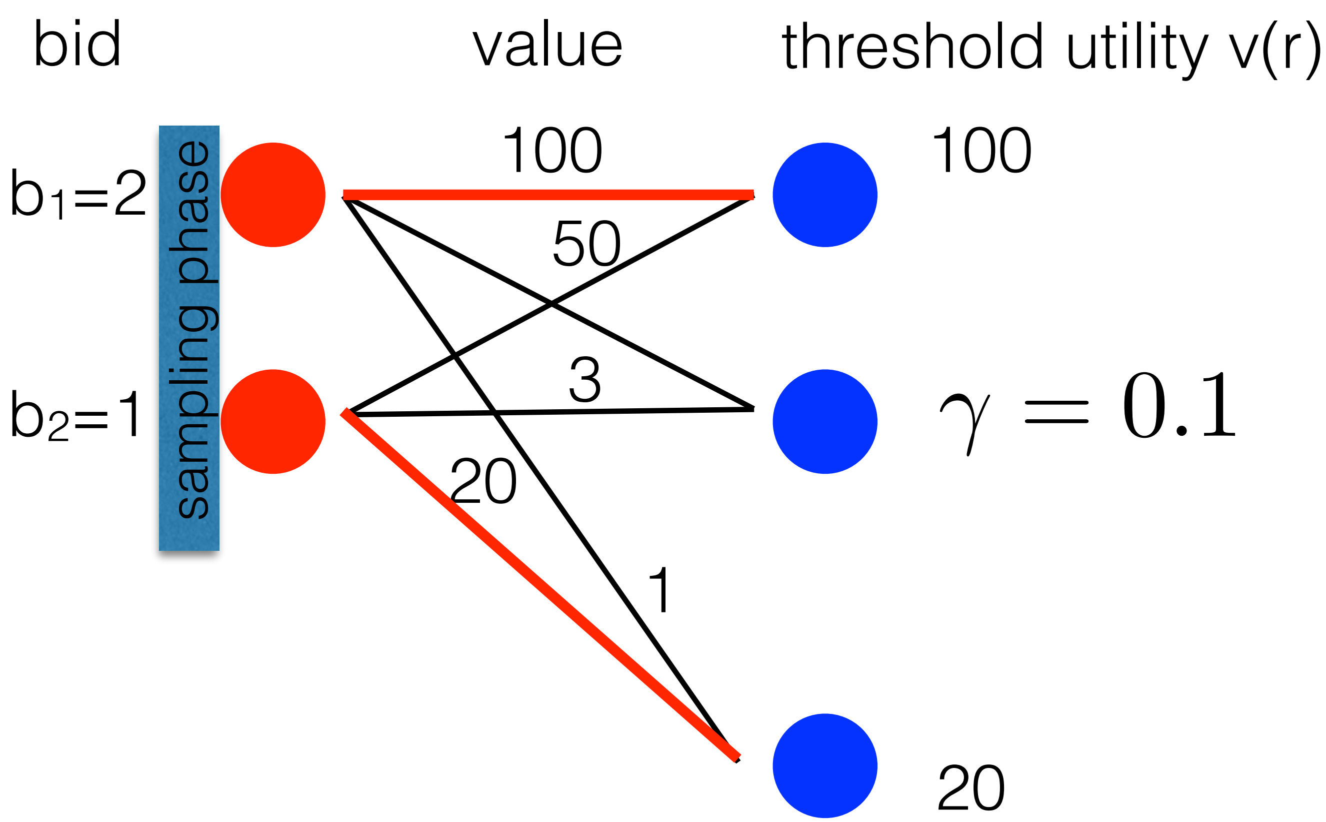
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of ON

satisfies the payment budget constraint



$$\gamma \sum_{e \in M_{\mathrm{OFF}}} v(r) \leq B/\beta$$

$$p_\ell = \gamma \beta v(r)$$

Since we are looking for matching

Algorithm - OFF

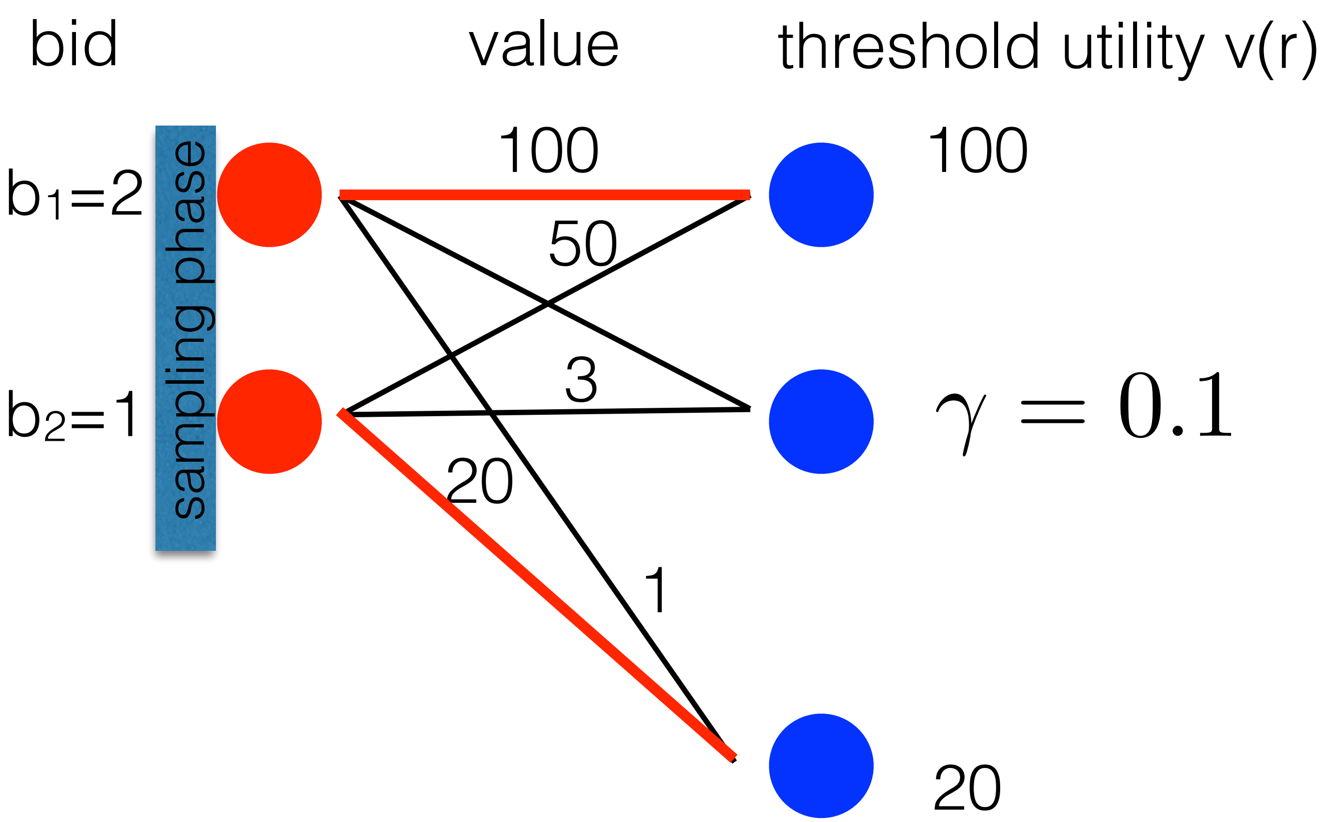
good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **ON**

satisfies the payment budget constraint



$\gamma = 0.1$

$$\gamma \sum_{e \in M_{\text{OFF}}} v(r) \leq B/\beta$$

$$p_\ell = \gamma \beta v(r)$$

Since we are looking for matching

$$\sum_{e=(\ell,r) \in M_{\text{ON}}} p_\ell \leq \gamma\beta \sum_{e \in M_{\text{OFF}}} v(r) \leq B$$

Algorithm - OFF

good Graph $\mathrm{G}(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

$\mathrm{M}(\gamma)$ be greedy matching over $G(\gamma)$

Find largest $\gamma \sum_{e \in M(\gamma)} v(e) \leq B$

Properties of **ON**

For truthfulness we just check

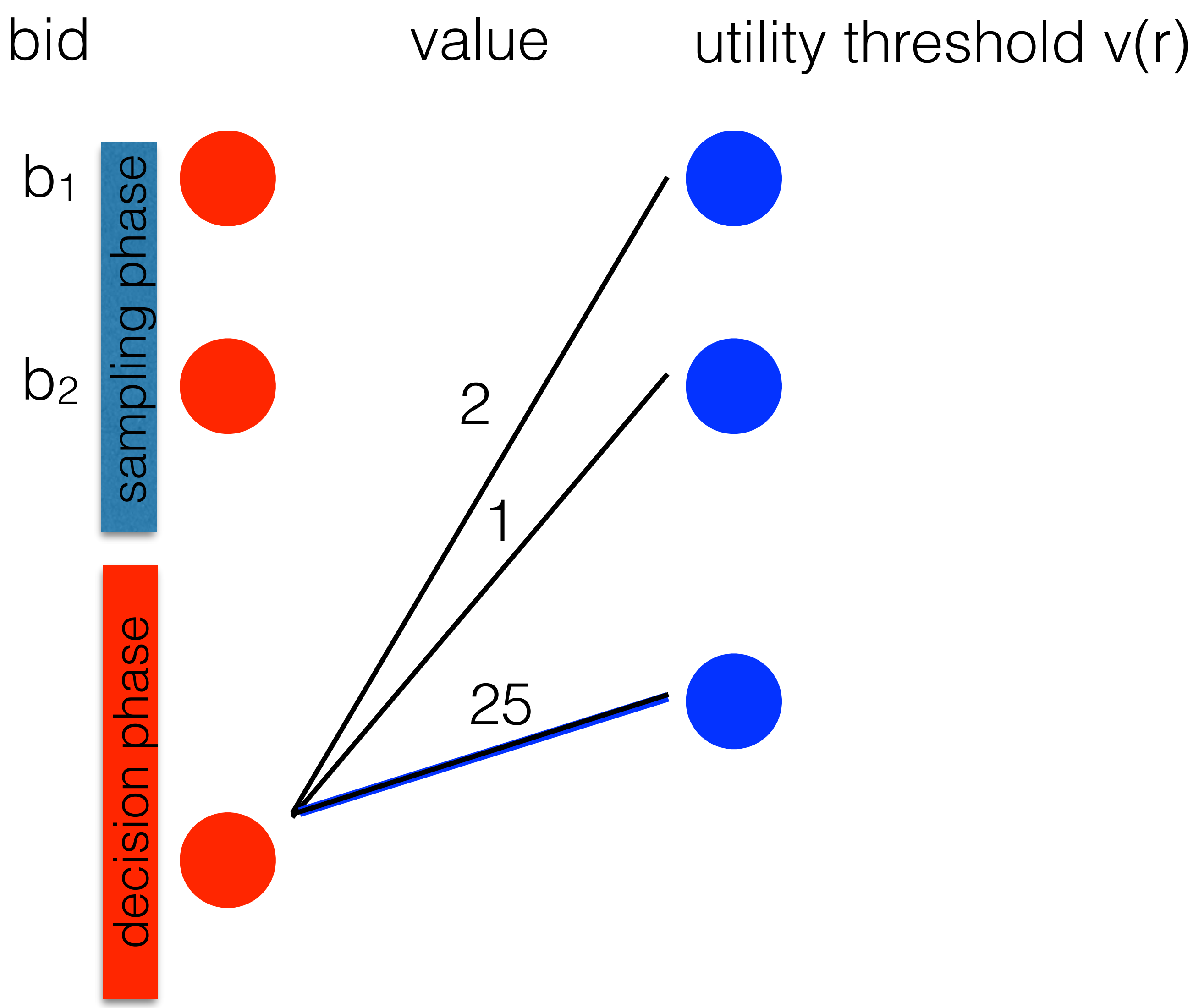
R. Myerson

Monotonicity - if an agent is selected with bid b, then he is always selected if he bids below b

Critical Price - there exists a threshold price such that if an agent bids above it, he is never selected

Properties of `ON` – truthfulness

γ and threshold utility v(r) obtained from **`OFF`**



on arrival of a new left vertex

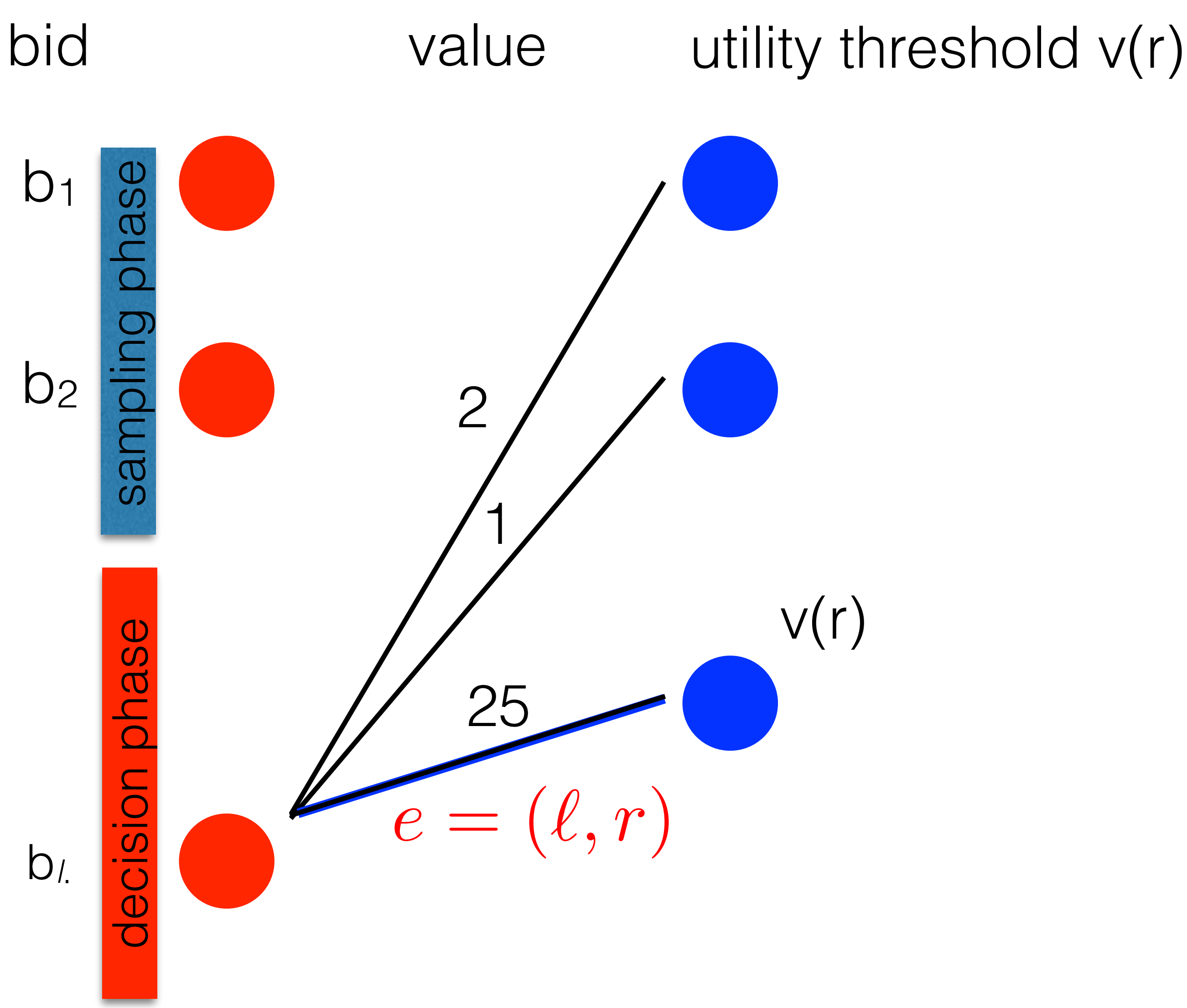
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of **ON** – truthfulness

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

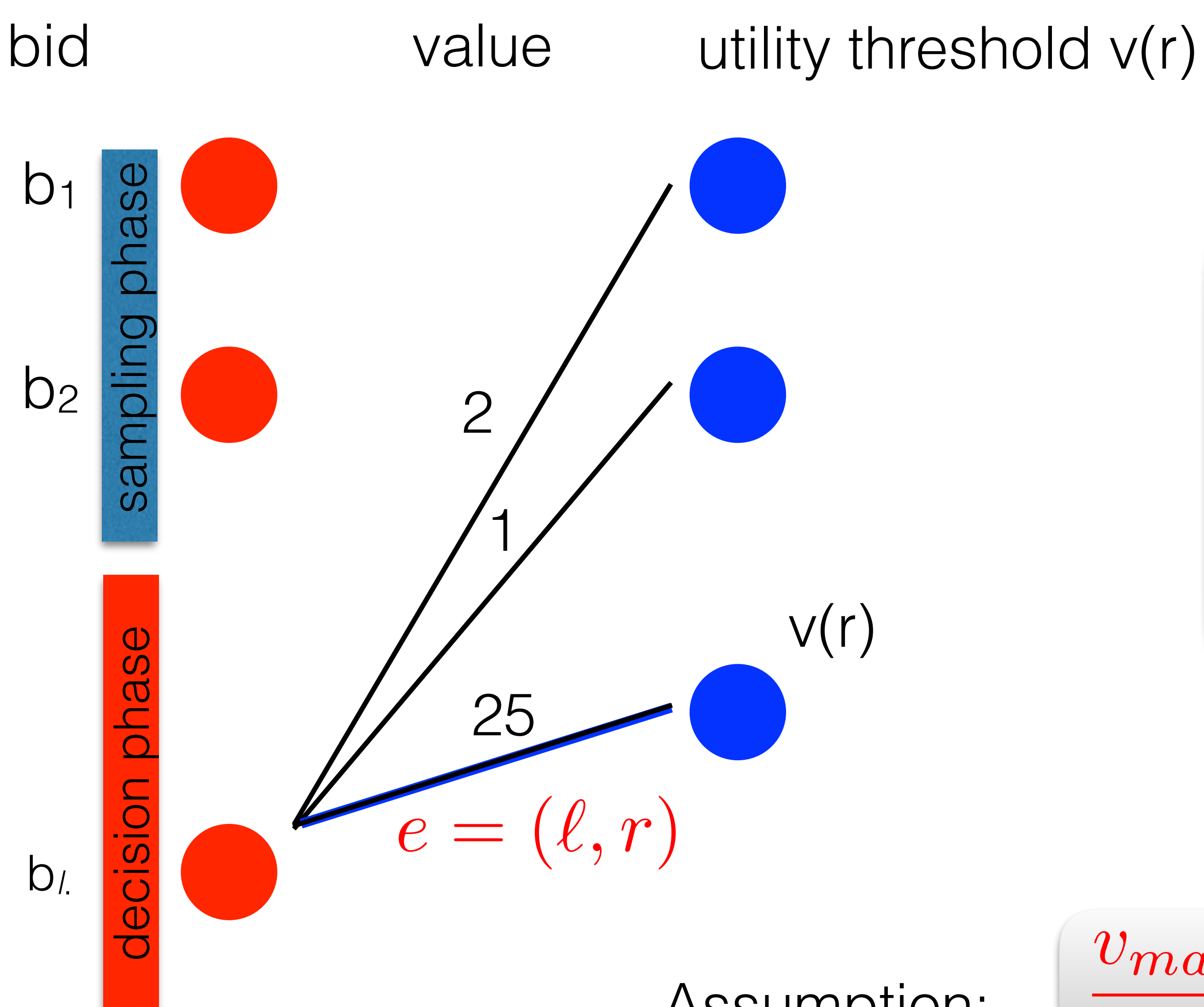
remove all edges with **bid to benefit** ratio > γ

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Properties of ON – truthfulness

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

remove all edges with **bid to benefit** ratio > γ

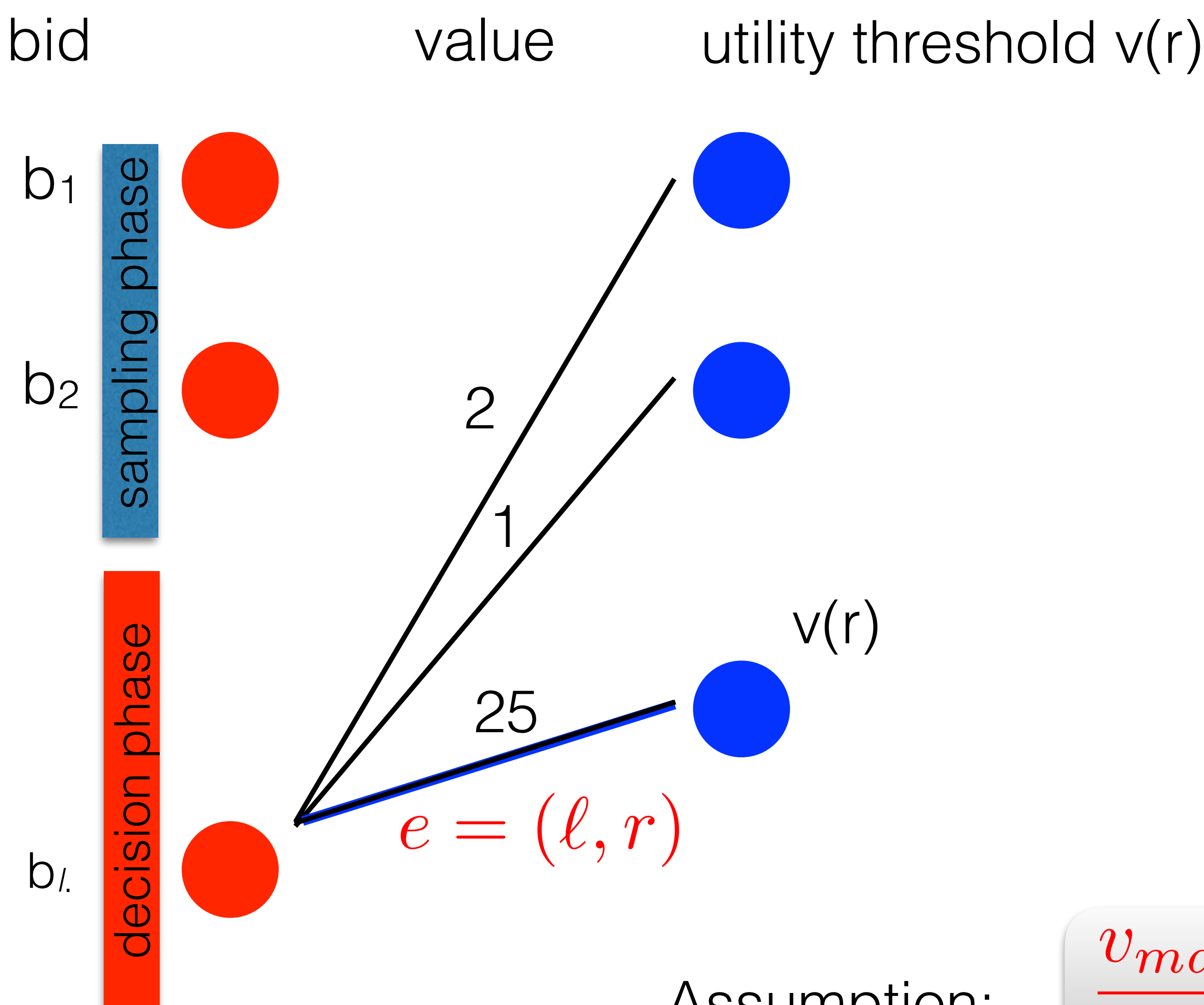
match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Assumption: $\frac{v_{max}}{v_{min}} \leq \beta$ $\quad u(e) \leq \beta v(r)$

Properties of ON – truthfulness

γ and threshold utility v(r) obtained from **OFF**



on arrival of a new left vertex

remove all edges with **bid to benefit** ratio $> \gamma$

match the right vertex with max w > utility thres.

Payment for selected left vertex $p_\ell = \beta\gamma v(r)$

Assumption: $\boxed{\frac{v_{max}}{v_{min}} \leq \beta}$ $\qquad u(e) \leq \beta v(r)$

If $b_l > p_l$ then bid to benefit ratio of that edge $\frac{b_\ell}{u(e)} > \gamma$

Important Extension

Important Extension

If instead of hard budget constraint, its an expected budget constraint

$$\mathbf{E}\{p_{1} + p_{2} + \ldots + p_{m}\} < B$$

Important Extension

If instead of hard budget constraint, its an expected budget constraint

$$\mathbf{E}\,\{p_{1} + p_{2} + \ldots + p_{m}\} < B$$

- a small variation of proposed algorithm is constant (288) competitive

Peynet Museum' Antibes