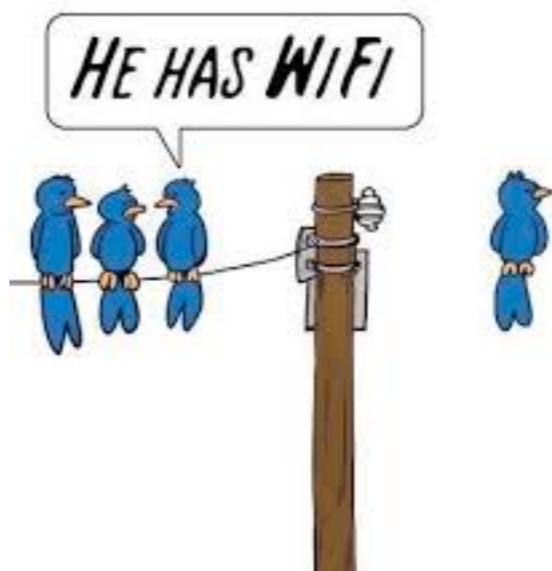


# Graph Matchings and Wireless Communication

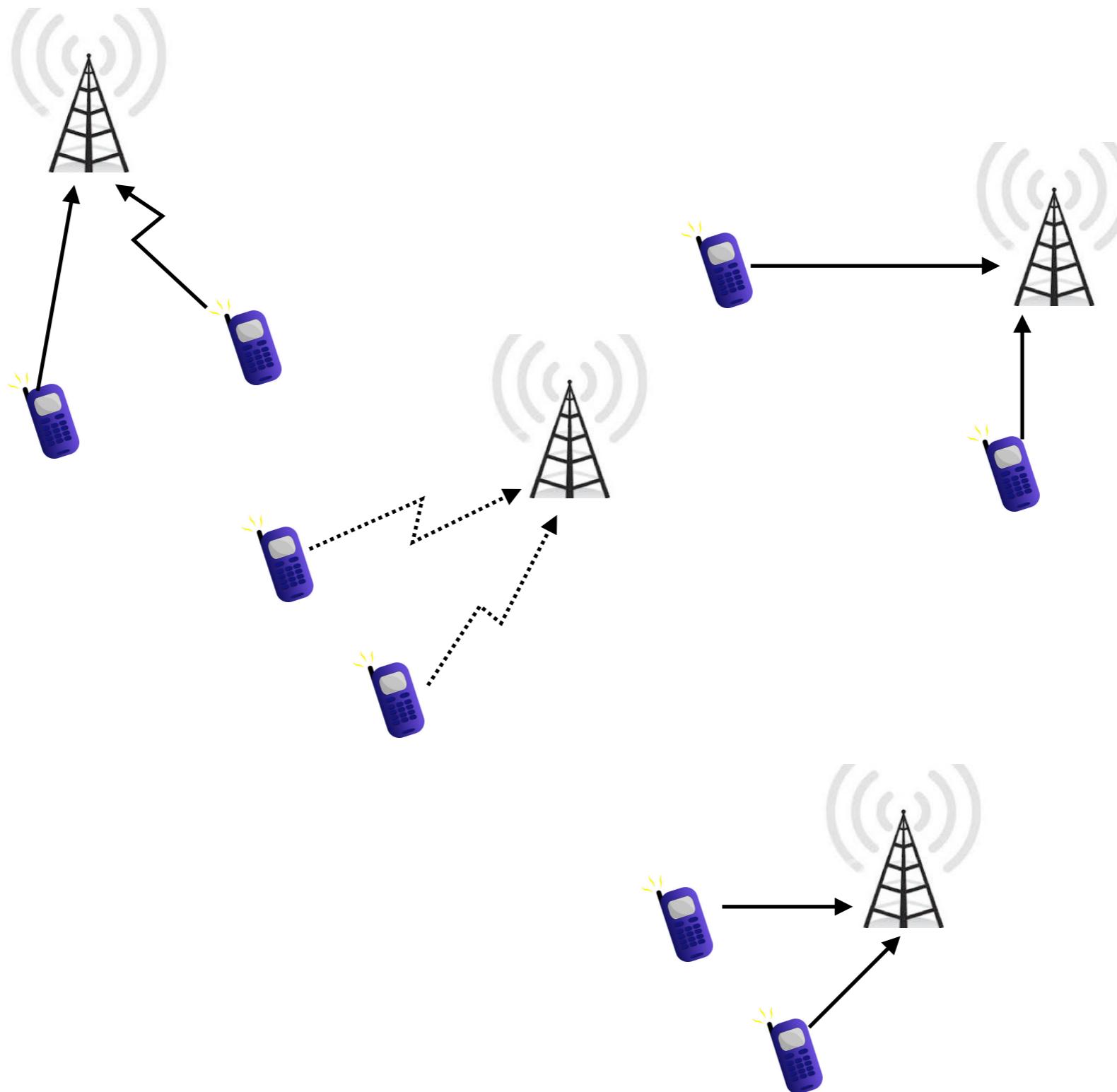
Rahul Vaze

# Graph Matchings and Wireless Communication

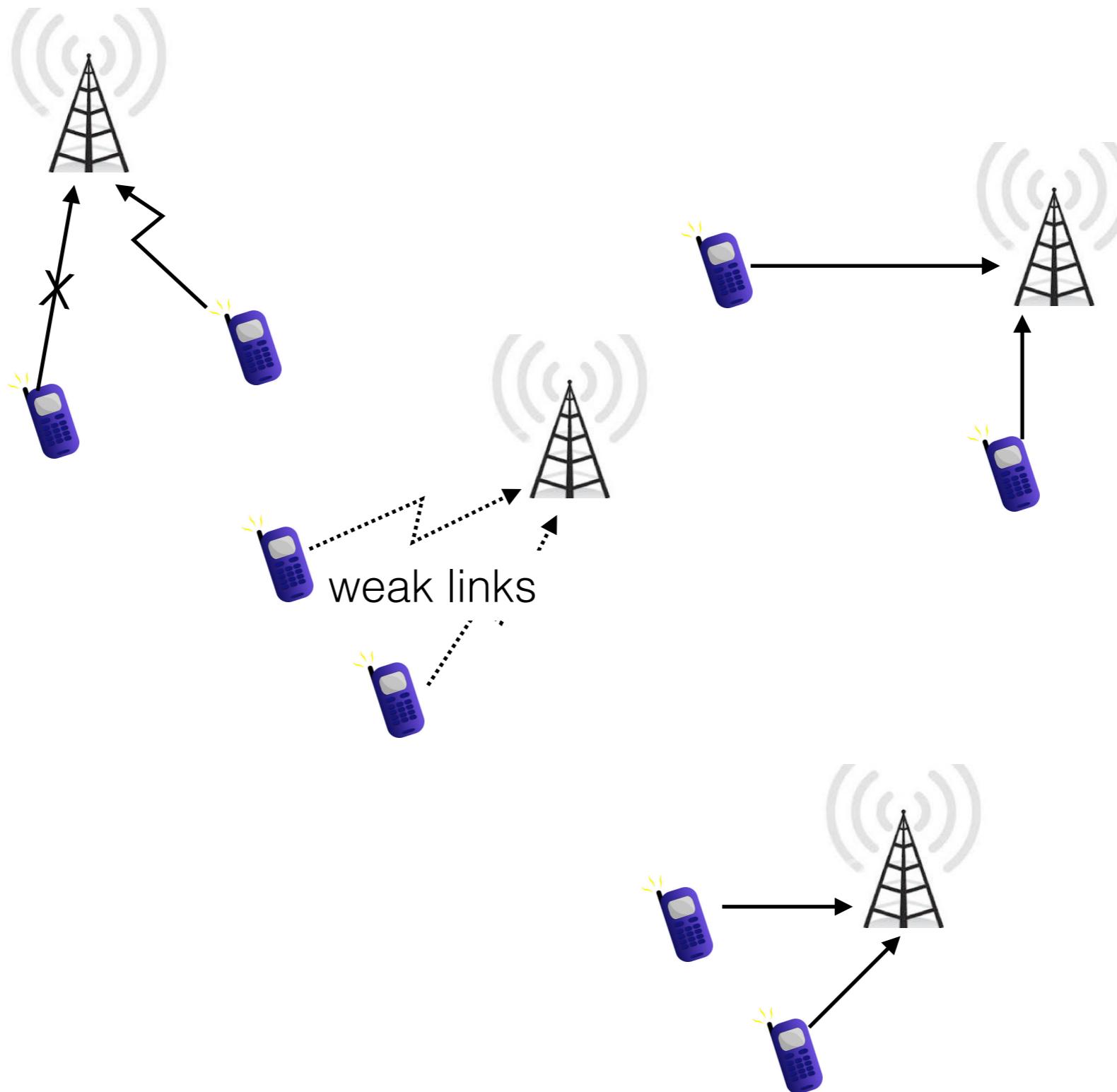


Rahul Vaze

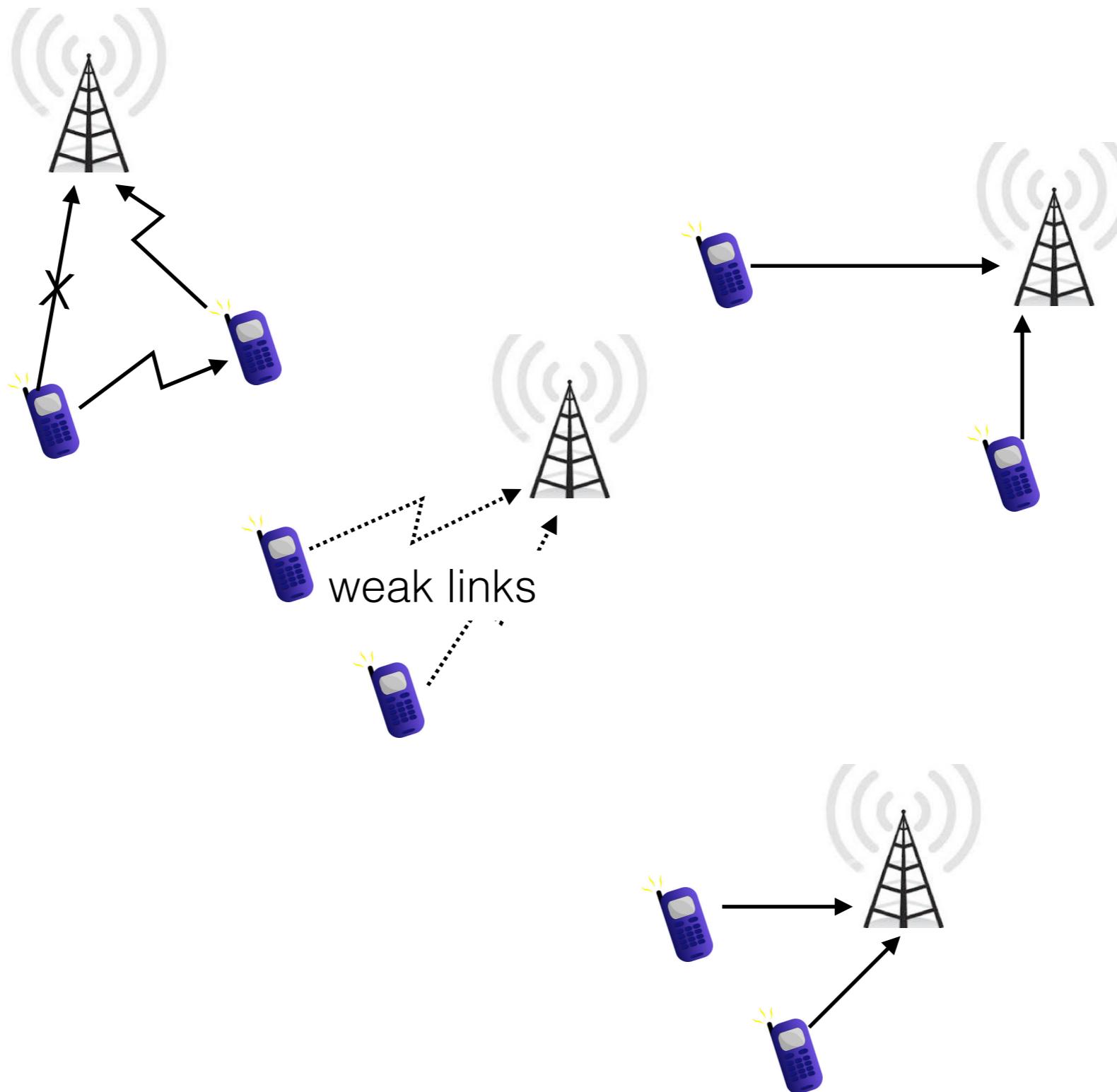
# Device-2-Device Communication



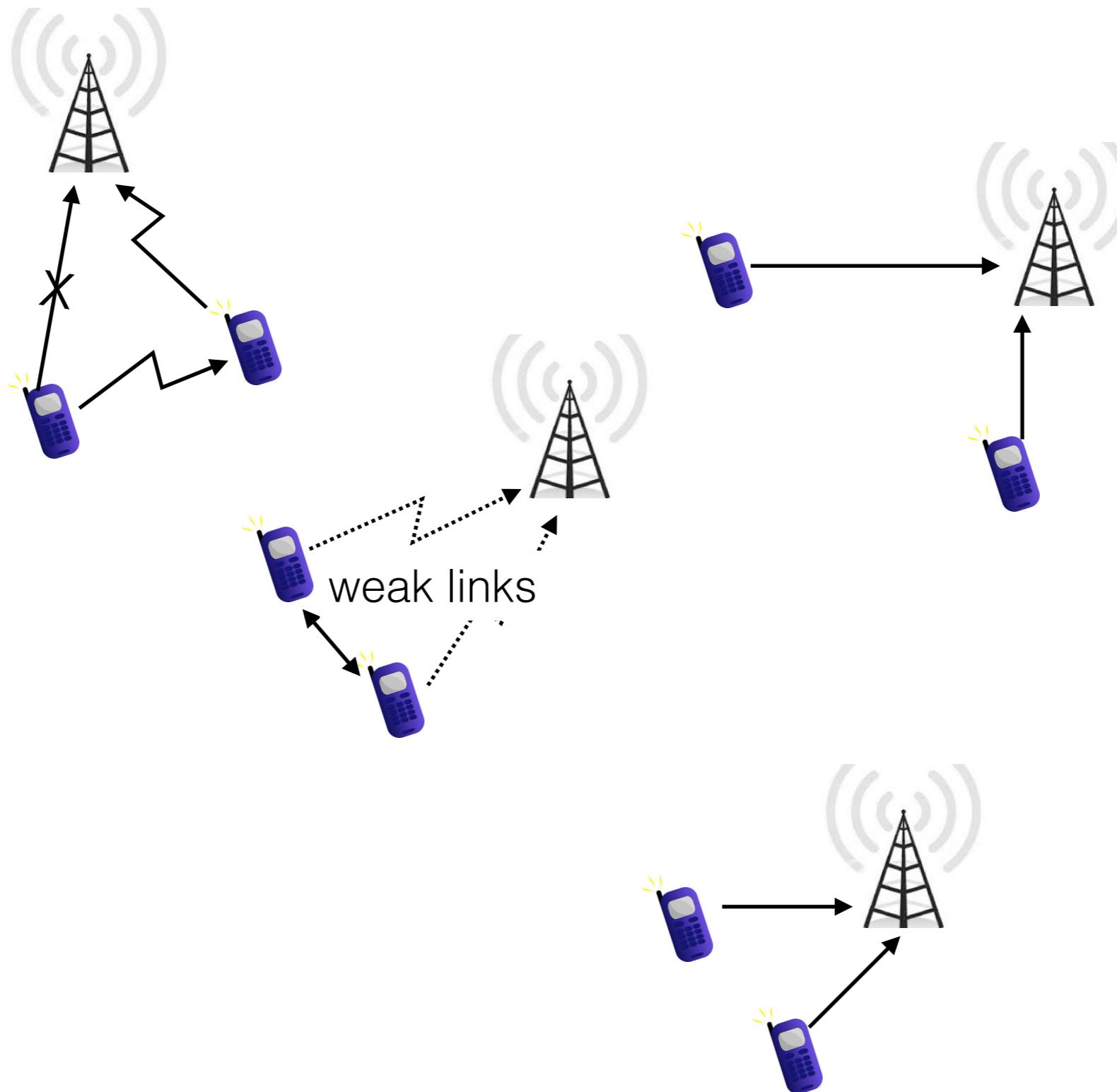
# Device-2-Device Communication



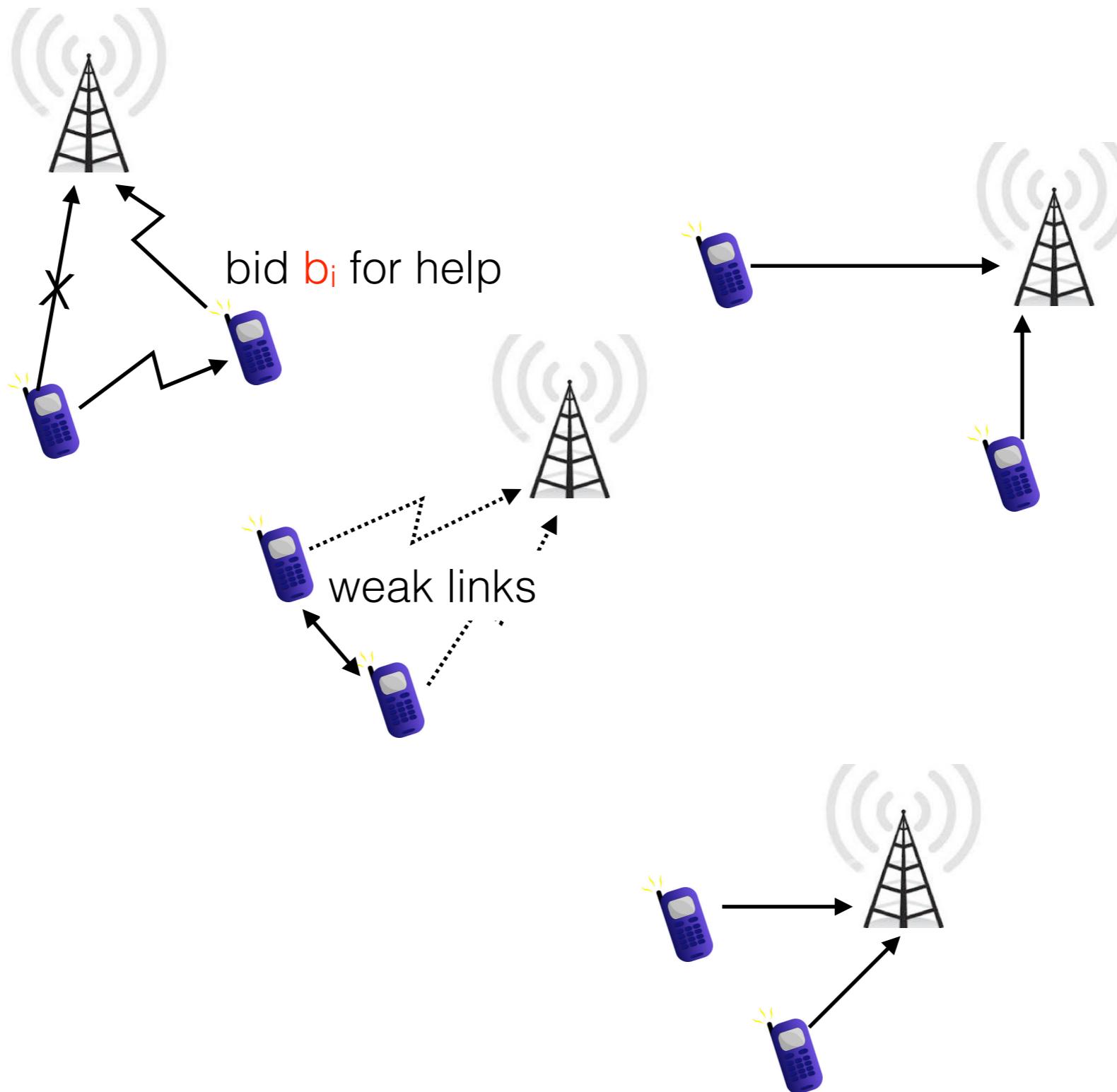
# Device-2-Device Communication



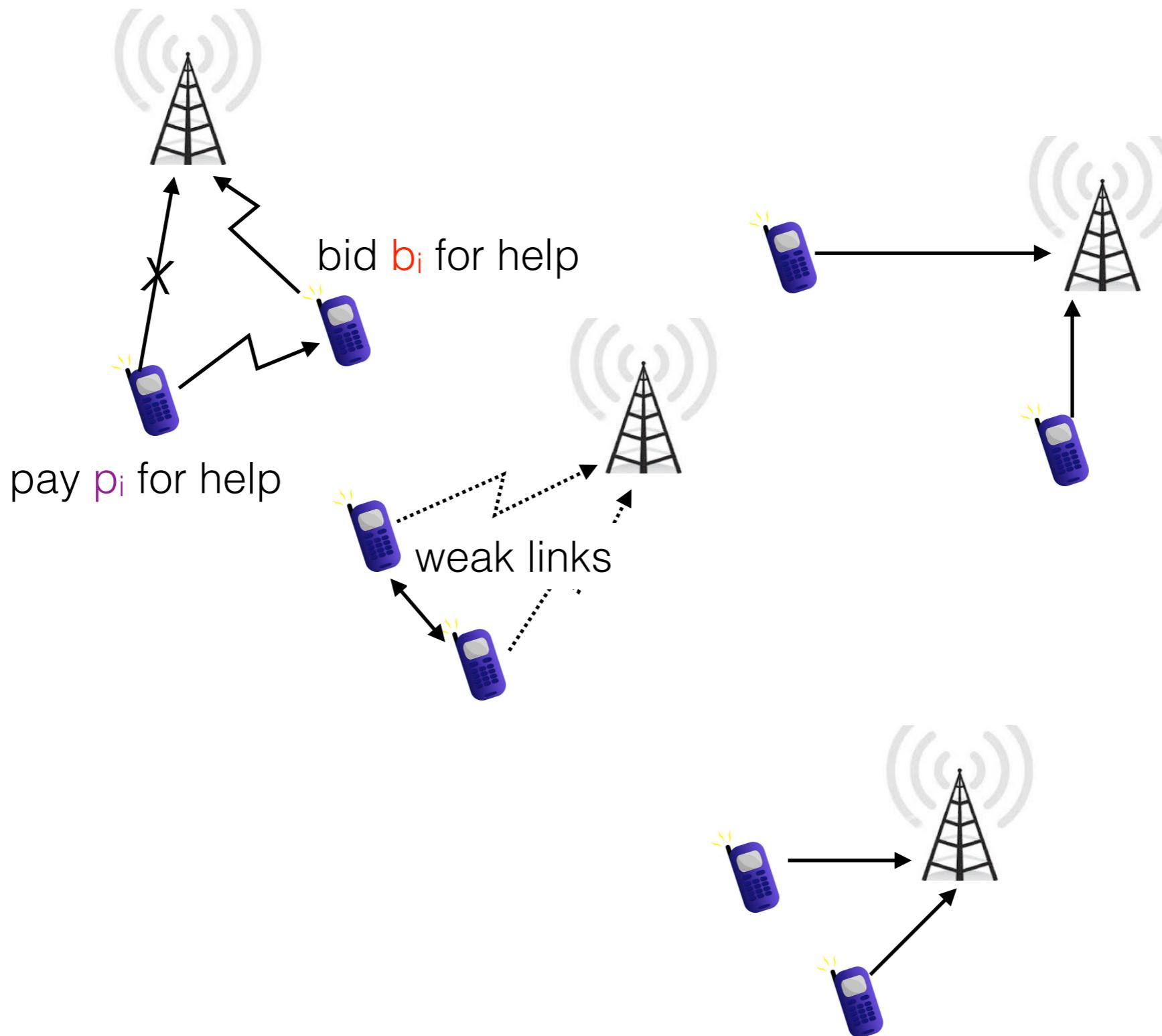
# Device-2-Device Communication



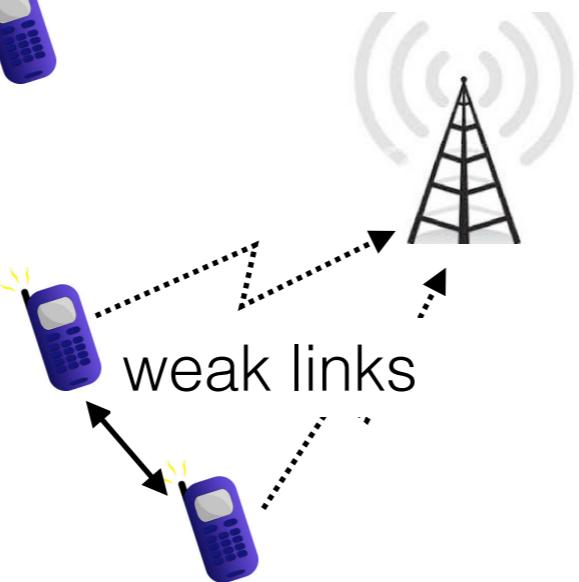
# Device-2-Device Communication



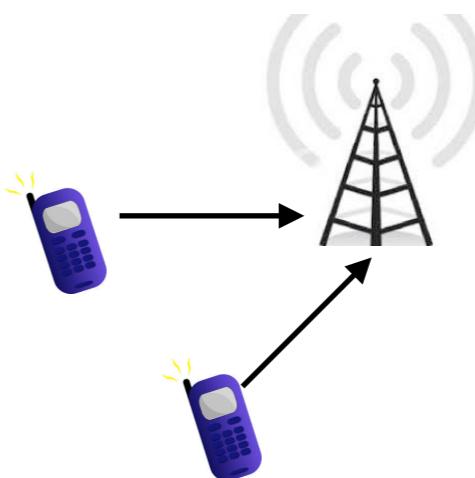
# Device-2-Device Communication



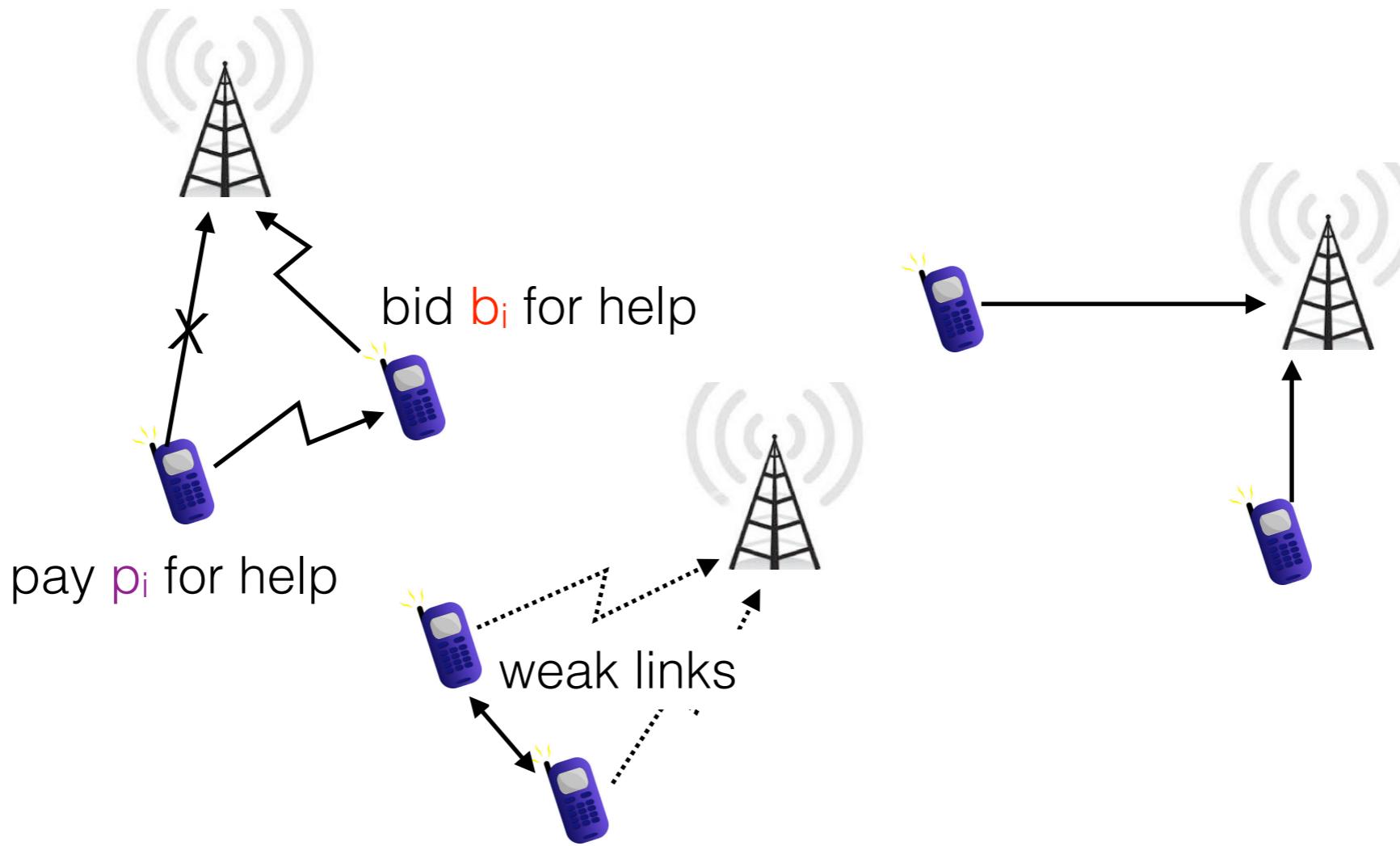
# Device-2-Device Communication



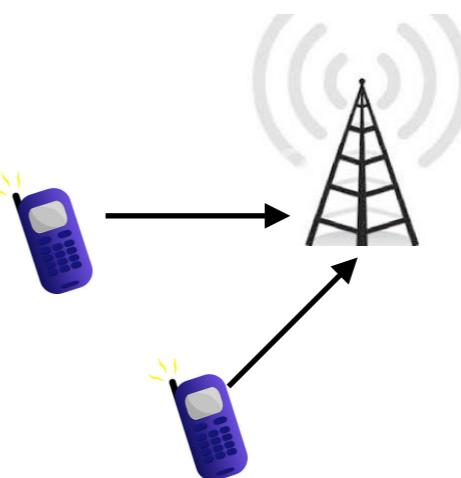
mechanism to avoid cheating



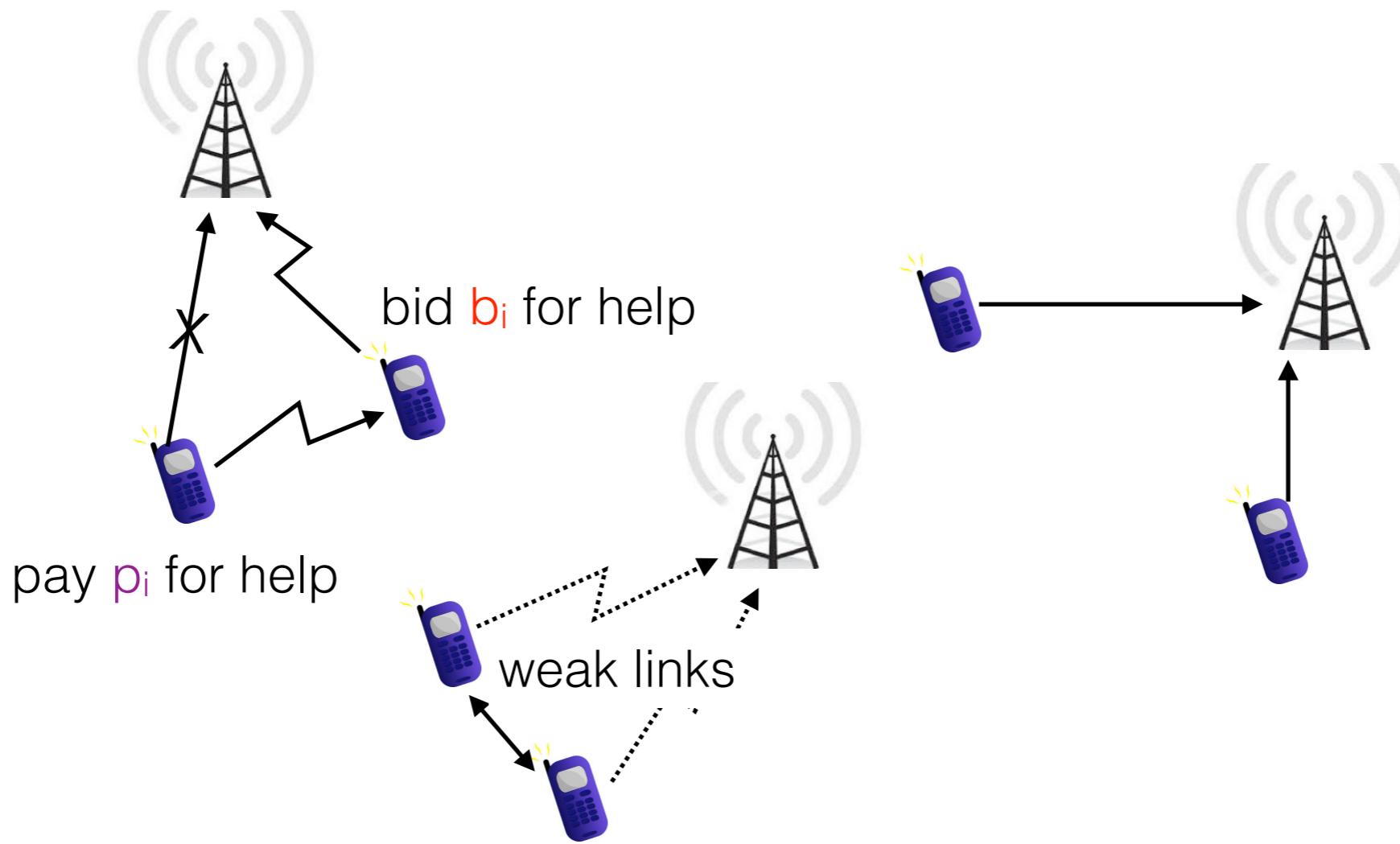
# Device-2-Device Communication



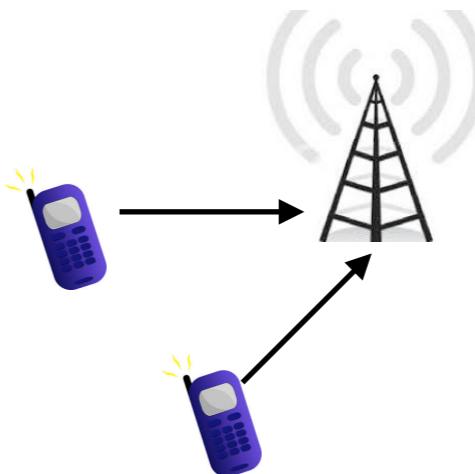
mechanism to avoid cheating  
ensure maximum throughput



# Device-2-Device Communication



mechanism to avoid cheating  
ensure maximum throughput

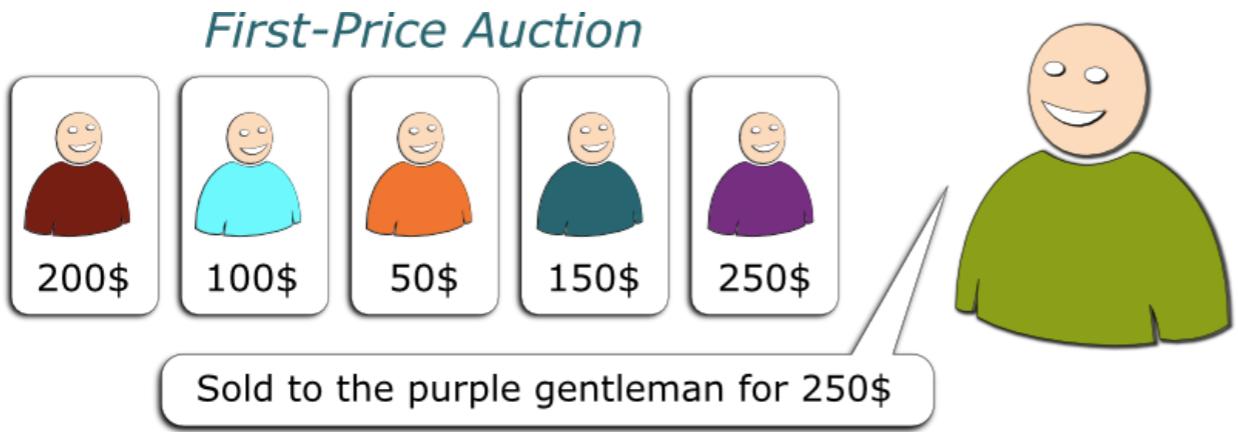


**Find ONLINE optimal helper association and incentive rule that is truthful**

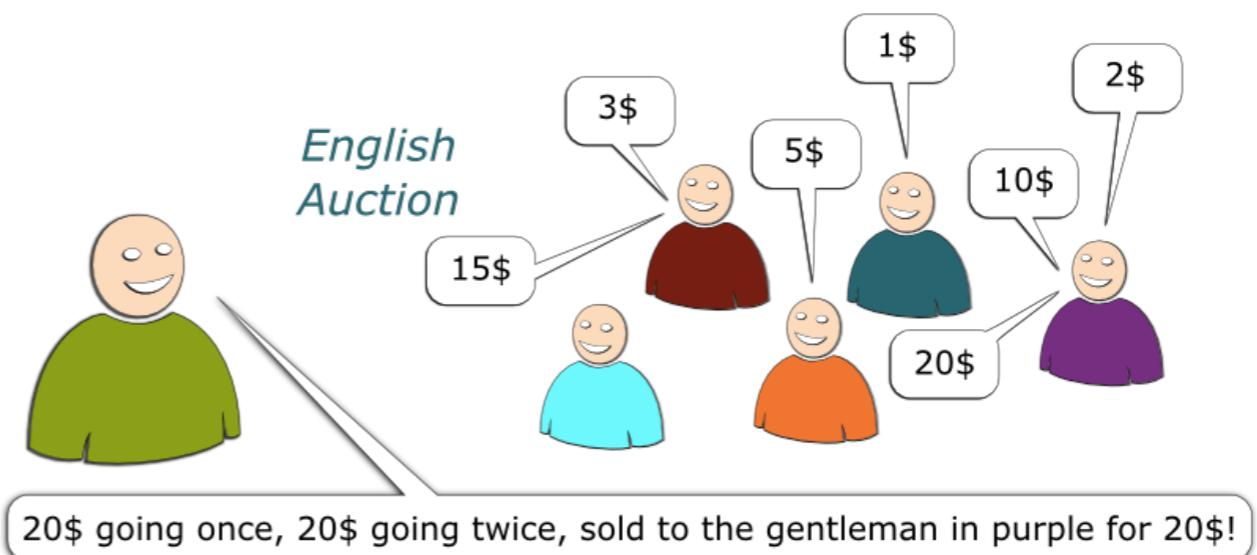
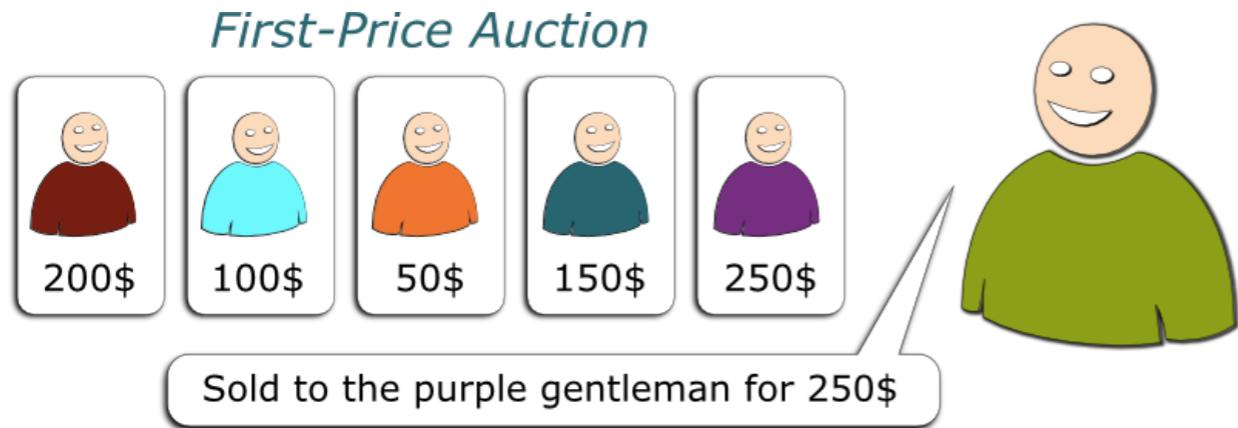
# Truthful Auction



# Truthful Auction



# Truthful Auction



small detour

at the height of their popularity.

For more stories

For more stories

## The mathematics behind a perfect match

**I**t's a problem that has had many names—the secretary problem, the Sultan's dowry problem and the optimal stopping problem, but it's also one that can be used to decide when to stop dating and settle down. "Skepticism works and you might forge the chance of a more perfect match later on. But it's too long to wait around, and all the good ones might be gone. You don't want to marry the first person you meet, but you also don't want to wait too long," writes Anna Svitanski, as she describes the mathematics behind finding the best spouse.

The spouse problem is an old one, and has a solution—a simple mathematical rule. "In the scenario, you're choosing from a set number of options. For example, let's say there is a total of 10 potential mates who you could seriously date and settle down with in your lifetime. If you could only see them all together at the same time, you'd have no problem picking out the best. But this isn't how a lifetime of dating works, obviously," writes Svitanski.



**MAGIC NUMBER:** Dump the first 37% of the suitors you may have over your lifetime.

The problem, of course, is that nations don't appear together for months. And of course, once you've rejected someone, it can be awkward—*if not impossible*, to go back.

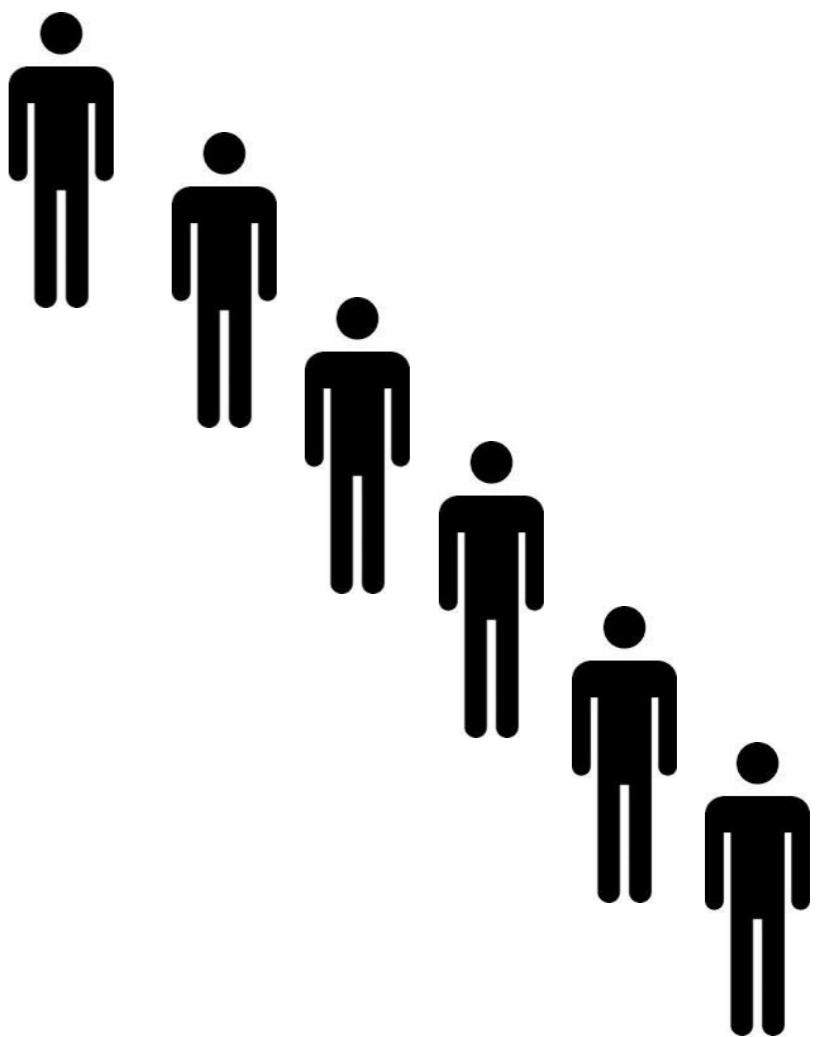
The answer lies in probability—and its mathematics. To find the perfect partner, you have to date and reject the first 37% of the group you've met. This means that the next person who is better than anyone you've ever dated before. Naturally, for this to work in real life, you need to know how many suitors you will have, but somehow make a guess.

There are risks, of course. There's the risk, for example, that the first person you date really is your perfect partner. If you follow the rule, you'll reject that person right away. And as you continue to date other people, no one will ever measure up to your first love, and you'll end up rejecting everyone, and end up alone with your cats. (Of course, some people may find cats preferable to boyfriends or girlfriends anyway), writes Svitanski.

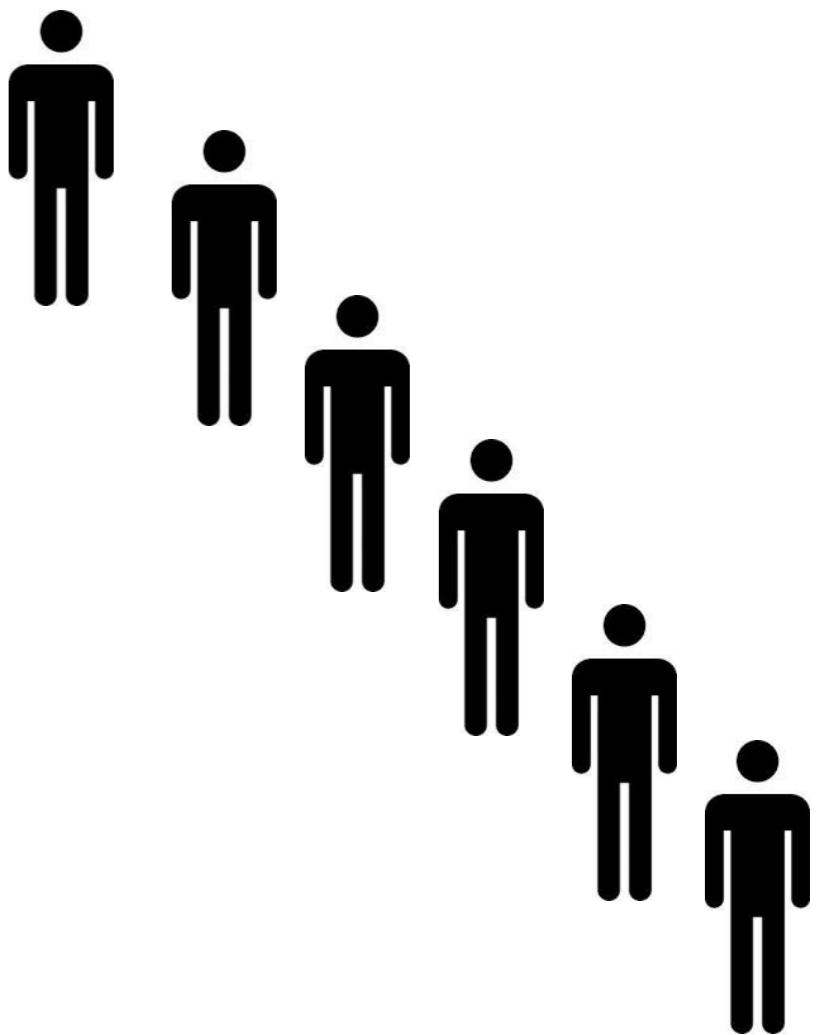
For more stories

how many to date before committing !

# Hiring impatient staff



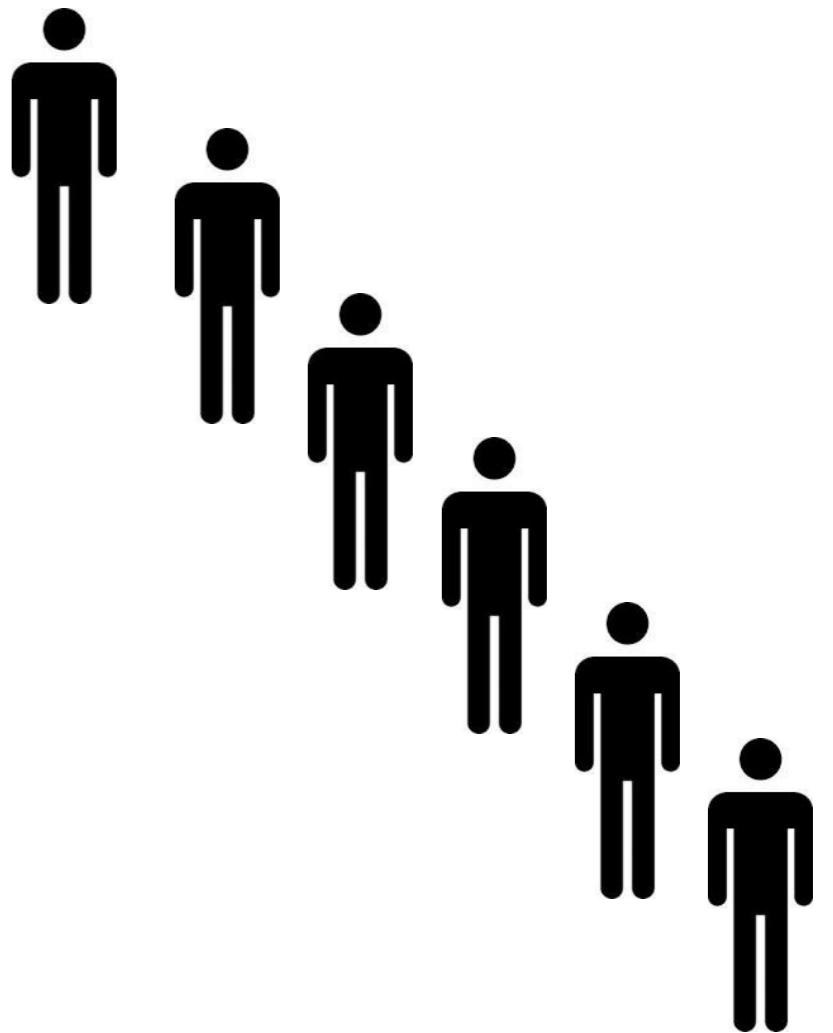
# Hiring impatient staff



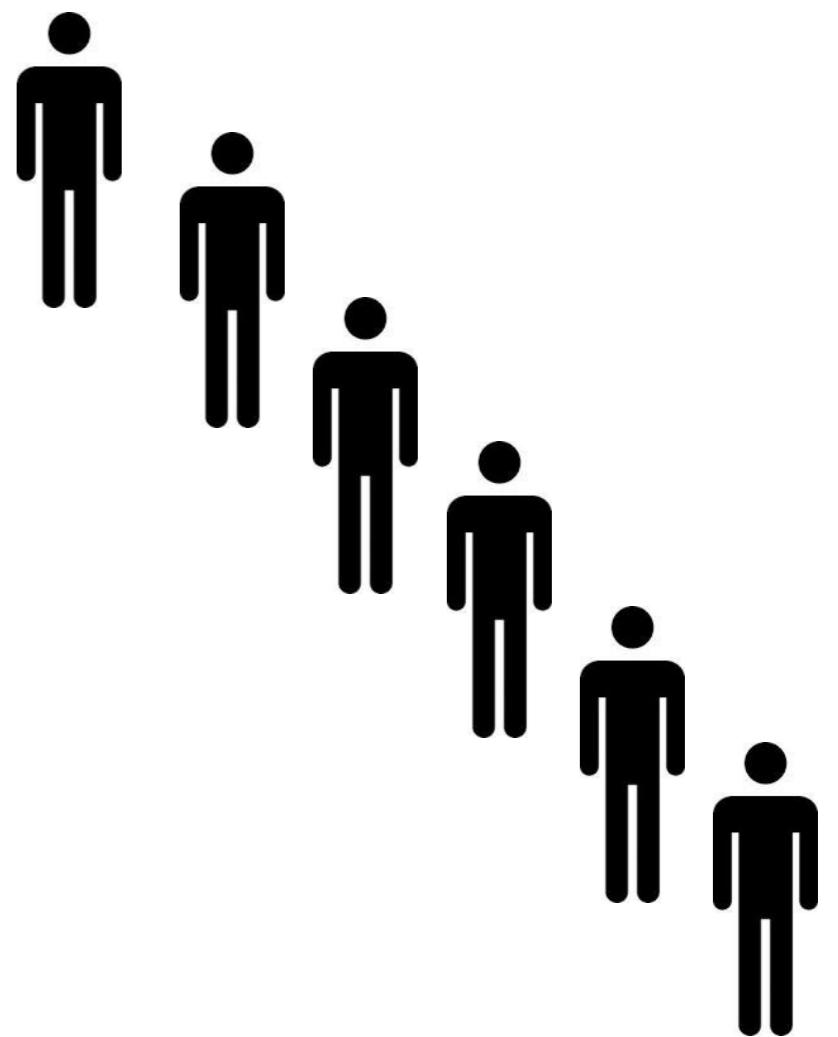
# Hiring impatient staff



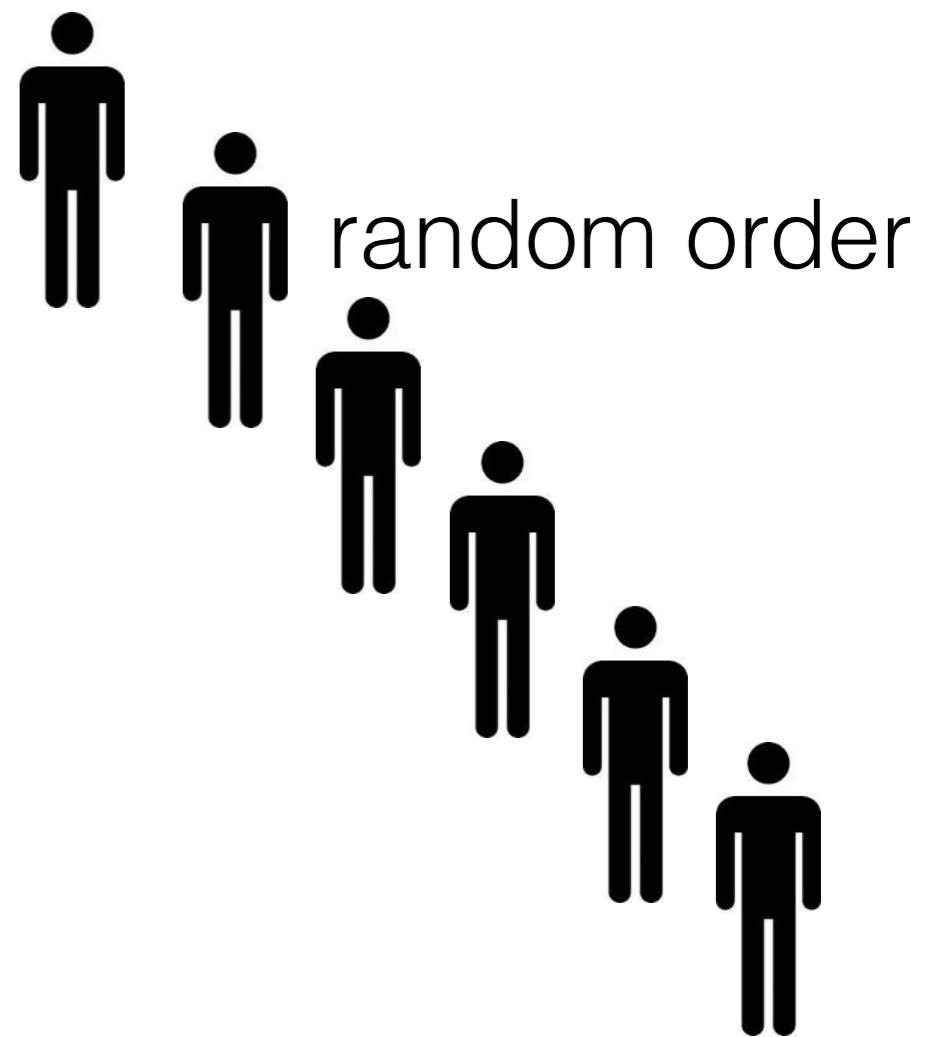
prob. of choosing best candidate is  $1/n$



# Hiring staff - not adversarial



# Hiring staff - not adversarial

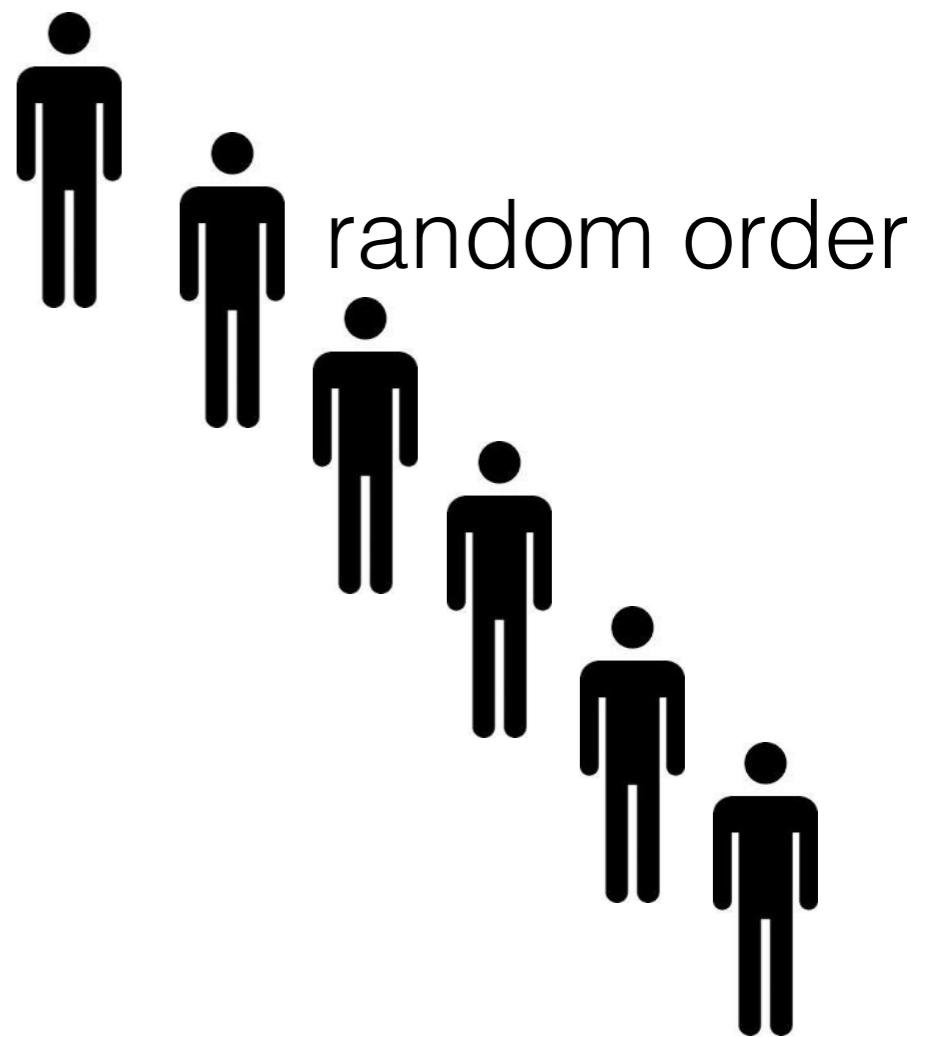


# Hiring staff - not adversarial

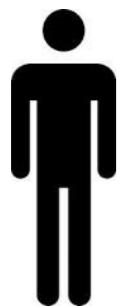


sampling phase

first half

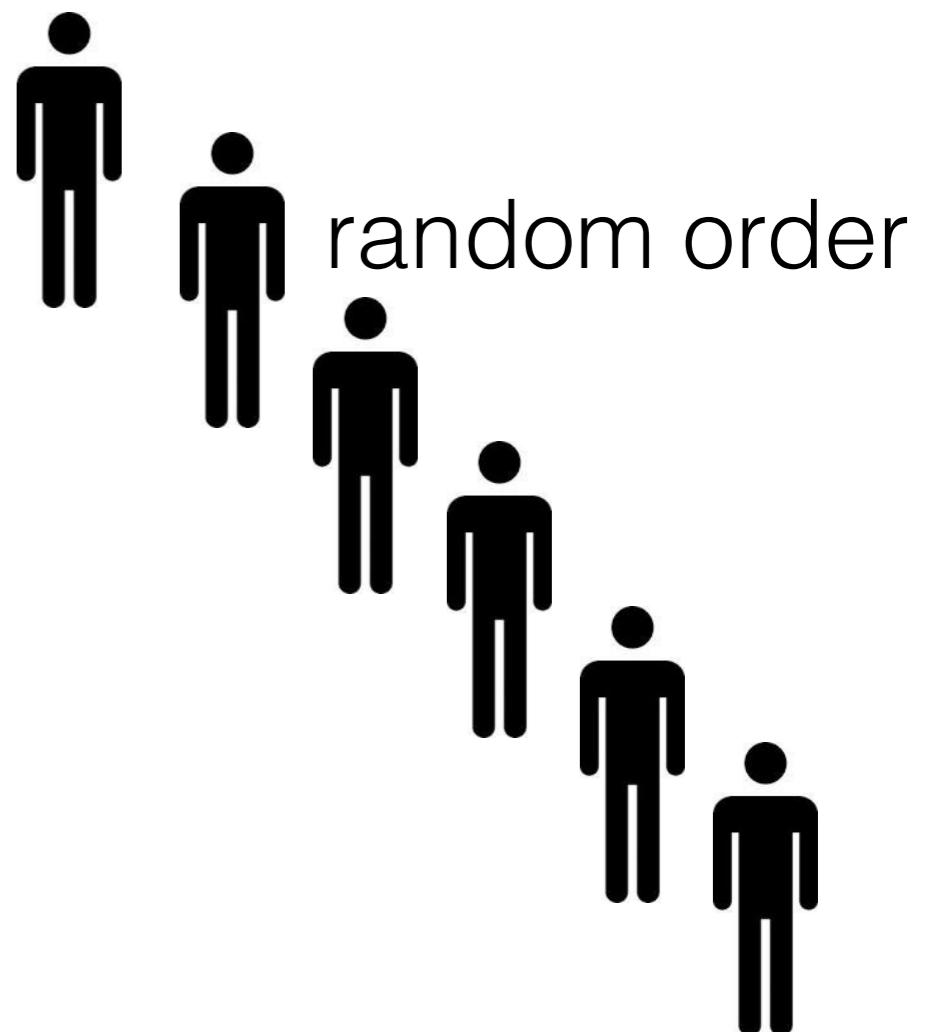


# Hiring staff - not adversarial

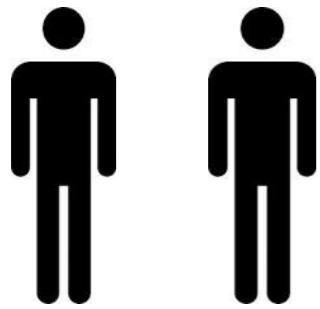


sampling phase

first half

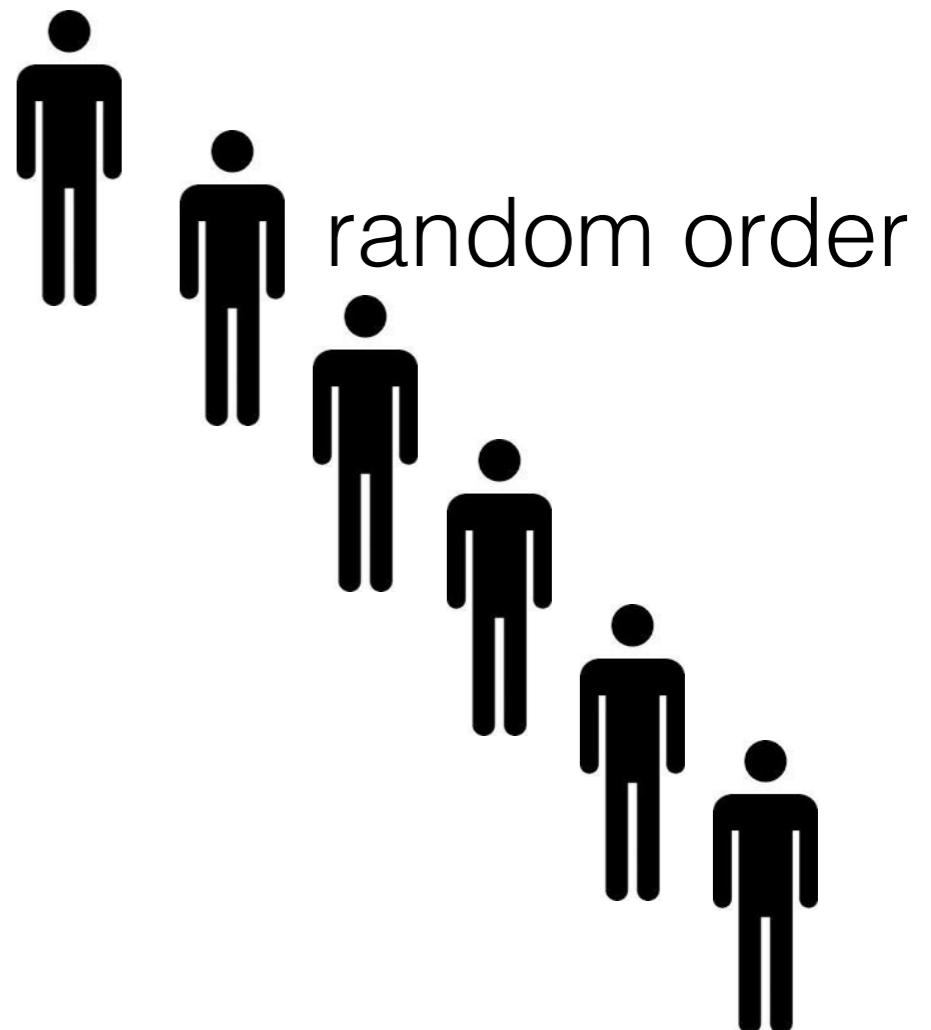


# Hiring staff - not adversarial

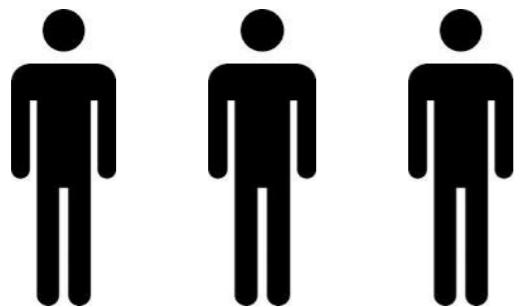


sampling phase

first half

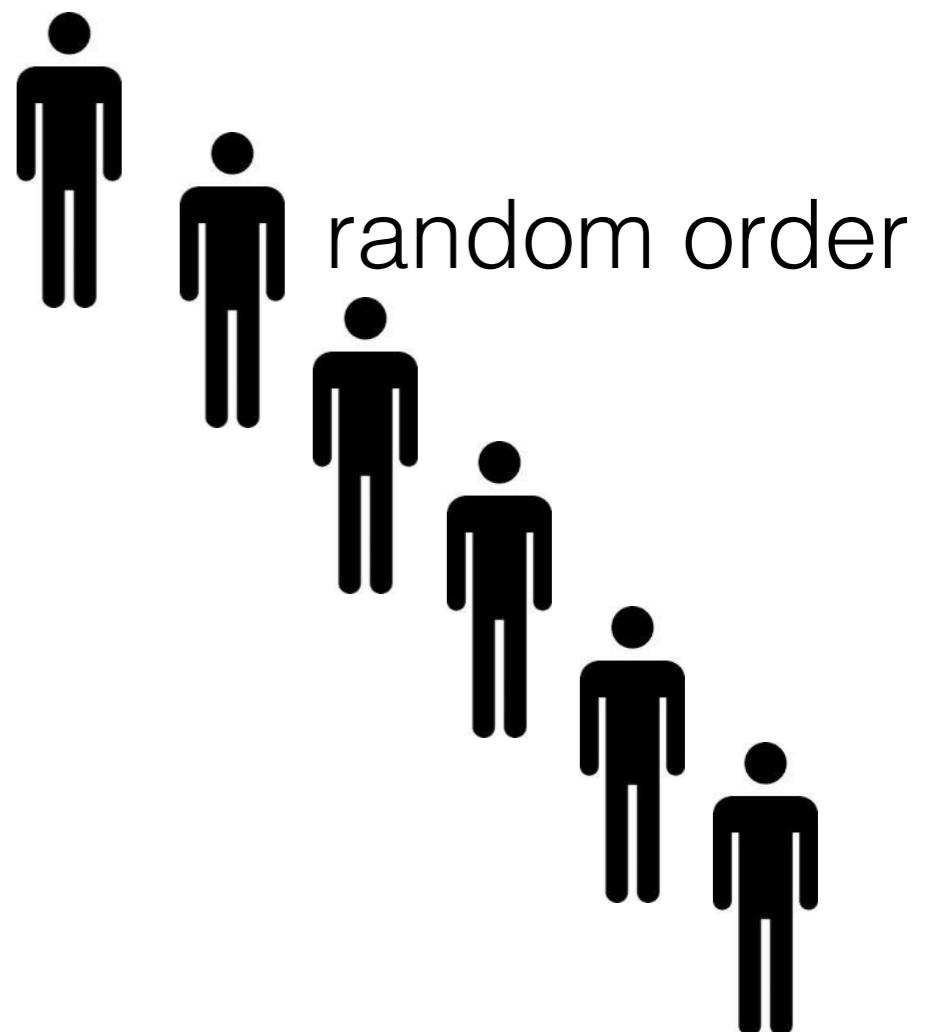


# Hiring staff - not adversarial

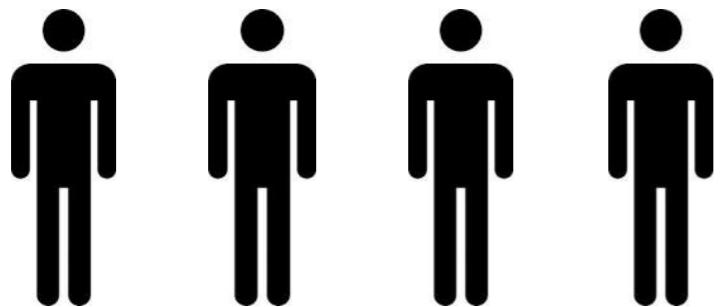


sampling phase

first half

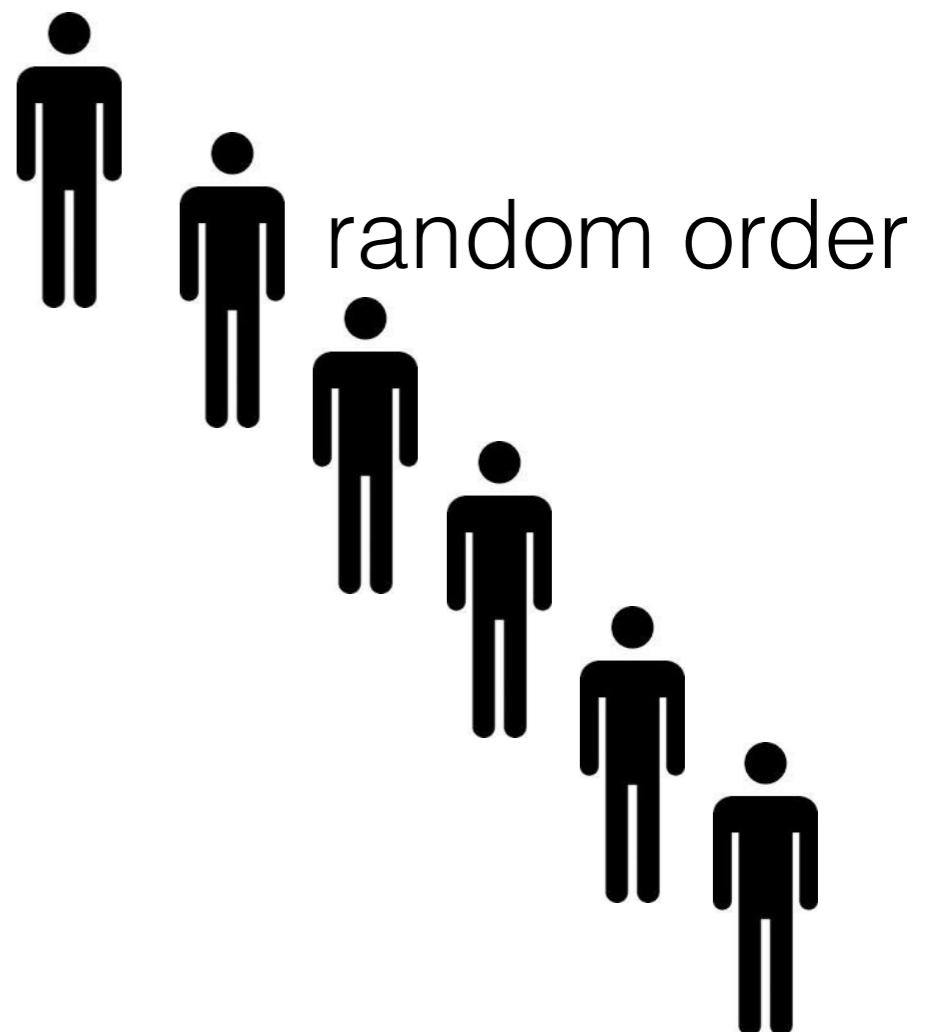


# Hiring staff - not adversarial

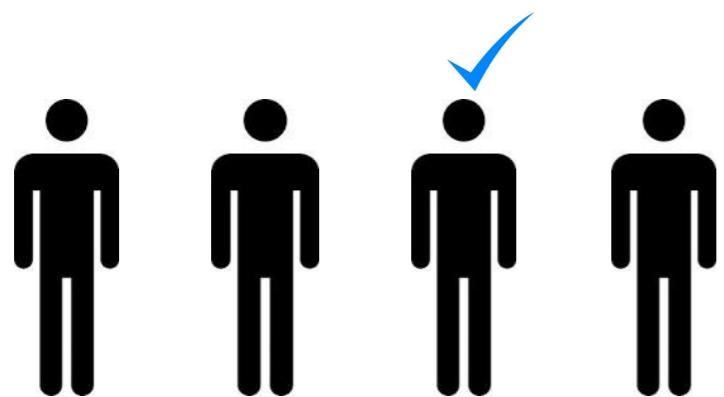


sampling phase

first half

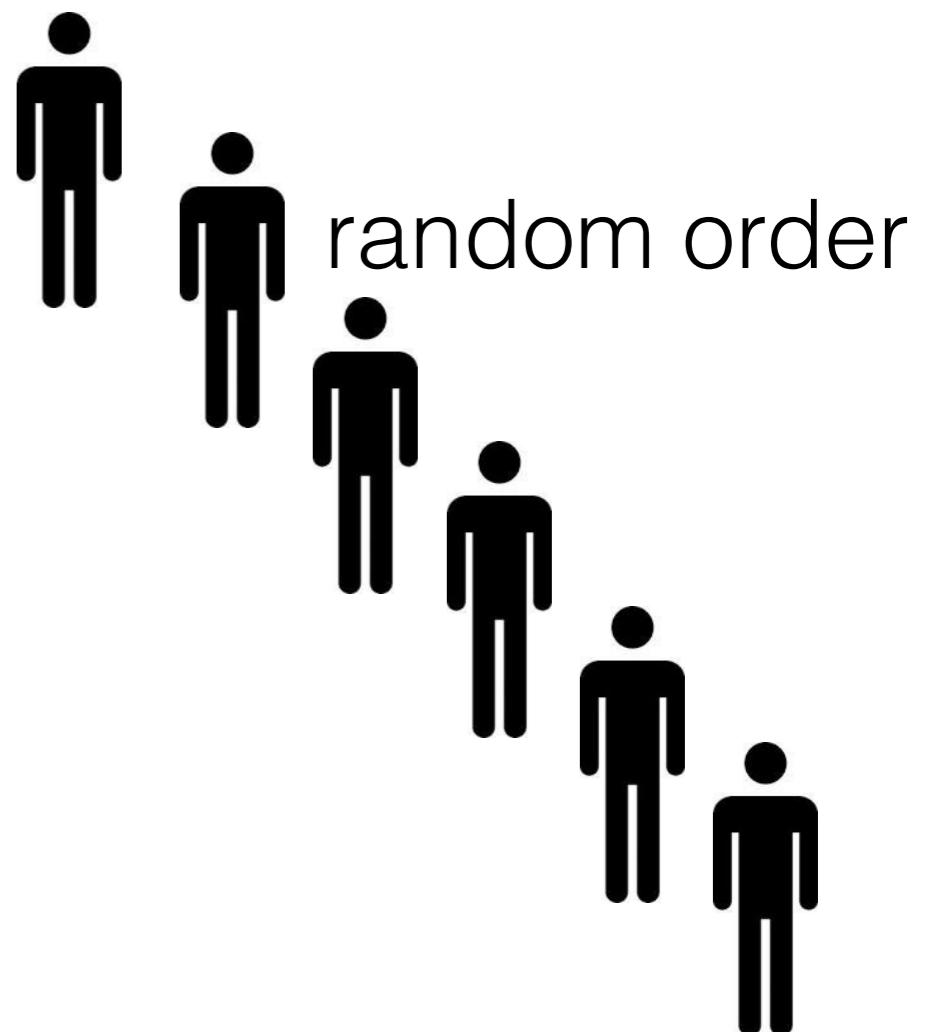


# Hiring staff - not adversarial



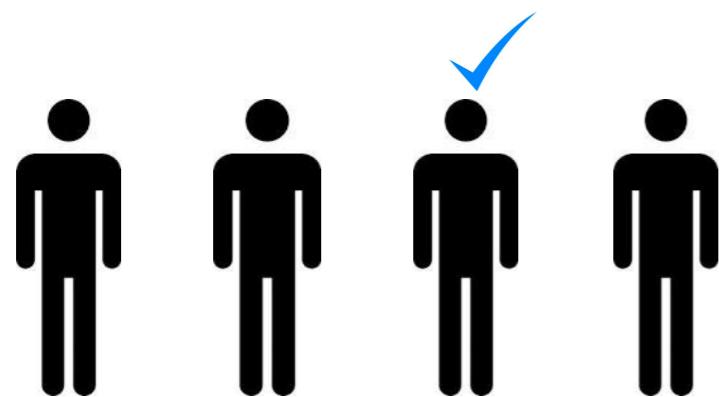
sampling phase

first half



random order

# Hiring staff - not adversarial

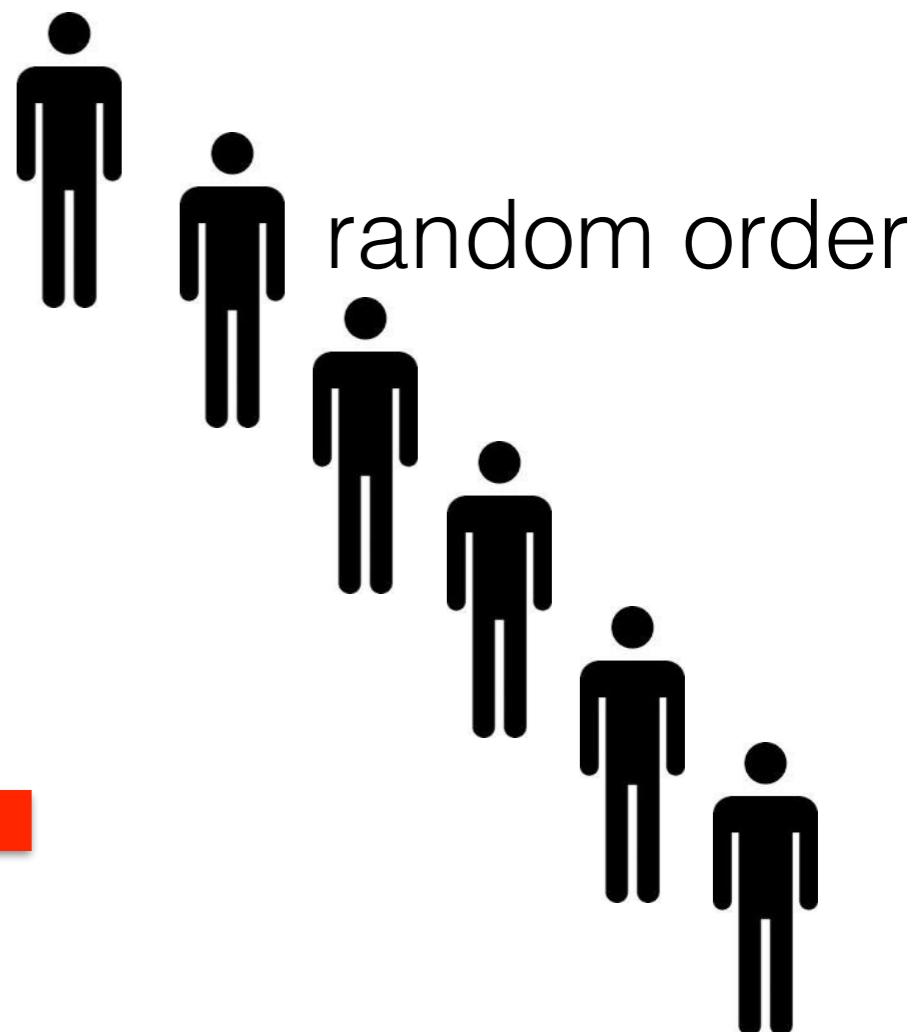


sampling phase

first half

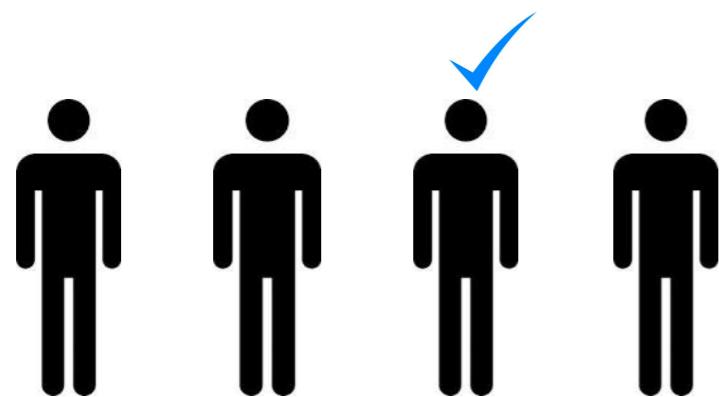
decision phase

second half



random order

# Hiring staff - not adversarial



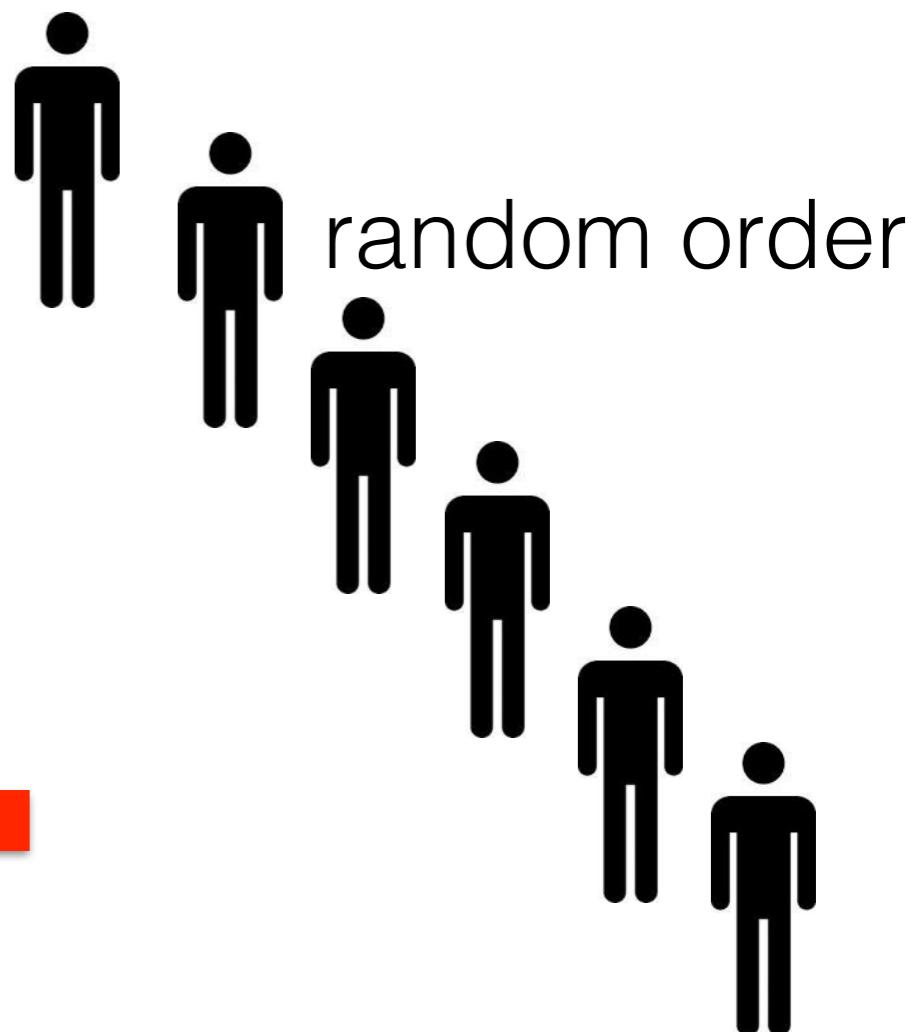
sampling phase

first half



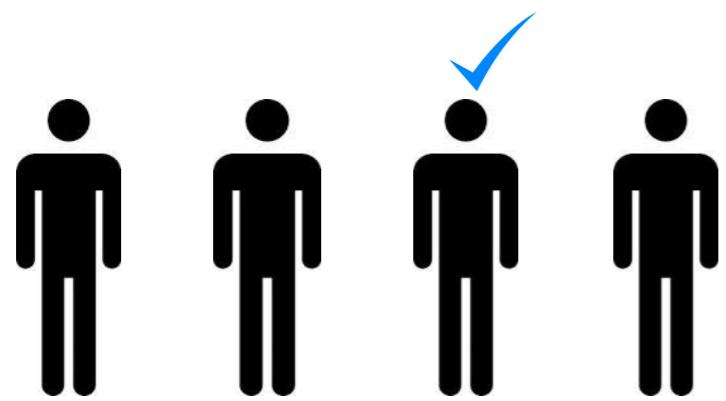
decision phase

second half



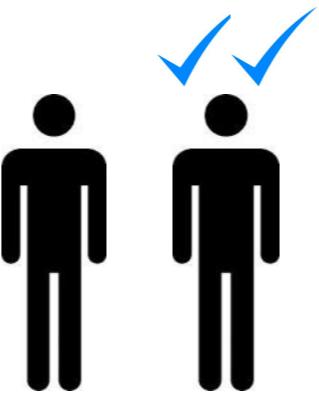
random order

# Hiring staff - not adversarial



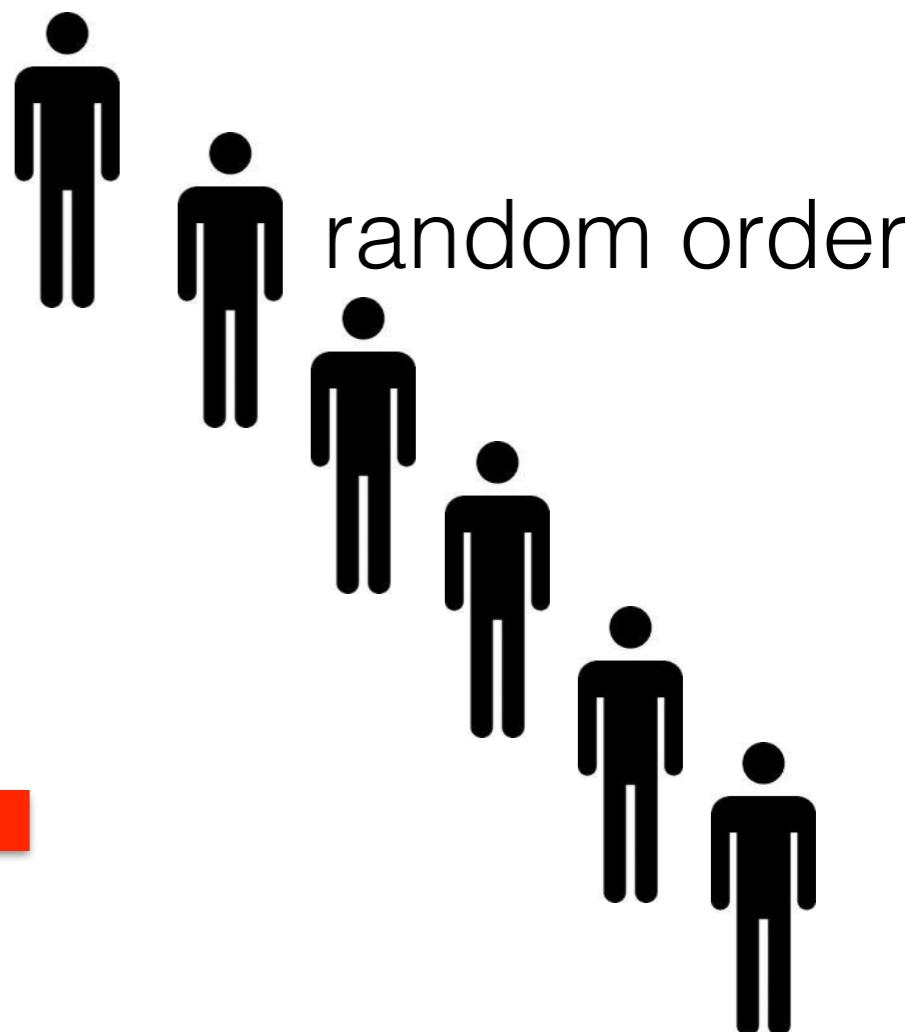
sampling phase

first half



decision phase

second half

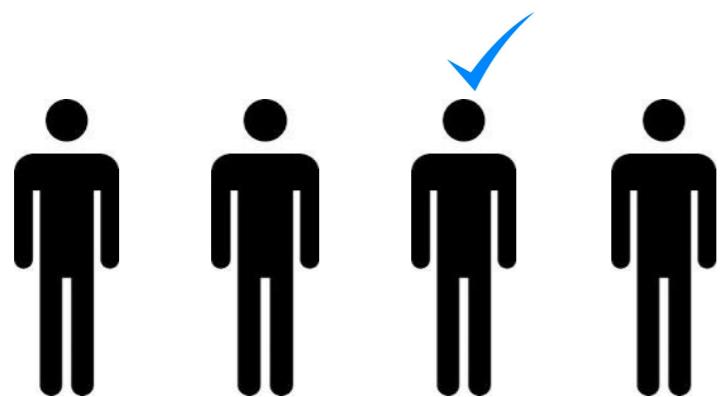


random order

# Hiring staff - not adversarial

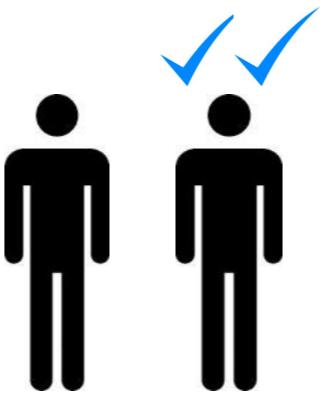


Success with prob > 1/4



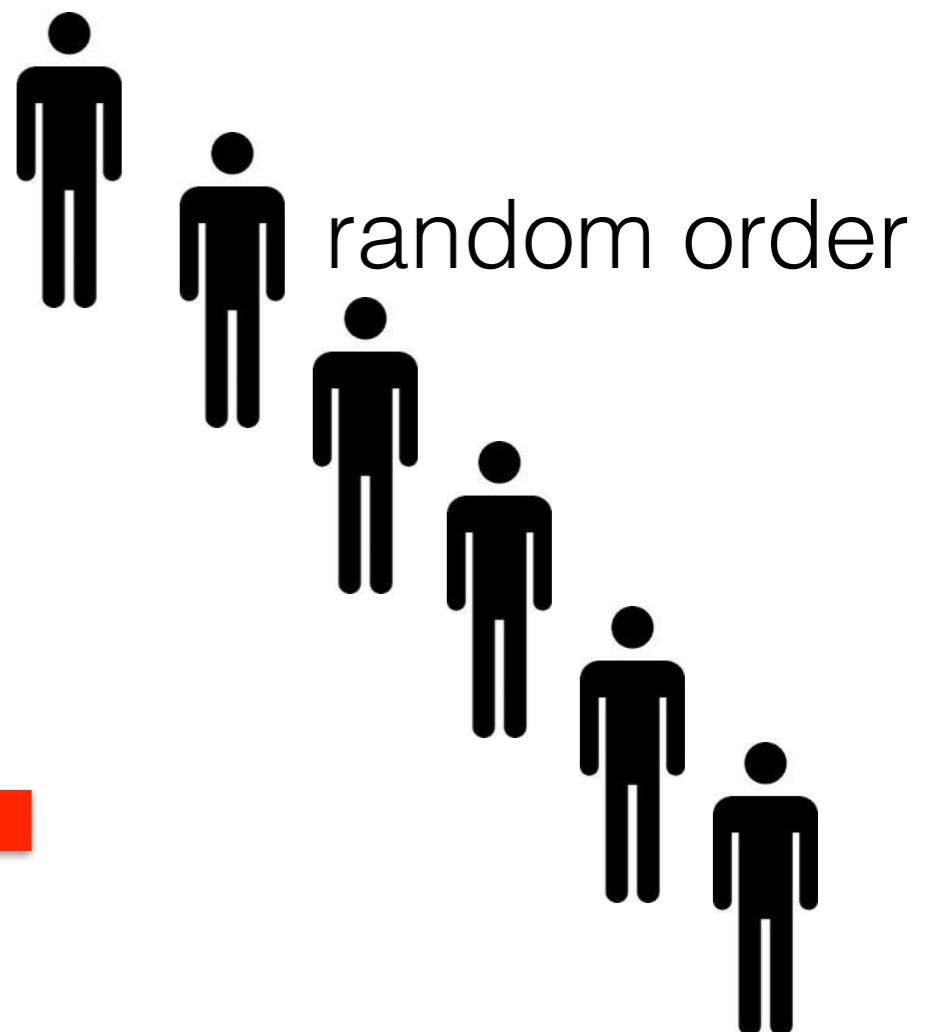
sampling phase

first half



decision phase

second half

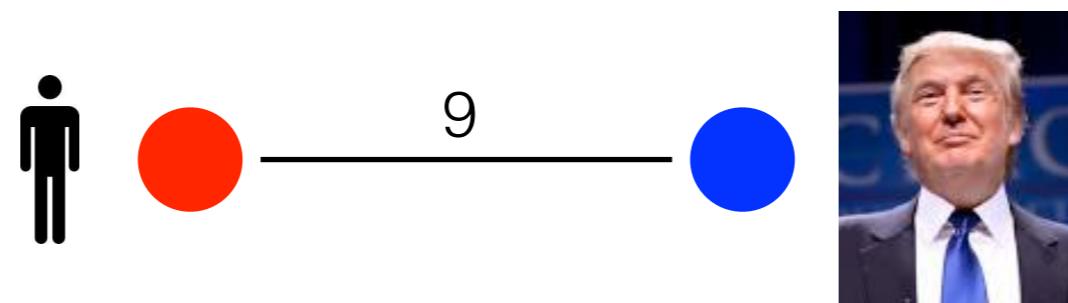


random order

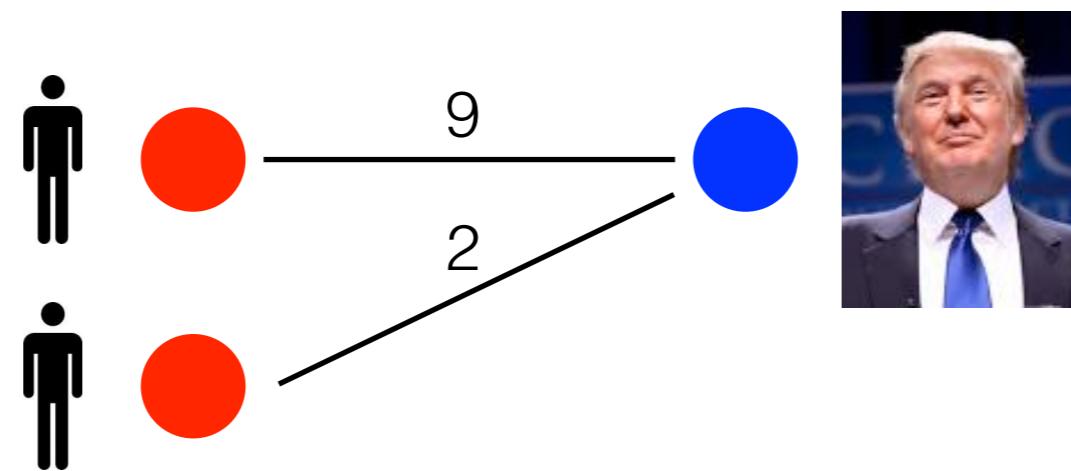
# Actually Matching



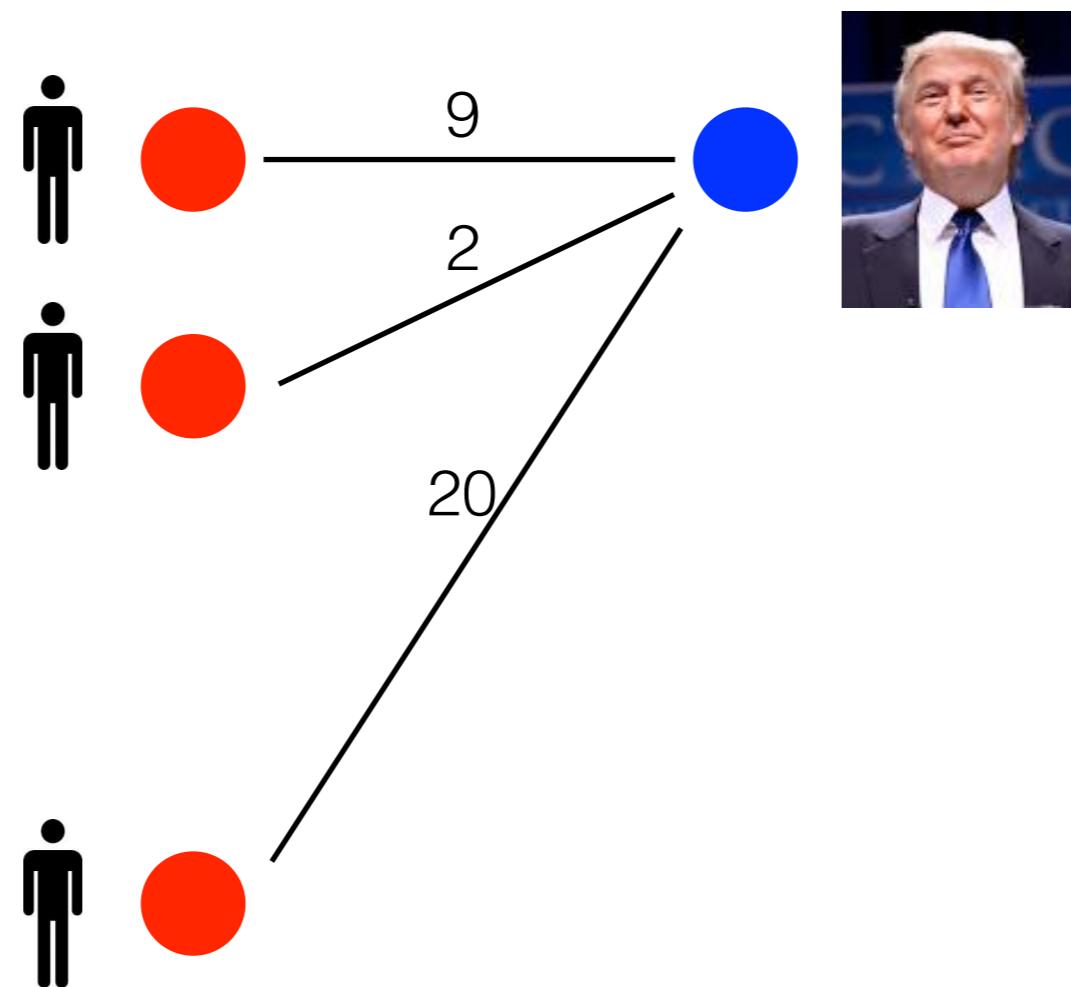
# Actually Matching



# Actually Matching

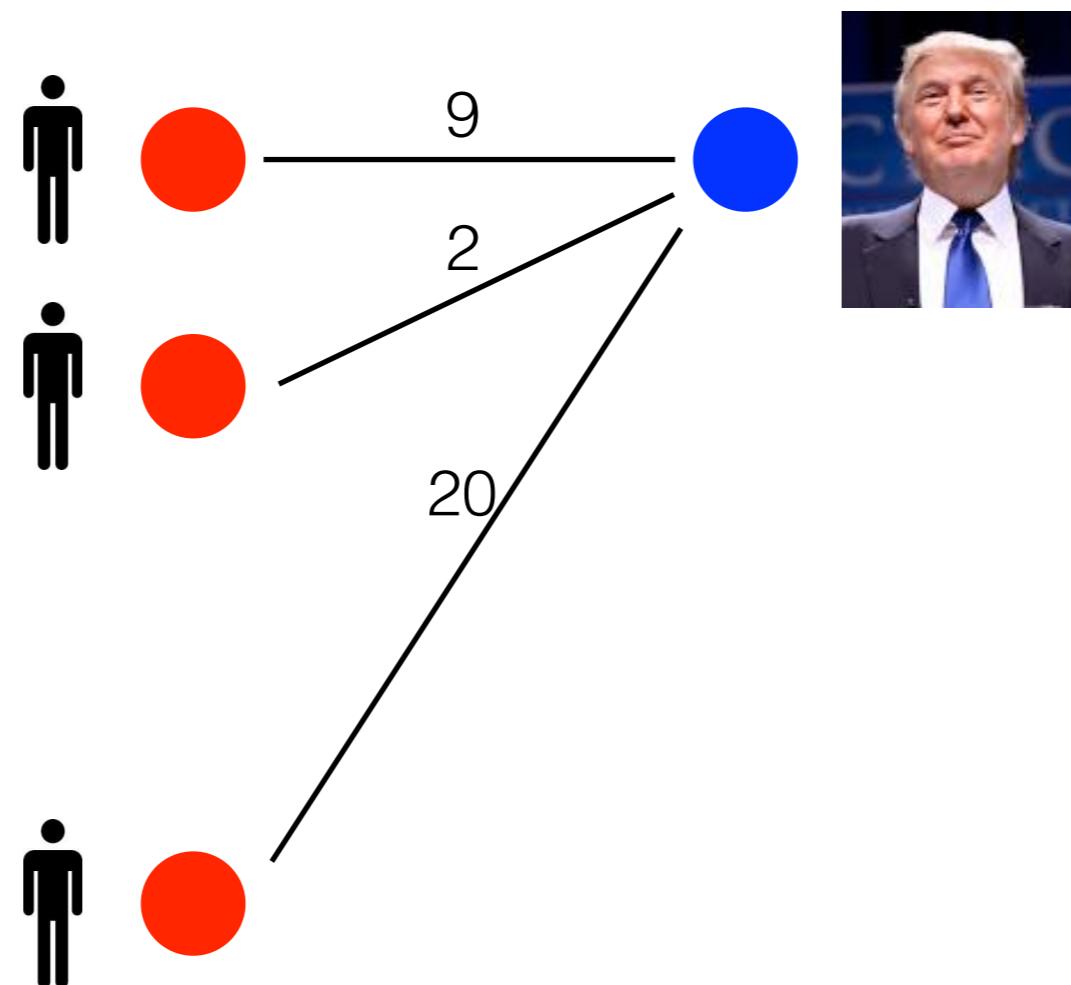


# Actually Matching



# Actually Matching

accept the edge with the largest weight instantaneously



# Natural Generalization



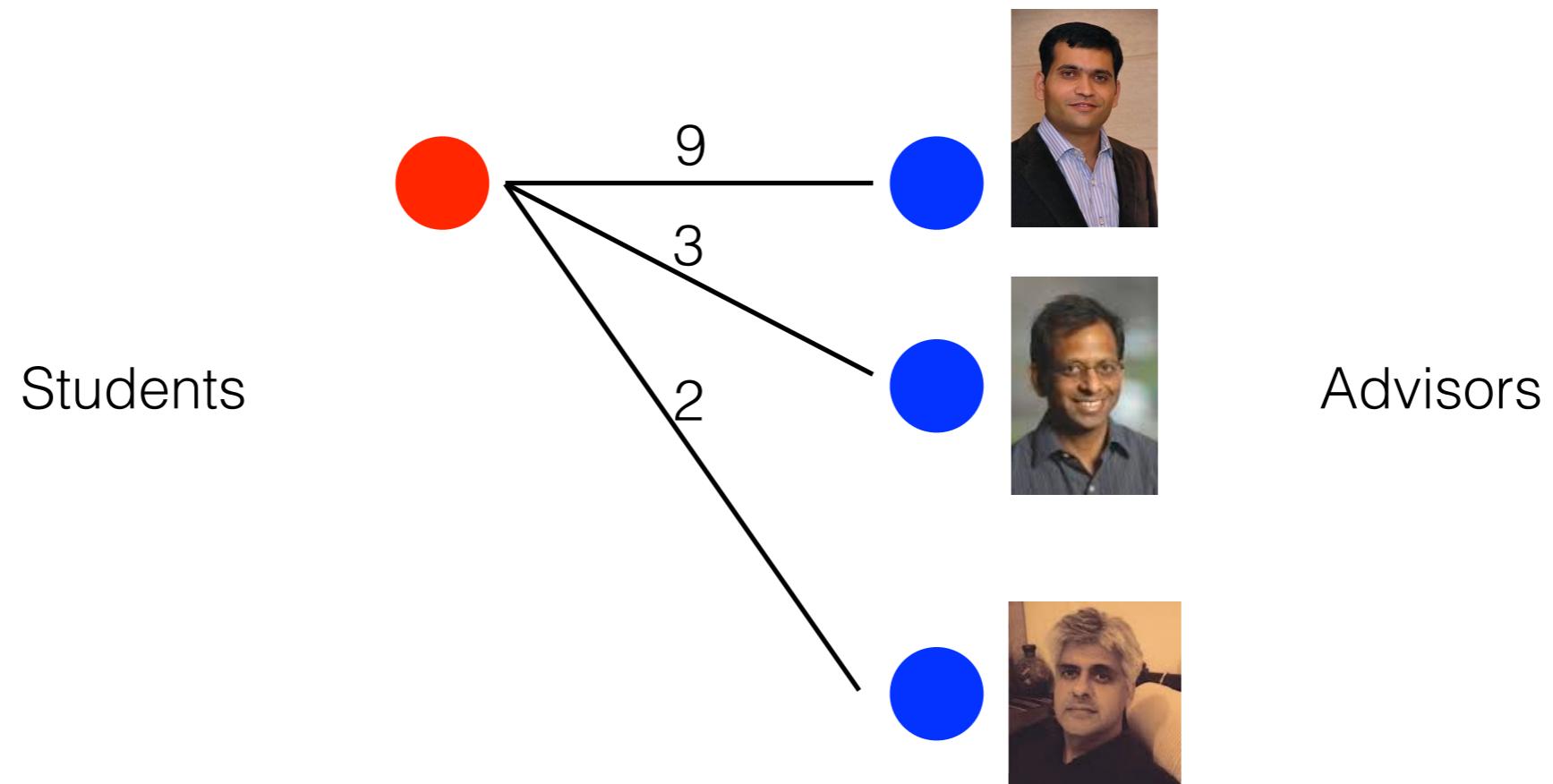
Students



Advisors

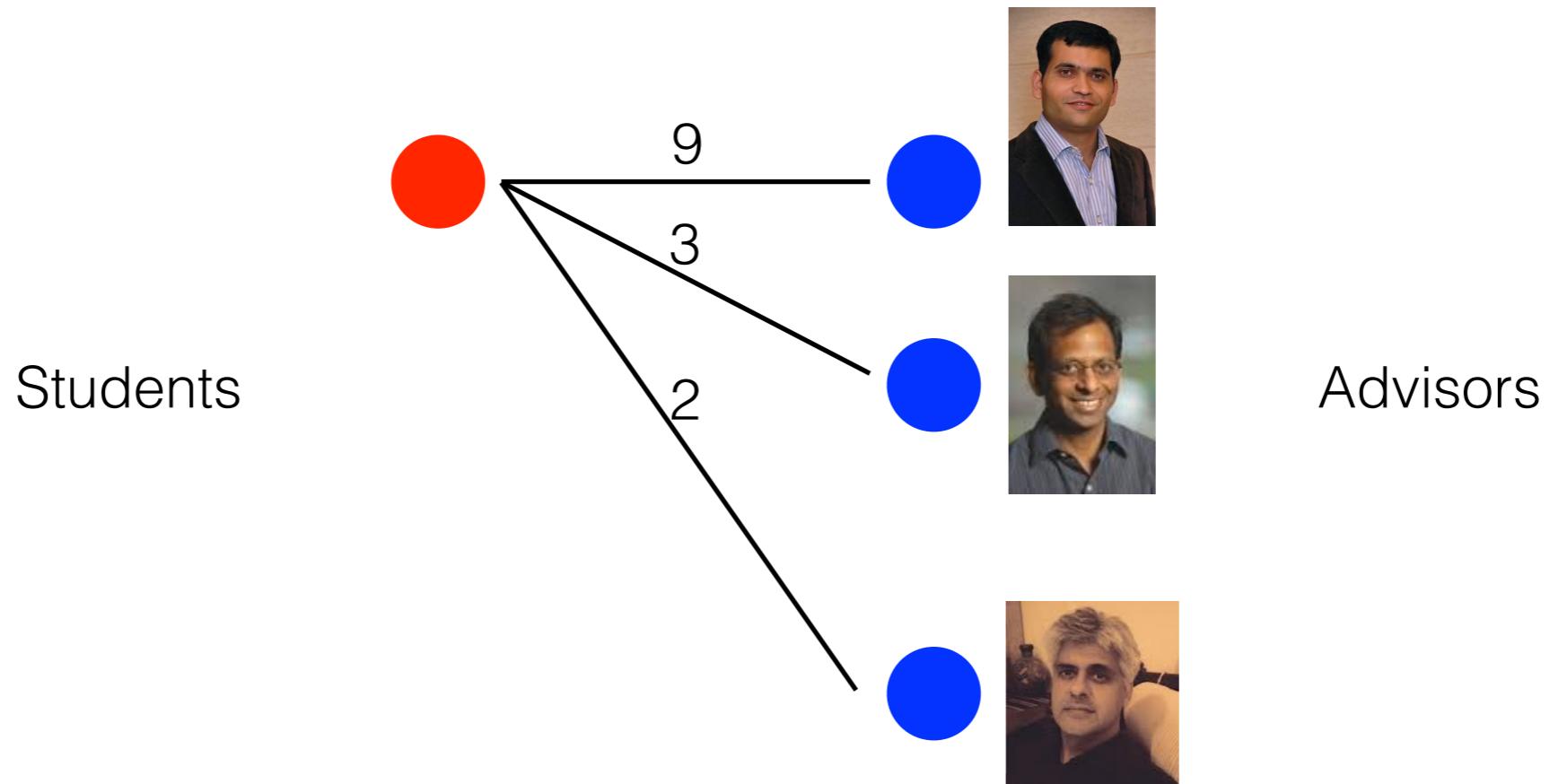


# Natural Generalization



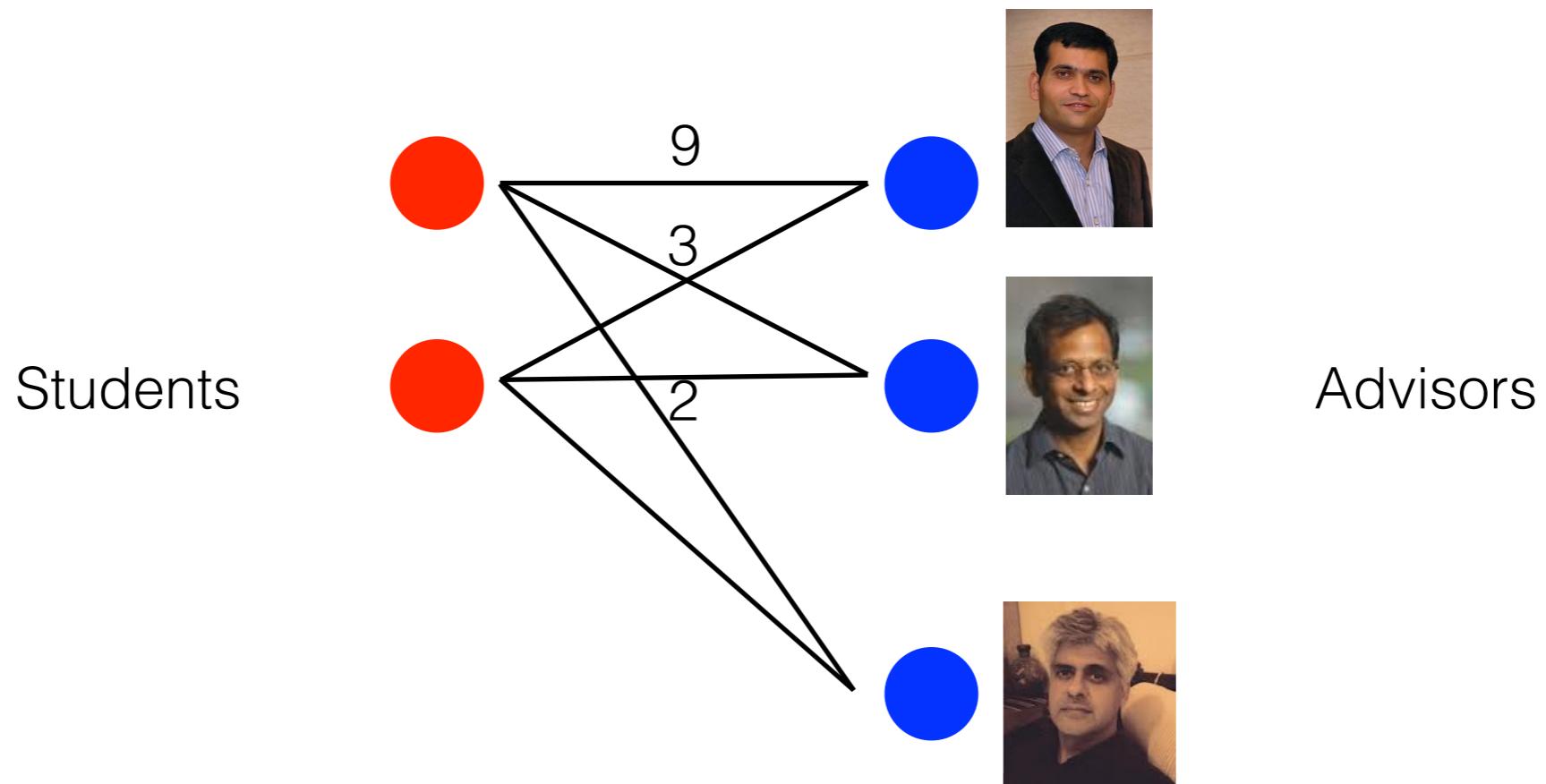
# Natural Generalization

each advisor gets at most one student



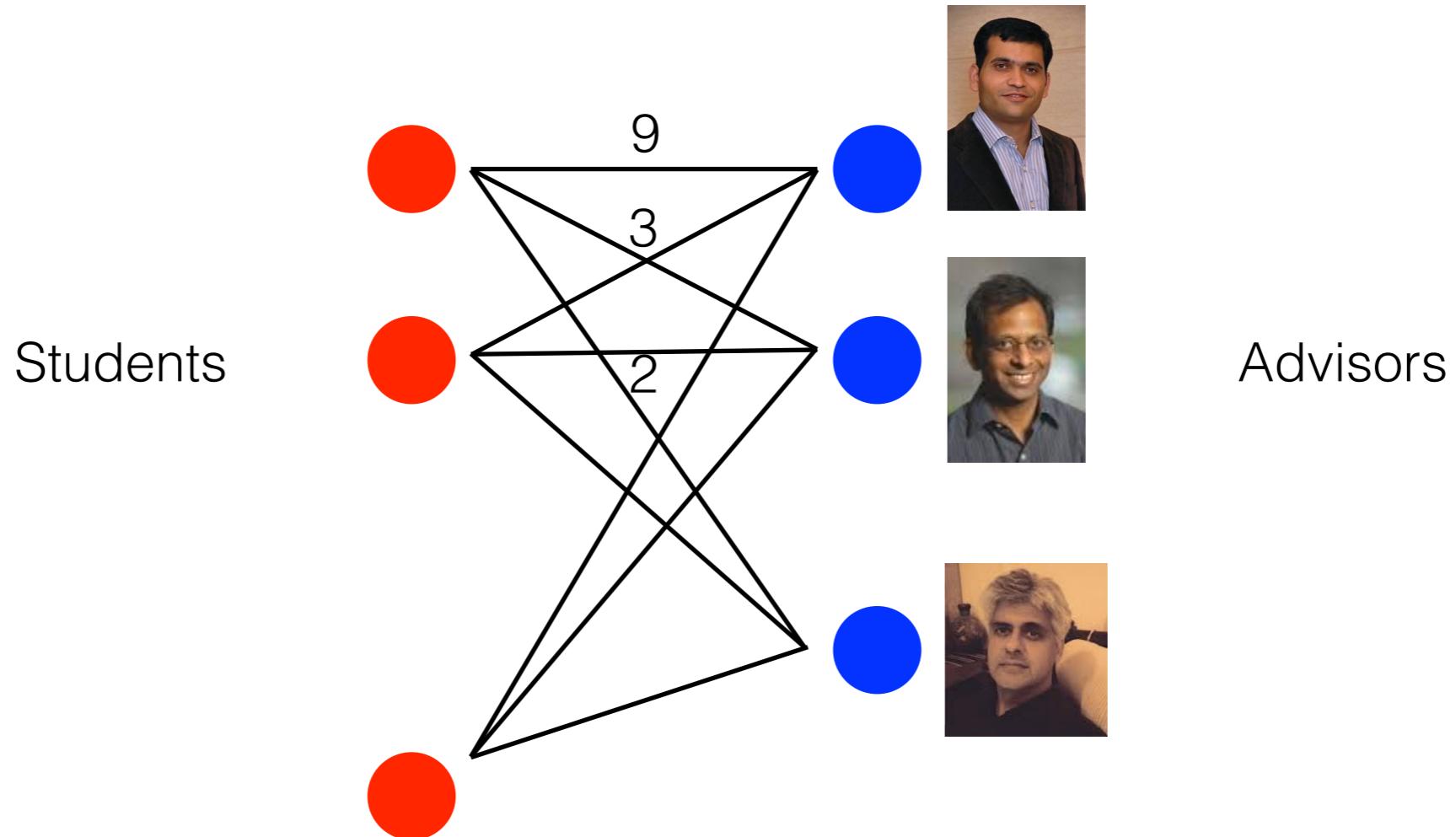
# Natural Generalization

each advisor gets at most one student



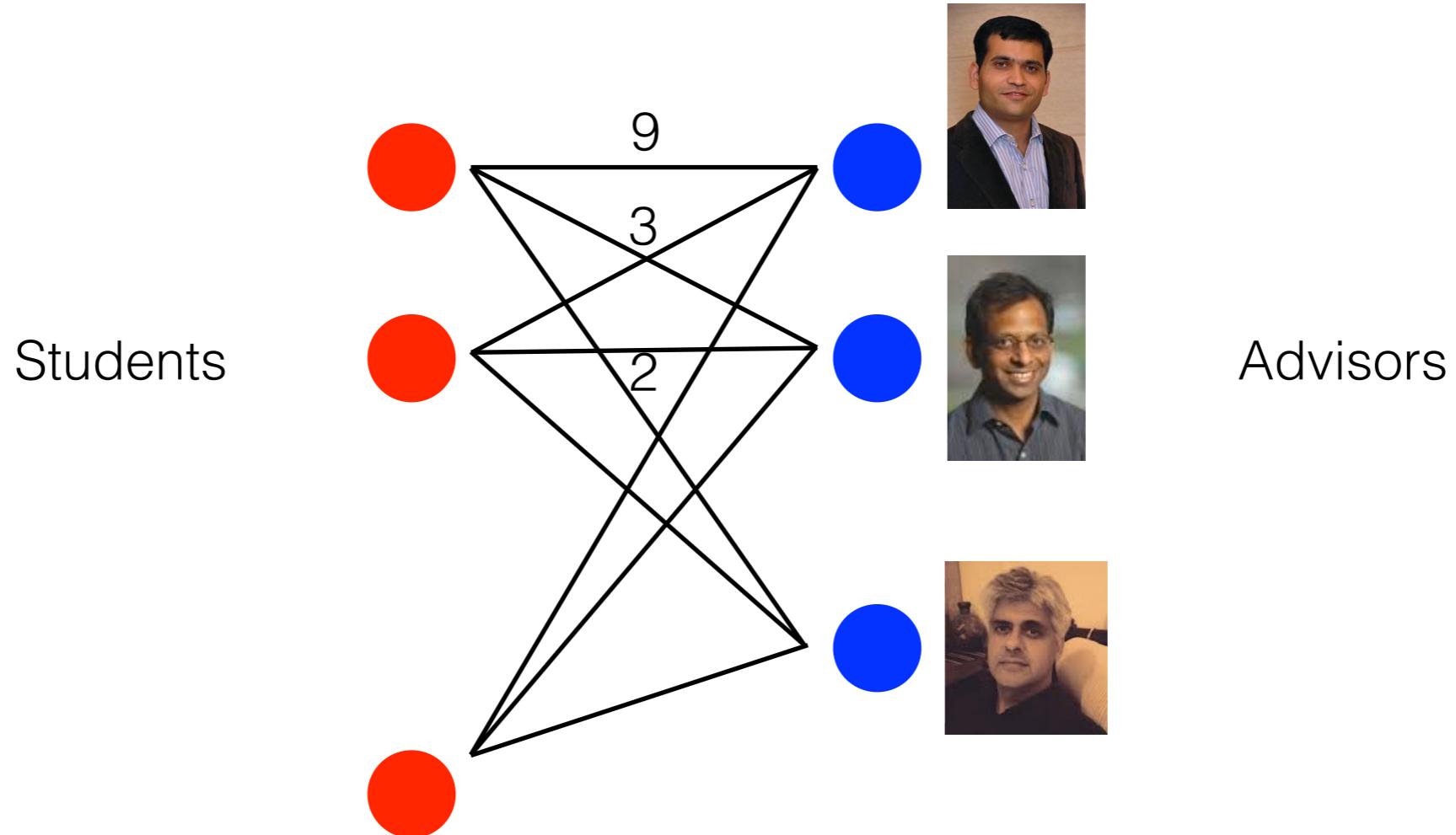
# Natural Generalization

each advisor gets at most one student



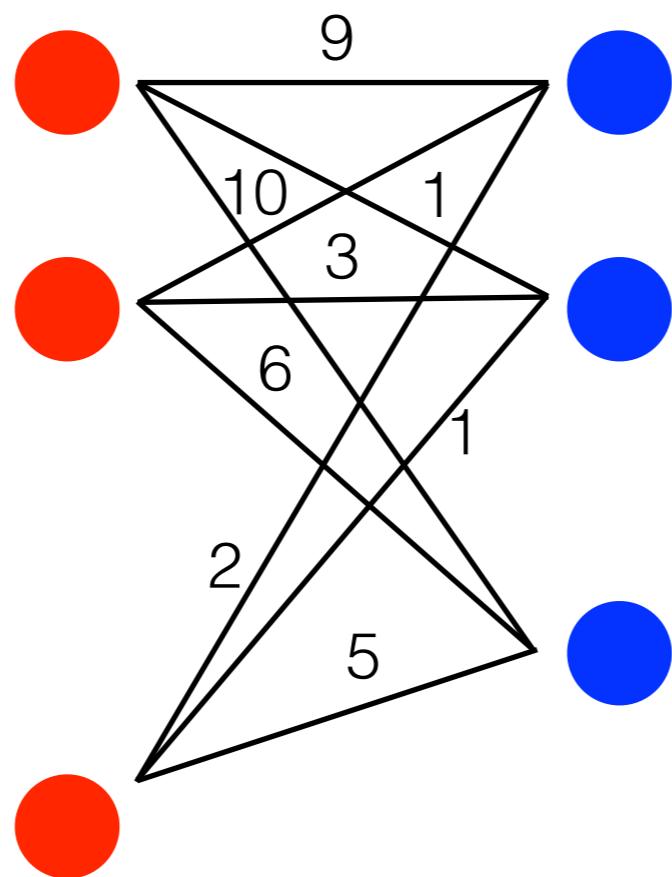
# Natural Generalization

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



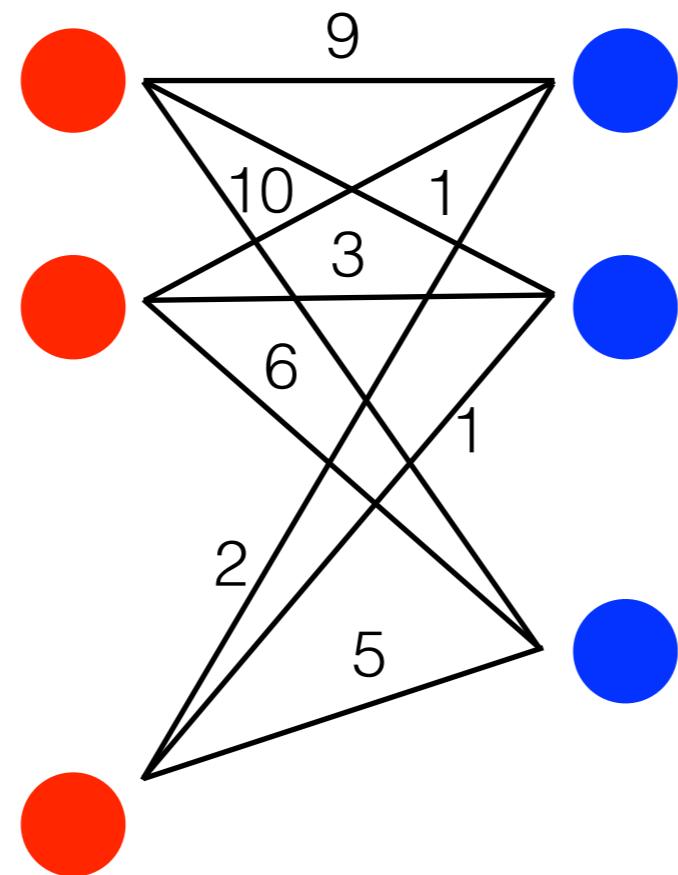
Objective: Matching with largest sum weight

# Example



# Example

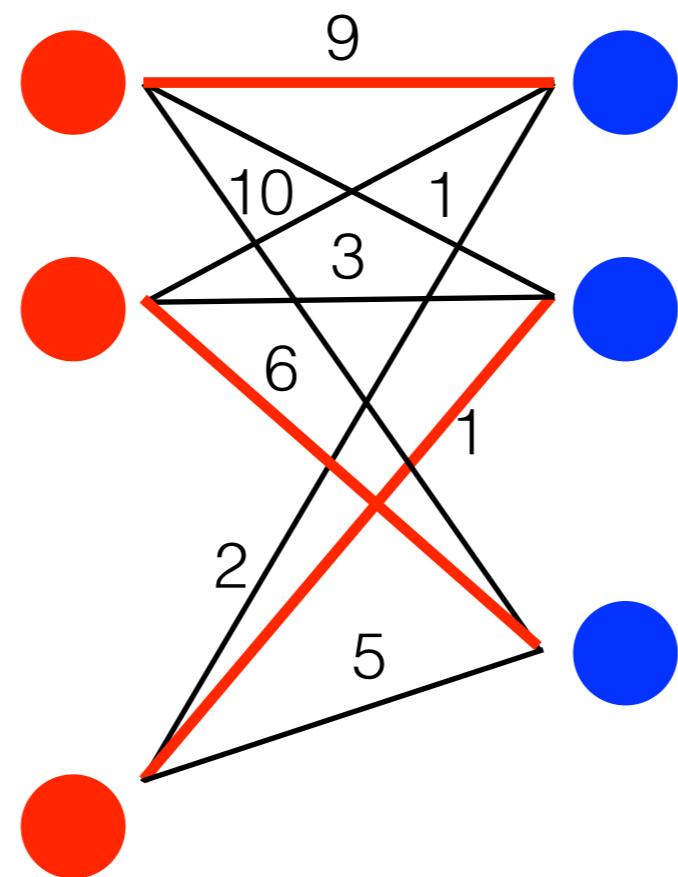
at most one accepted edge



Objective: Matching with largest sum weight

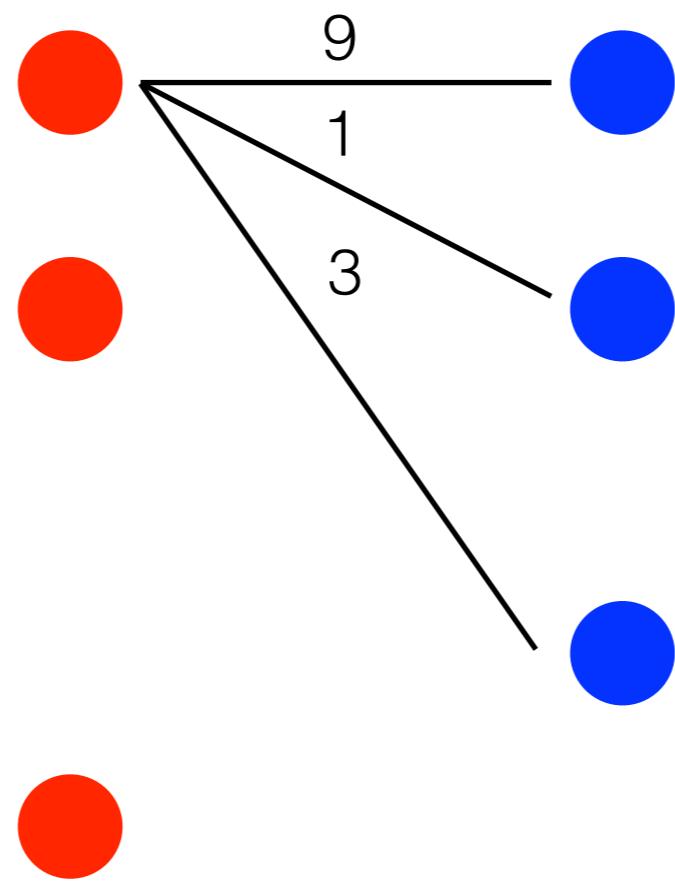
# Example

at most one accepted edge

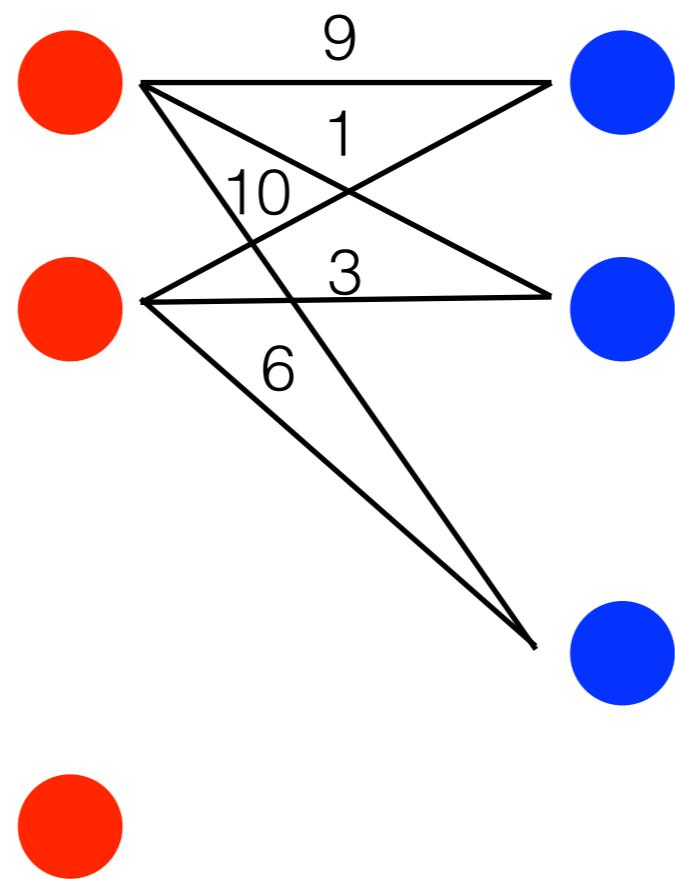


Objective: Matching with largest sum weight

# How to solve this ONLINE

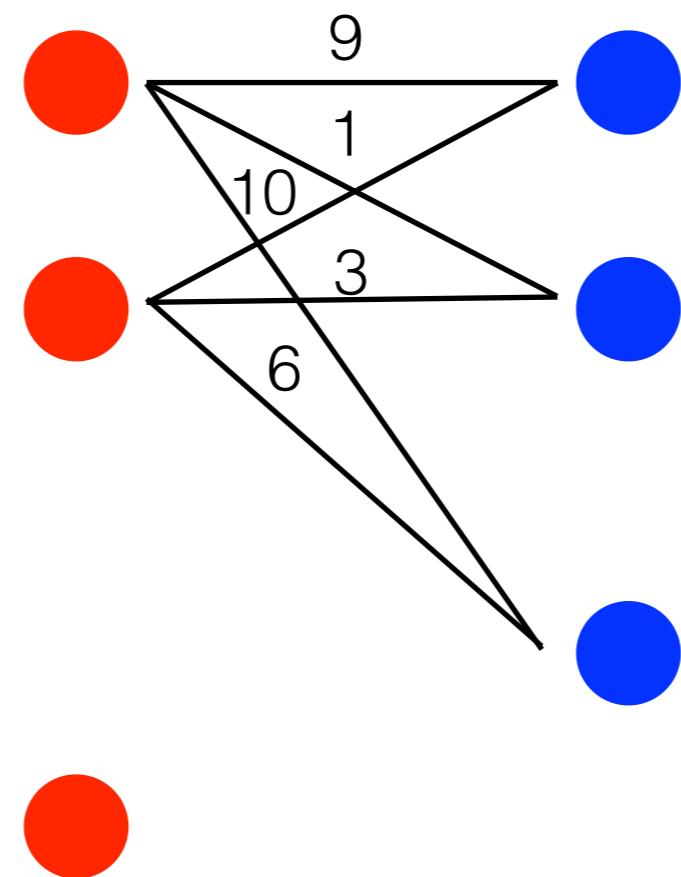


# How to solve this ONLINE



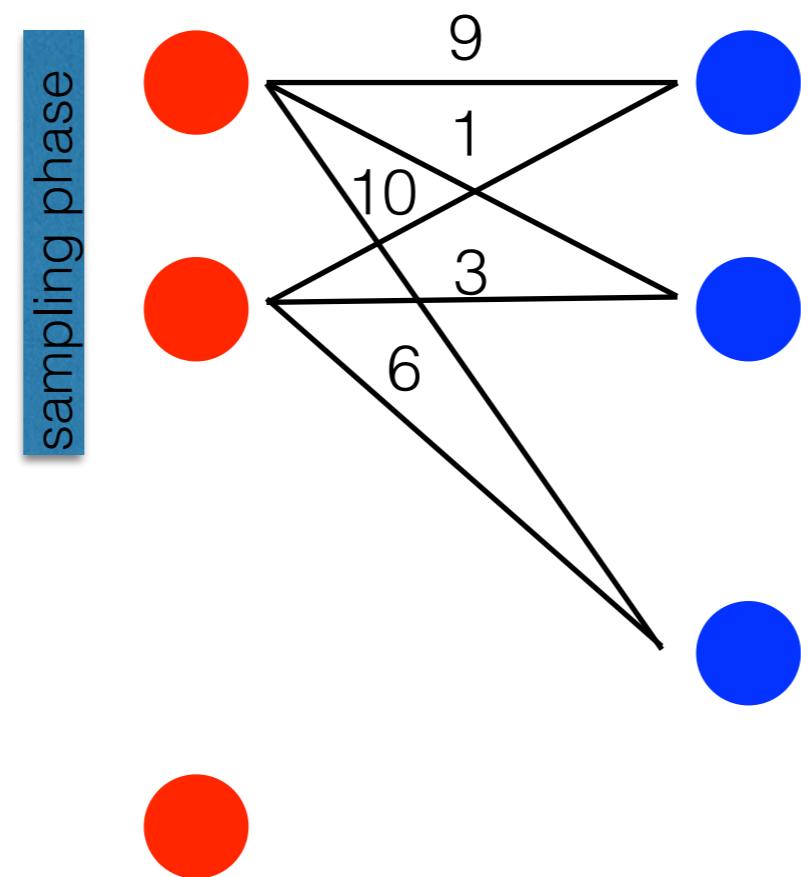
# How to solve this ONLINE

Sampling idea as before



# How to solve this ONLINE

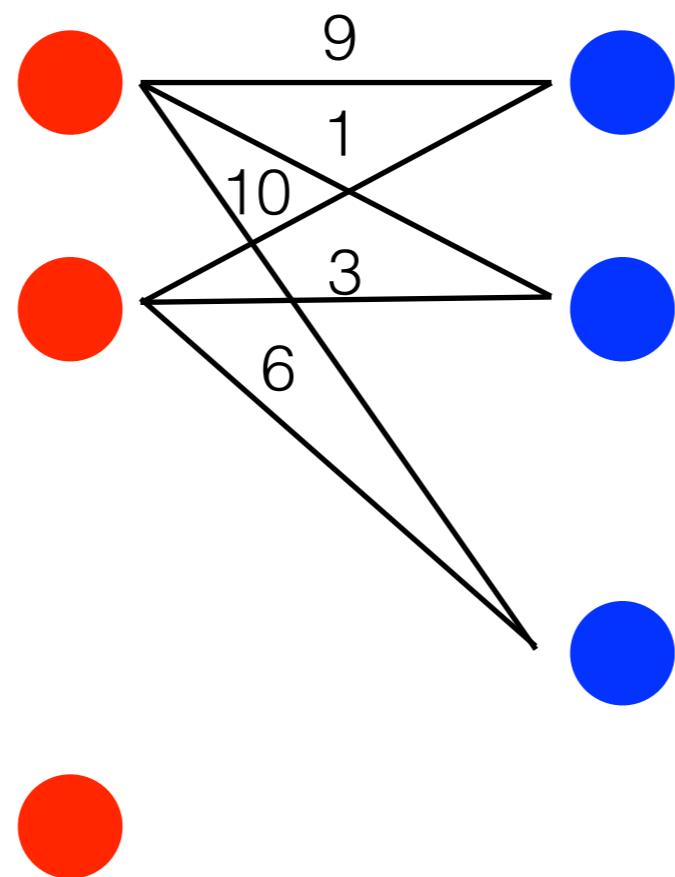
Sampling idea as before



# How to solve this ONLINE

Find best matching

sampling phase

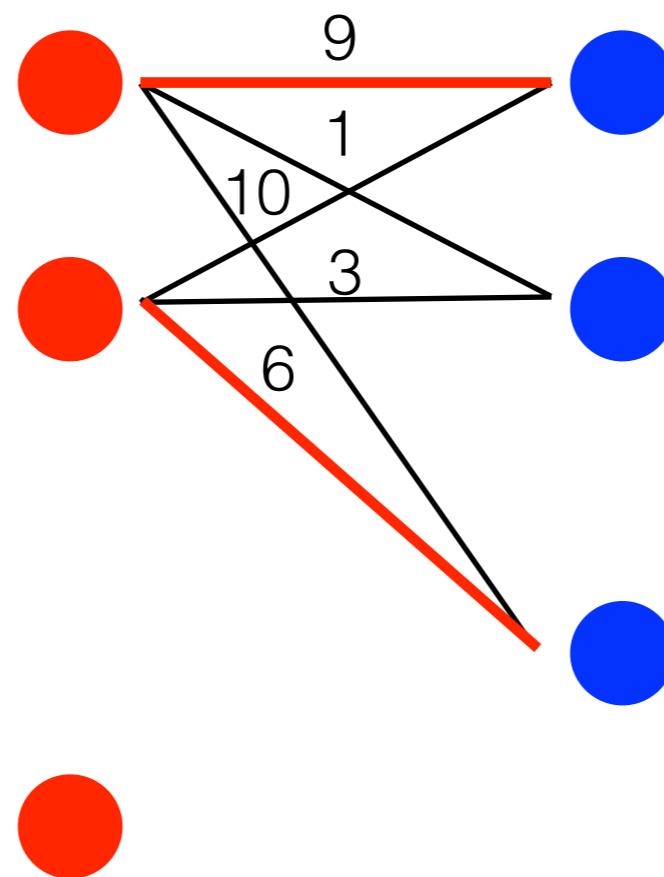


Sampling idea as before

# How to solve this ONLINE

Find best matching

sampling phase

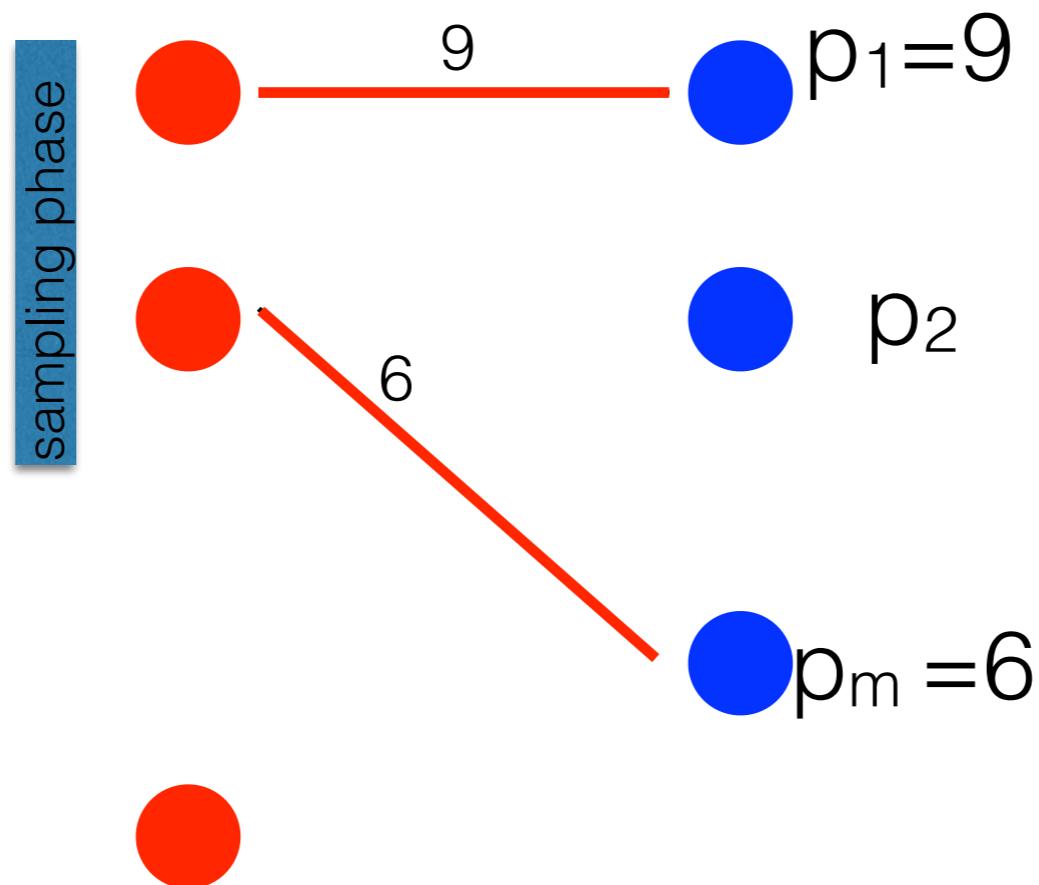


Sampling idea as before

# How to solve this ONLINE

Sampling idea as before

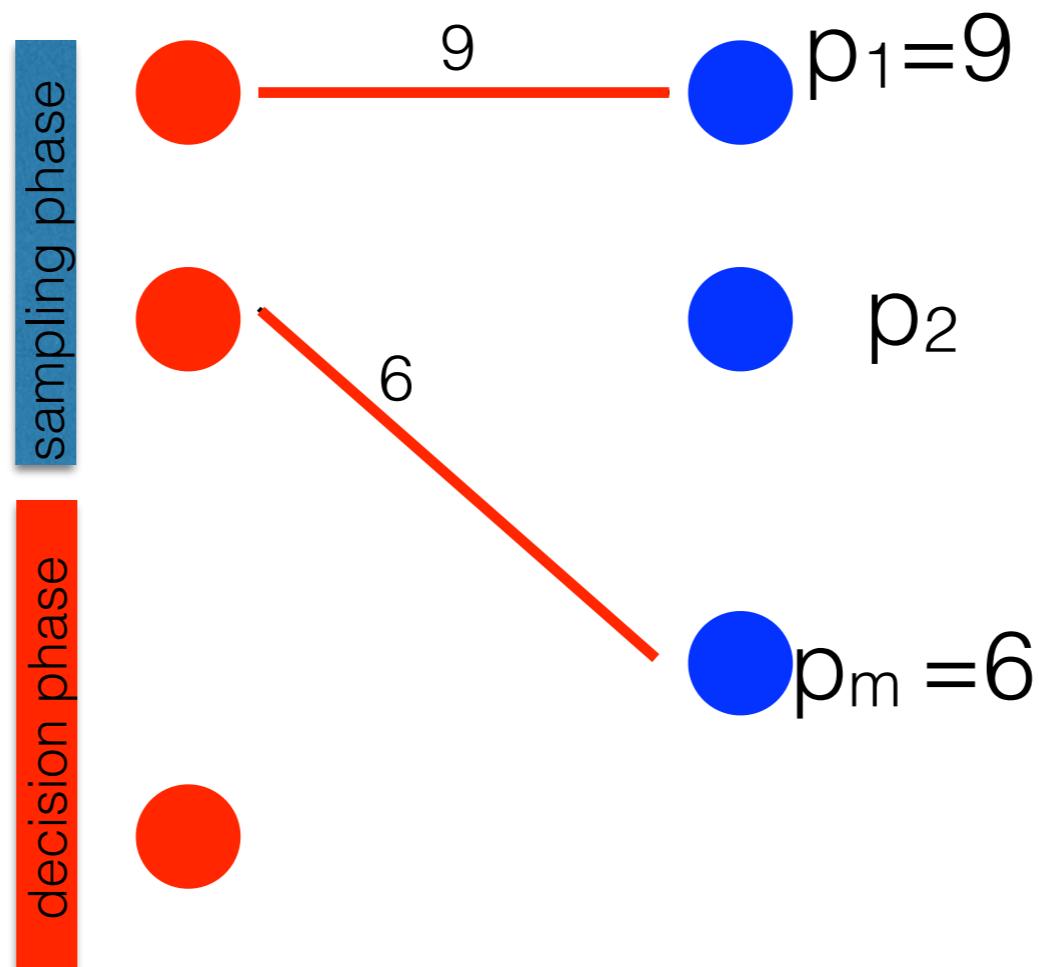
Find best matching



# How to solve this ONLINE

Sampling idea as before

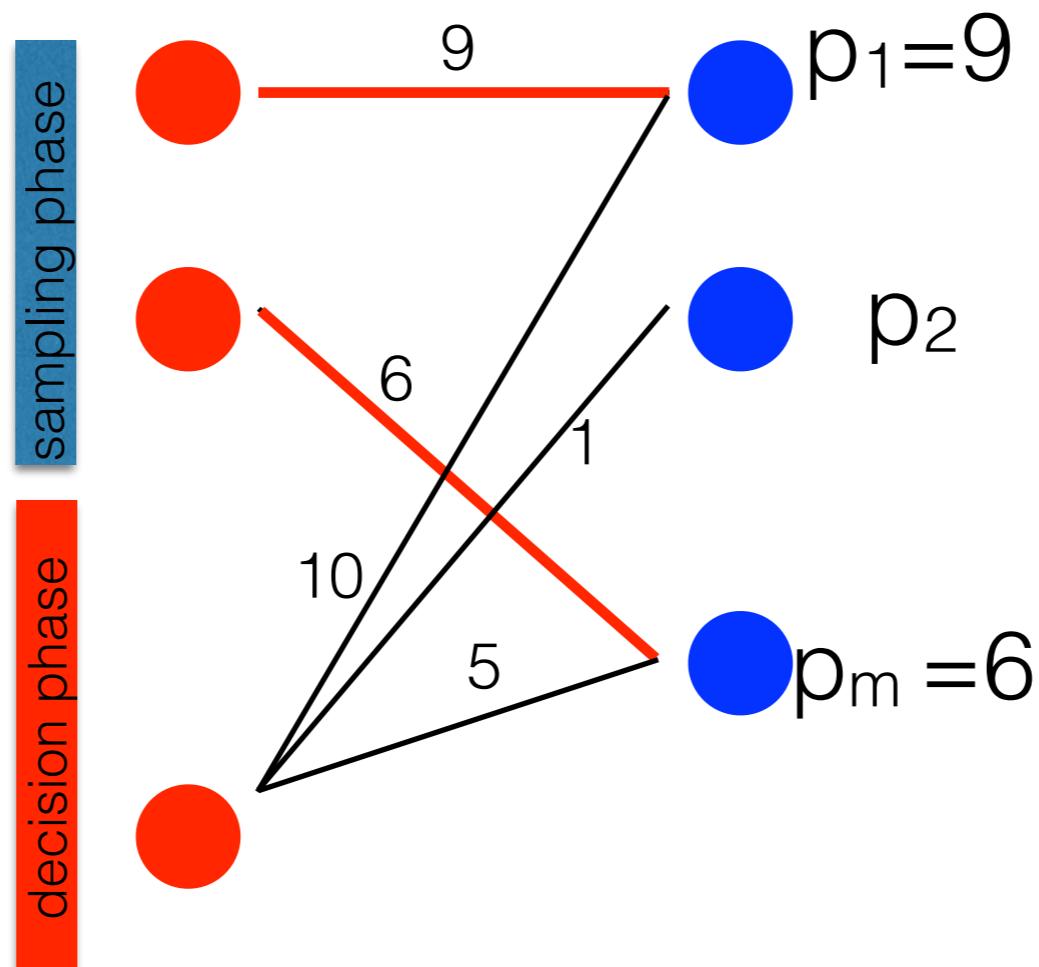
Find best matching



# How to solve this ONLINE

Sampling idea as before

Find best matching

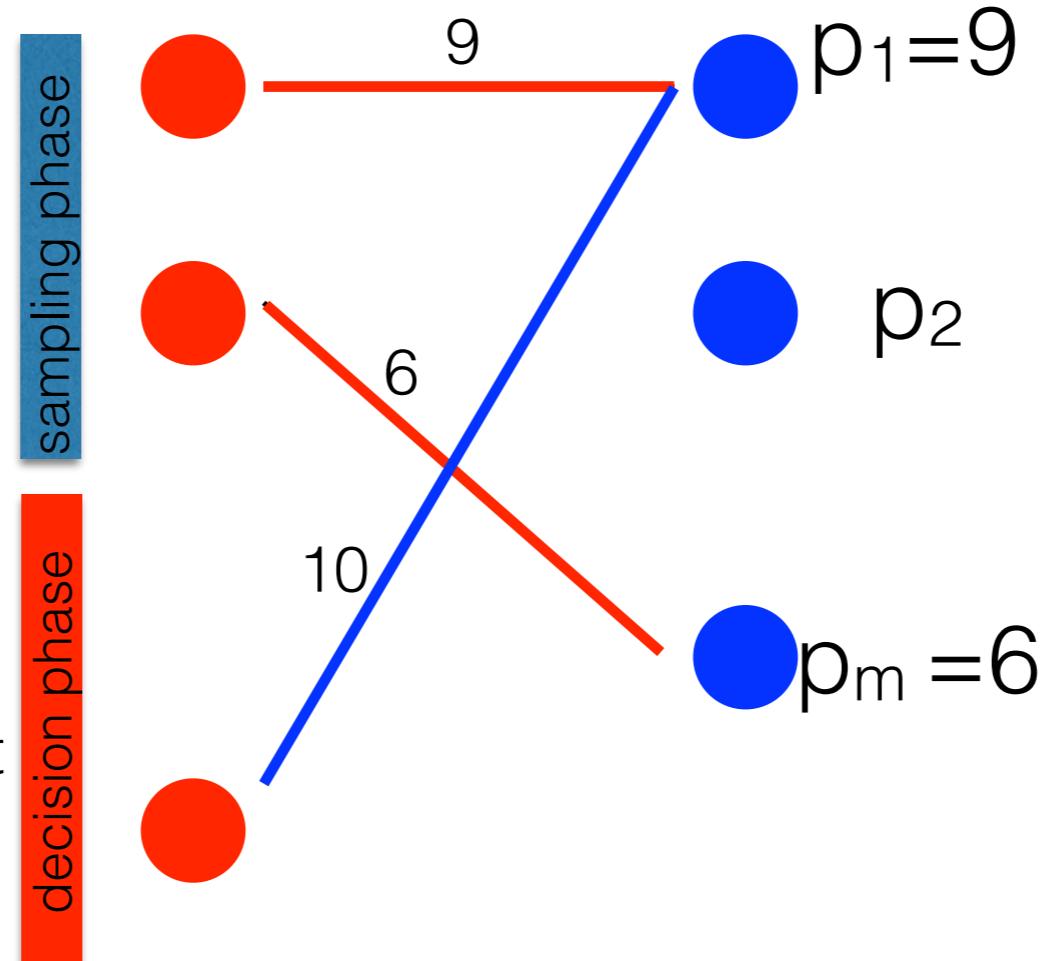


# How to solve this ONLINE

Sampling idea as before

Find best matching

match the largest weight  
more than price

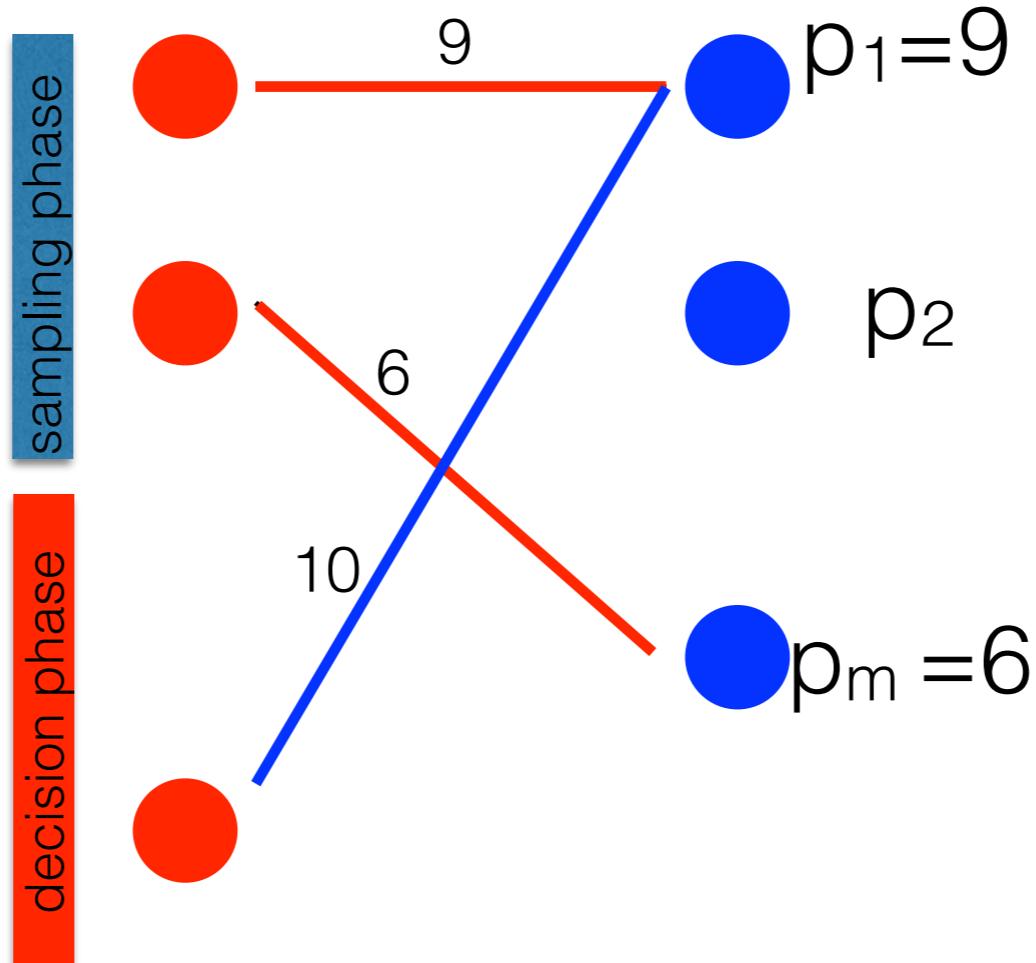


# How to solve this ONLINE

Sampling idea as before

Find best matching

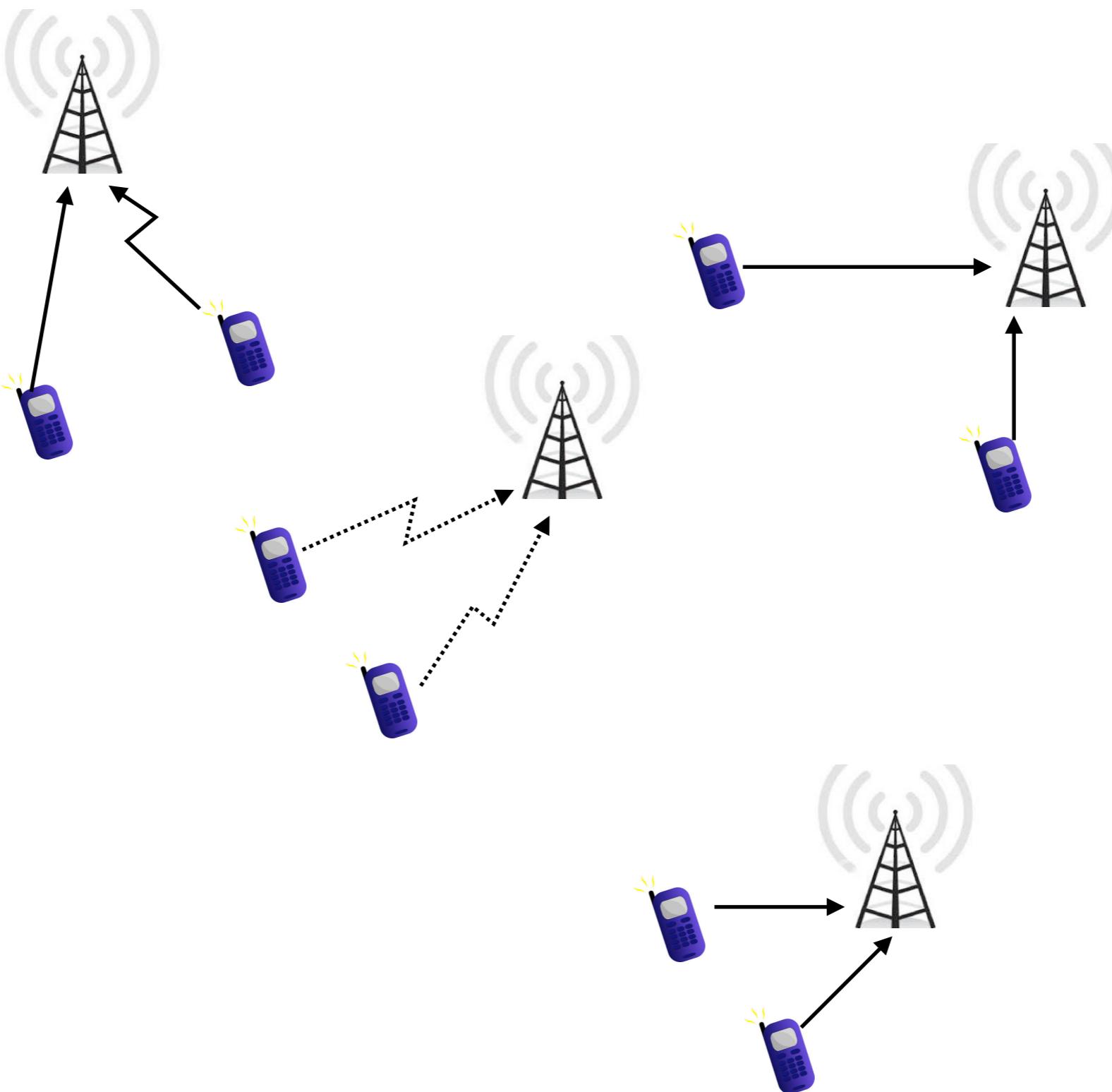
match the largest weight  
more than price



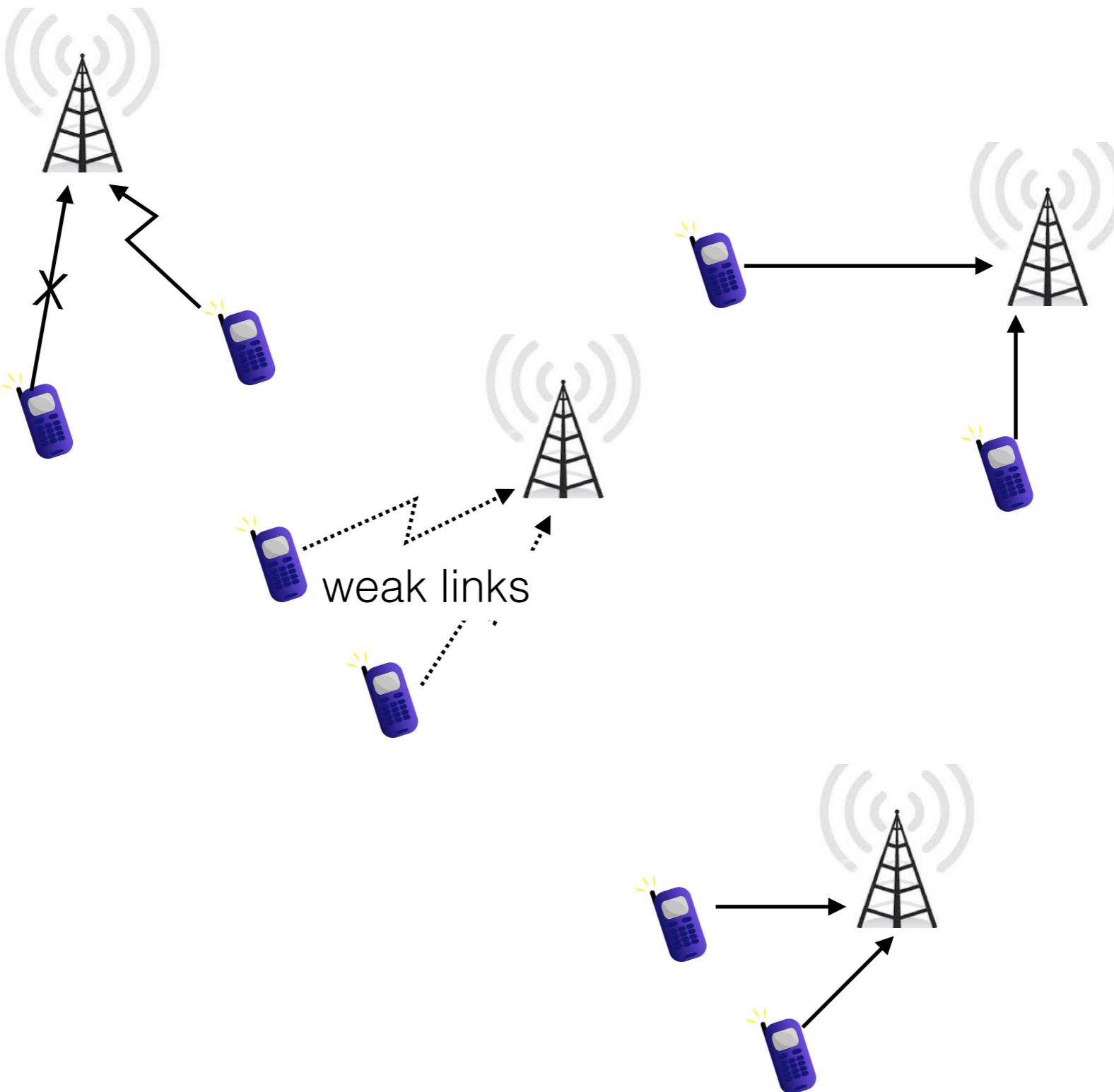
$$u(M_{ON}) \geq \frac{u(M_{OFF}(\text{half}))}{2}$$

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

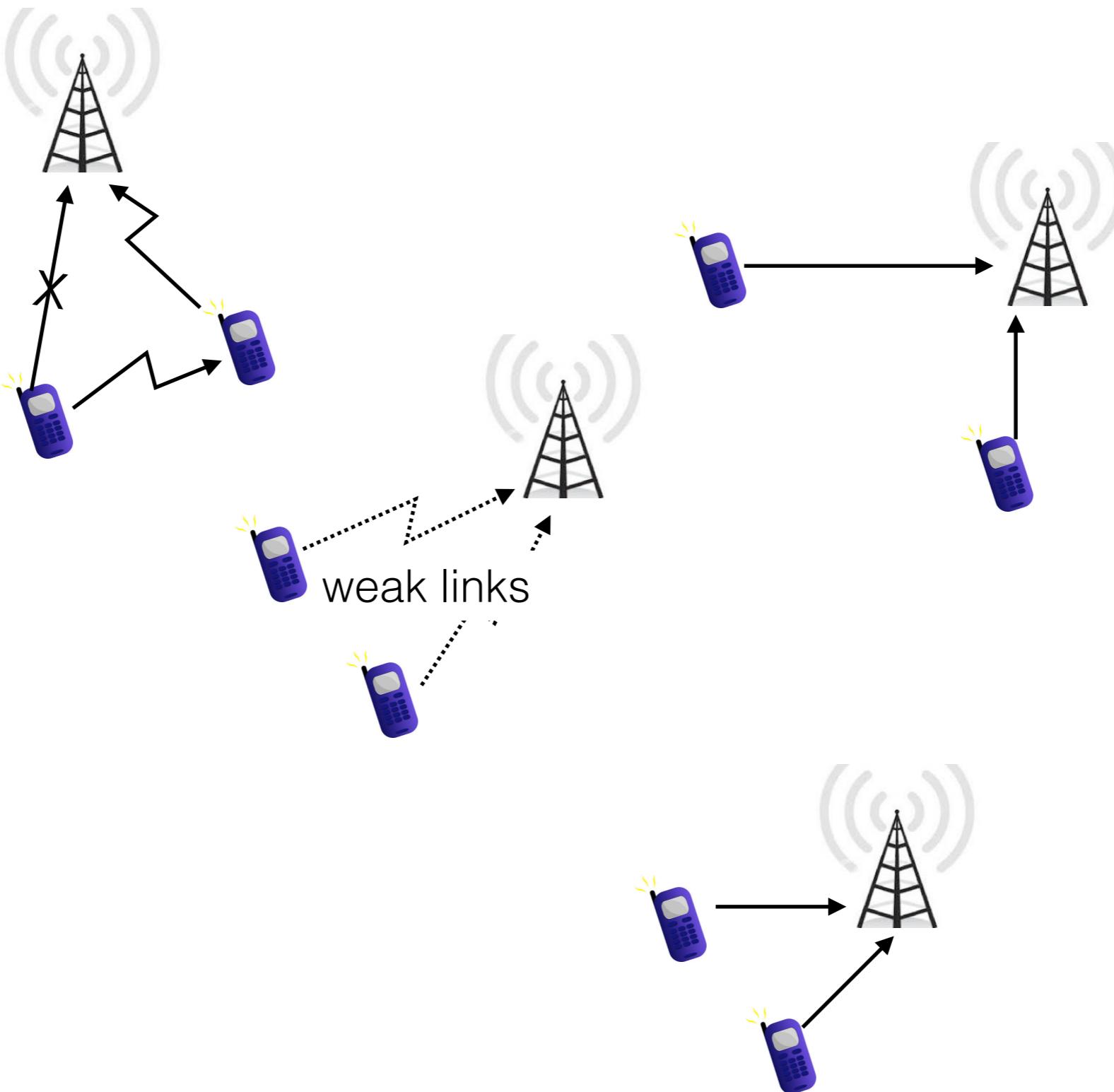
# Modern Problem Device-2-Device Communication



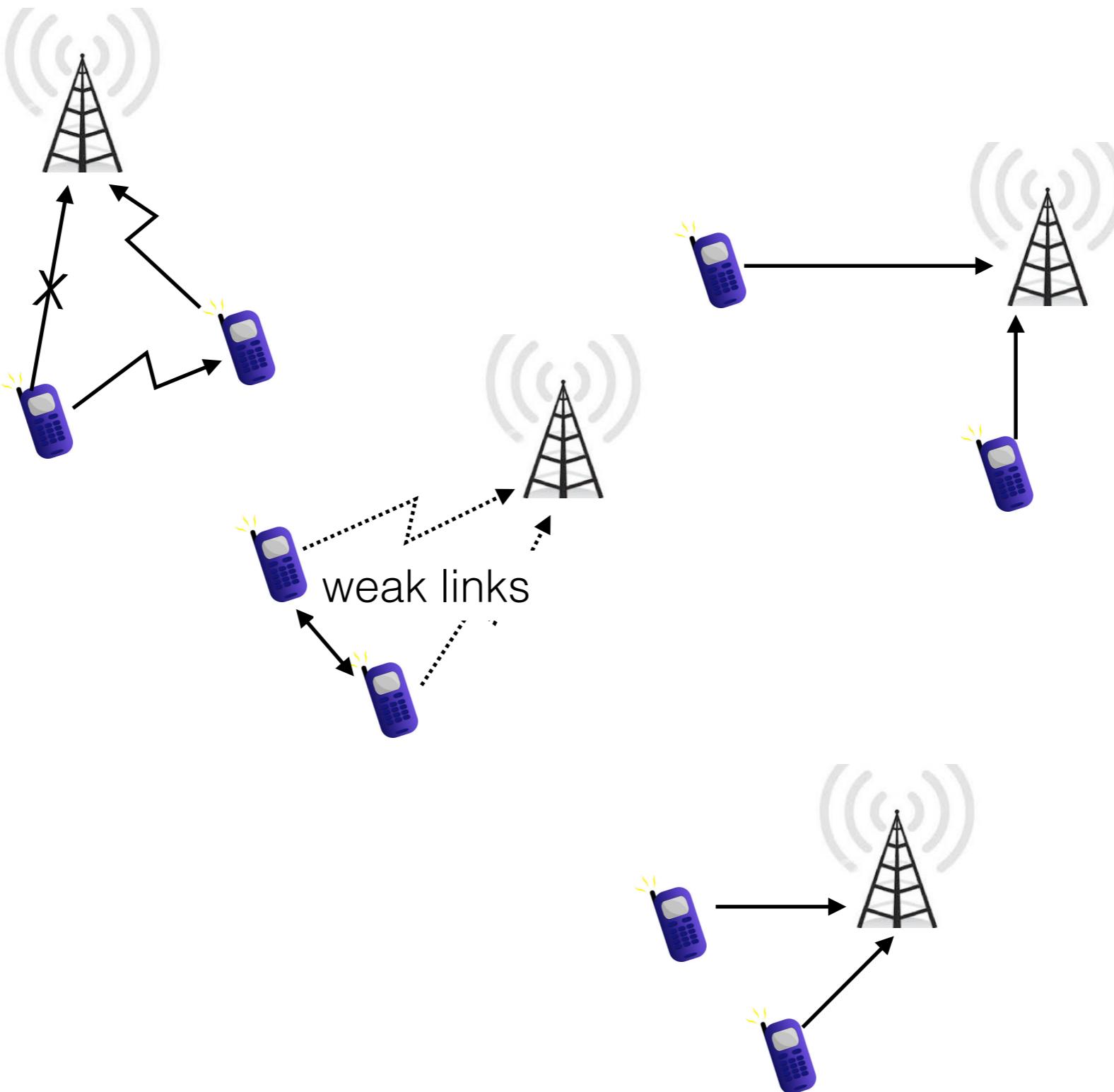
# Modern Problem Device-2-Device Communication



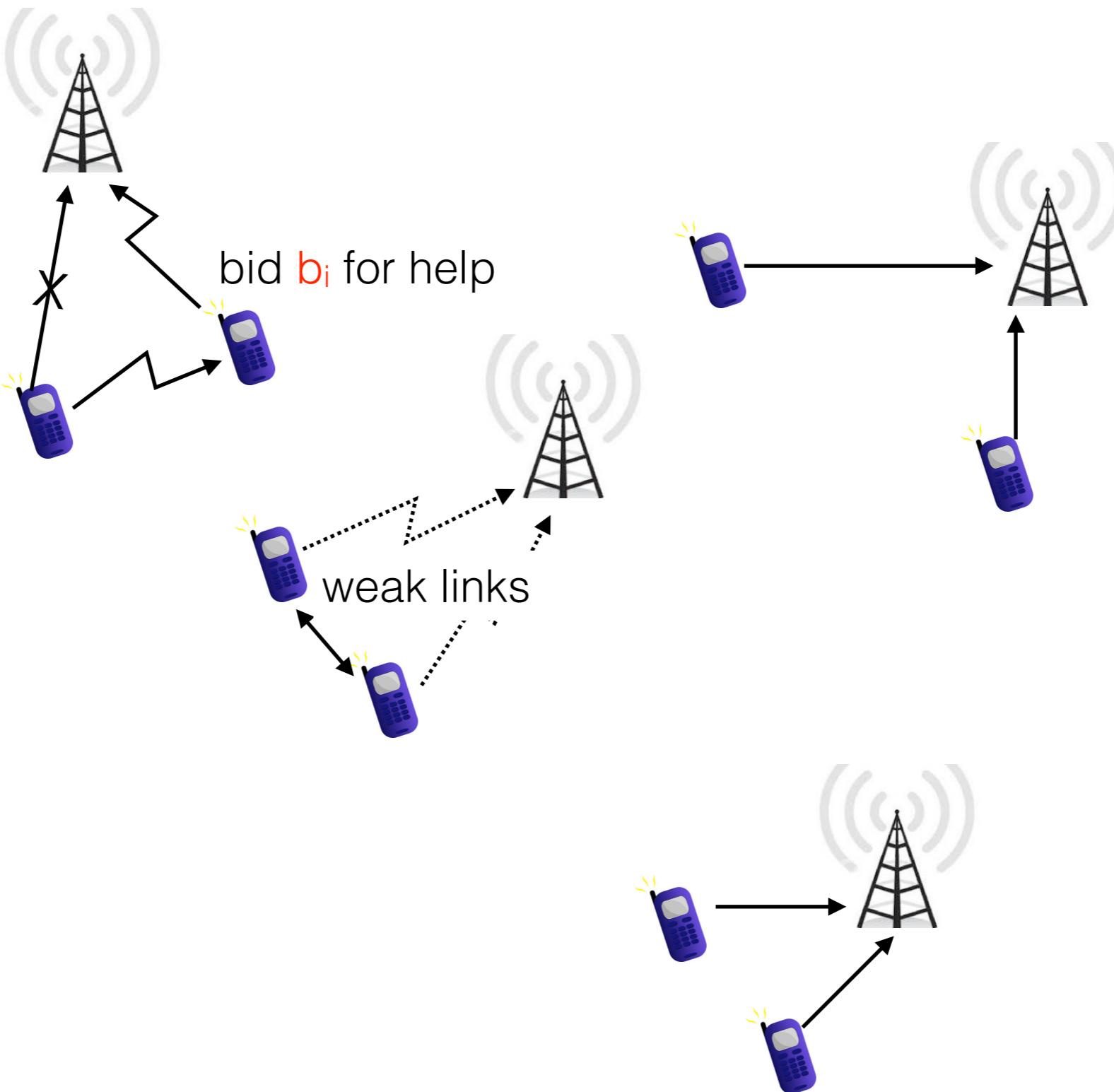
# Modern Problem Device-2-Device Communication



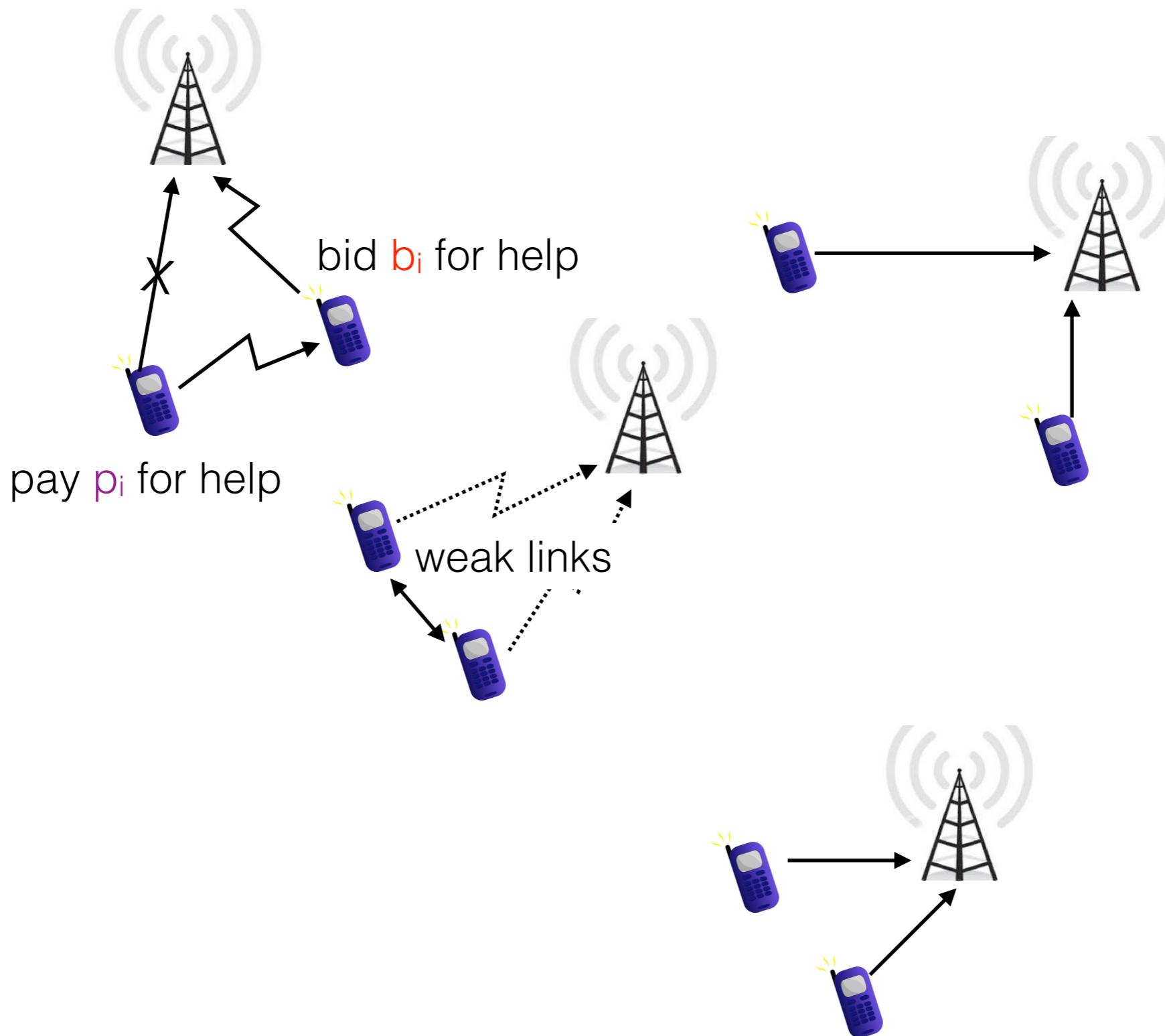
# Modern Problem Device-2-Device Communication



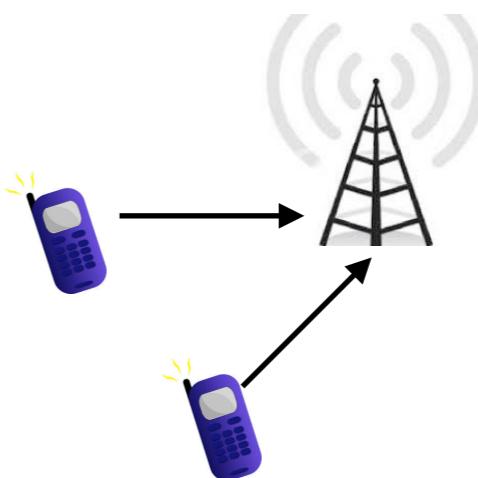
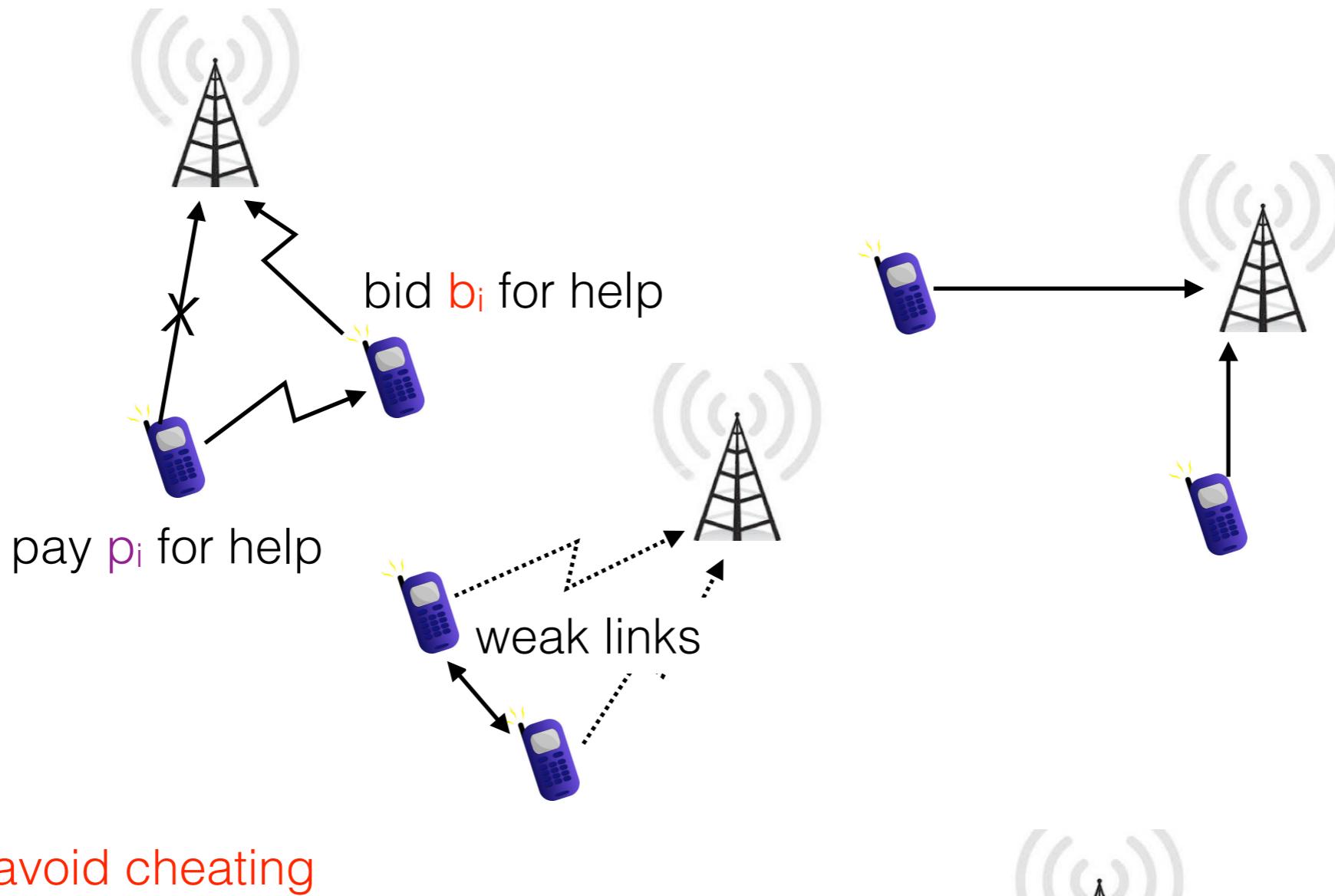
# Modern Problem Device-2-Device Communication



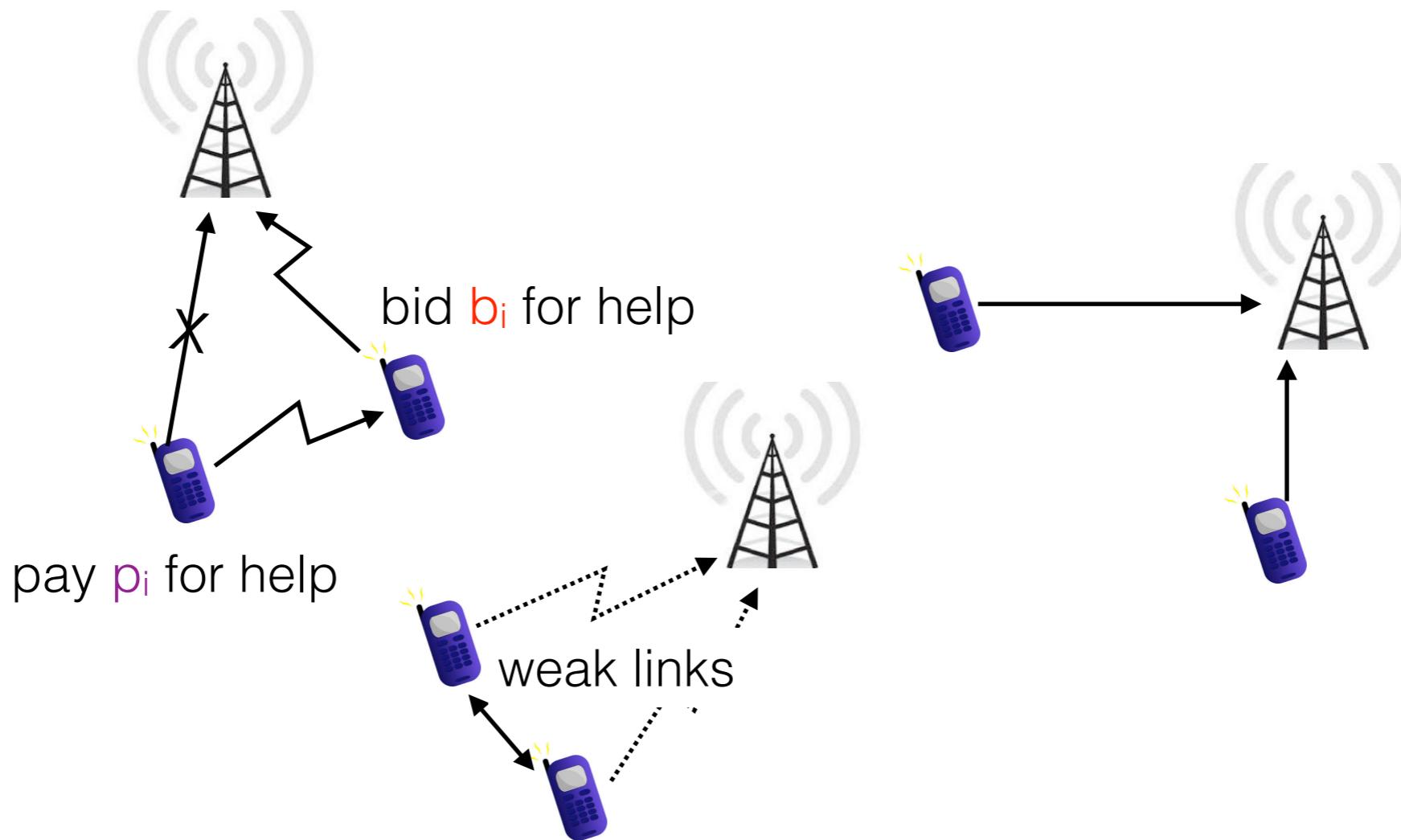
# Modern Problem Device-2-Device Communication



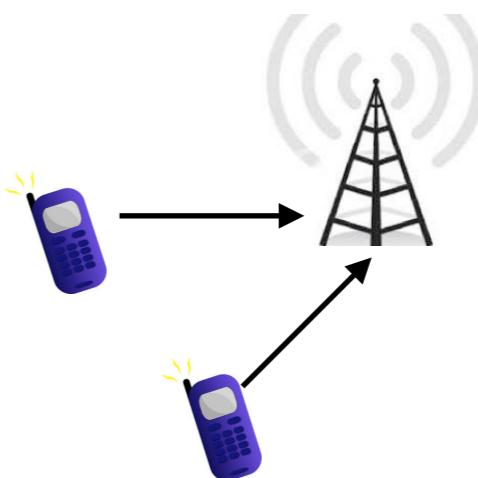
# Modern Problem Device-2-Device Communication



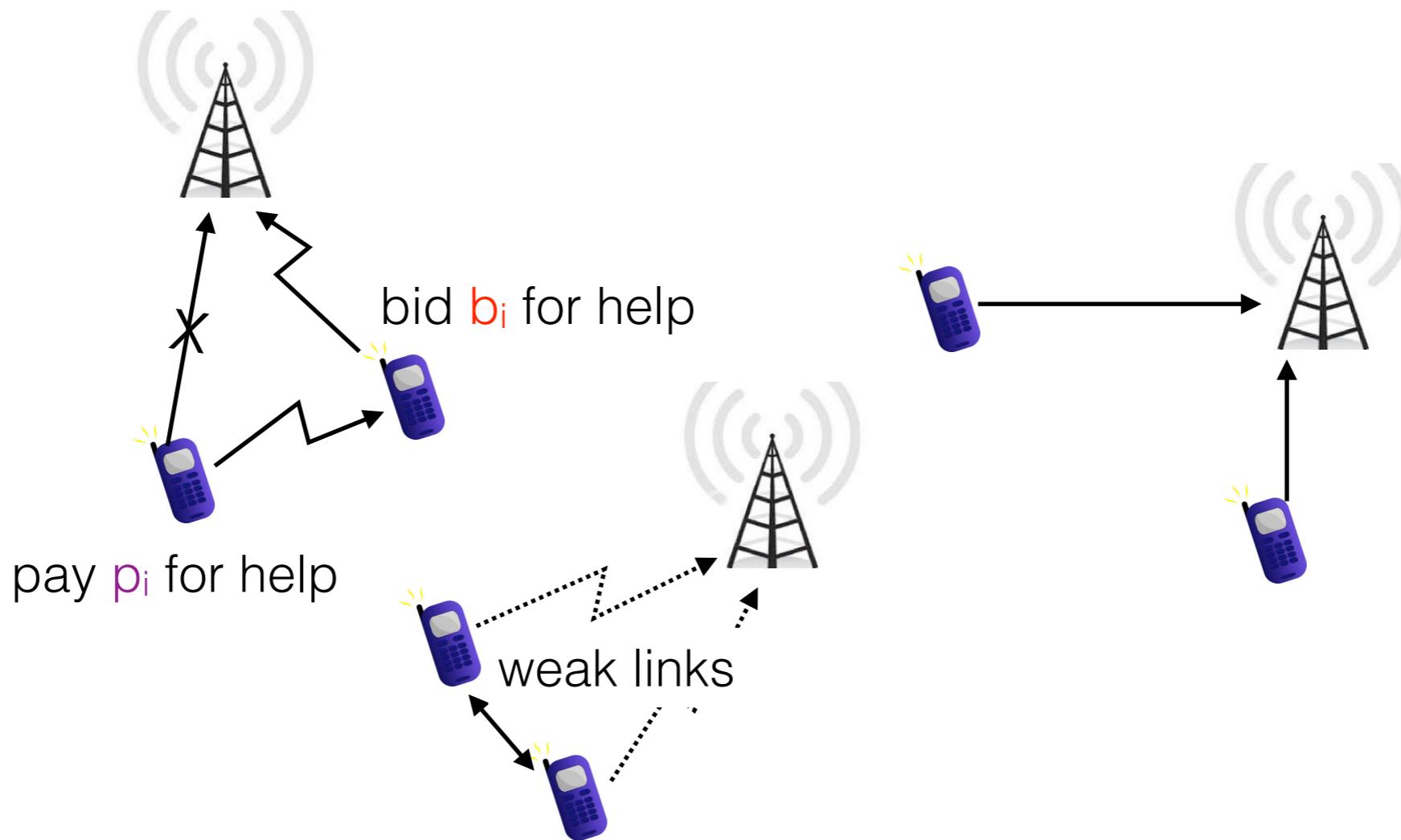
# Modern Problem Device-2-Device Communication



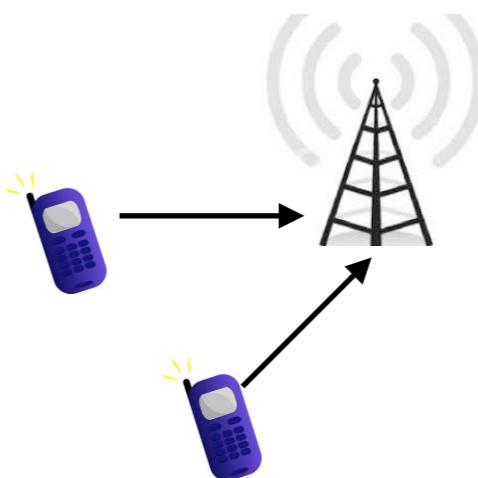
mechanism to avoid cheating  
ensure maximum throughput



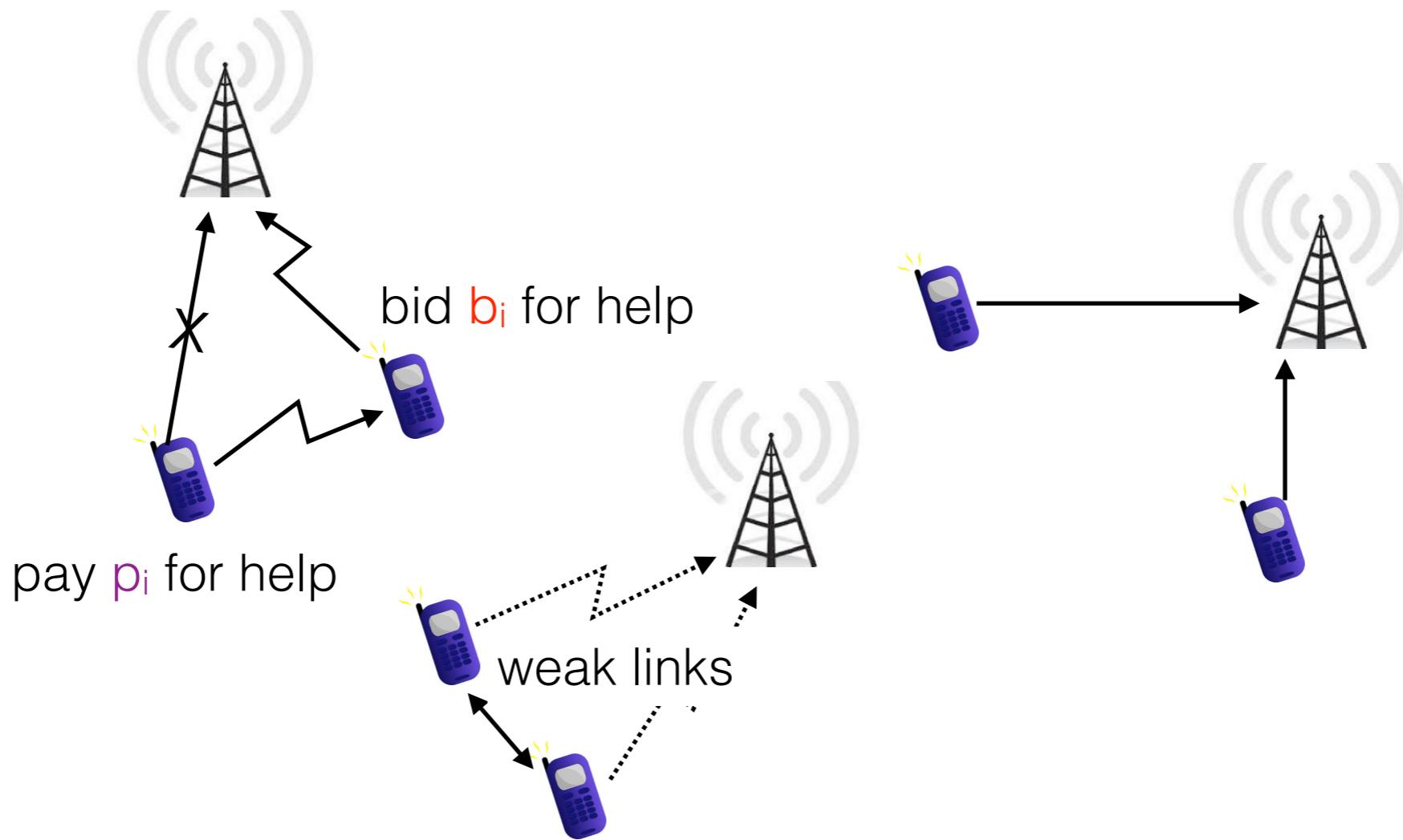
# Modern Problem Device-2-Device Communication



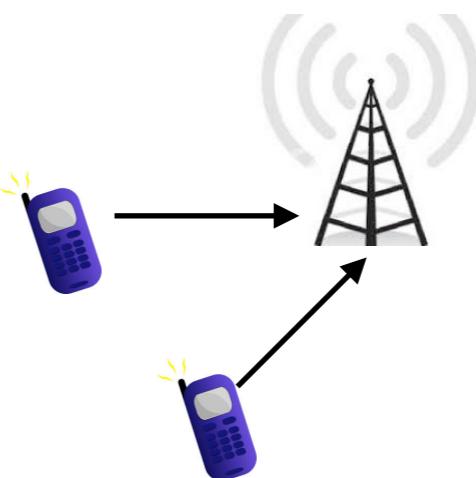
mechanism to avoid cheating  
ensure maximum throughput  
subject to payment budget



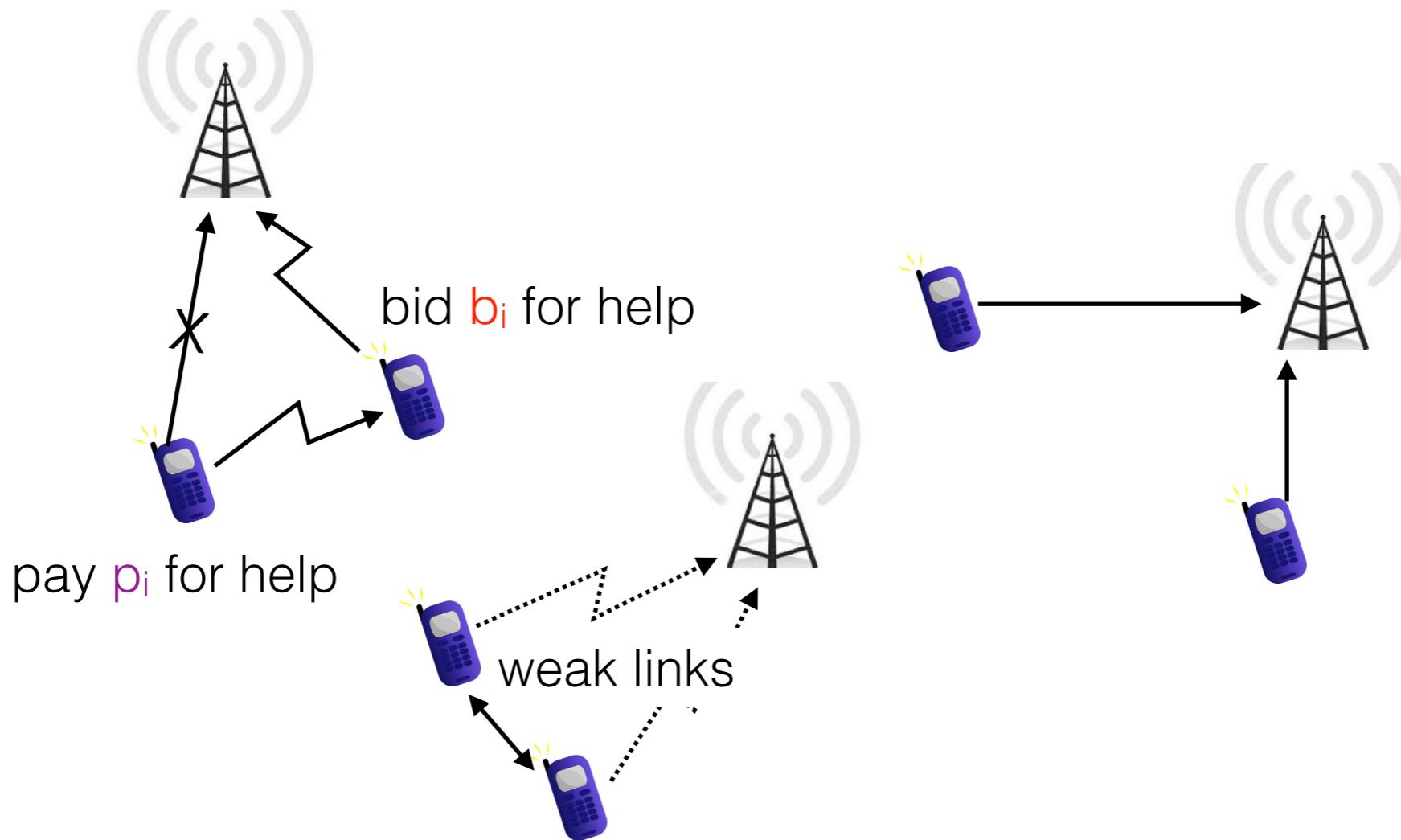
# Modern Problem Device-2-Device Communication



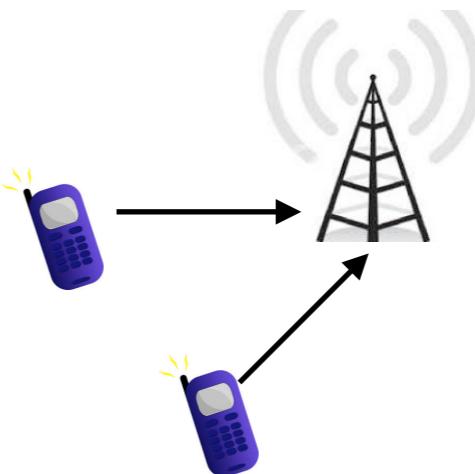
mechanism to avoid cheating  
ensure maximum throughput  
subject to payment budget  
unknown help opportunities



# Modern Problem Device-2-Device Communication

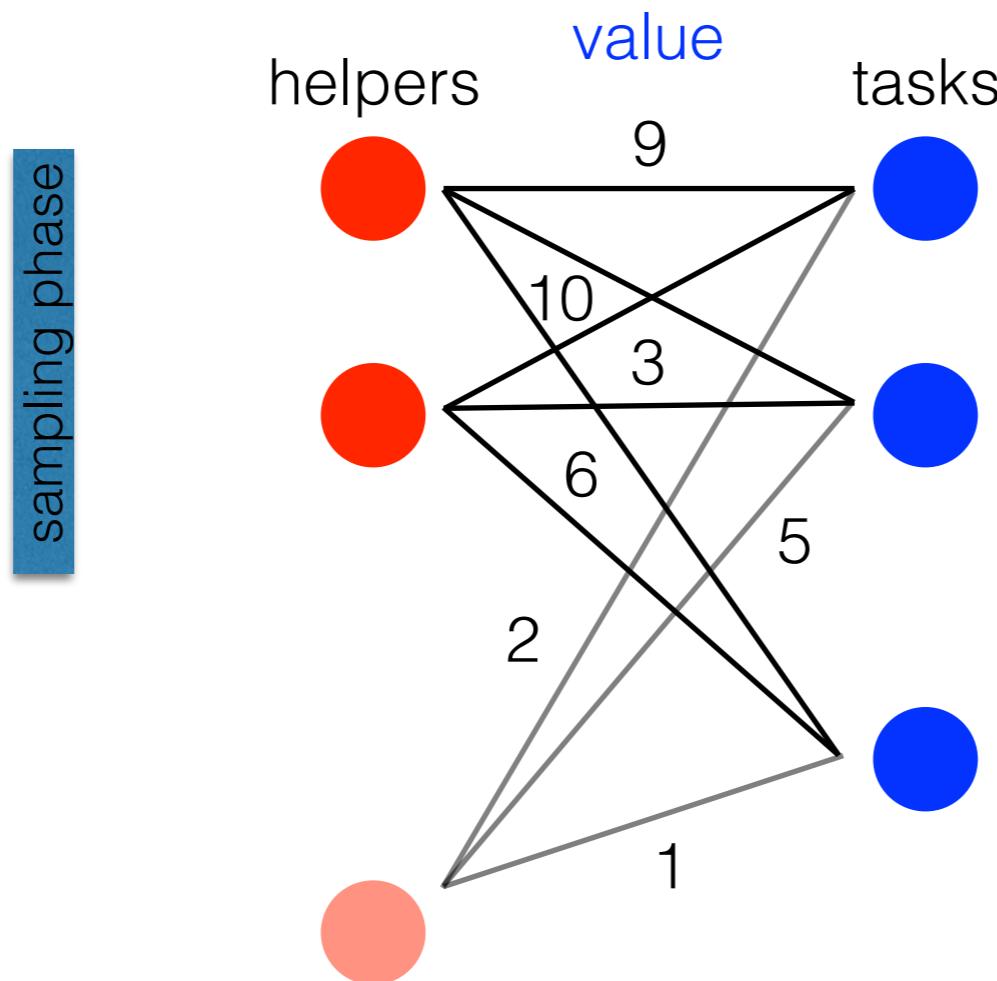


mechanism to avoid cheating  
ensure maximum throughput  
subject to payment budget  
unknown help opportunities



**Find optimal helper association and incentive rule that is truthful**

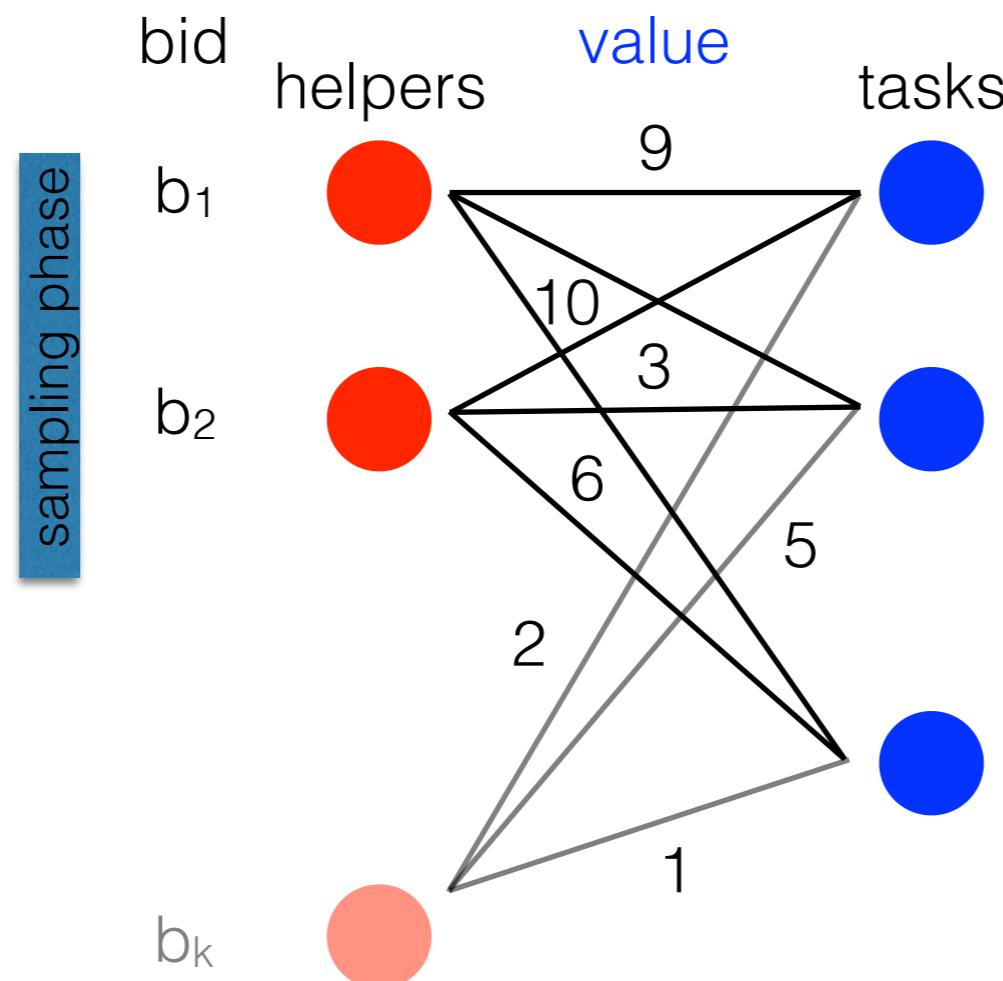
# Possible Strategy for Our Problem



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

# Possible Strategy for Our Problem

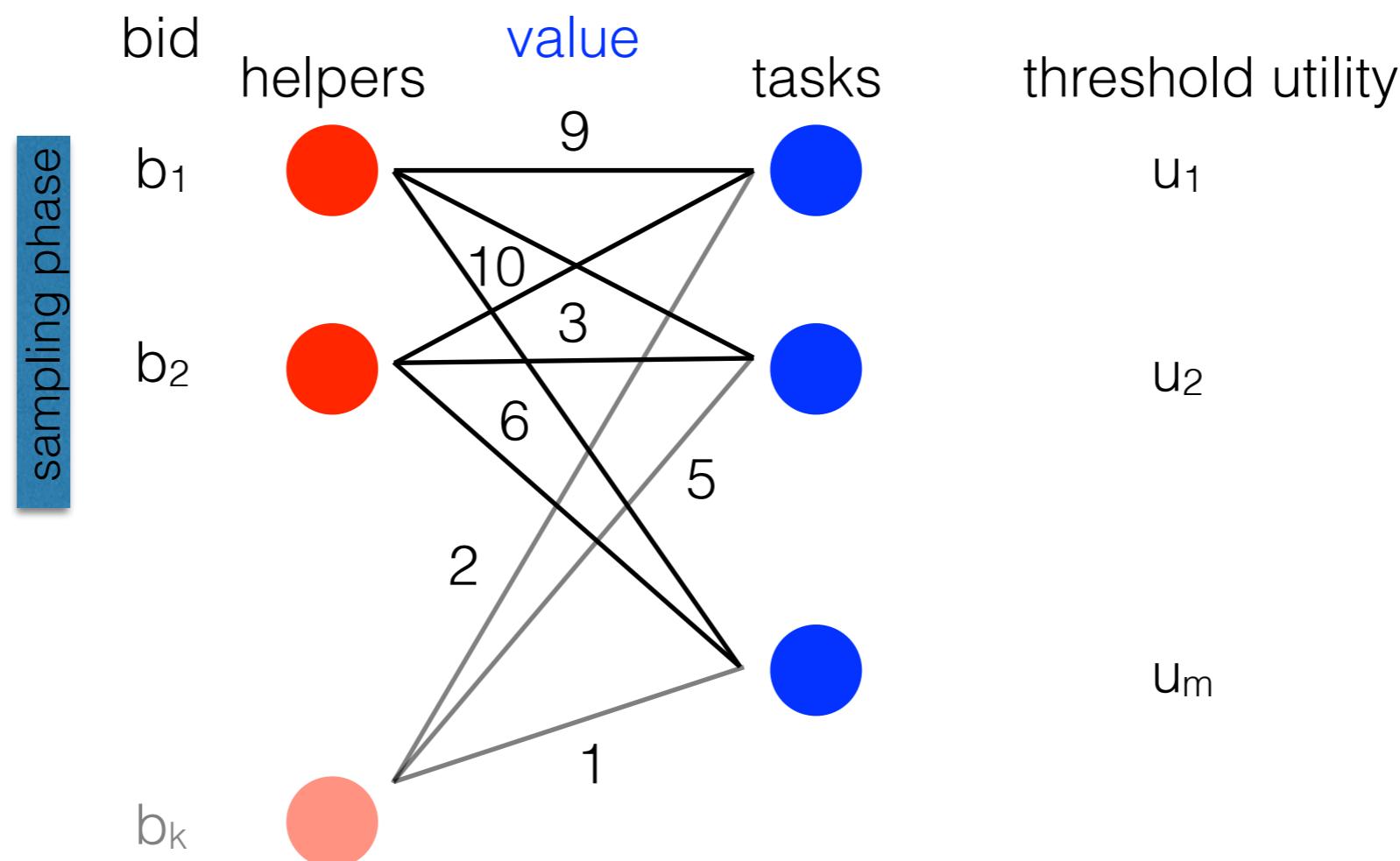
Offline Soln



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

# Possible Strategy for Our Problem

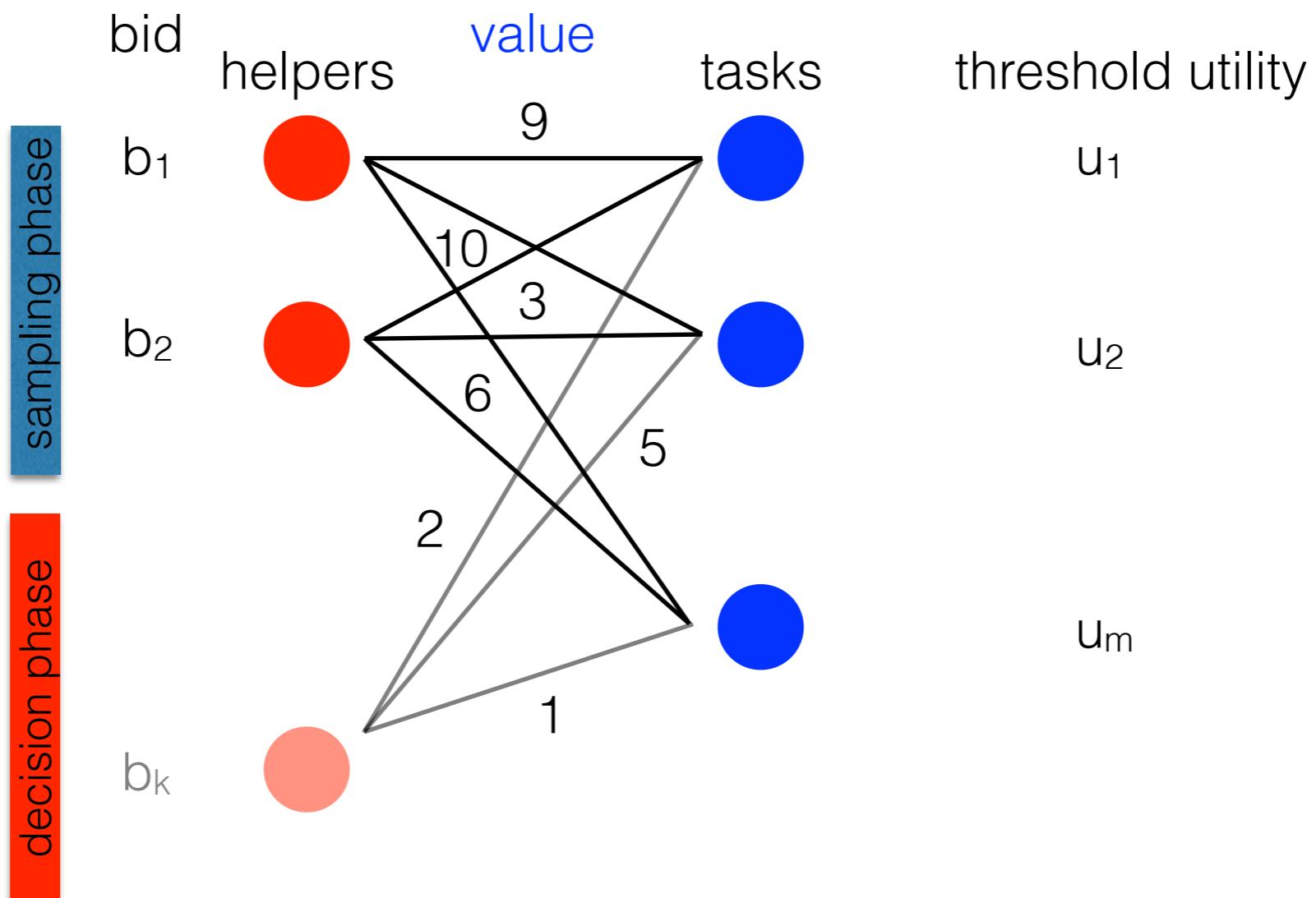
Offline Soln



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

# Possible Strategy for Our Problem

Offline Soln

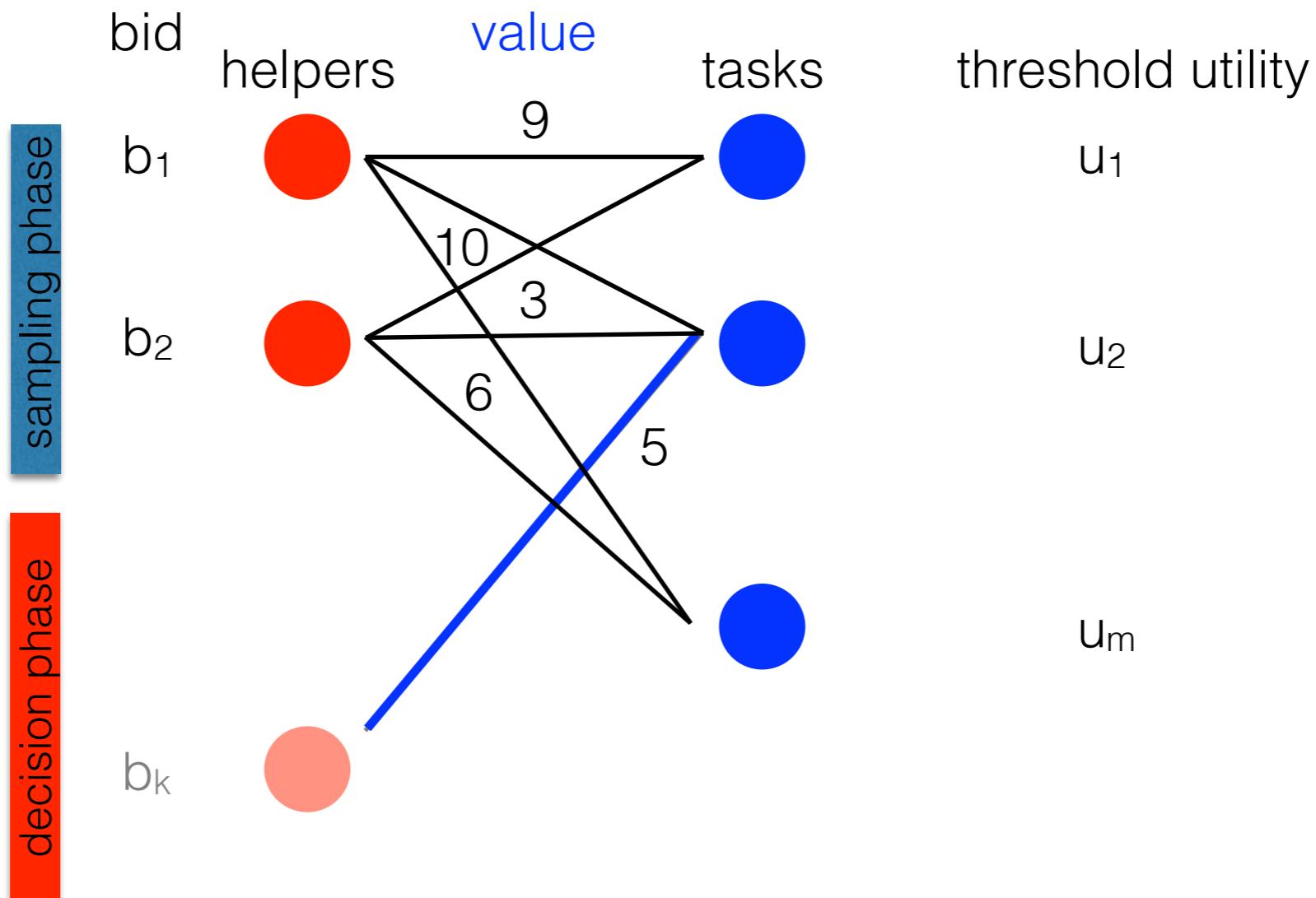


Online Soln

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

# Possible Strategy for Our Problem

Offline Soln

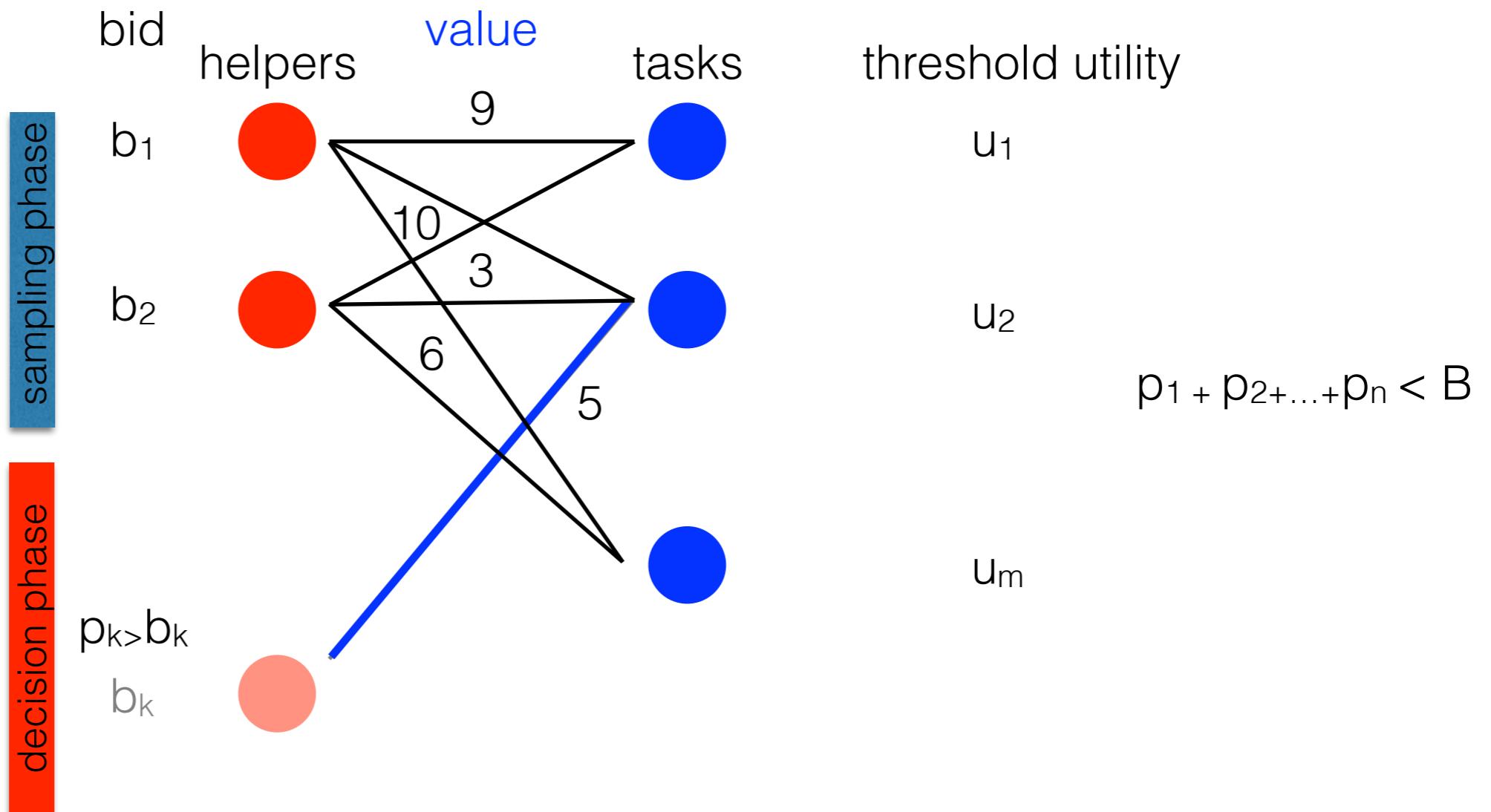


Online Soln

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

# Possible Strategy for Our Problem

Offline Soln



Online Soln

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  $\frac{b(e)}{v(e)} > \frac{B}{u(M)}$

$G = G \setminus \{e\}$  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]

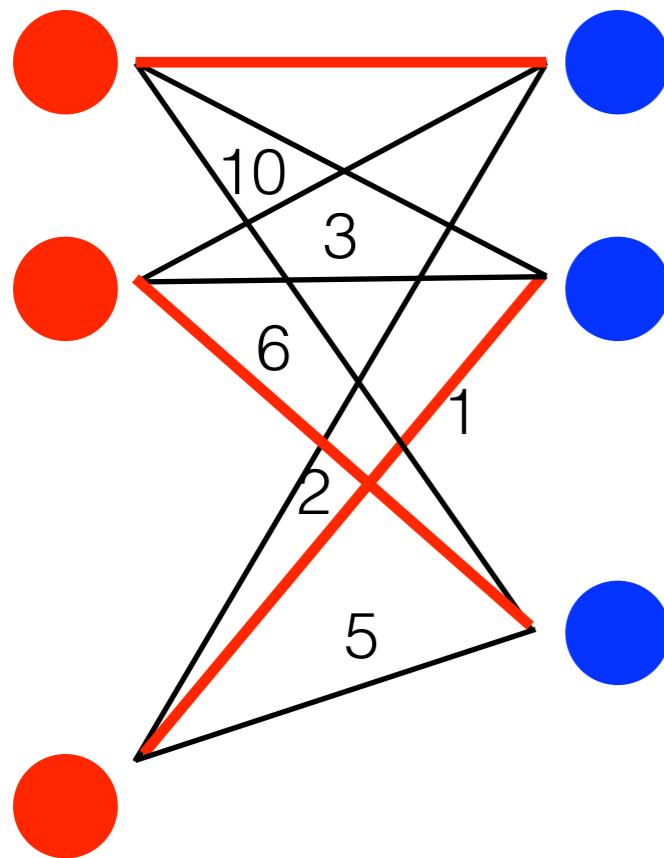
$M$  be greedy matching over  $G$

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

$G = G \setminus \{e\}$  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]

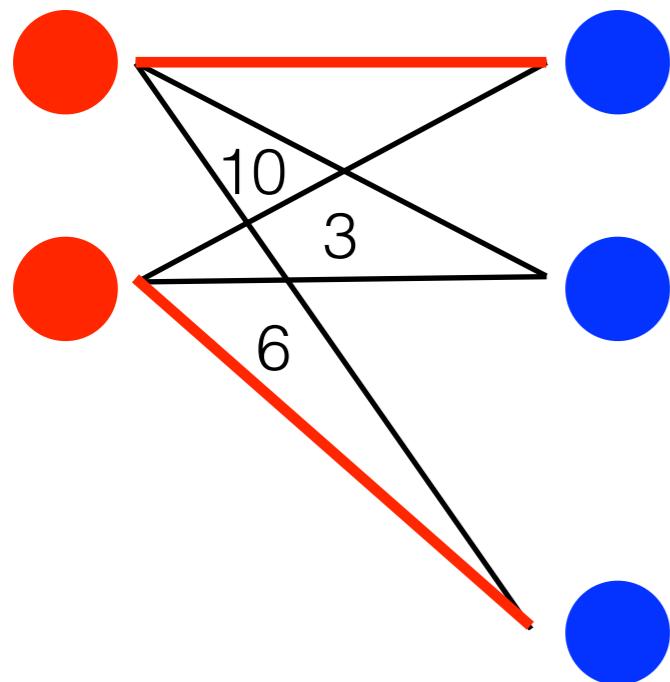
$M$  be greedy matching over  $G$

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

$G = G \setminus \{e\}$  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]

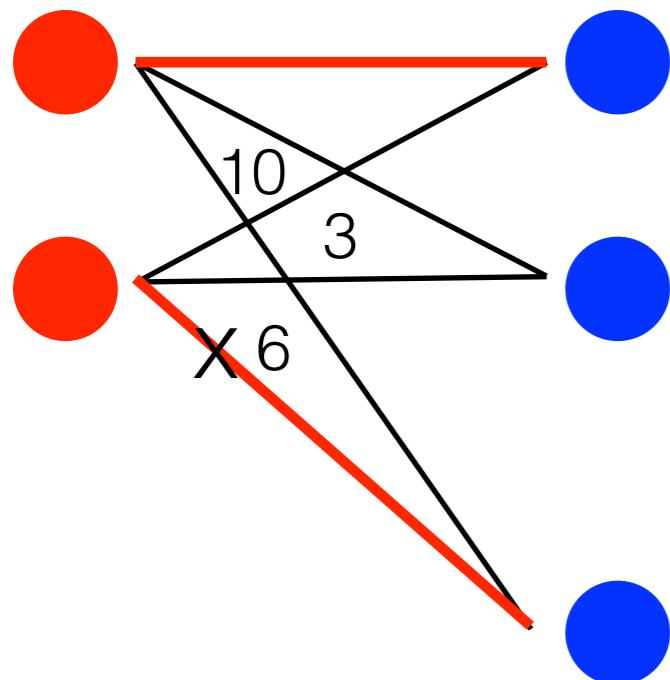
$M$  be greedy matching over  $G$

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

$G = G \setminus \{e\}$  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

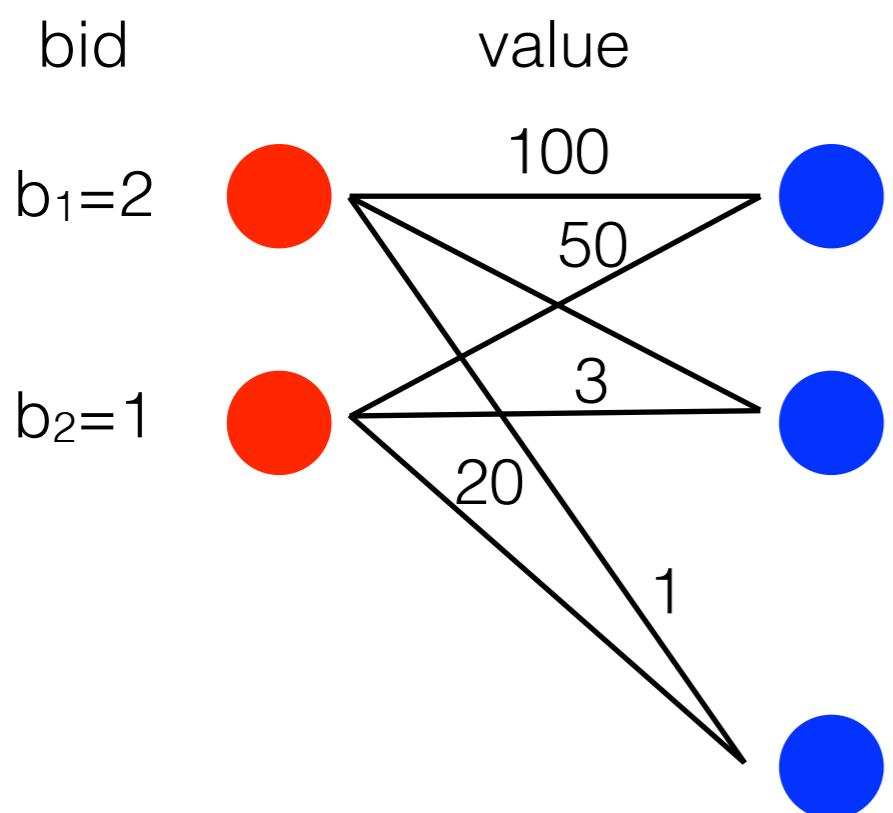
**Algorithm** [Goel, Nikzad, Singla '14]

$M$  be greedy matching over  $G$

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

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

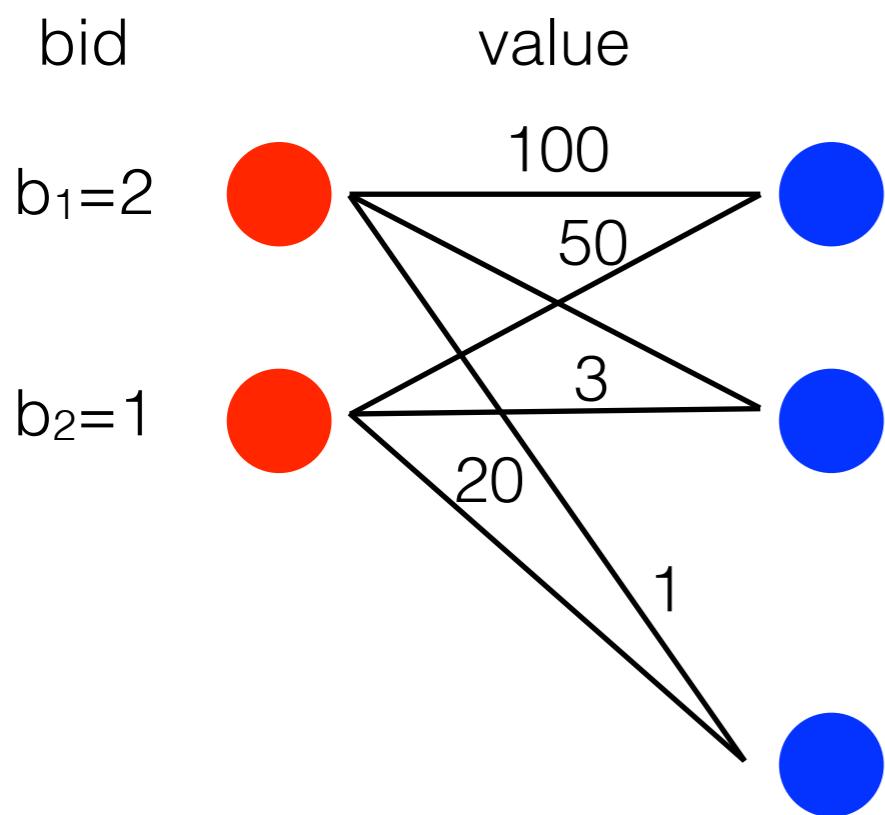
# Our Offline Algorithm OFF



# Our Offline Algorithm OFF



# Our Offline Algorithm OFF

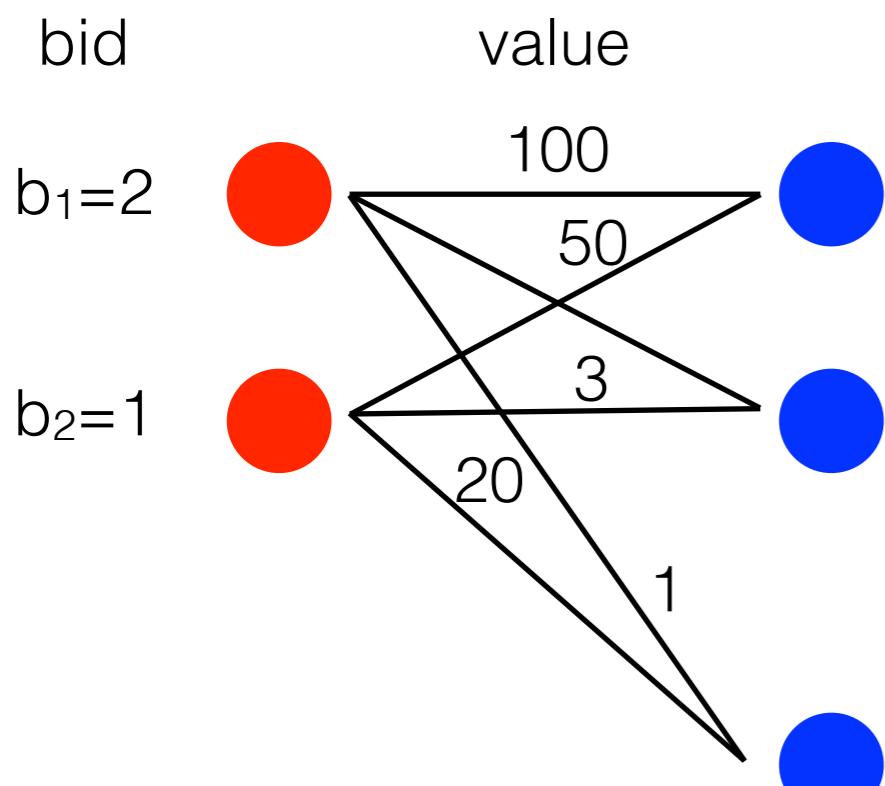


**bid to benefit** ratio of an edge

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

Algorithm

# Our Offline Algorithm OFF



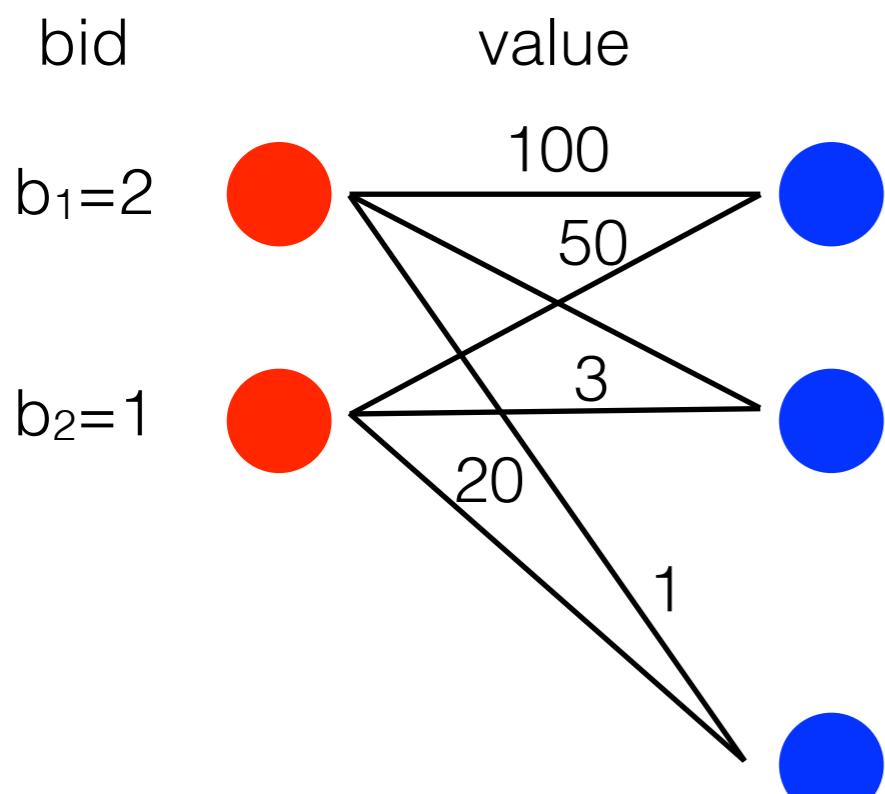
**bid to benefit** ratio of an edge

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

## Algorithm

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

# Our Offline Algorithm OFF



**bid to benefit** ratio of an edge

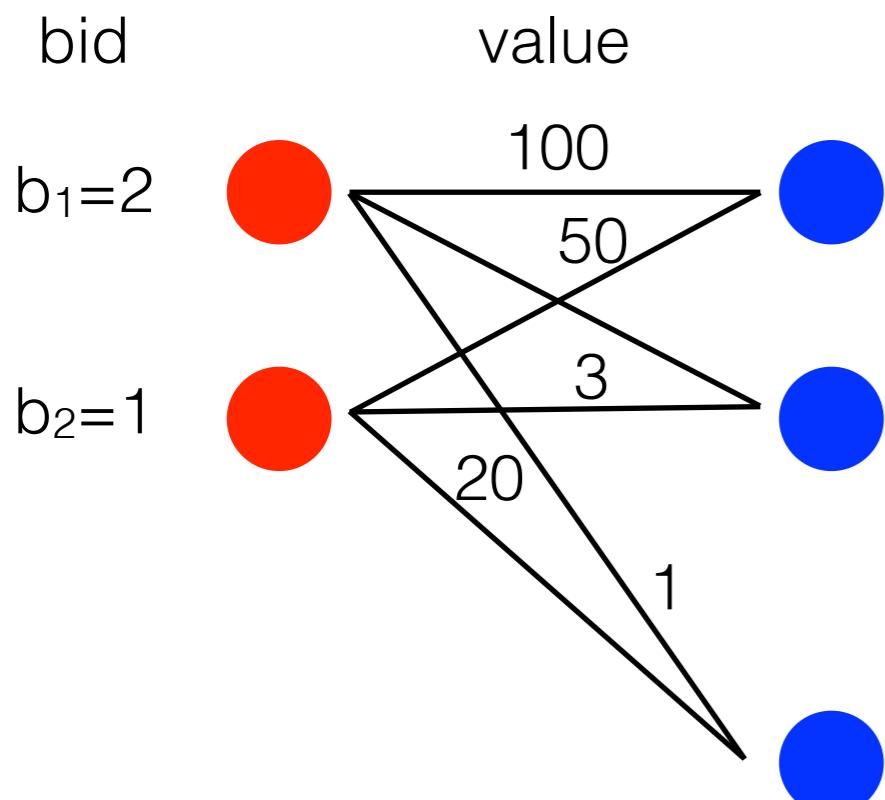
$$\frac{b(e)}{v(e)}$$

## Algorithm

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

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

# Our Offline Algorithm OFF



**bid to benefit** ratio of an edge

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

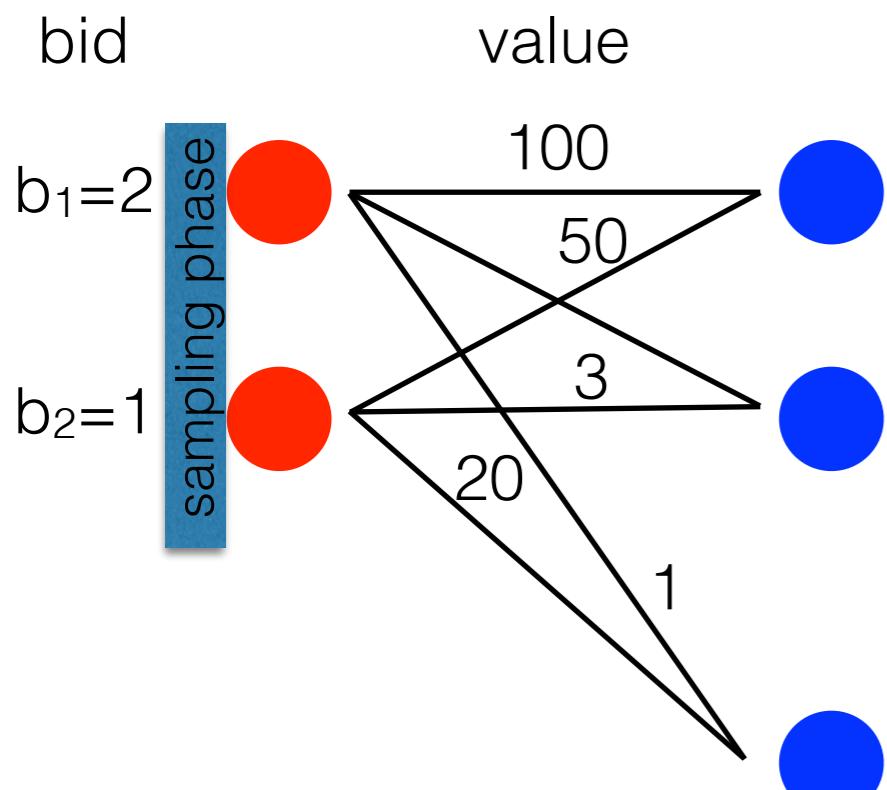
## Algorithm

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

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

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

# Our Offline Algorithm OFF



**bid to benefit** ratio of an edge

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

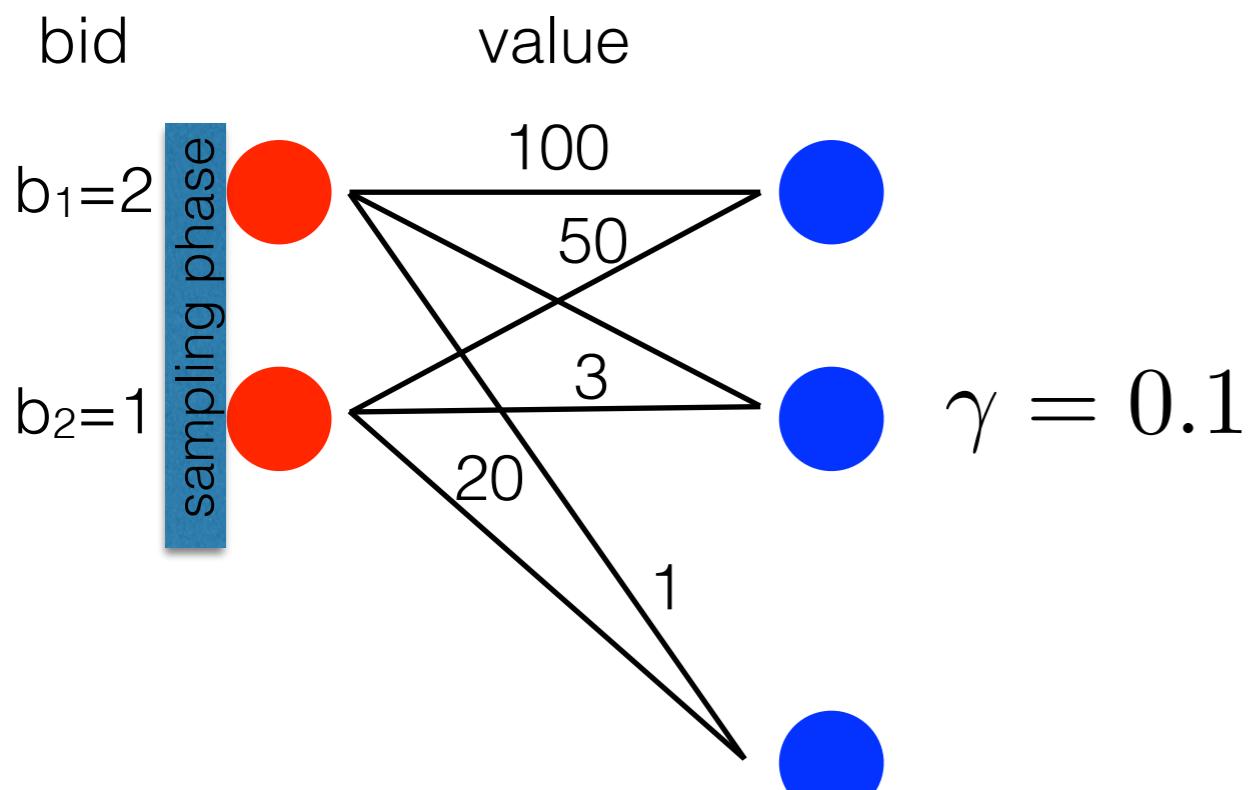
## Algorithm

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



**bid to benefit** ratio of an edge

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

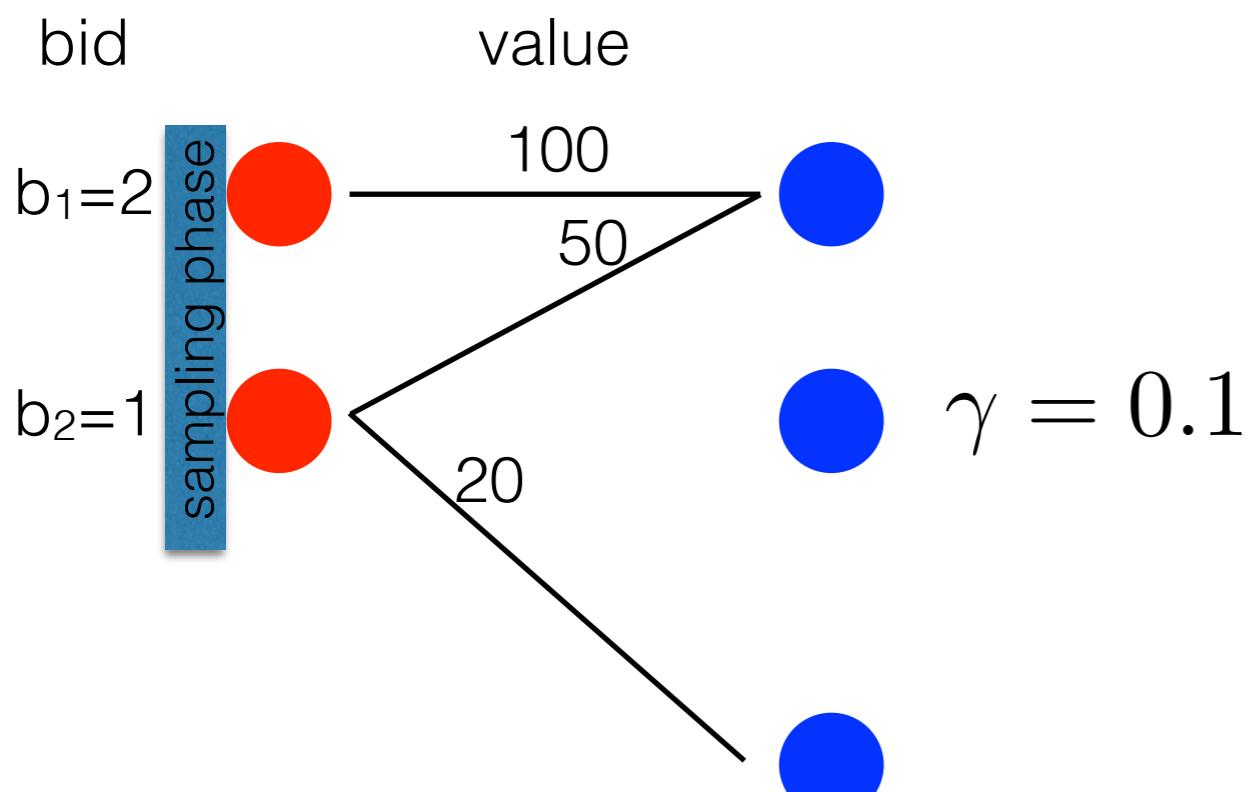
## Algorithm

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



**bid to benefit** ratio of an edge

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

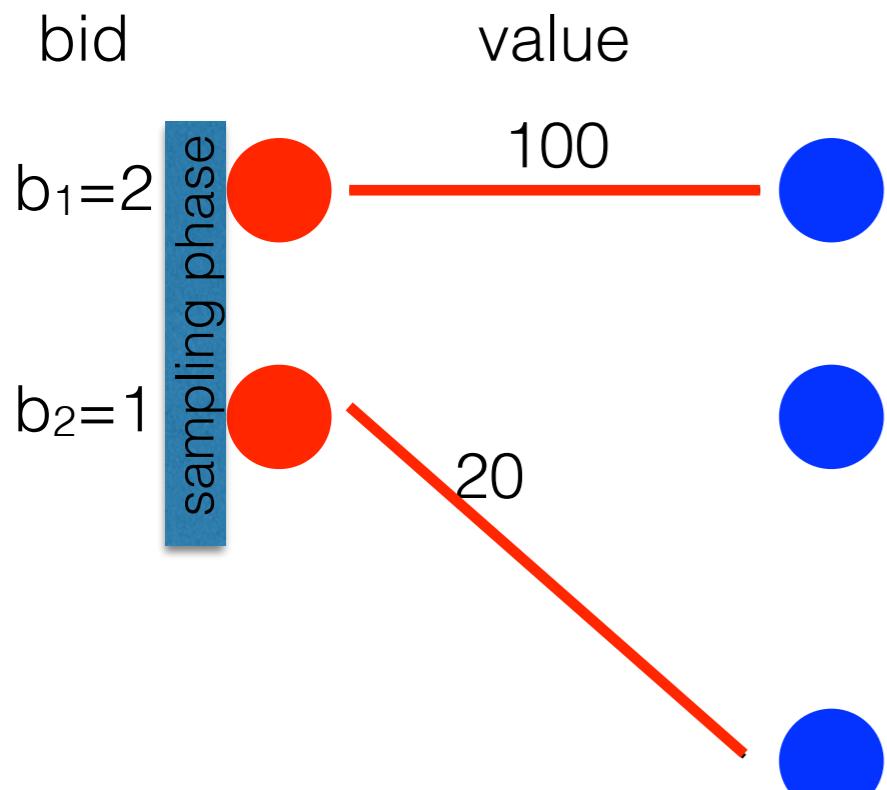
## Algorithm

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



**bid to benefit** ratio of an edge

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

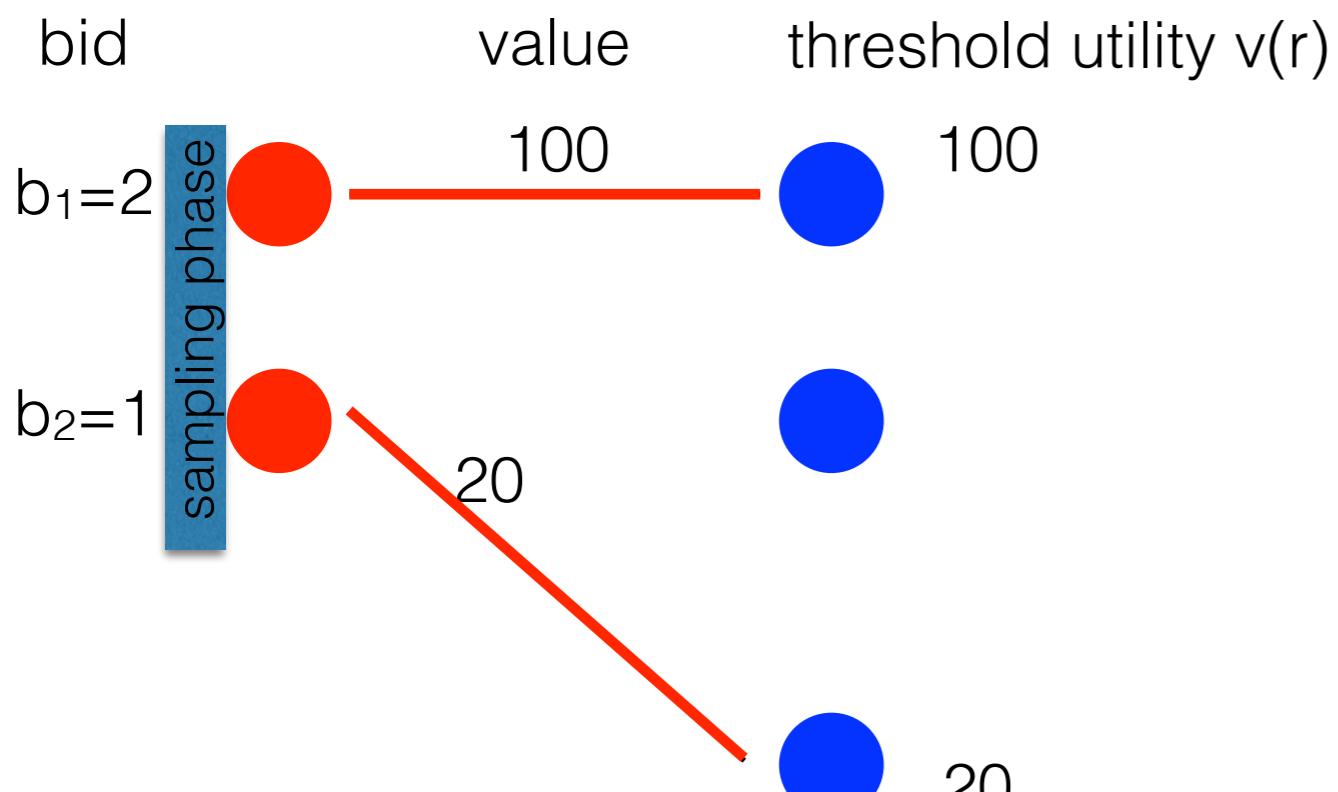
## Algorithm

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



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

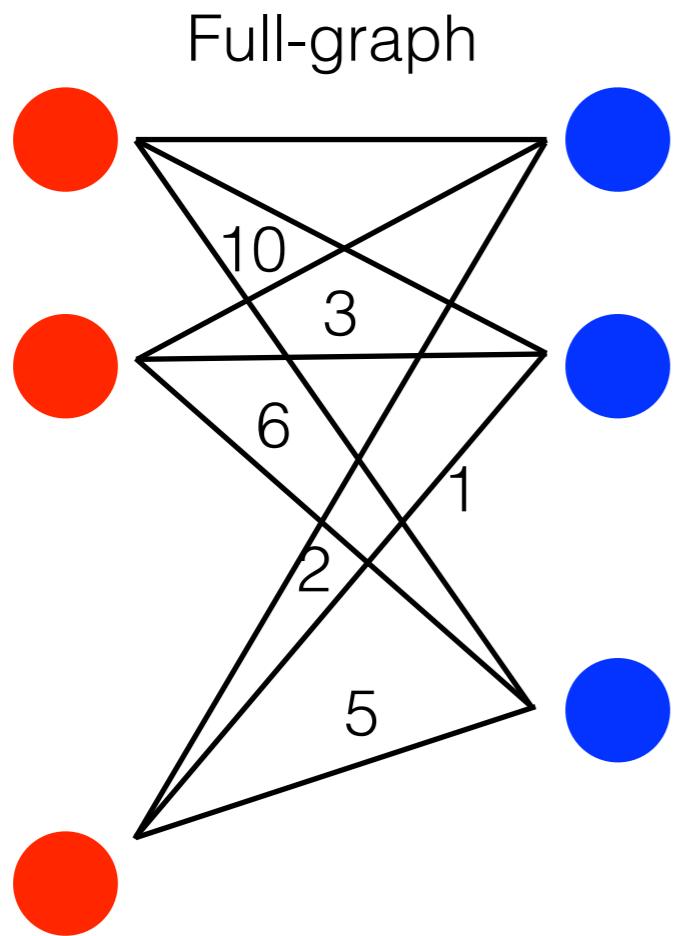
$M(\gamma)$  be greedy matching over  $G(\gamma)$

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

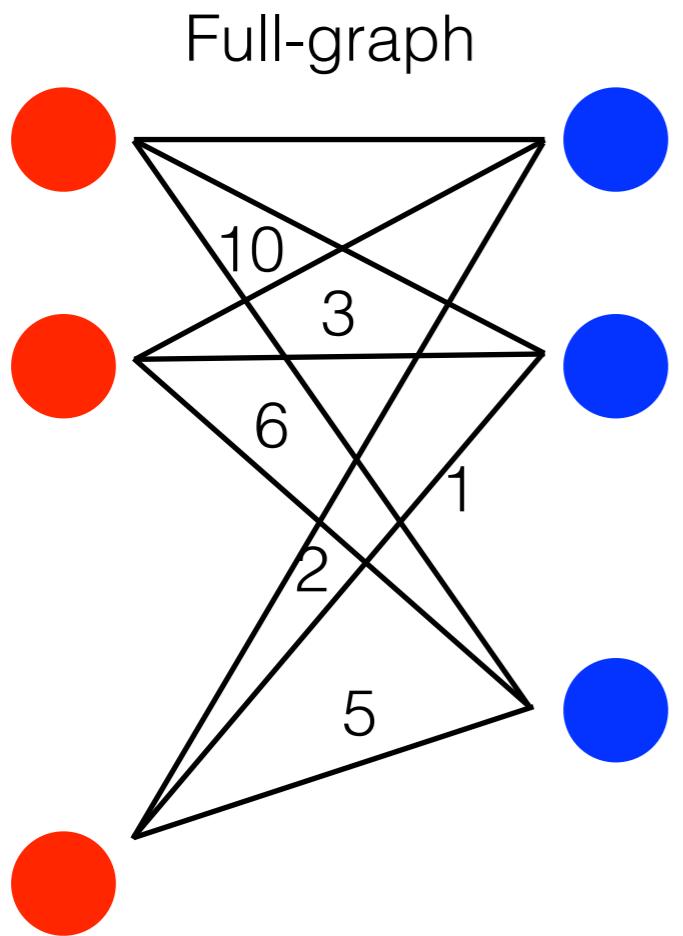
Claim : OFF is 12 - approximate in expectation

# Properties of OFF

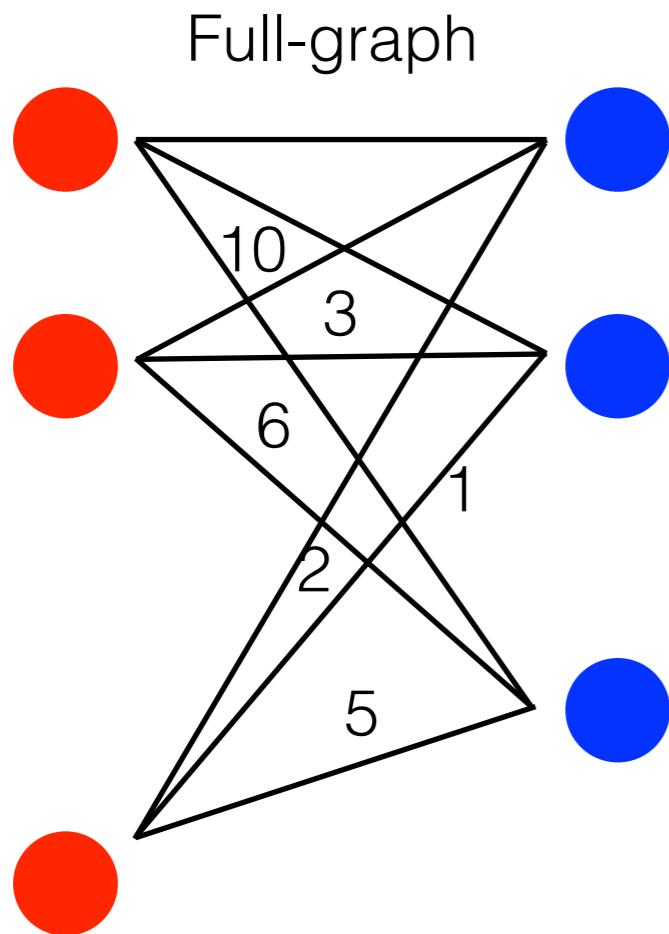
# Properties of OFF



# Properties of OFF



# Properties of OFF

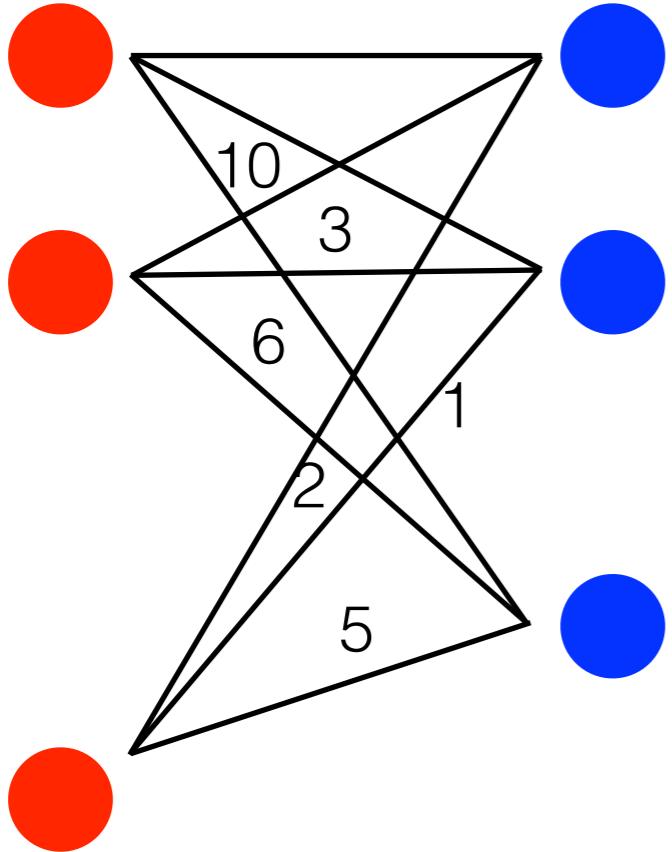


Greedy matching with edges

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

# Properties of OFF

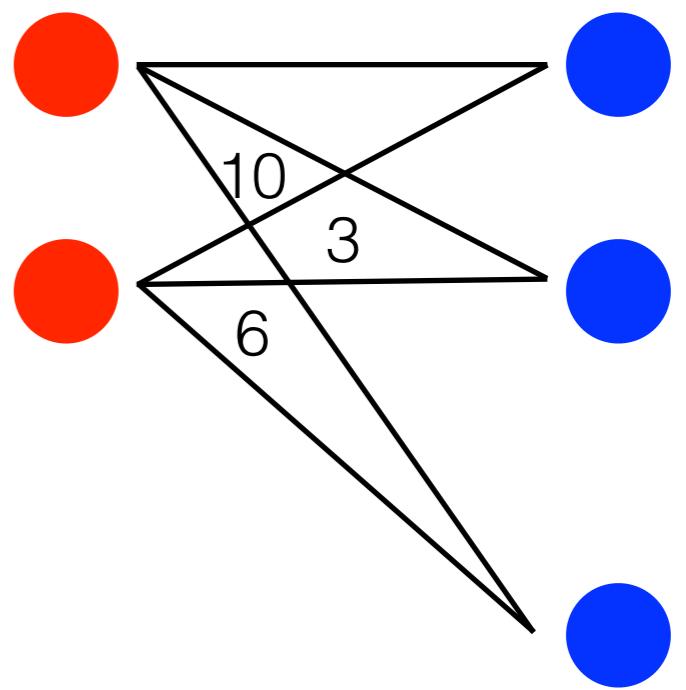
Full-graph



Greedy matching with edges

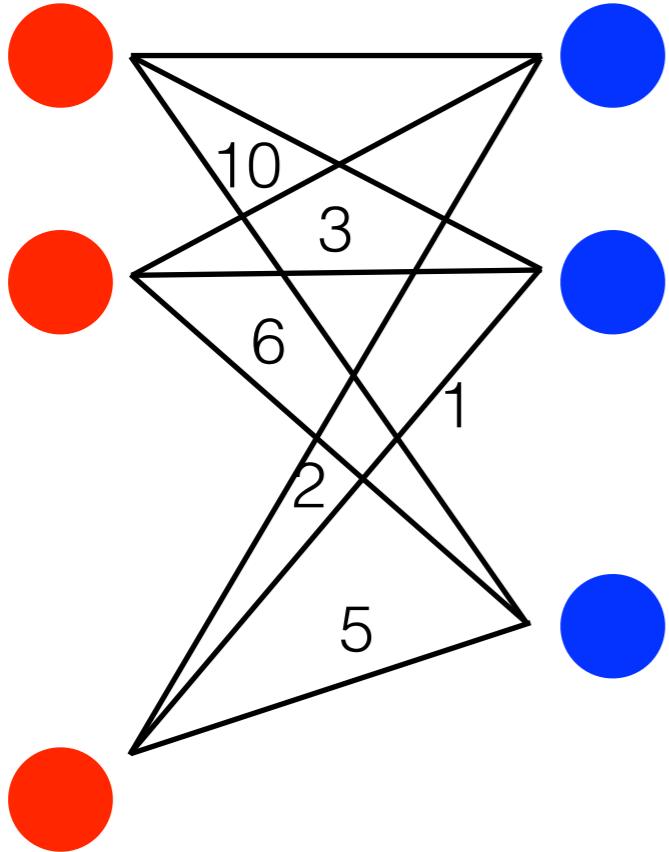
$$\frac{b(e)}{v(e)} < \gamma$$

half-graph



# Properties of OFF

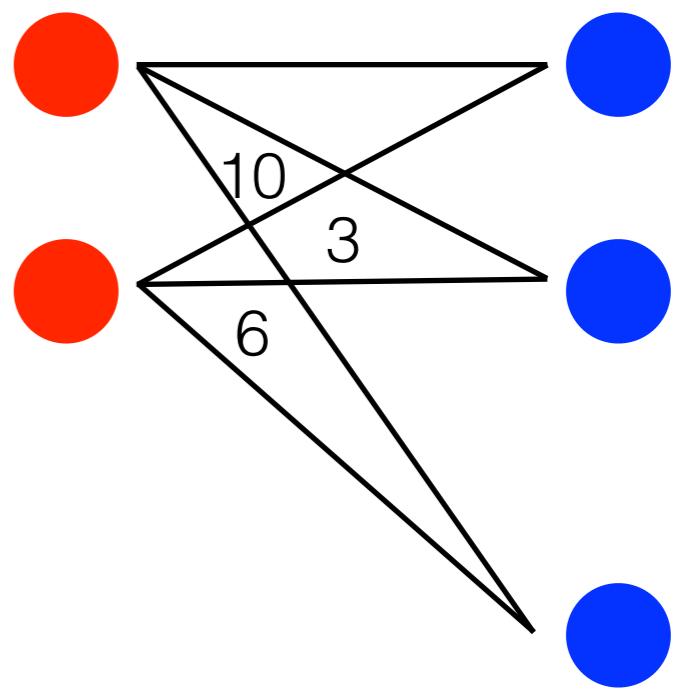
Full-graph



Greedy matching with edges

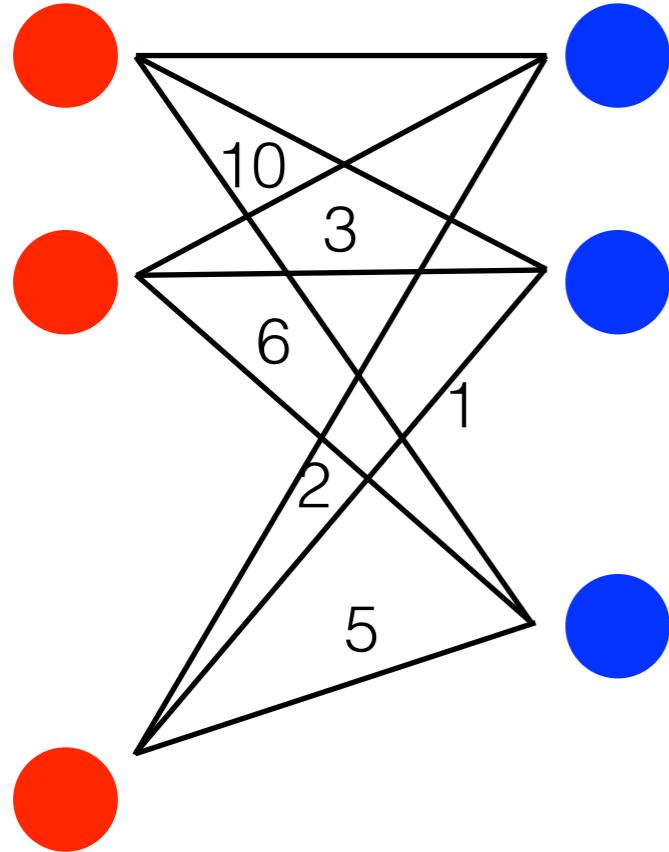
$$\frac{b(e)}{v(e)} < \gamma$$
$$\gamma_f \leq \gamma_h$$

half-graph



# Properties of OFF

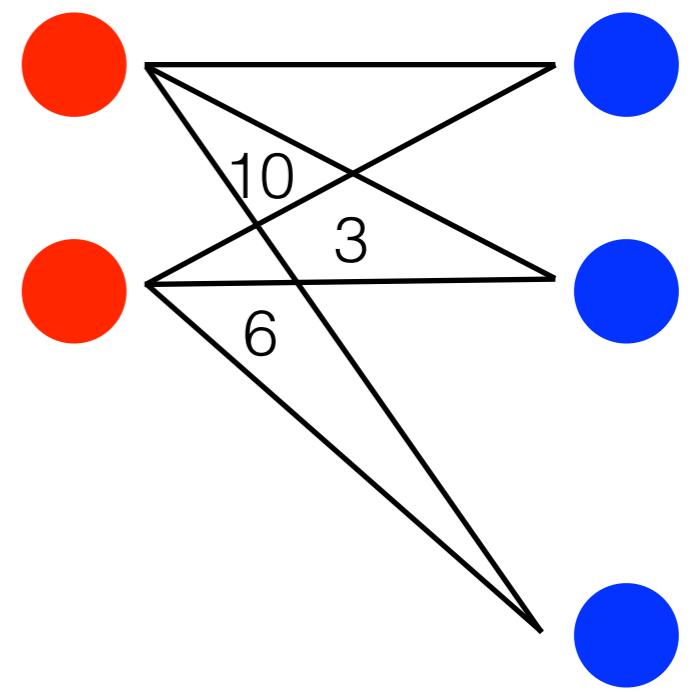
Full-graph



Greedy matching with edges

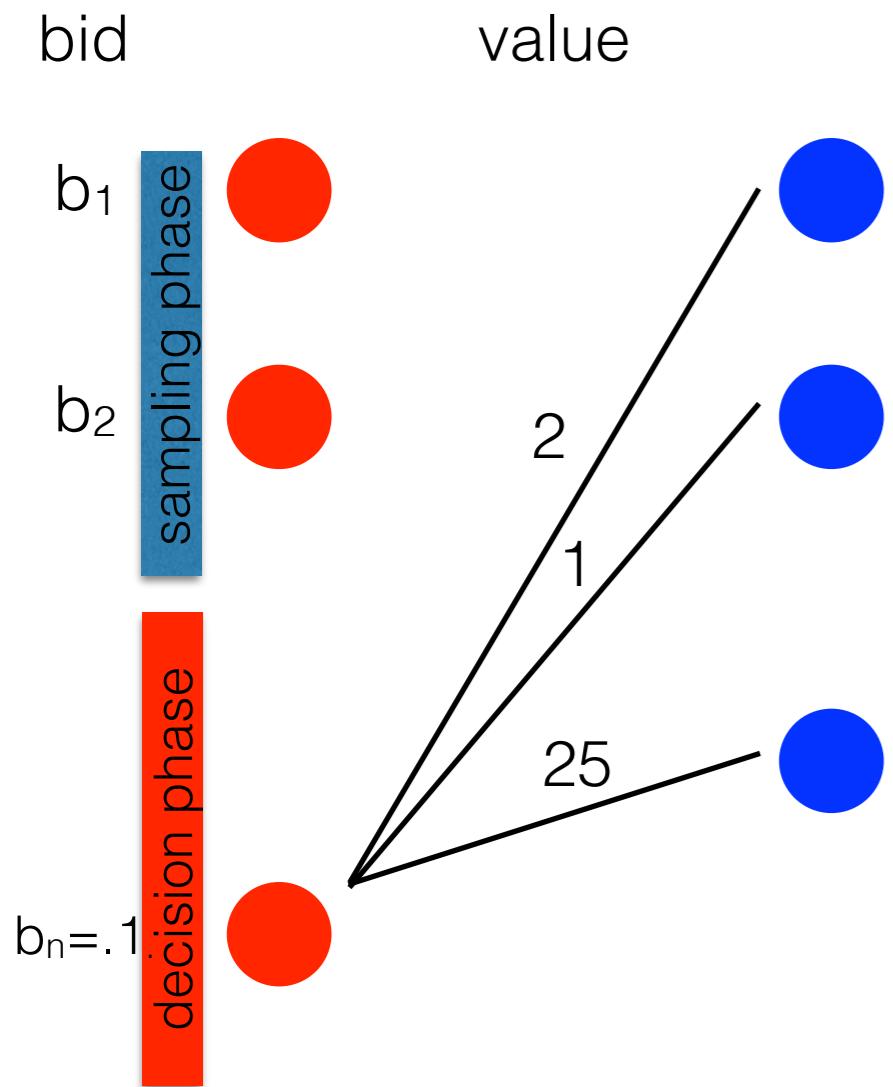
$$\frac{b(e)}{v(e)} < \gamma$$
$$\gamma_f \leq \gamma_h$$

half-graph



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

# Algorithm - ON

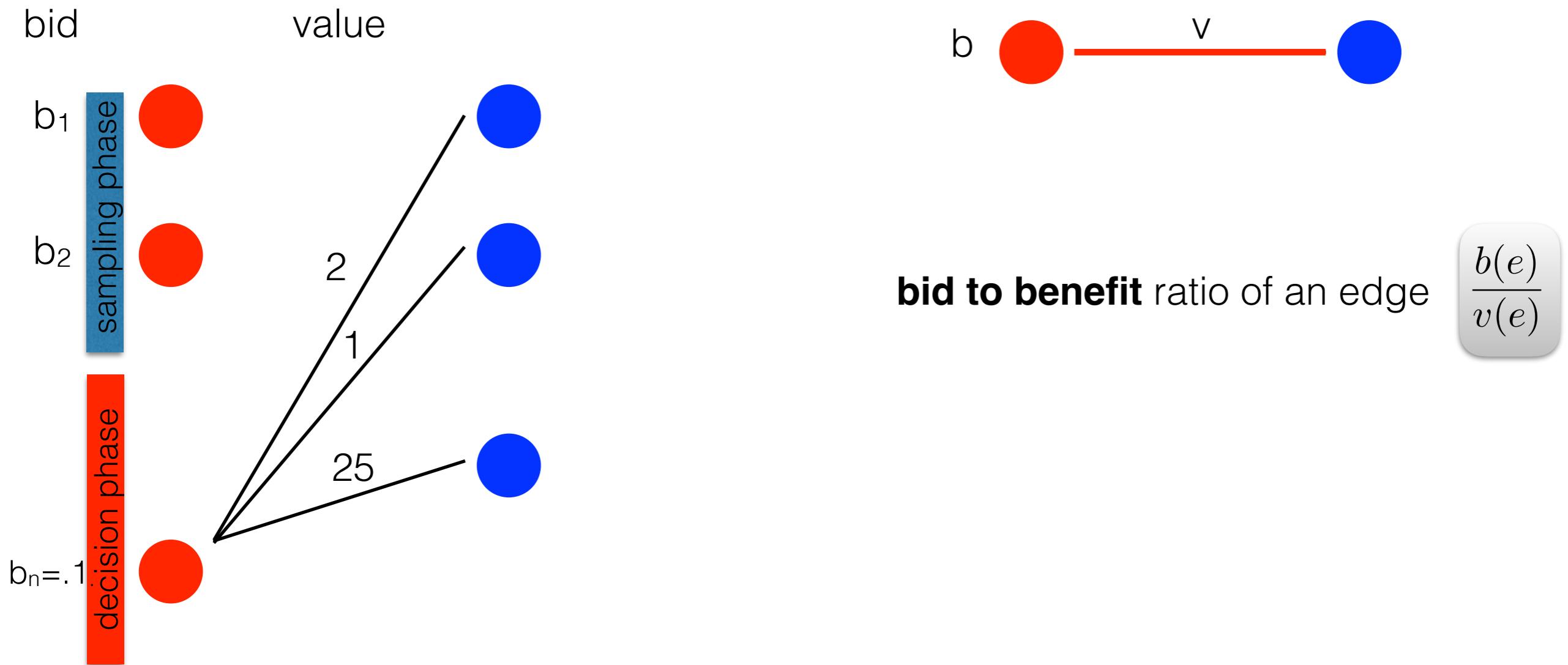


**bid to benefit** ratio of an edge

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

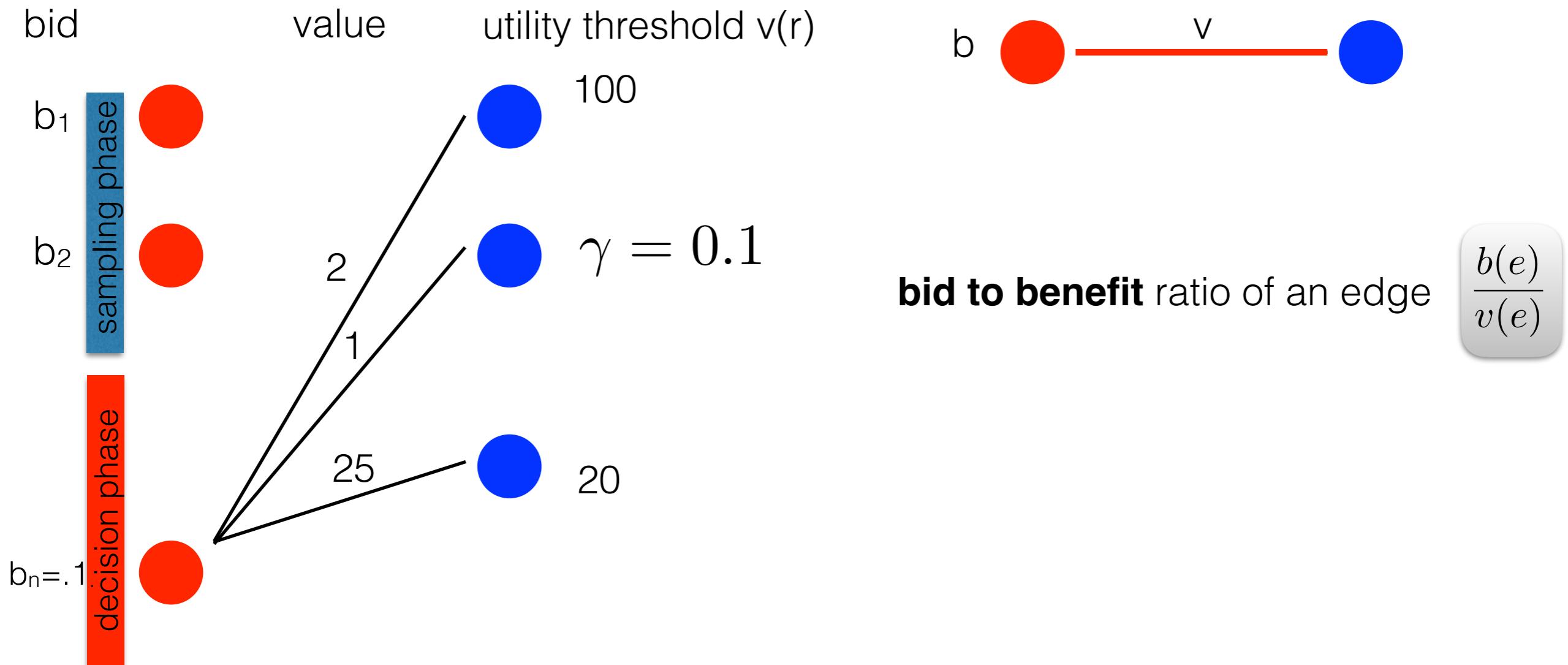
# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



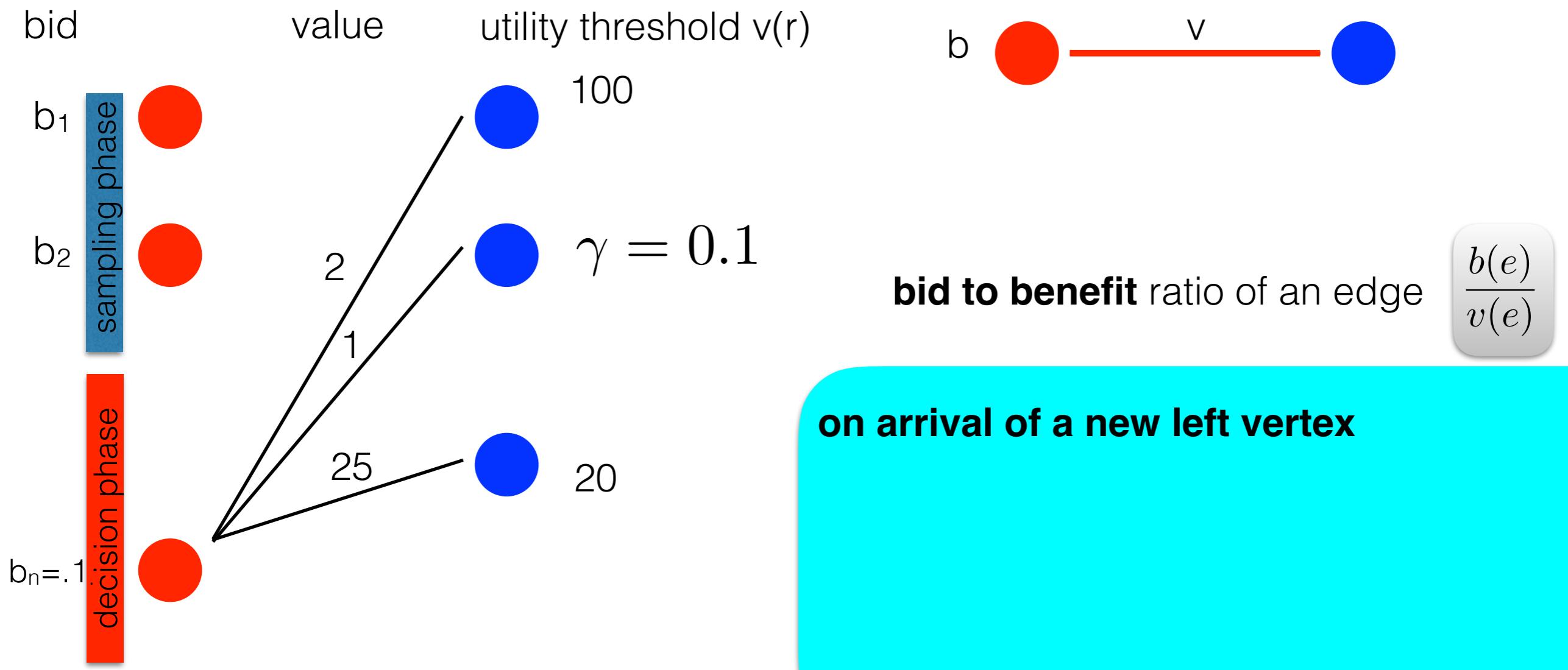
# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



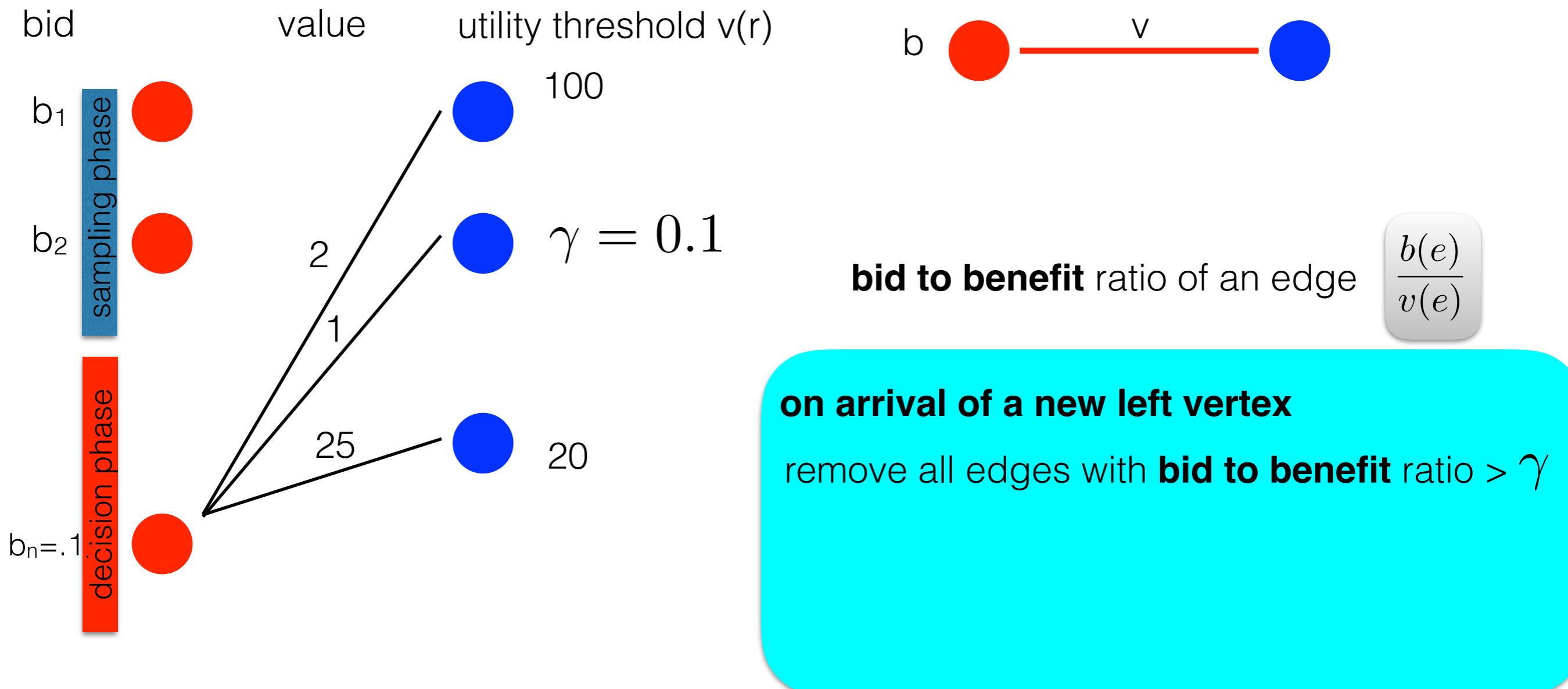
# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



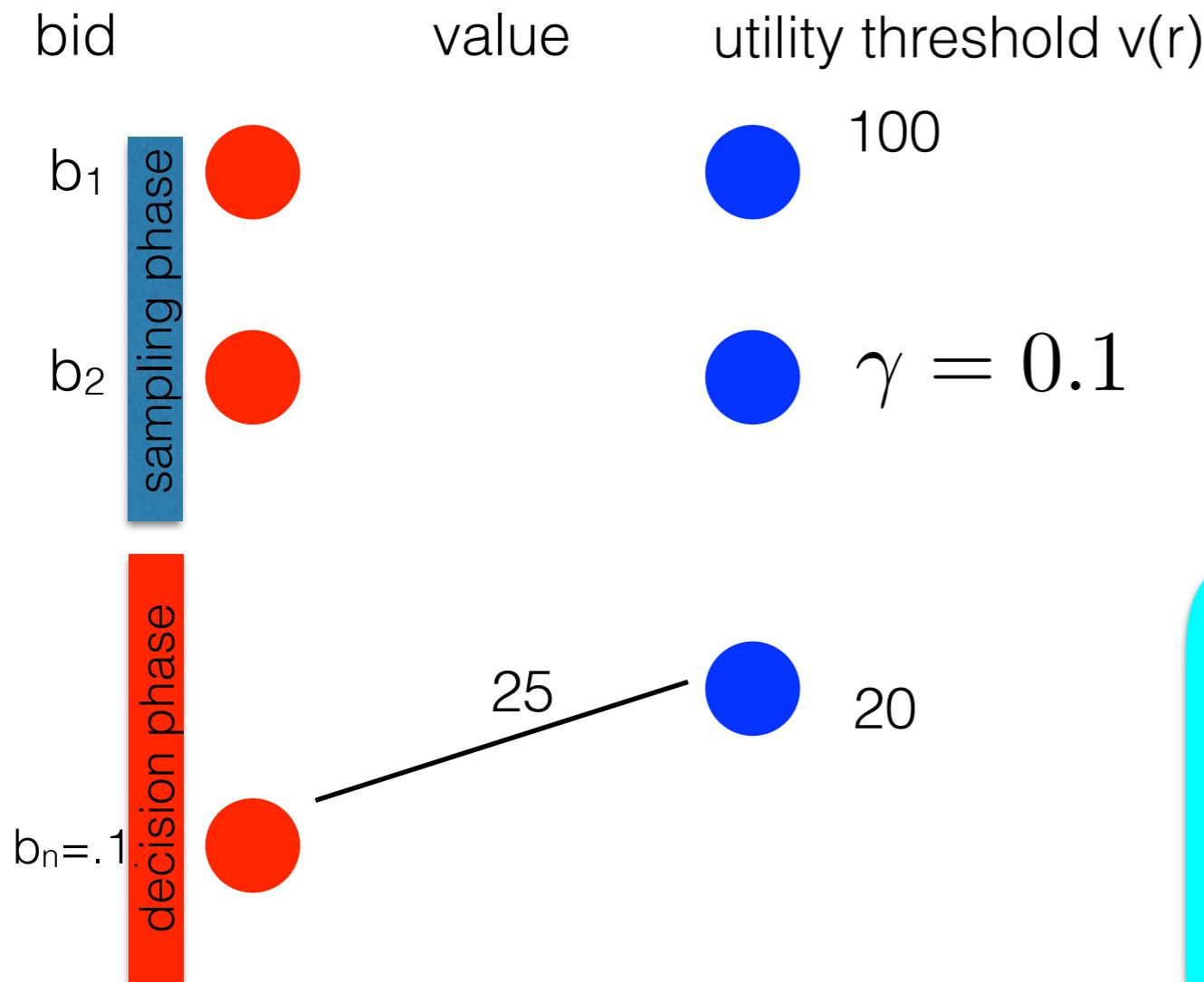
# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



**bid to benefit** ratio of an edge

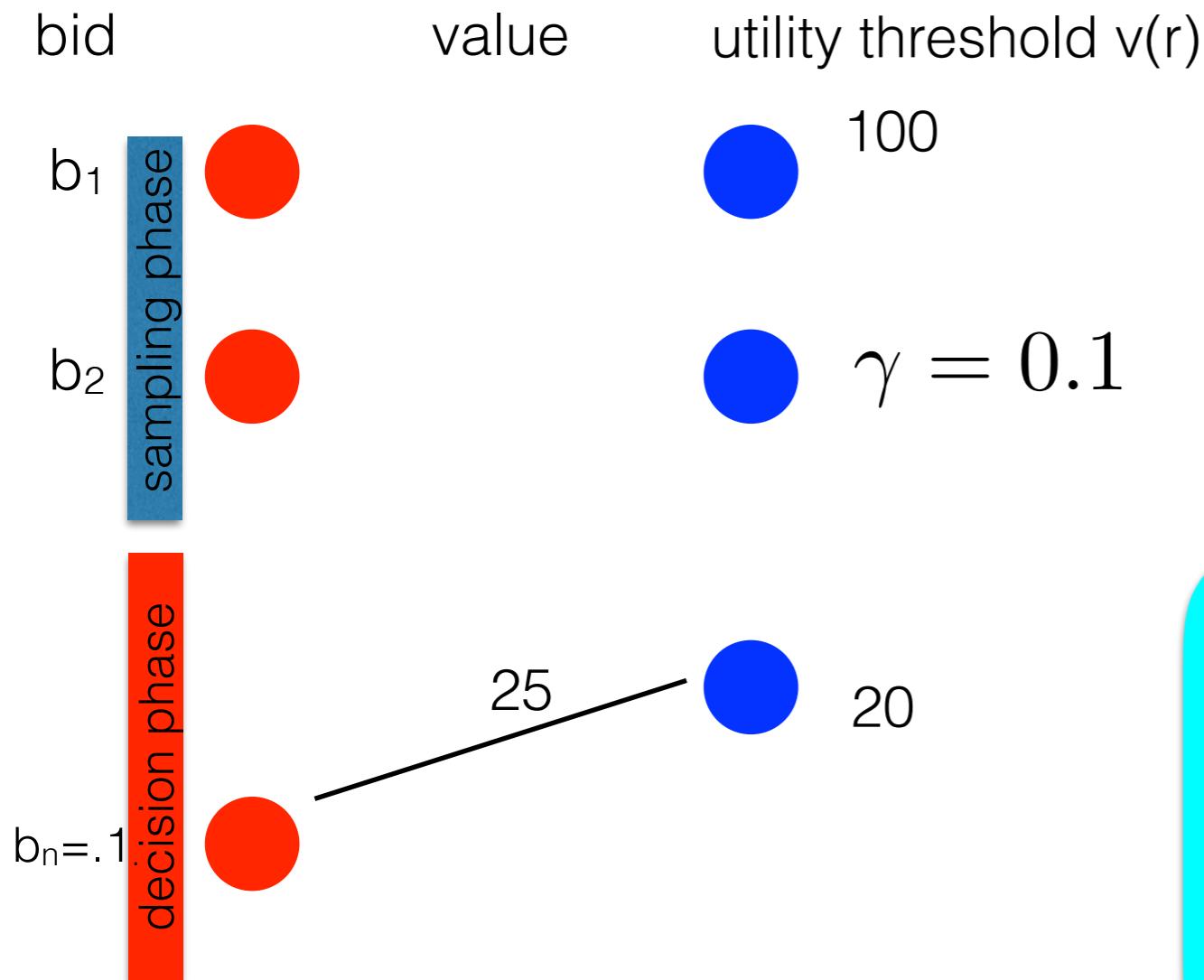
$$\frac{b(e)}{v(e)}$$

**on arrival of a new left vertex**

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

# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



**bid to benefit** ratio of an edge

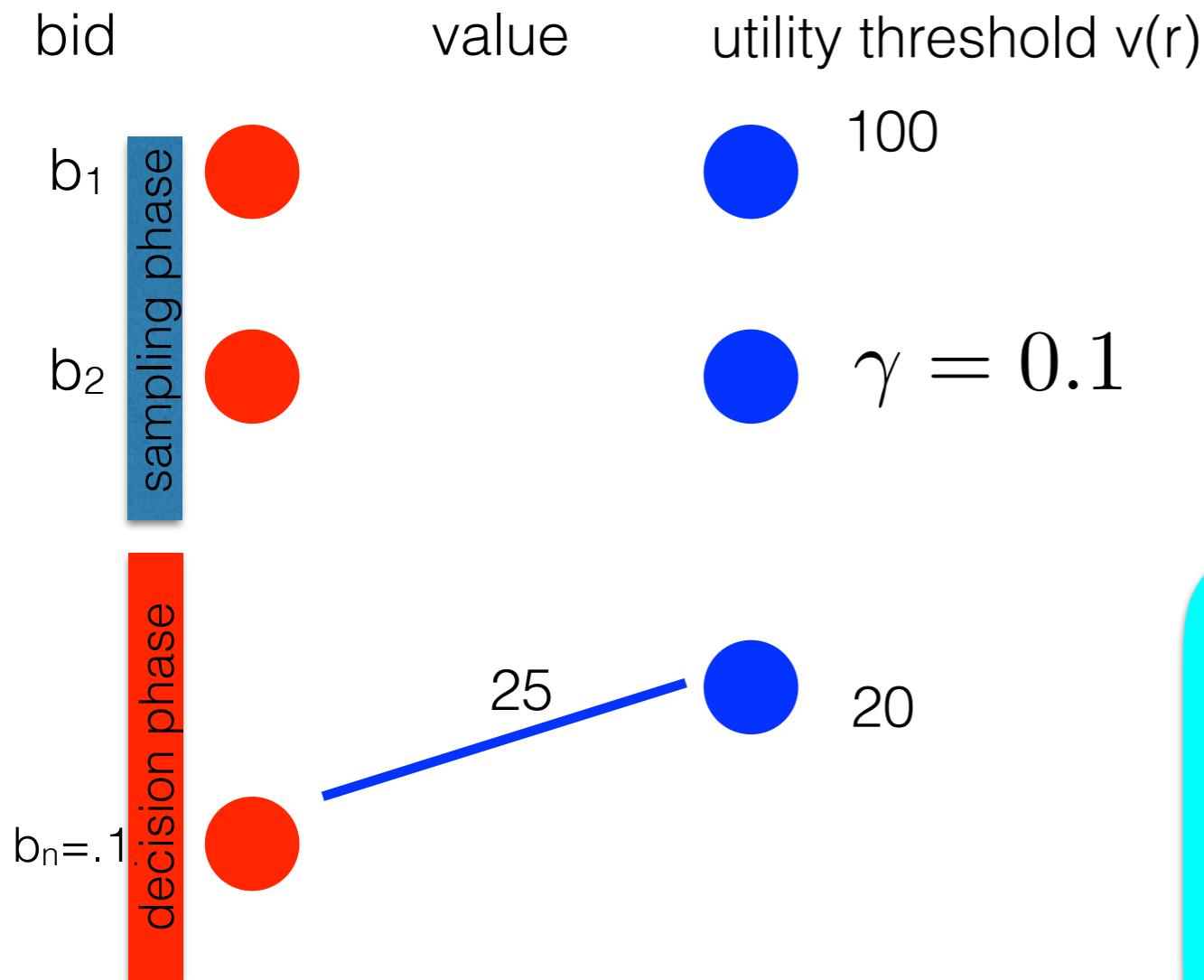
$$\frac{b(e)}{v(e)}$$

**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.

# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



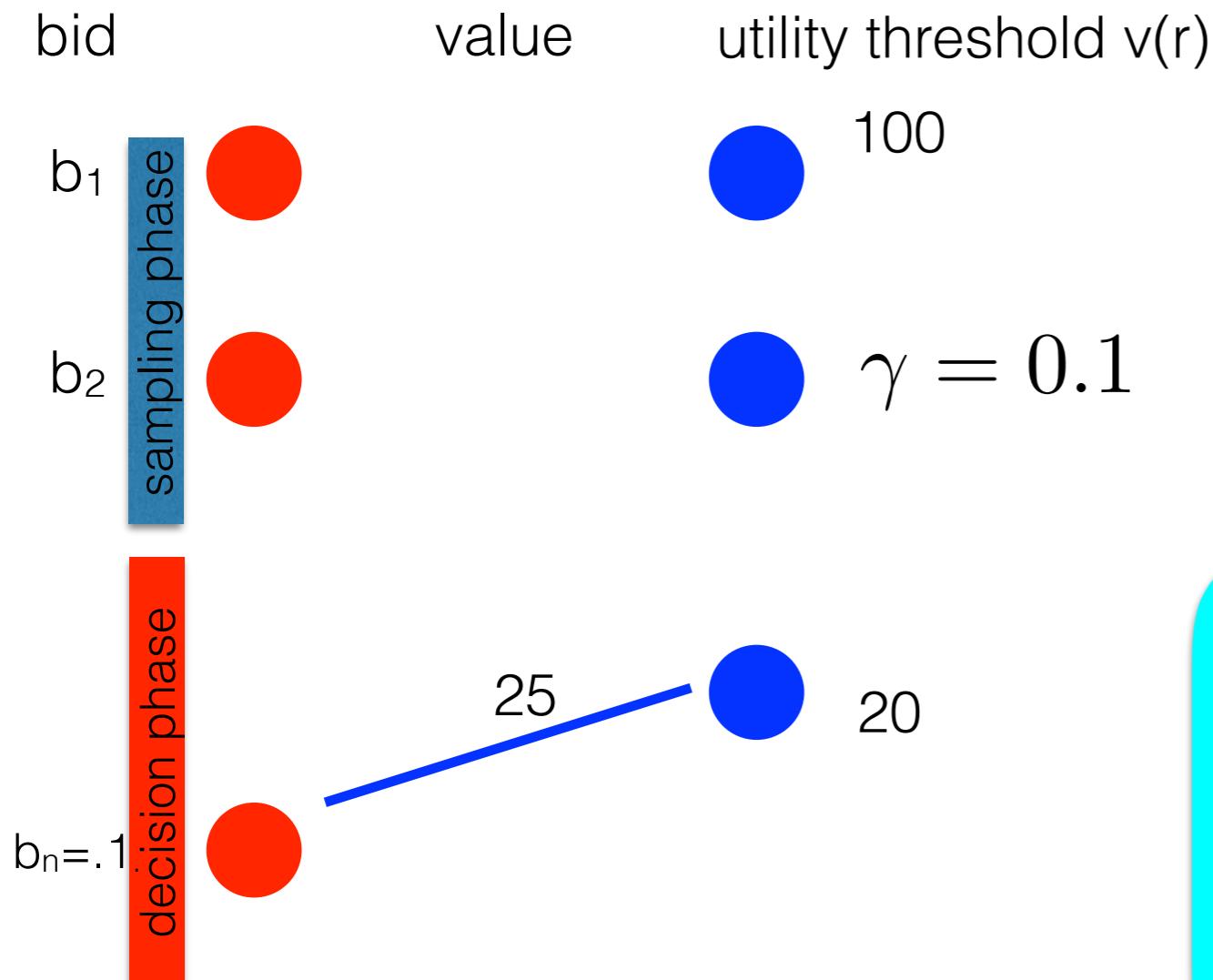
$$\frac{b(e)}{v(e)}$$

**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.

# Algorithm - ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF **with budget**  $B/\beta$



**bid to benefit** ratio of an edge

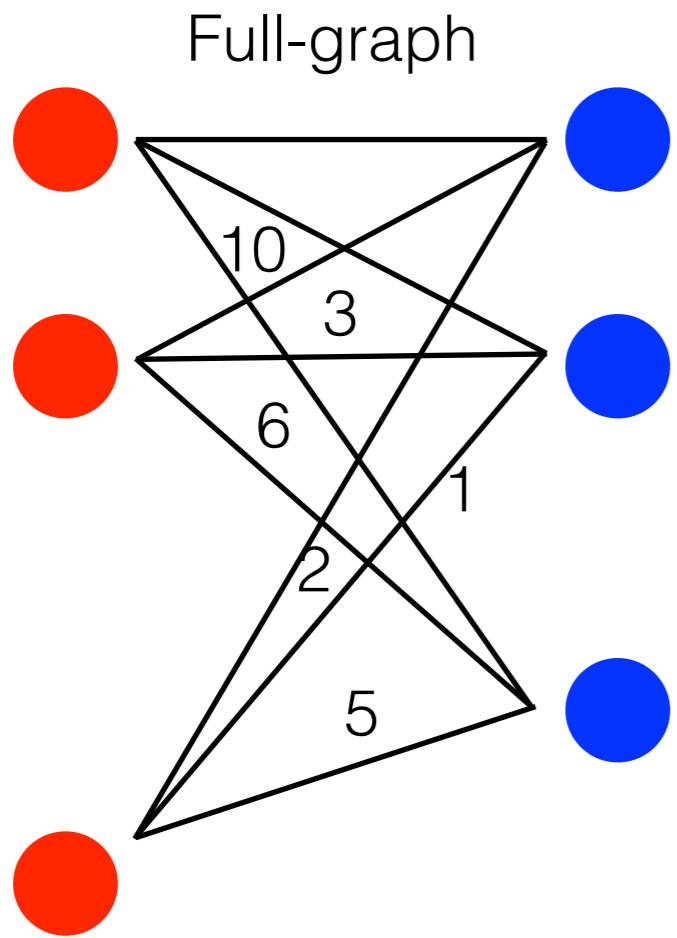
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
 match the right vertex with max  $w >$  utility thres.  
 Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON

- 24 $\beta$  - competitive in expectation
- is truthful
- satisfies the payment budget constraint

# Properties of OFF



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

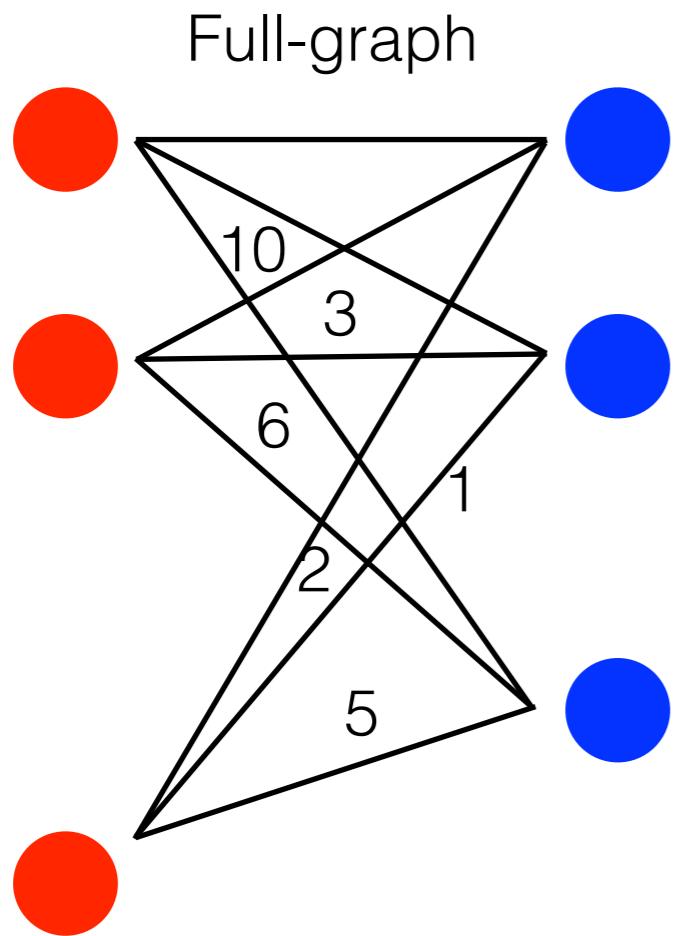
## Algorithm

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

# Properties of OFF



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

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

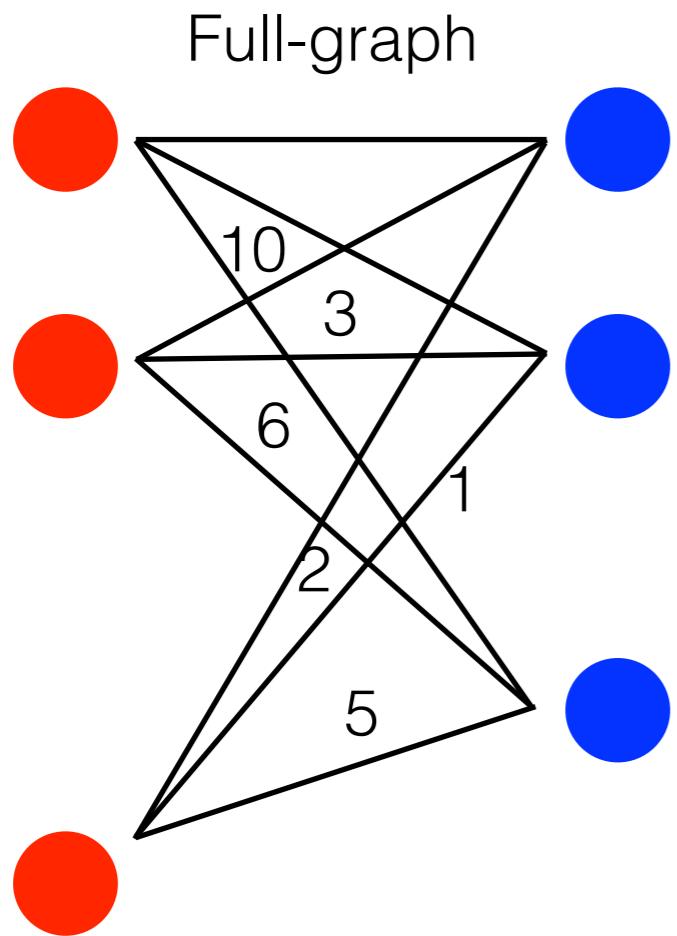
## Algorithm

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

# Properties of OFF



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

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

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

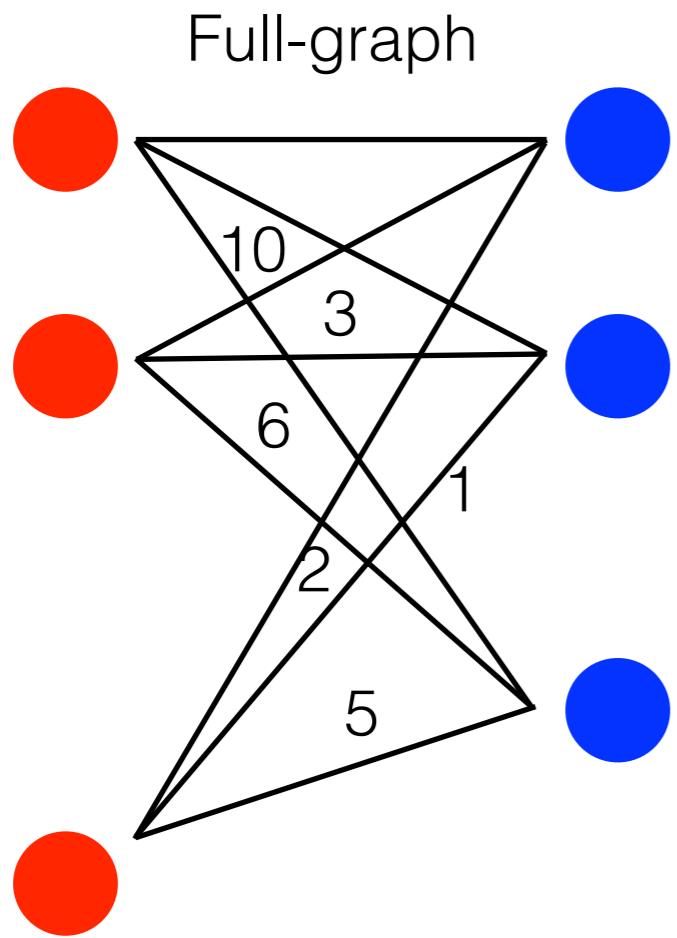
## Algorithm

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

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

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

# 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

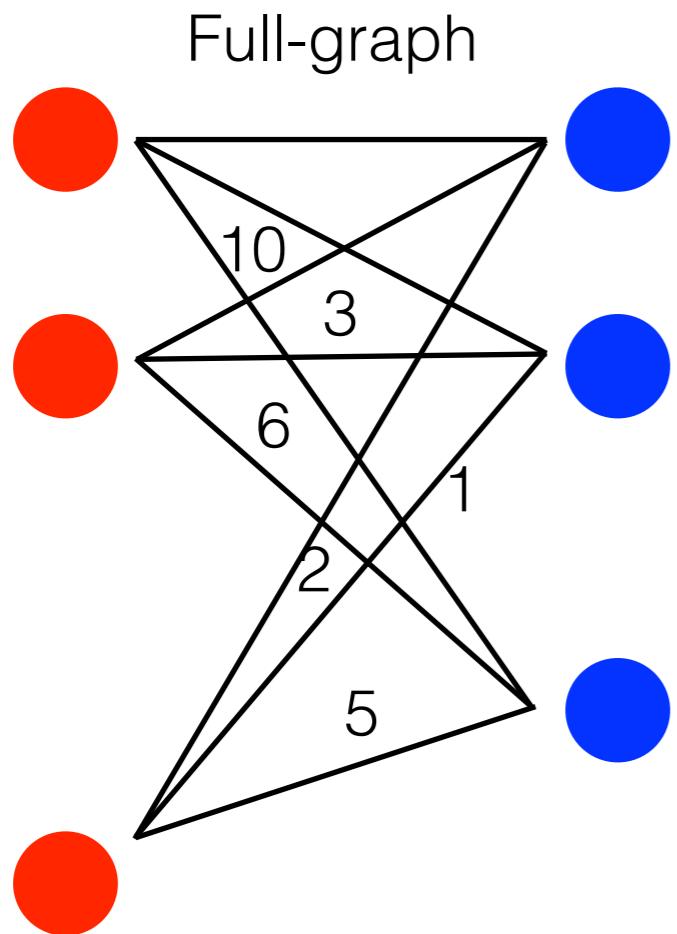
## Algorithm

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

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

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

# 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

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

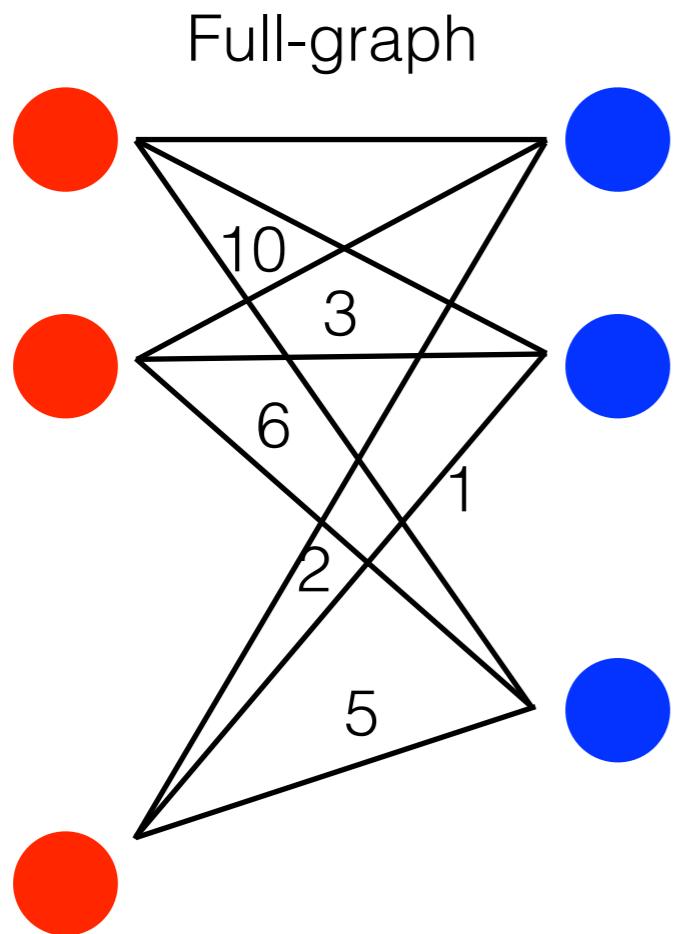
## Algorithm

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

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

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

# 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

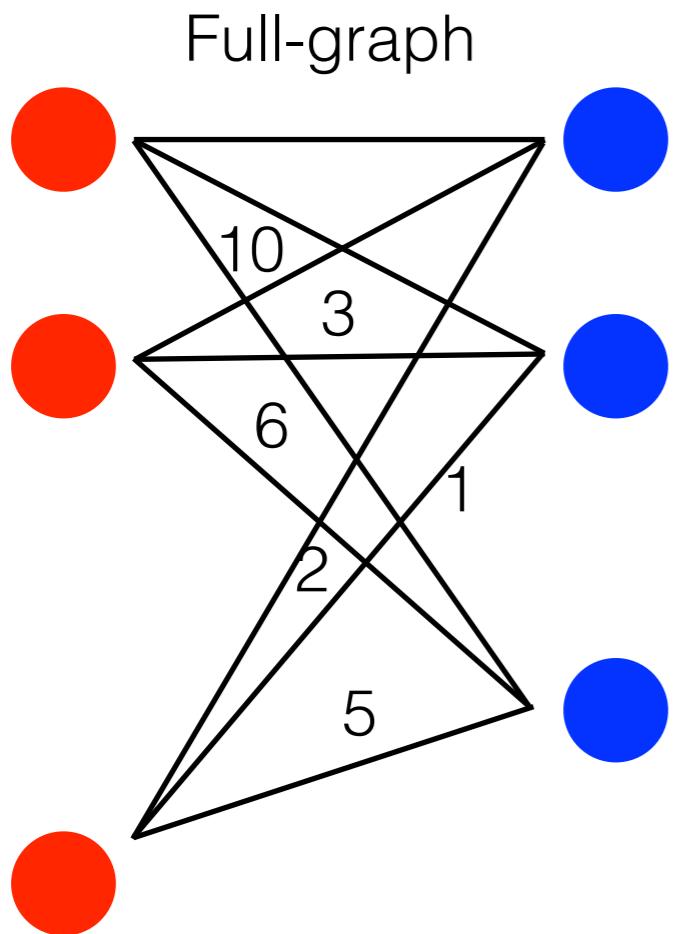
good Graph  $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

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

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

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

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

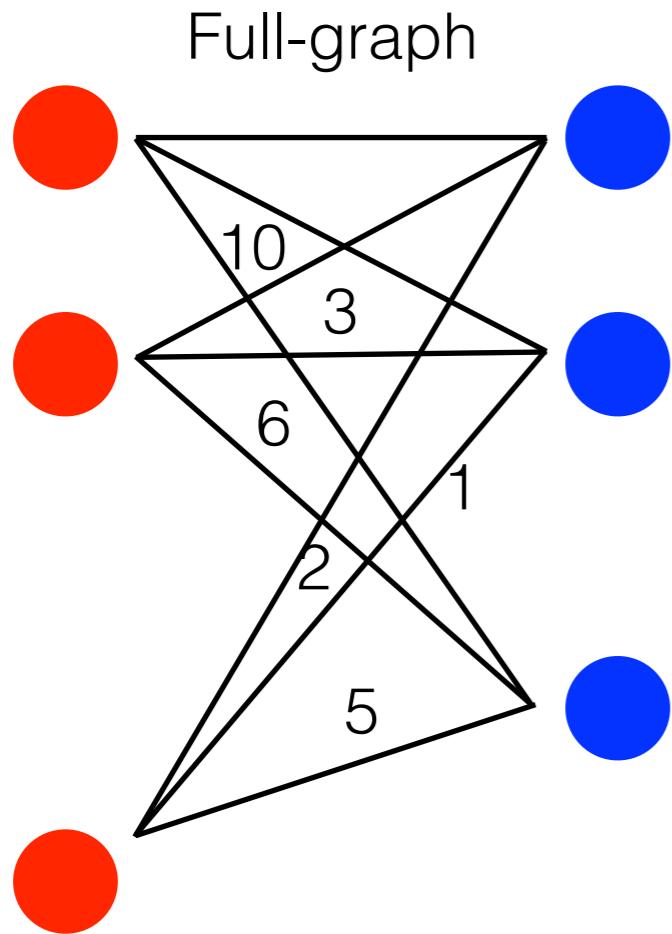
$M(\gamma)$  be greedy matching over  $G(\gamma)$

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

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

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

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

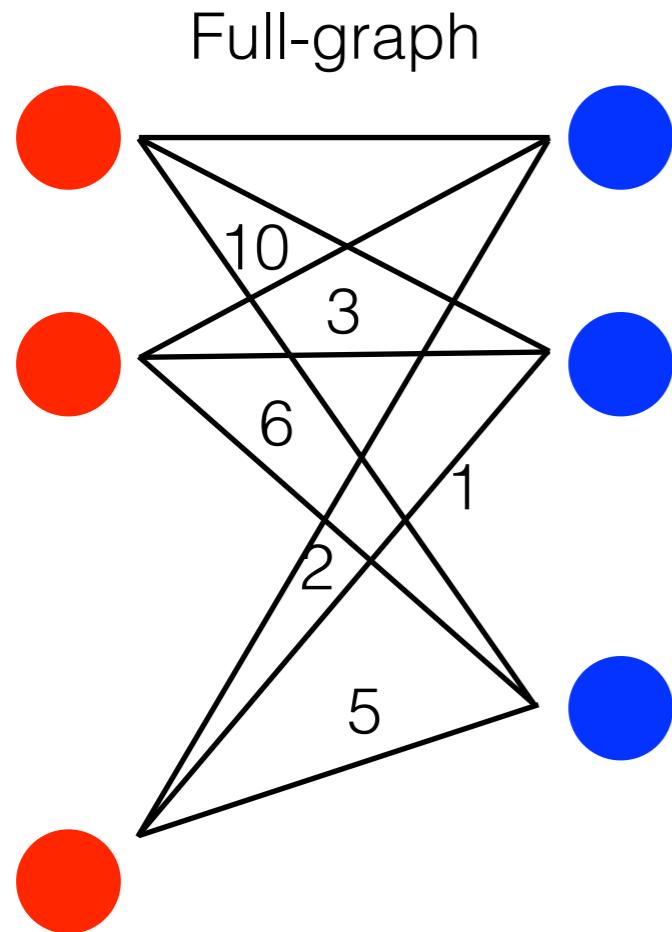
$M(\gamma)$  be greedy matching over  $G(\gamma)$

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

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

$$\begin{aligned} u(\text{OPT}^+) &\leq \sum_{e \in \text{OPT}^+} u(e) \leq \sum_e \frac{c_e}{\gamma} \\ &< \frac{B}{\gamma} \end{aligned}$$

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

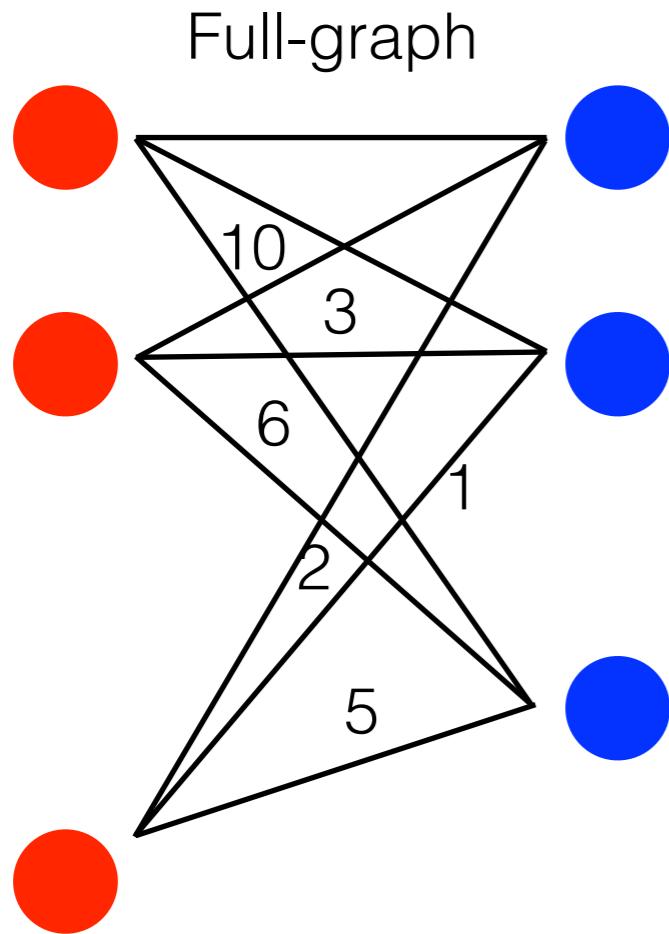
$M(\gamma)$  be greedy matching over  $G(\gamma)$

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

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

$$\begin{aligned} u(\text{OPT}^+) &\leq \sum_{e \in \text{OPT}^+} u(e) \leq \sum_e \frac{c_e}{\gamma} \\ &< \frac{B}{\gamma} \\ &= u(M_{\text{OFF}}(\text{full})) \end{aligned}$$

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

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

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

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

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

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

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

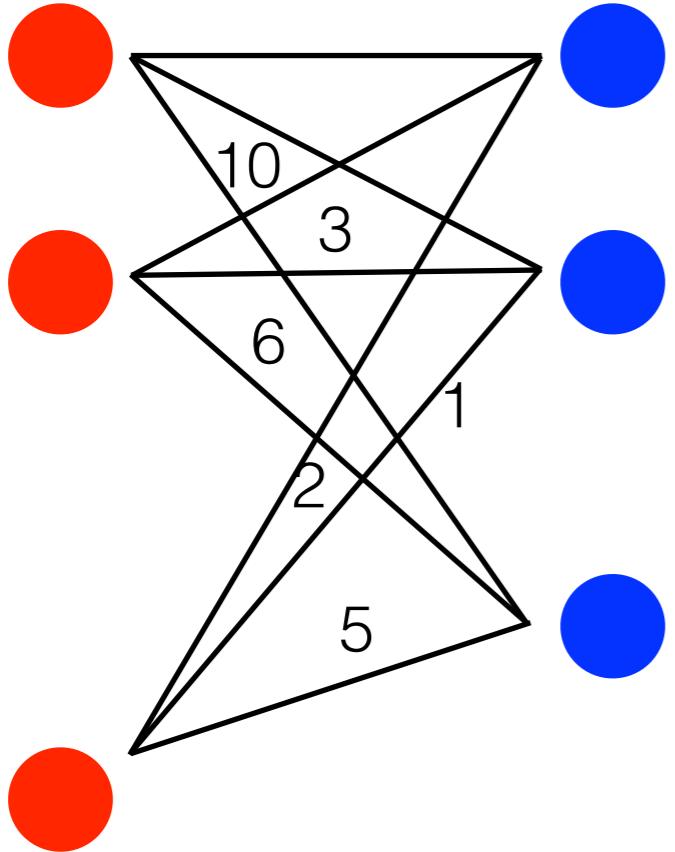
# Properties of OFF

12 - approximate in expectation in offline phase

# Properties of OFF

12 - approximate in expectation in offline phase

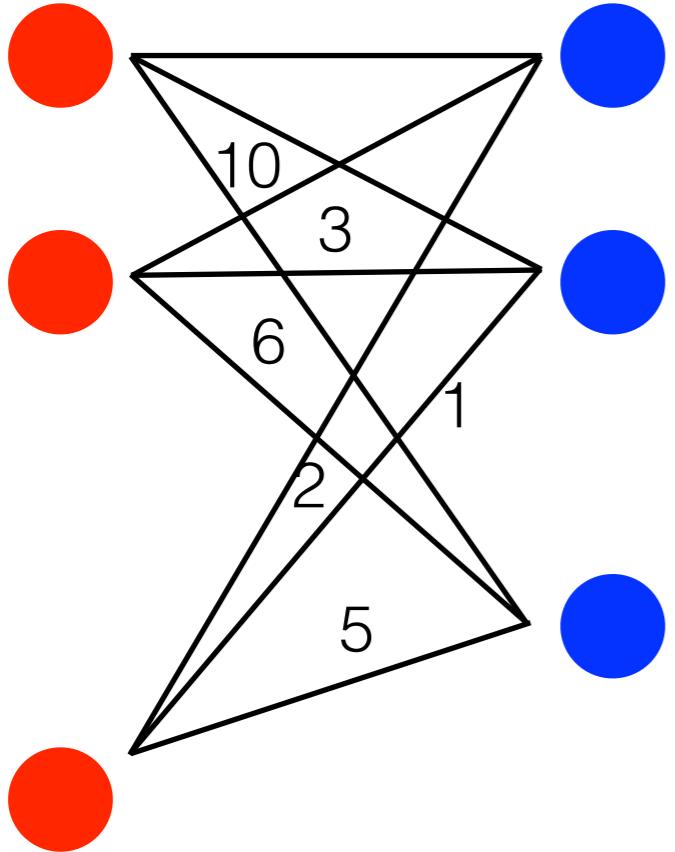
Full-graph



# Properties of OFF

12 - approximate in expectation in offline phase

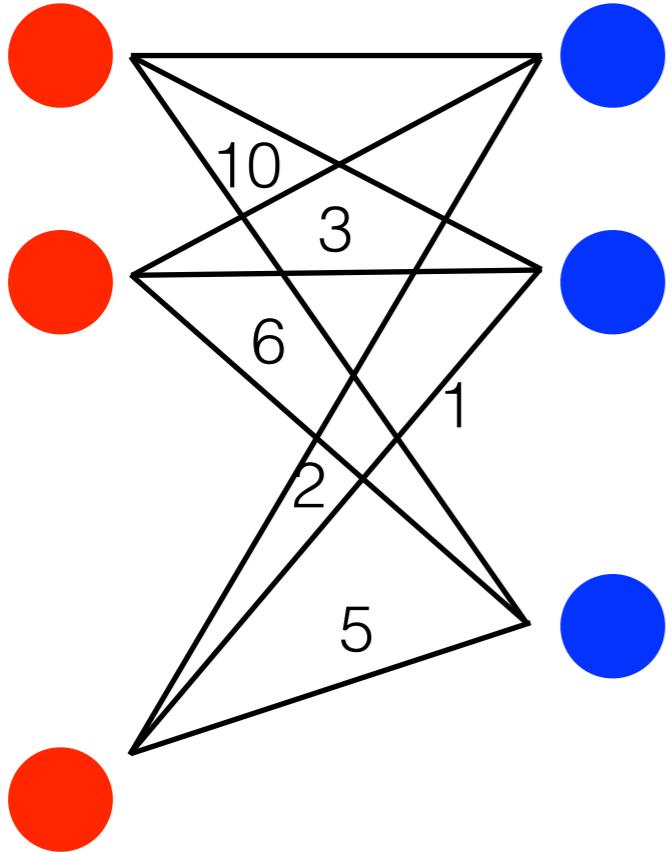
Full-graph



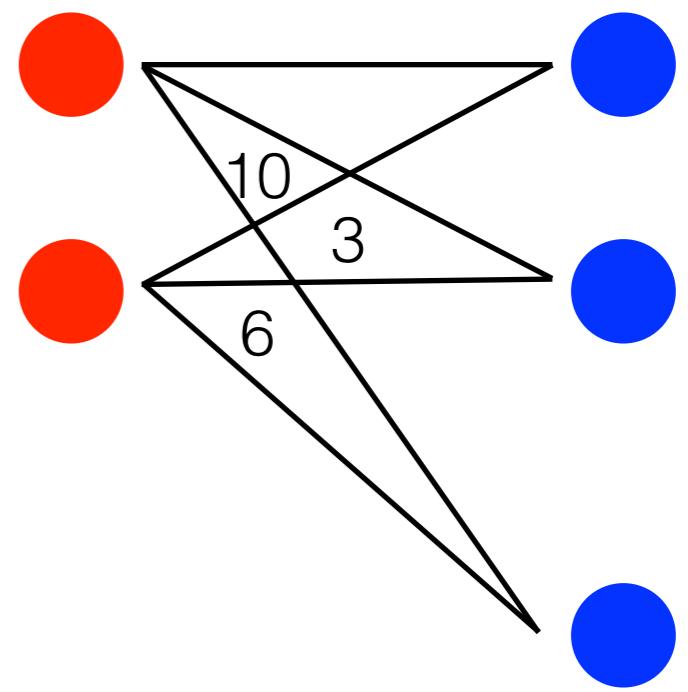
# Properties of OFF

12 - approximate in expectation in offline phase

Full-graph



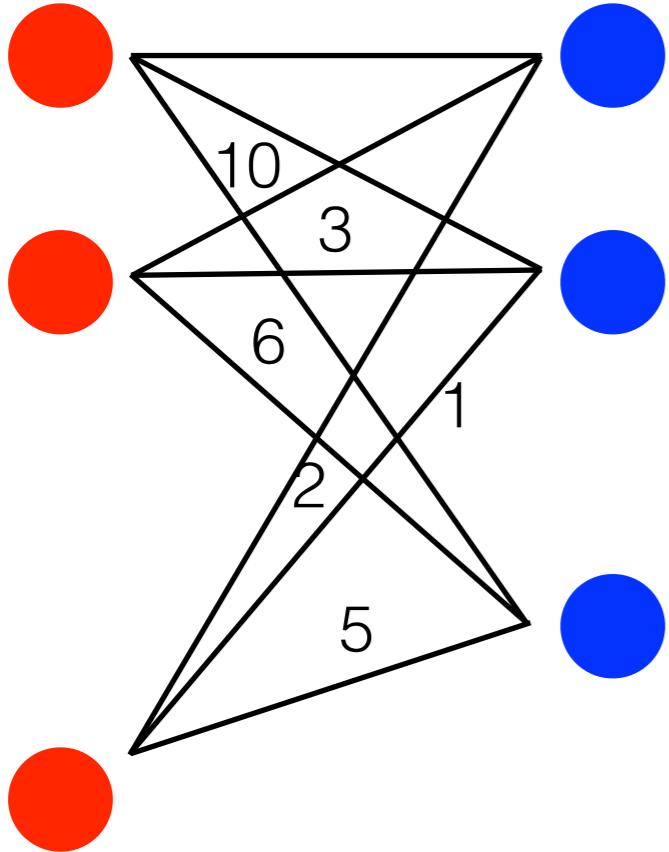
half-graph



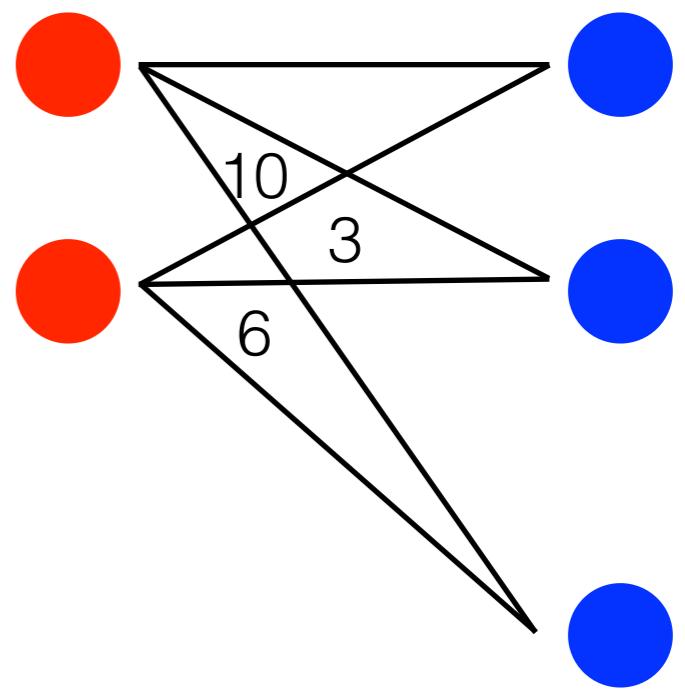
# Properties of OFF

12 - approximate in expectation in offline phase

Full-graph



half-graph



## Algorithm

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

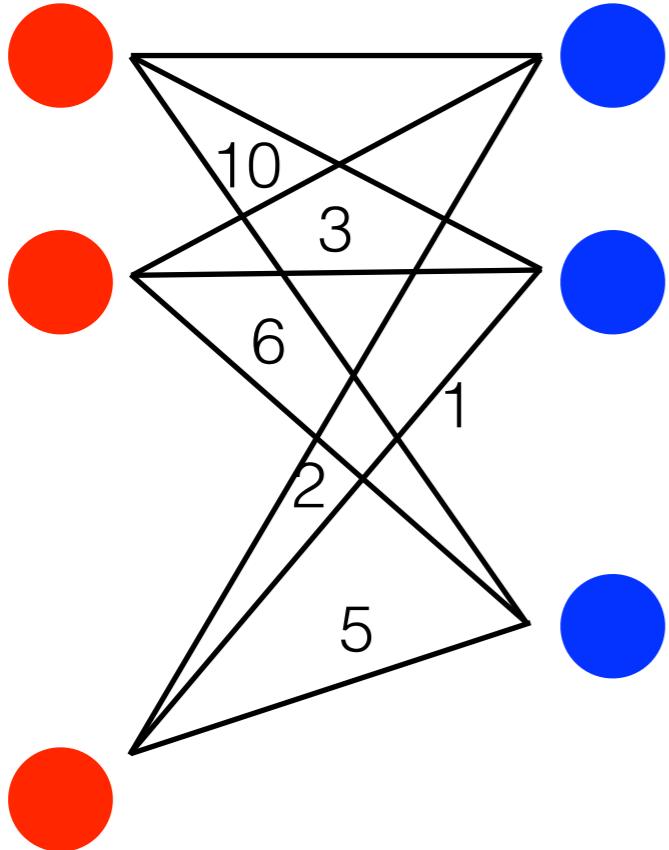
$M(\gamma)$  be greedy matching over  $G(\gamma)$

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

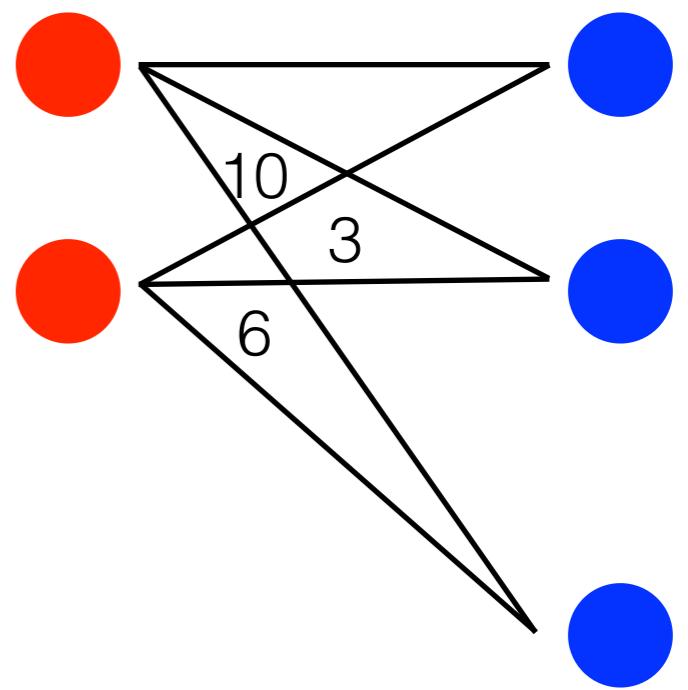
# Properties of OFF

12 - approximate in expectation in offline phase

Full-graph



half-graph



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

## Algorithm

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

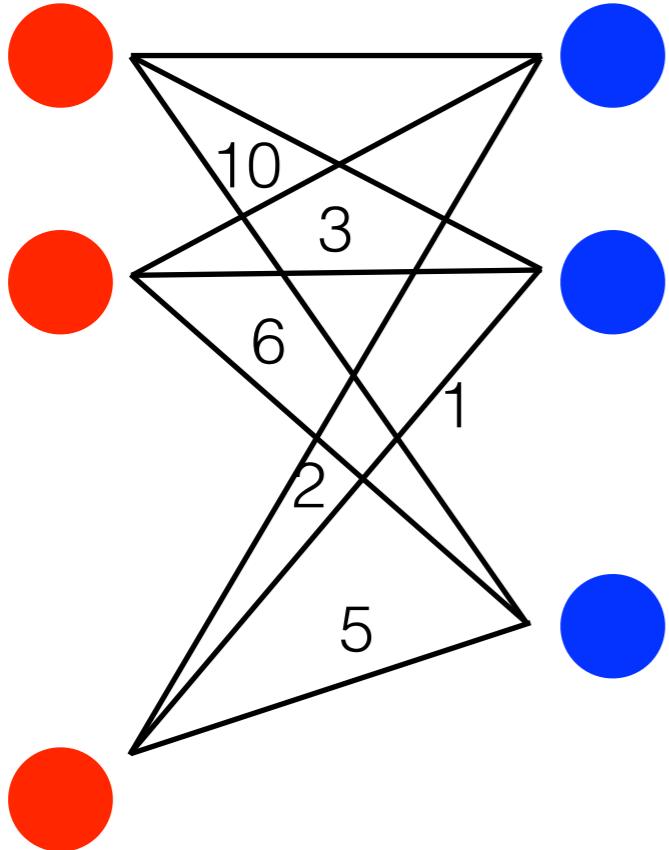
$M(\gamma)$  be greedy matching over  $G(\gamma)$

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

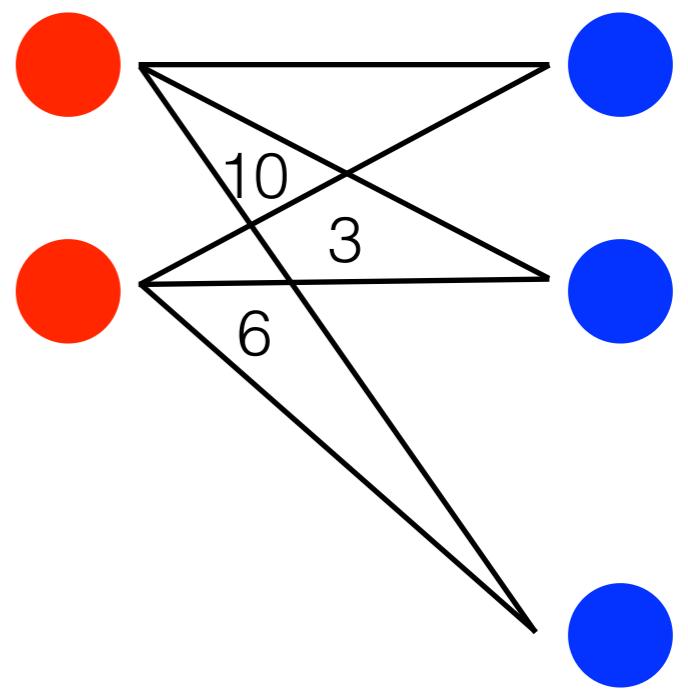
# Properties of OFF

12 - approximate in expectation in offline phase

Full-graph



half-graph



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

## Algorithm

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

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

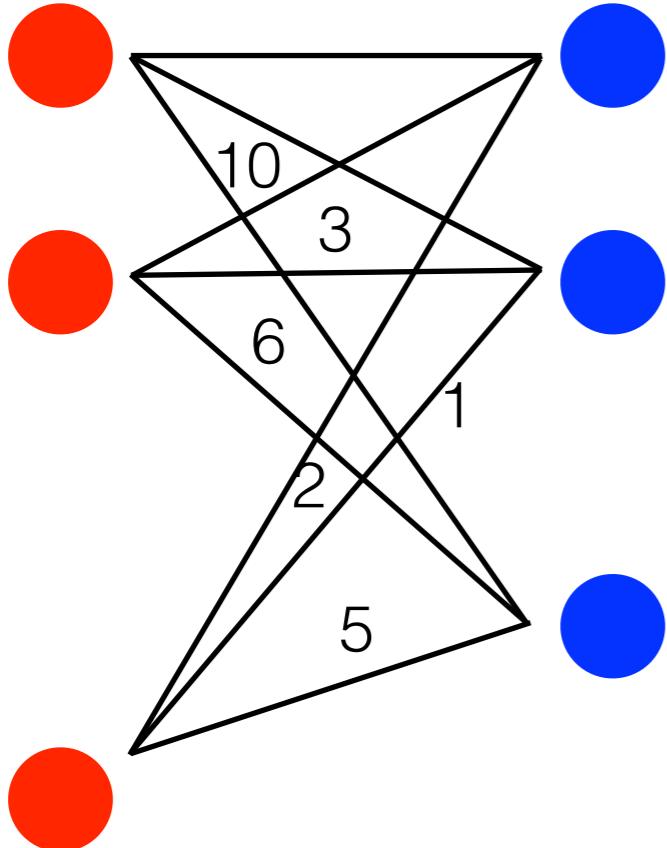
Find largest  $\gamma$   $\sum_{e \in M(\gamma)} v(e) \leq B$

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

# Properties of OFF

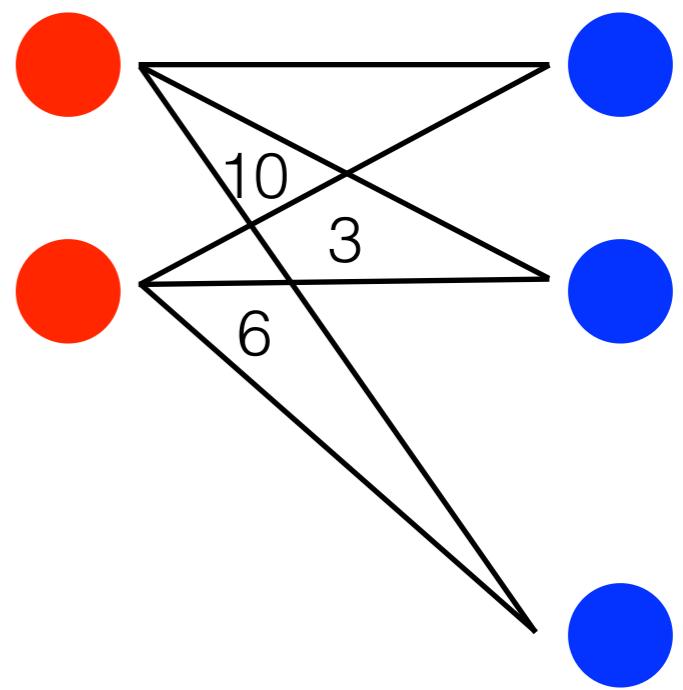
12 - approximate in expectation in offline phase

Full-graph



$$\gamma_f \leq \gamma_h$$

half-graph



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

## Algorithm

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

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

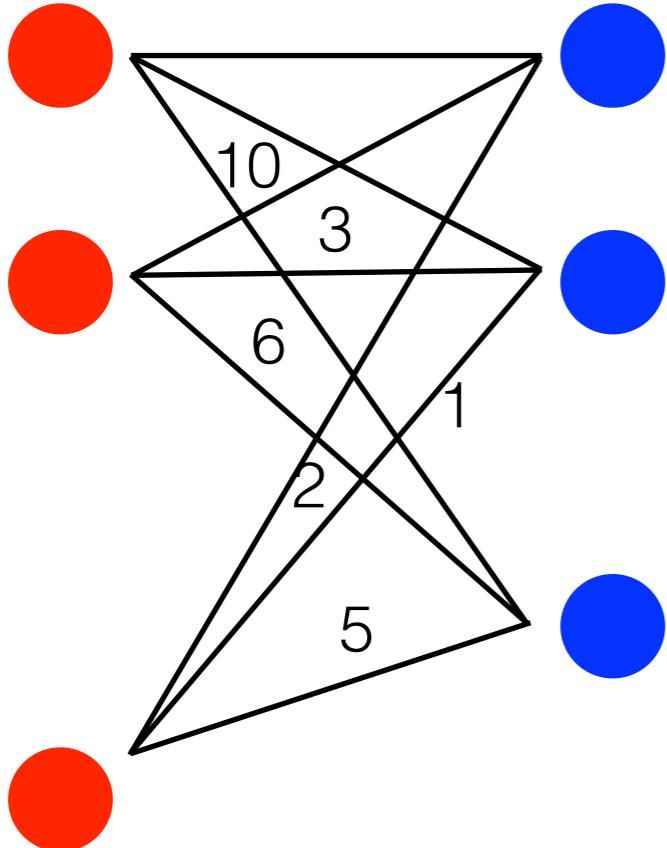
Find largest  $\gamma$   $\sum_{e \in M(\gamma)} v(e) \leq B$

$$E \{ u(M_{\text{OFF}}(\text{full}, \text{half})) \} \geq \frac{u(M_{\text{OFF}}(\text{full}))}{2}$$

# Properties of OFF

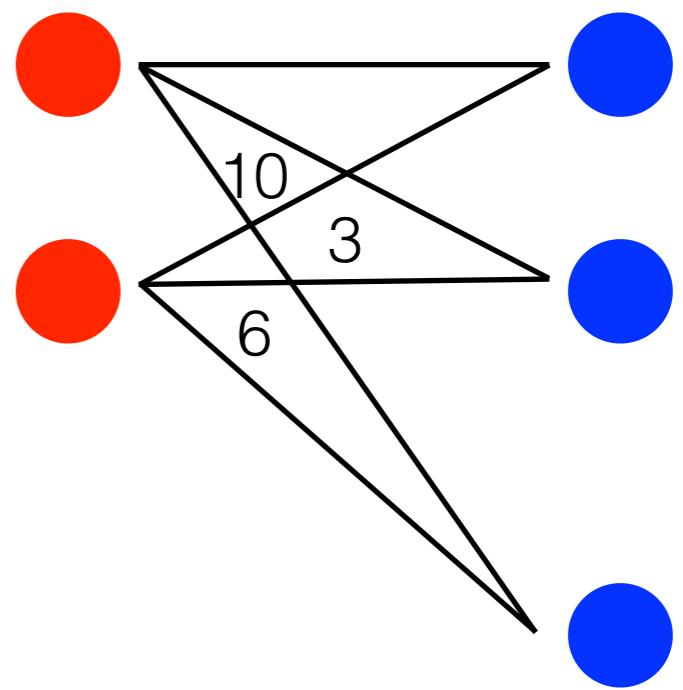
12 - approximate in expectation in offline phase

Full-graph



$$\gamma_f \leq \gamma_h$$

half-graph



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

## Algorithm

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

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

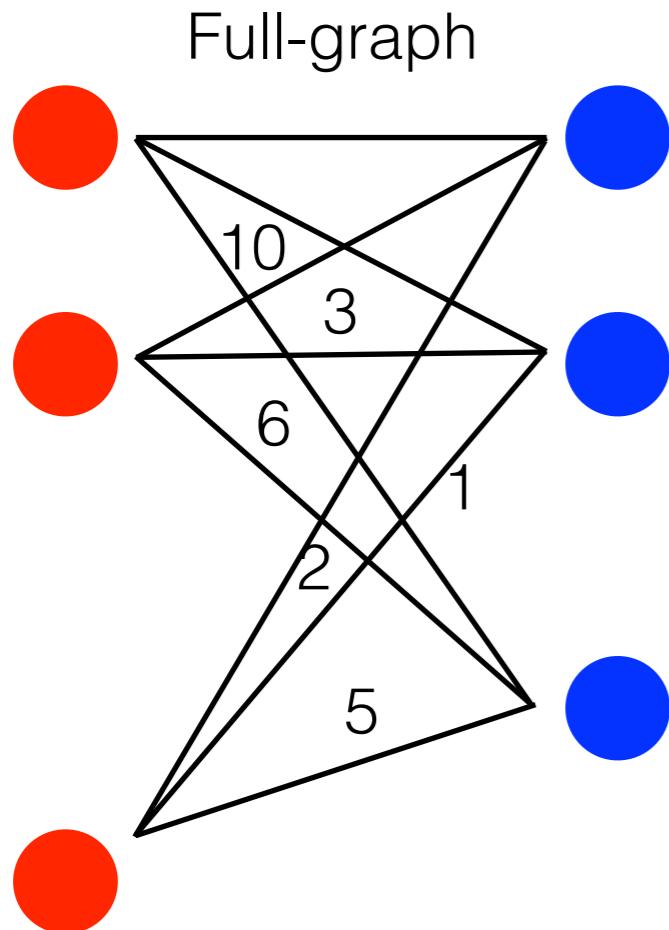
Find largest  $\gamma$   $\sum_{e \in M(\gamma)} v(e) \leq B$

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

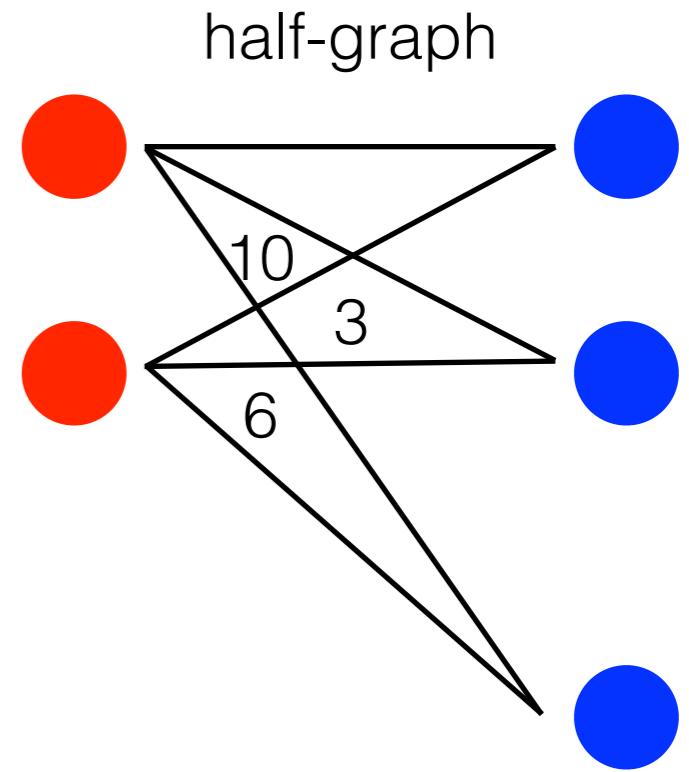
$$u(M_{\text{OFF}}(\text{half})) \geq \frac{u(\text{OPT}_{\text{half}})}{2}$$

# Properties of OFF

12 - approximate in expectation in offline phase



$$\gamma_f \leq \gamma_h$$



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

## Algorithm

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

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

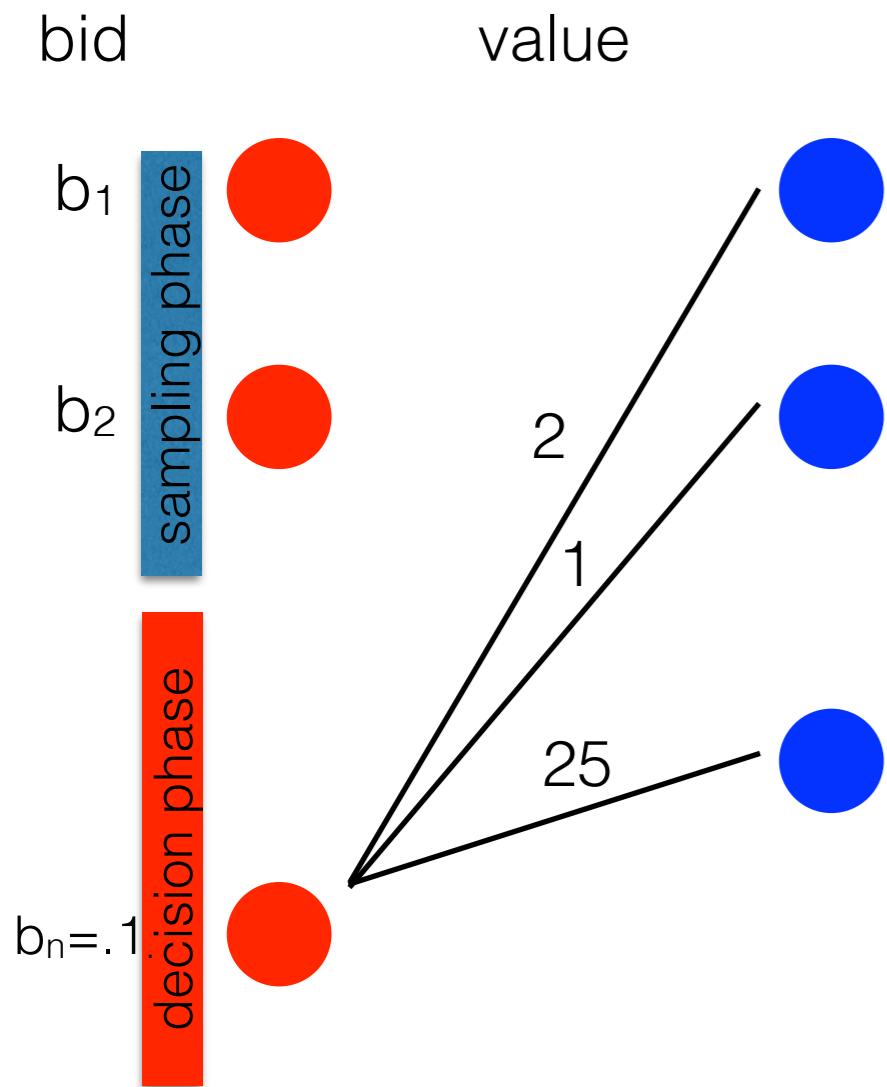
Find largest  $\gamma$   $\sum_{e \in M(\gamma)} v(e) \leq B$

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

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

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

# Properties of ON



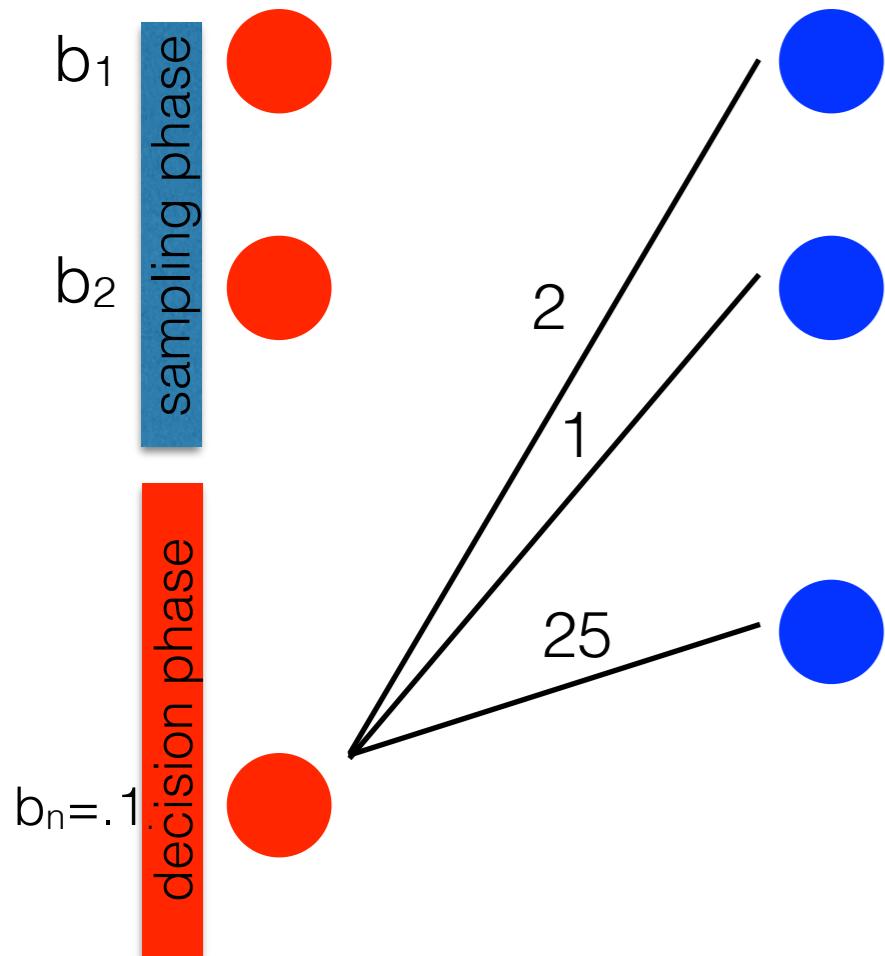
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max w > utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

bid value

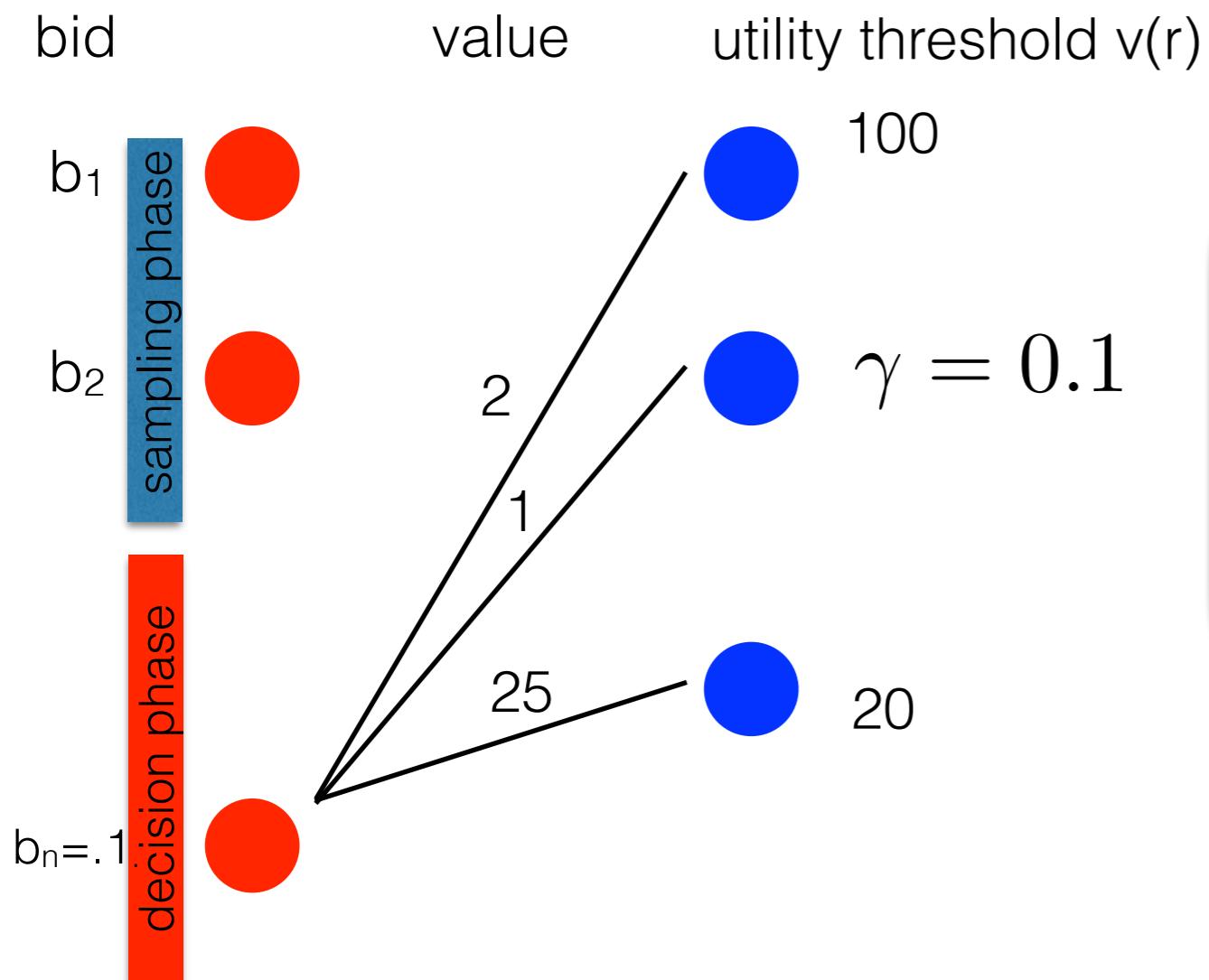


**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

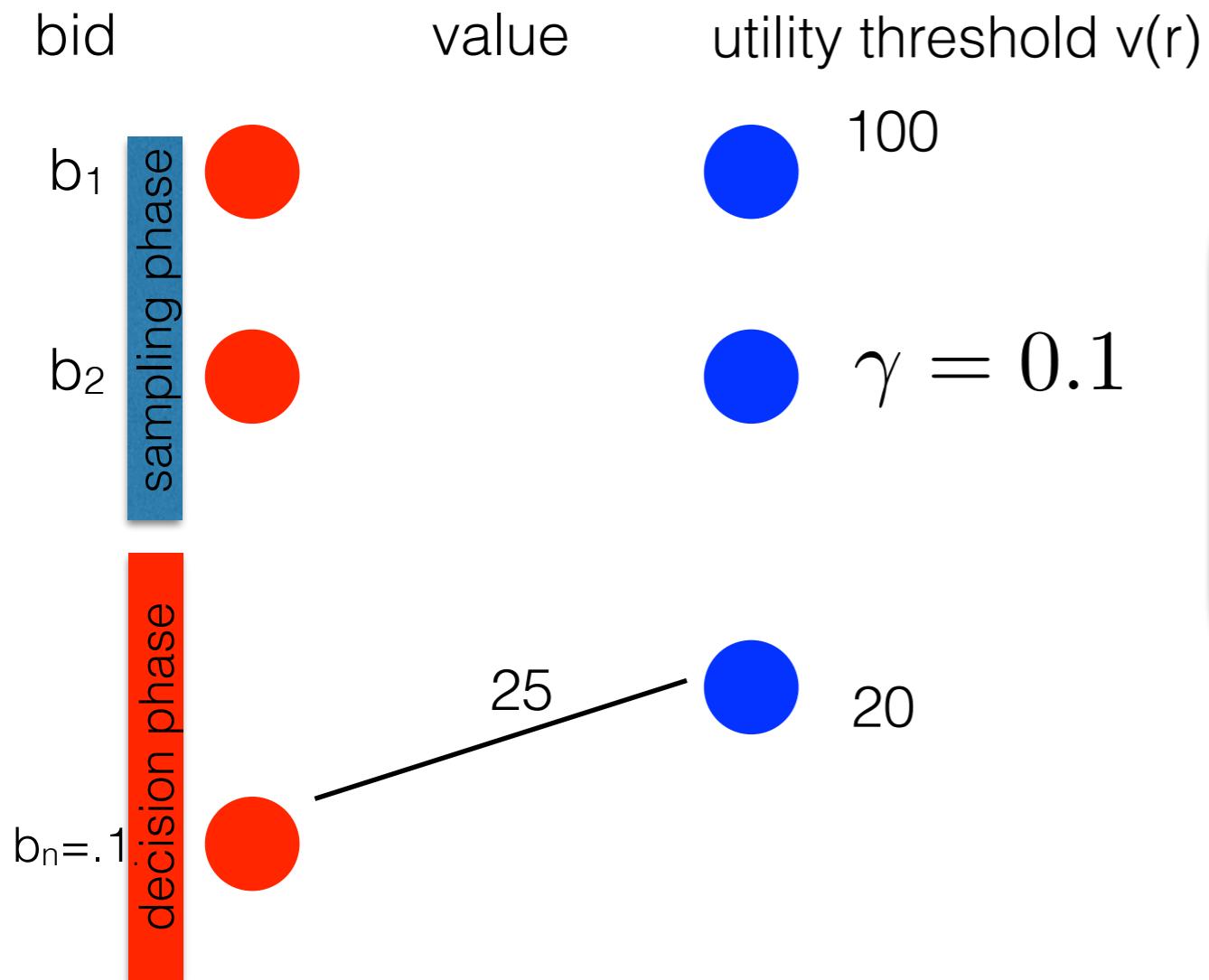


**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max w  $>$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

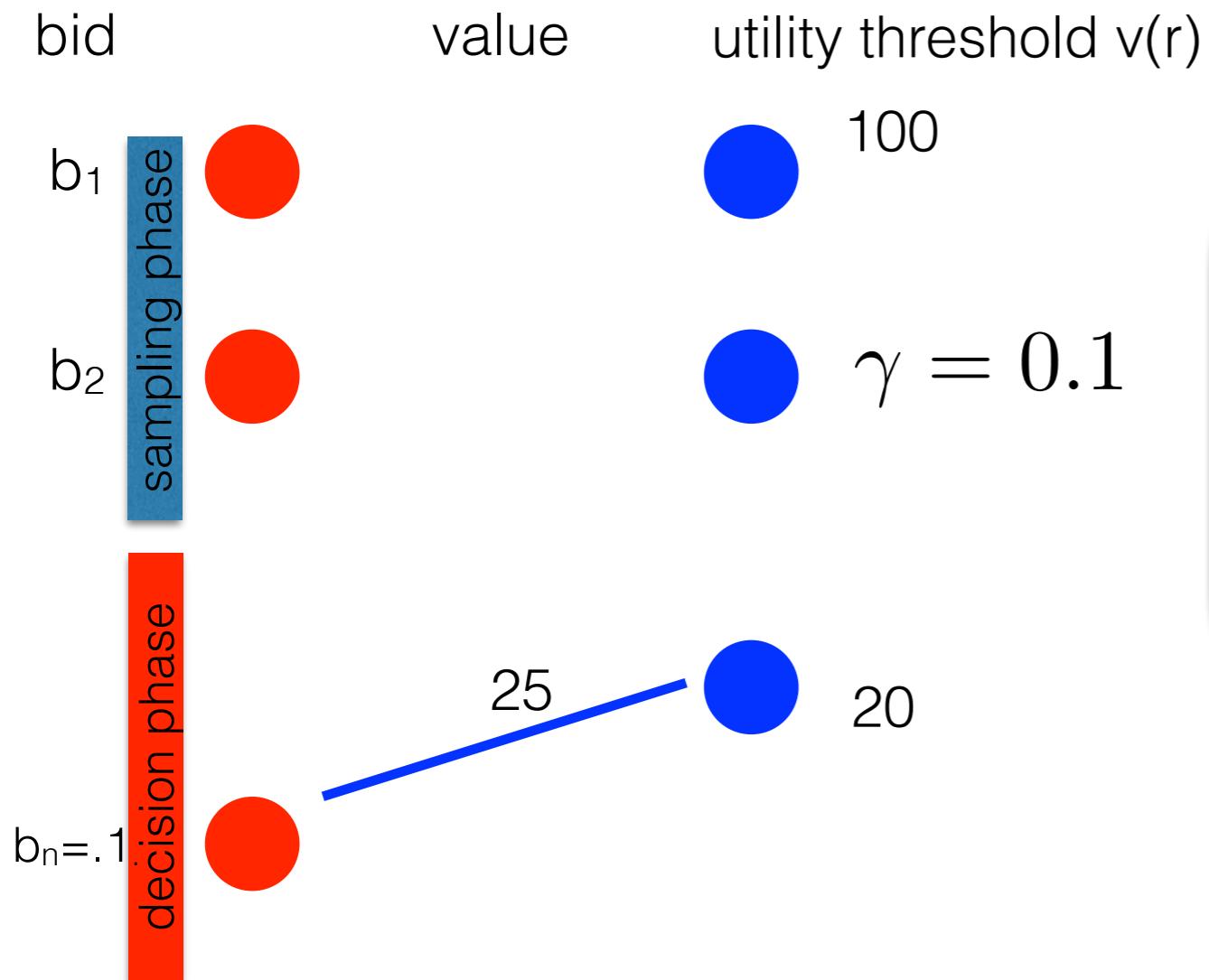


**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

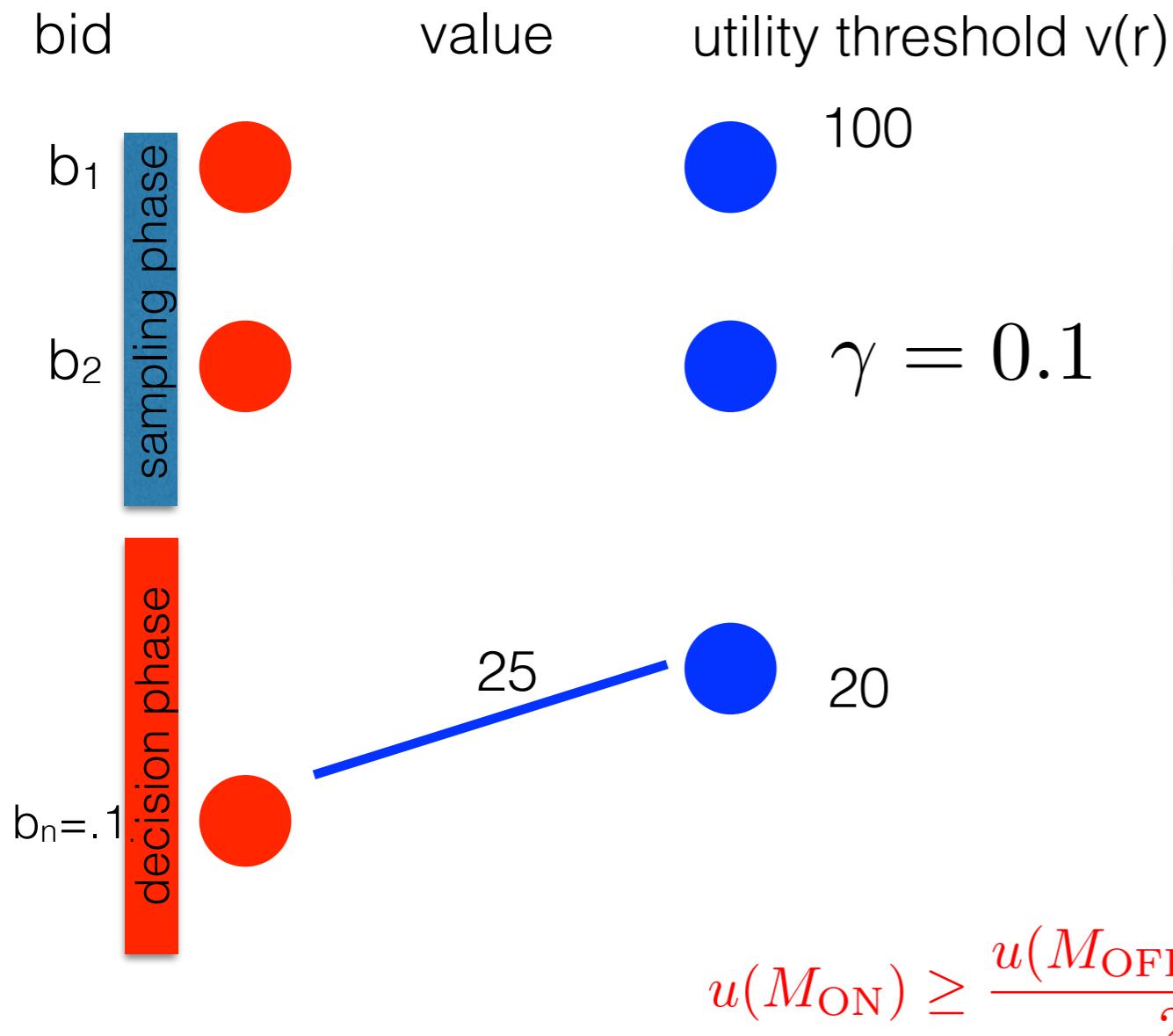


**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF



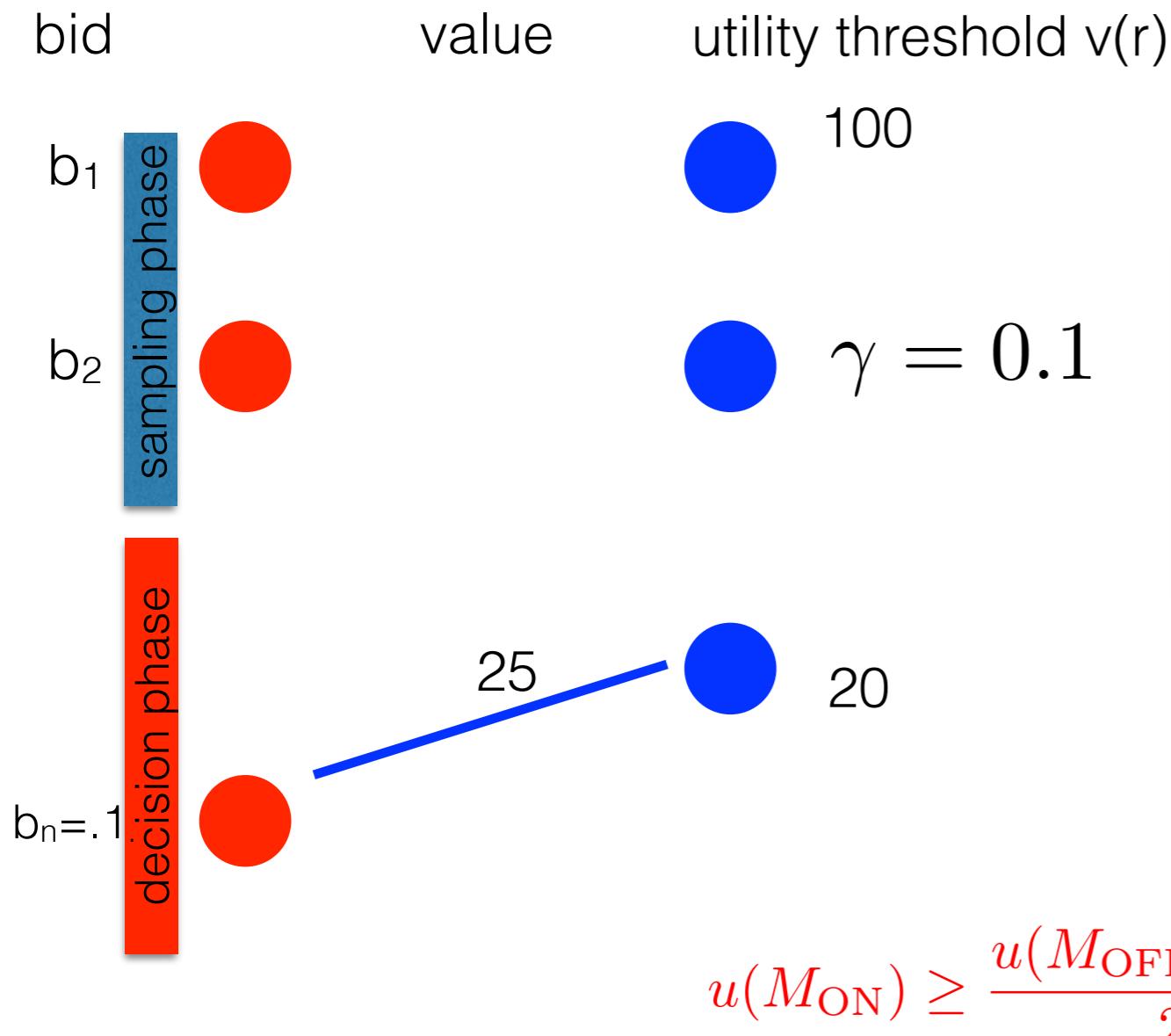
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

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

# Properties of ON

$\gamma$  and threshold utility  $v(r)$  obtained from OFF



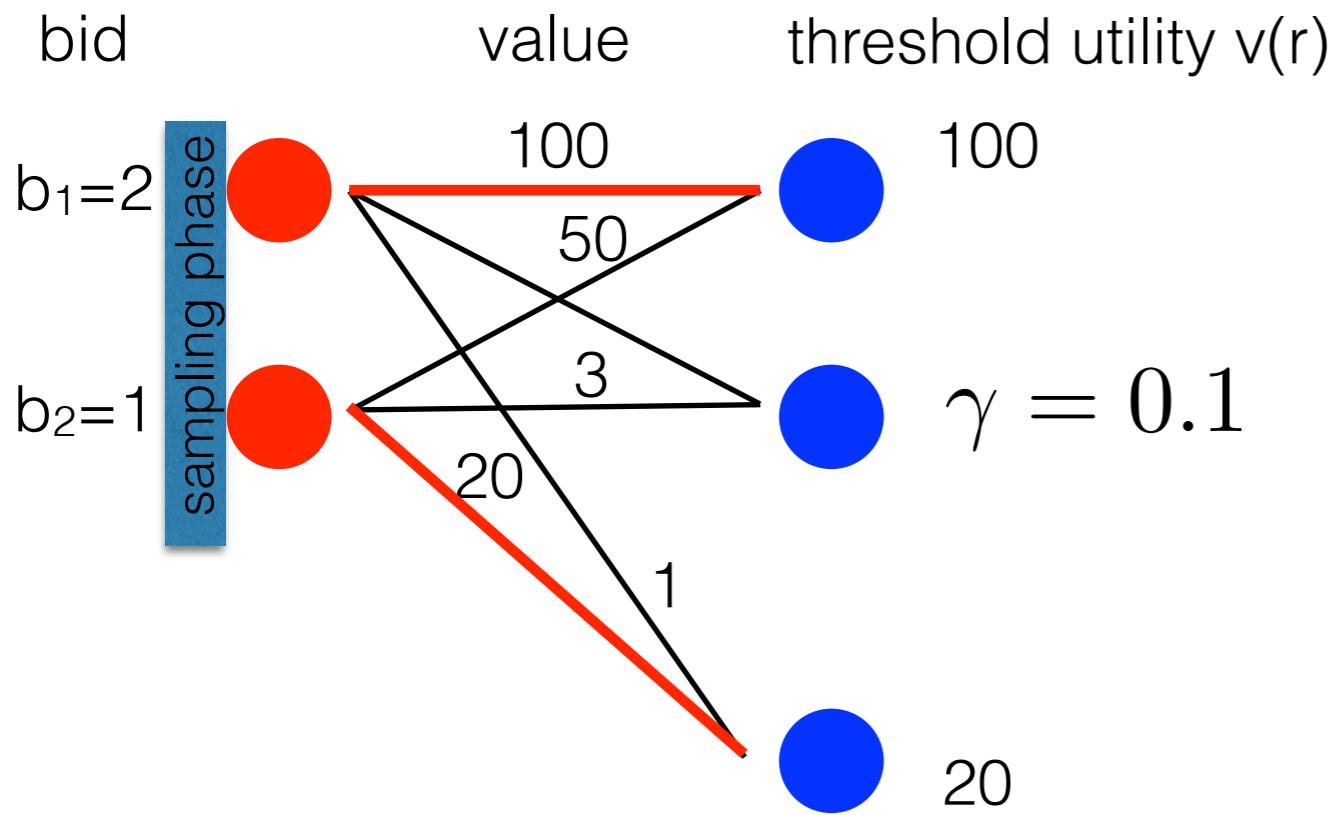
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
 match the right vertex with max  $w >$  utility thres.  
 Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

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

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

# Properties of ON



## Algorithm - OFF

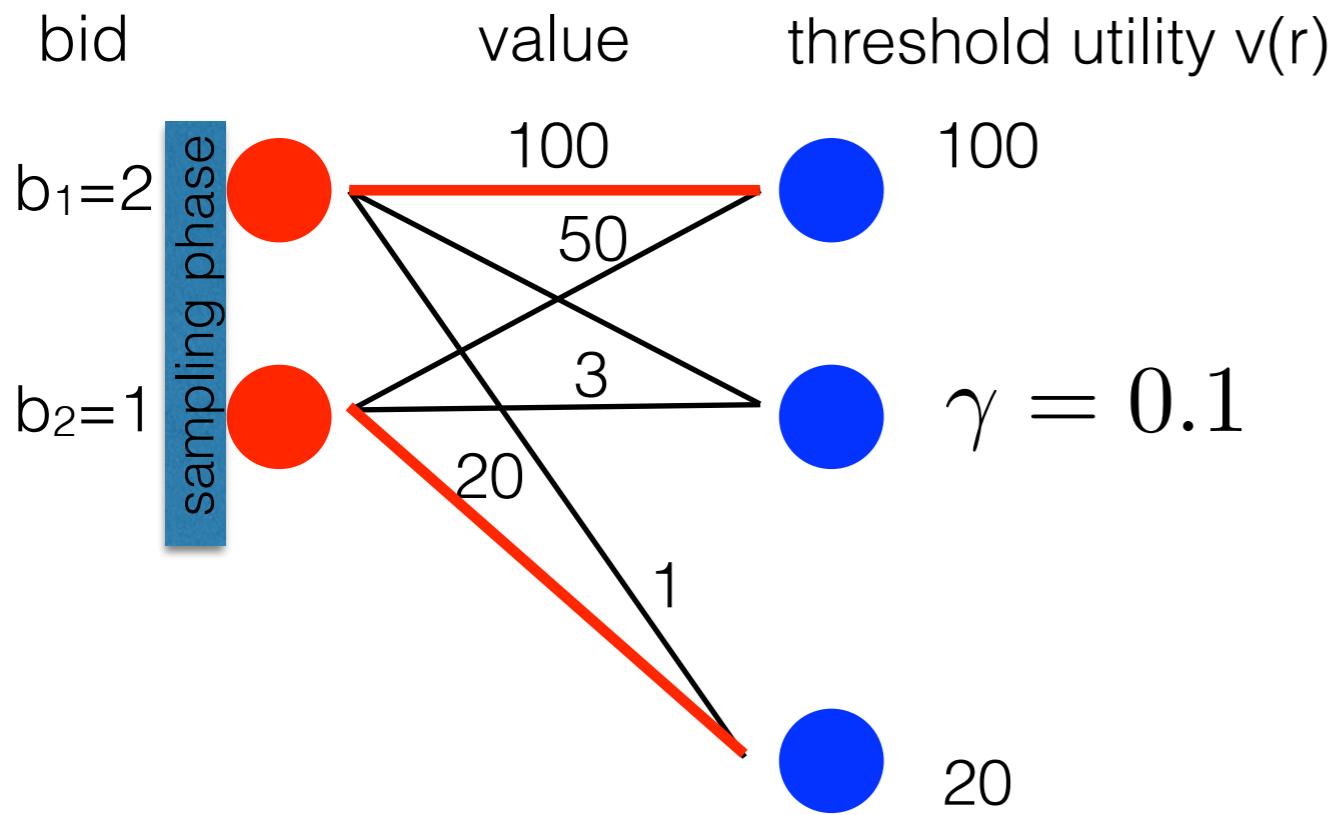
good Graph  $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

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

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

# Properties of ON

satisfies the payment budget constraint



## Algorithm - OFF

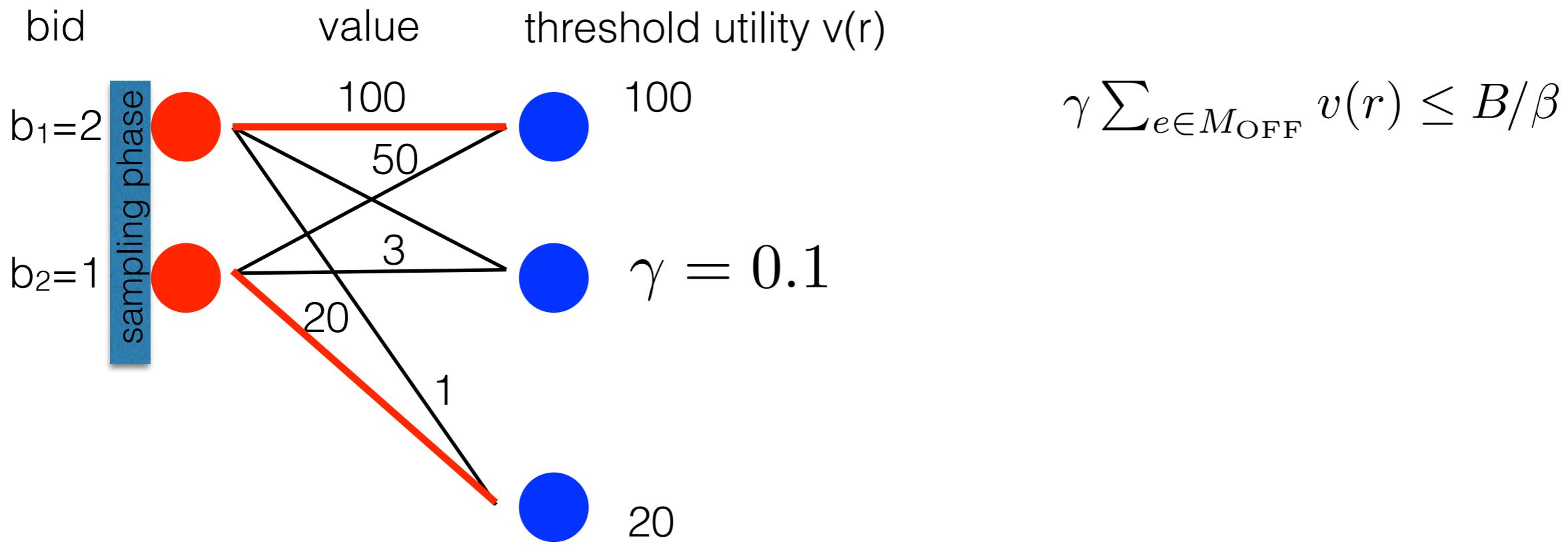
good Graph  $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

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

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

# Properties of ON

satisfies the payment budget constraint



## Algorithm - OFF

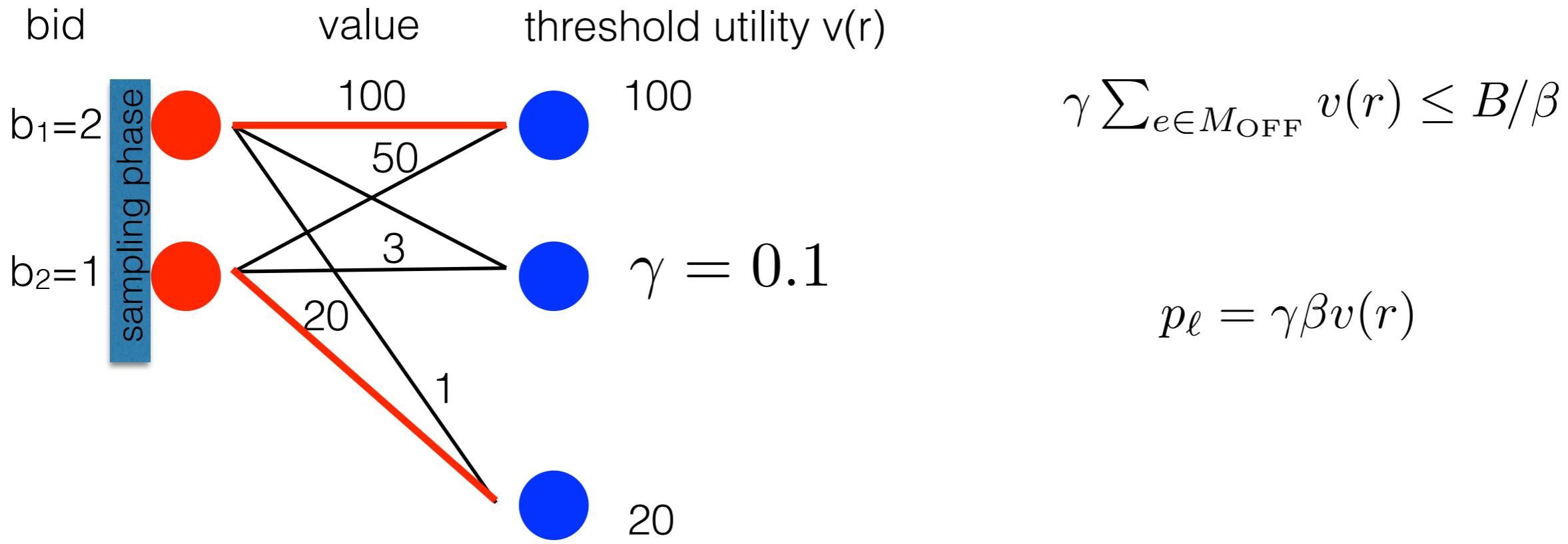
good Graph  $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

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

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

# Properties of ON

satisfies the payment budget constraint



## Algorithm - OFF

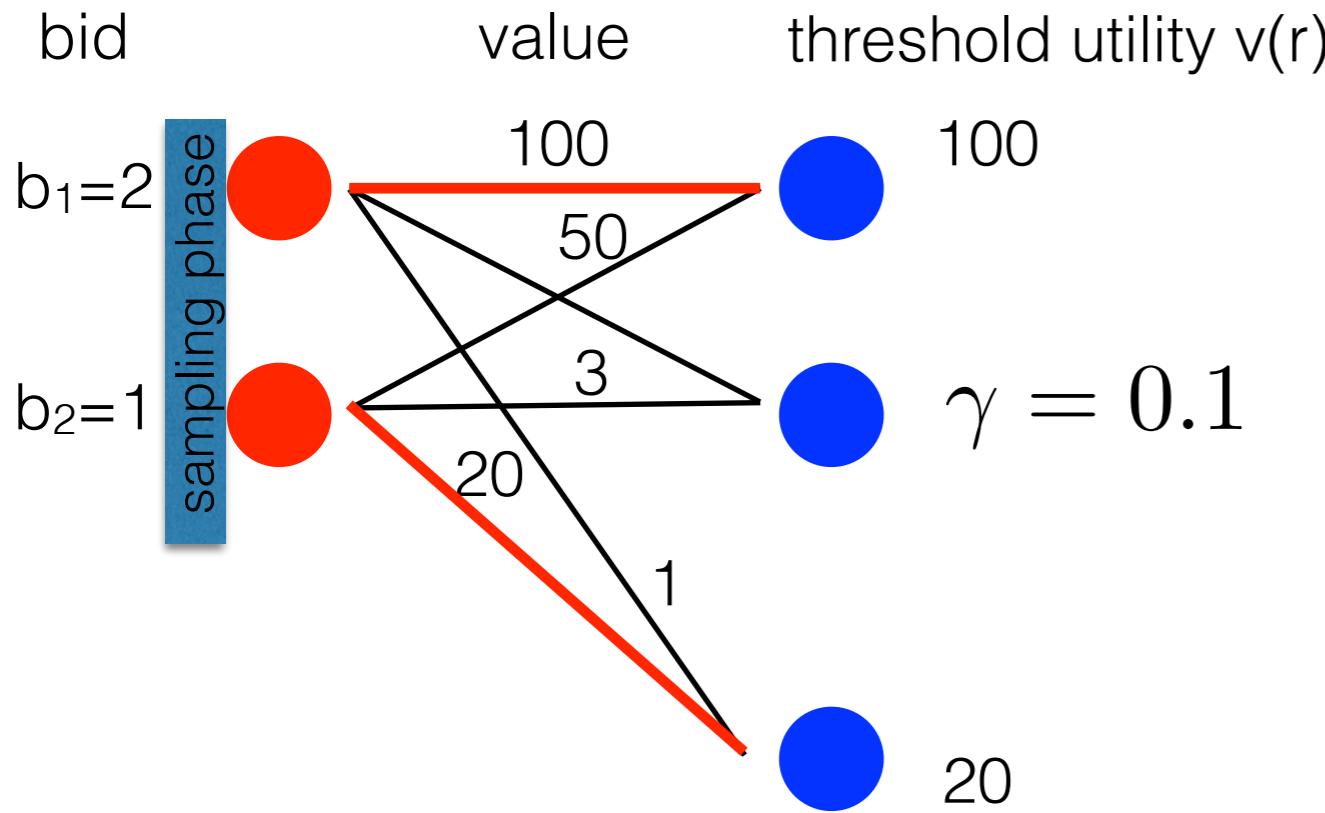
good Graph  $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

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

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

# Properties of ON

satisfies the payment budget constraint



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

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

Since we are looking for matching

## Algorithm - OFF

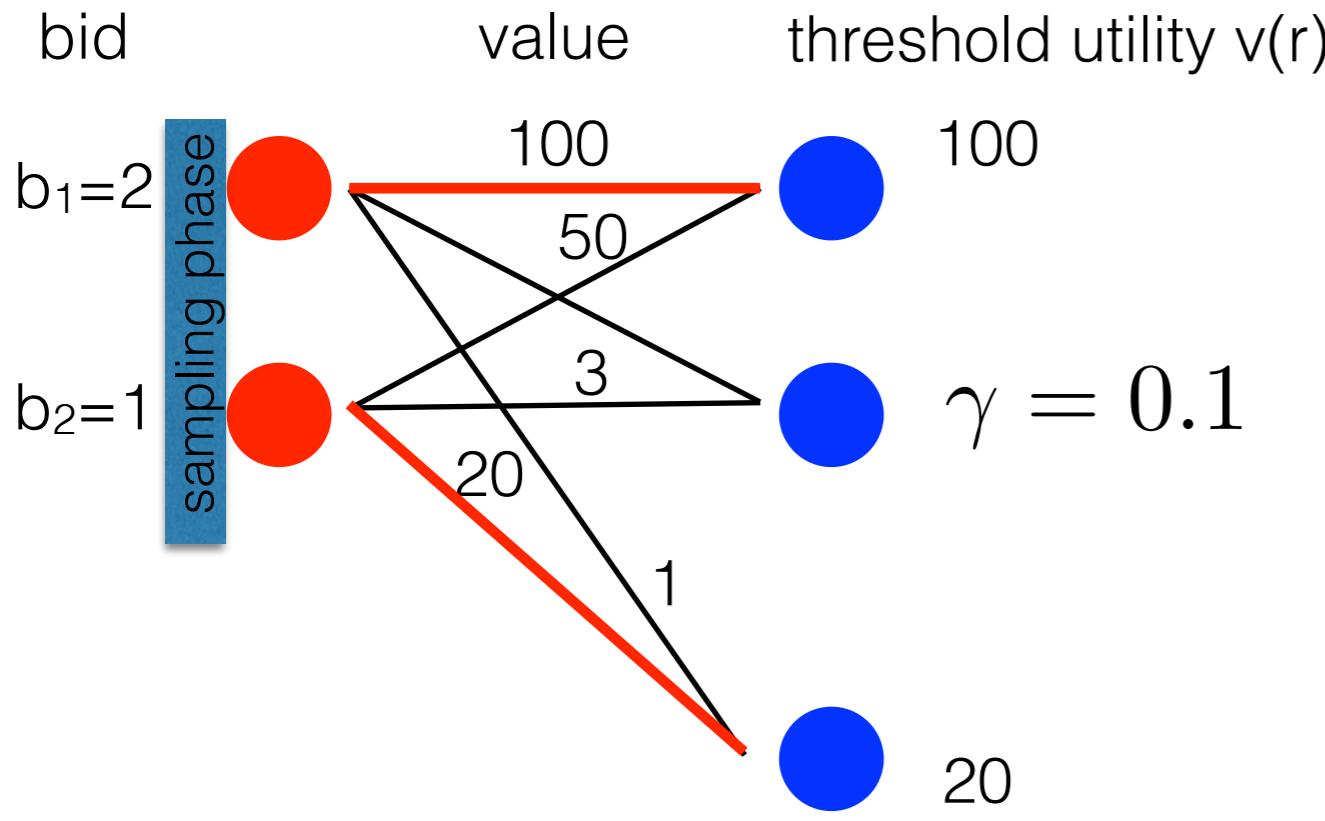
good Graph  $G(\gamma) = \left\{ e \in G : \frac{b(e)}{v(e)} < \gamma \right\}$

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

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

# Properties of ON

satisfies the payment budget constraint



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

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

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

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

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

# Properties of ON

For truthfulness we just check



R. Myerson

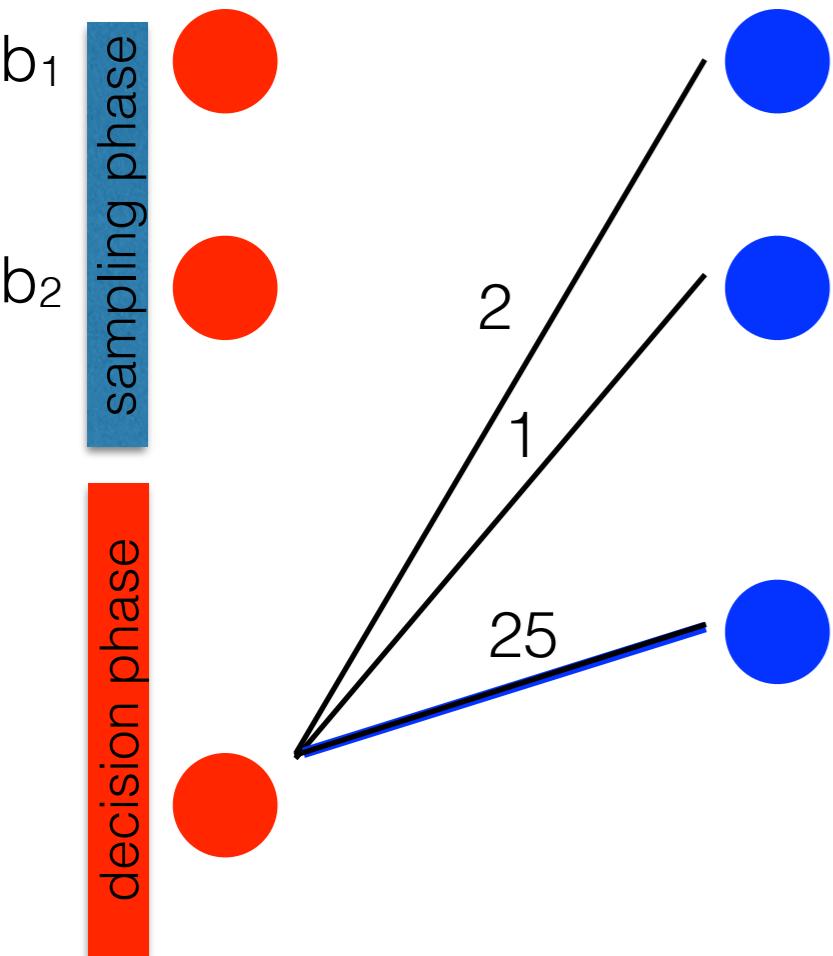
**Monotonicity** - if an agent is selected with bid  $b$ , then he is always selected if he bids below  $b$

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

# Properties of ON – truthfulness

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

bid                    value                    utility threshold  $v(r)$



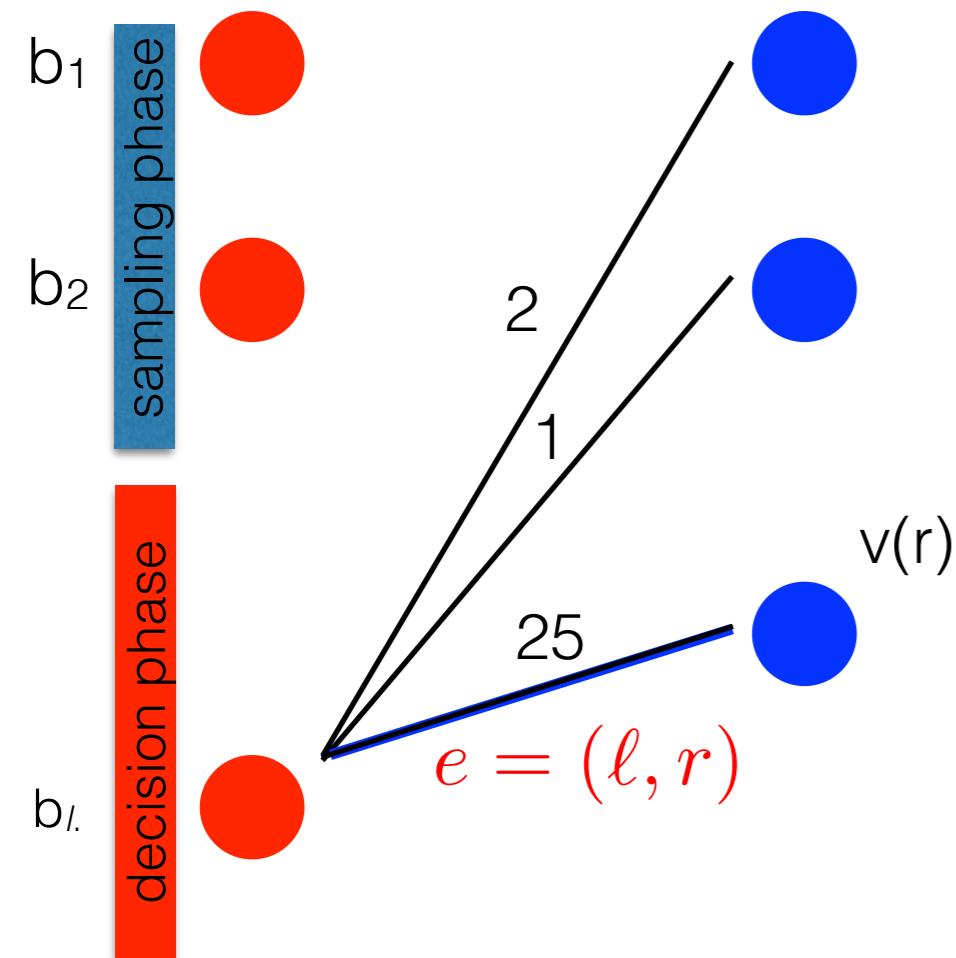
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max w  $>$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON – truthfulness

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

bid                    value                    utility threshold  $v(r)$



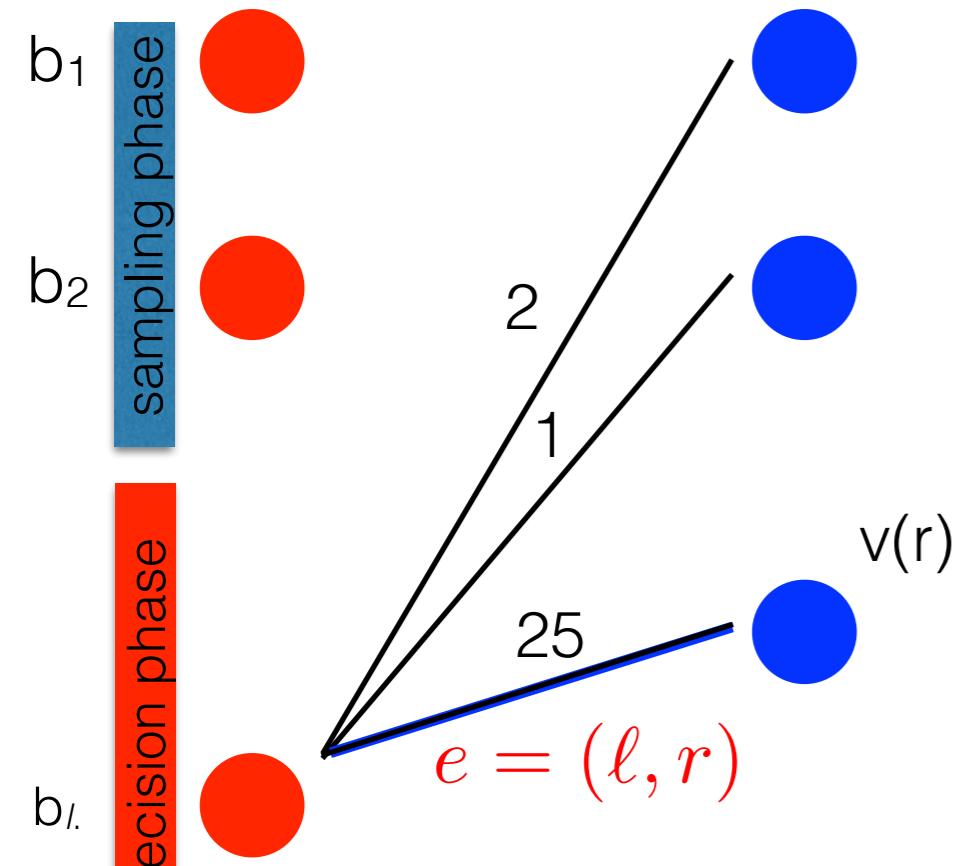
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta\gamma v(r)$

# Properties of ON – truthfulness

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

bid                    value                    utility threshold  $v(r)$



Assumption:

$$\frac{v_{max}}{v_{min}} \leq \beta$$

$$u(e) \leq \beta v(r)$$

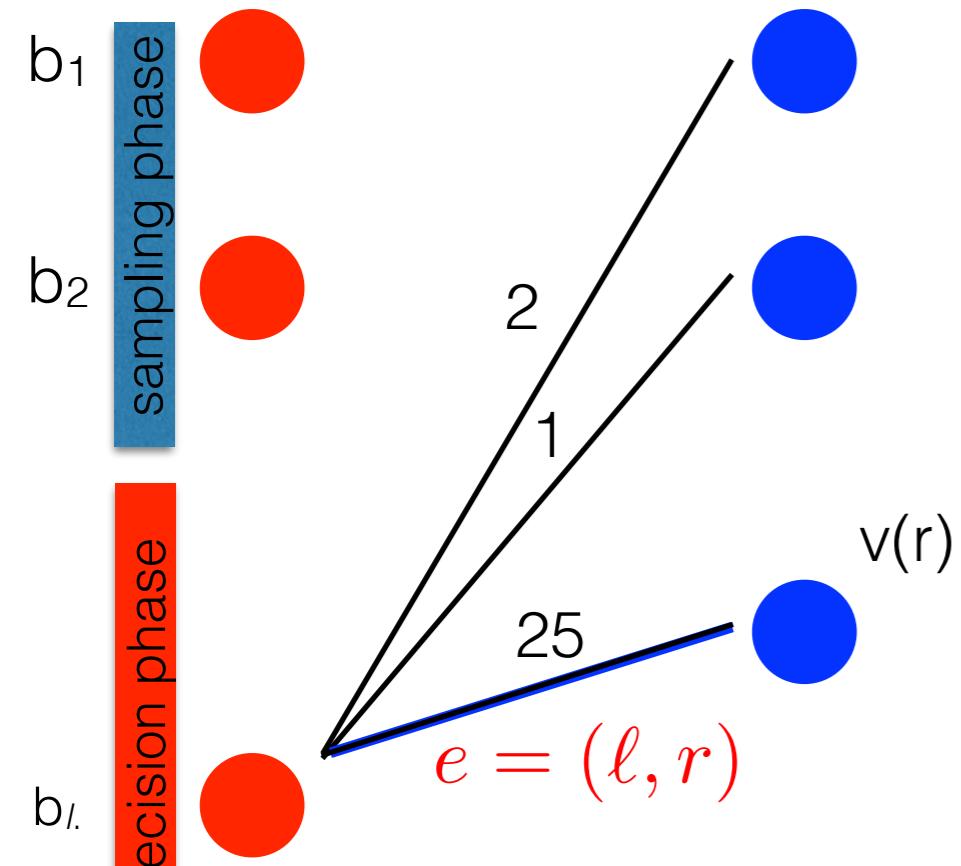
**on arrival of a new left vertex**

remove all edges with **bid to benefit** ratio  $> \gamma$   
match the right vertex with max  $w >$  utility thres.  
Payment for selected left vertex  $p_\ell = \beta \gamma v(r)$

# Properties of ON – truthfulness

$\gamma$  and threshold utility  $v(r)$  obtained from OFF

bid                    value                    utility threshold  $v(r)$



**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:

$$\frac{v_{max}}{v_{min}} \leq \beta$$

$$u(e) \leq \beta v(r)$$

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

$$\mathbf{E} \{p_1 + p_2 + \dots + p_m\} < B$$

# Important Extension

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

$$\mathbf{E} \{p_1 + p_2 + \dots + p_m\} < B$$

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



Peynet Museum' Antibes