Social Learning in Multi Agent Multi Armed Bandits

Abishek Sankararaman, UC Berkeley

April 9, 2020

<u>Joint Work with</u>

- Sanjay Shakkottai, Ronshee Chawla, UT Austin
- Ayalvadi Ganesh, University of Bristol

Multi Armed Bandit Problem

A set of possible drugs with a-priori unknown cure rates

Multi Armed Bandit Problem

A set of possible drugs with a-priori unknown cure rates

Task - Prescribe one of these to new incoming patients to both

(i) cure them and
(ii) collect data about their cure rates

Multi Armed Bandit Problem




A set of possible drugs with a-priori unknown cure rates

Task - Prescribe one of these to new incoming patients to both

(i) cure them and
(ii) collect data about their cure rates

Explore/Exploit Tradeoff for each new patient [Thompson' 33]

Exploit Prescribe a drug that has shown the best promise so far

Explore Try a new drug to discover more promising alternatives

Run a risk of not curing these patients

Outline

1. Single Agent MAB

2. The Multi-Agent Setup

3. The Gossiping Insert-Eliminate (Gosine) Algorithm

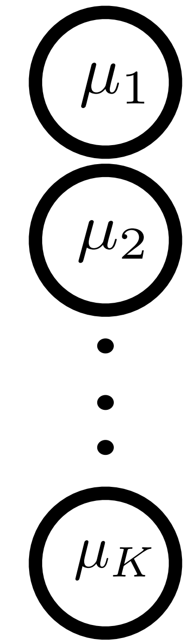
4. Insights

Multi Armed Bandit Problem

At each time, $t \in \{1, \cdots, T\}$ an agent

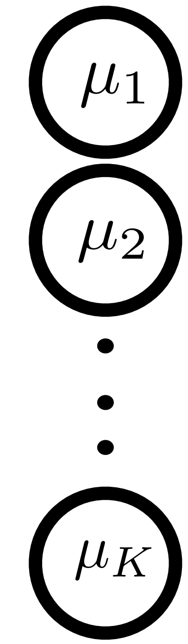
- chooses an arm $I_t \in \{1, \cdots, K\}$
- receives a stochastic reward $X_t \in \{0, 1\}$

$\mathbb{P}[X_t = 1 | I_t] = \mu_{I_t}$ independent of everything else



Each arm corresponds to a drug in the previous example

Multi Armed Bandit Problem



At each time, $t \in \{1, \cdots, T\}$ an agent

- chooses an arm $I_t \in \{1, \cdots, K\}$
- receives a stochastic reward $X_t \in \{0, 1\}$

  $\mathbb{P}[X_t = 1 | I_t] = \mu_{I_t}$ independent of everything else

Each arm corresponds to a drug in the previous example

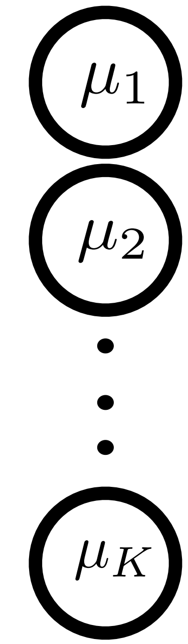
Goal - Sequentially choose arms to maximize total reward $\mathbb{E}[\sum_{t=1}^{T} X_t]$

Multi Armed Bandit Problem

At each time, $t \in \{1, \cdots, T\}$ an agent

- chooses an arm $I_t \in \{1, \cdots, K\}$
- receives a stochastic reward $X_t \in \{0, 1\}$

  $\mathbb{P}[X_t = 1 | I_t] = \mu_{I_t}$ independent of everything else



Each arm corresponds to a drug in the previous example

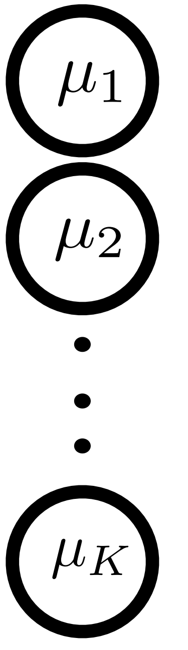
Goal - Sequentially choose arms to maximize total reward $\mathbb{E}[\sum_{t=1}^{T} X_t]$

Choosing $I_t = \arg\max\{\mu_k : k \in \{1, \cdots, K\}\}$ at all times is optimal

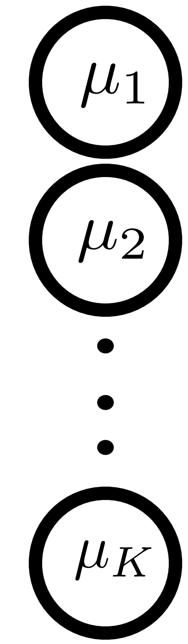
Challenge The arm-means $(\mu_i)_{i=1}^{K}$ initially unknown

Multi Armed Bandit Problem

As we play arms, can learn $(\mu_i)_{i=1}^K$



Multi Armed Bandit Problem



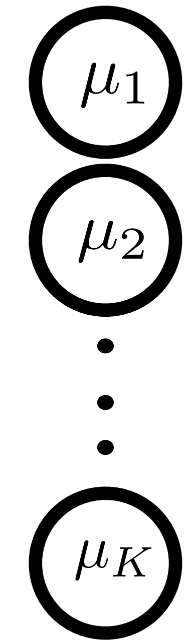
As we play arms, can learn $(\mu_i)_{i=1}^K$

<u>Explore-Exploit Tradeoff</u>

Exploit Play the arm that has been best so far

Explore Play an arm played few times so as to see if it is good

Multi Armed Bandit Problem



As we play arms, can learn $(\mu_i)_{i=1}^K$

<u>Explore-Exploit Tradeoff</u>

Exploit Play the arm that has been best so far

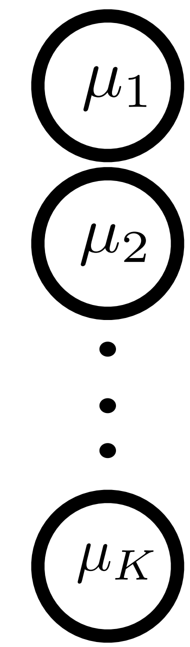
Explore Play an arm played few times so as to see if it is good

<u>Performance Metric</u> - Regret $R_T = \mu^* T - \mathbb{E}[\sum_{t=1}^T X_t]$

$$\mu^* = \max\{\mu_1, \cdots, \mu_K\}$$

How much loss due to lack of knowledge ?

Multi Armed Bandit Problem



Modern Day Applications

Internet Advertising

Which among the set of ads to display for a particular key-word

Recommendation Engines

Which category of items to recommend to a user

Packet Routing in Networks

On which route of the network to route packets to meet delay constraints

Multi Armed Bandit Problem

Suppose $1 \geq \mu_1 > \mu_2 \cdots \geq \mu_K$ Arm Gap $\Delta_i := \mu_1 - \mu_i$

Lower Bound [Lai and Robbins '85]

Under any "reasonable" strategy, any sub-optimal arm j , will be played

at-least $\mathbb{E}[N_k(T)] \geq \frac{\log(T)}{\mathrm{kl}(\mu_j, \mu_1)}$ times, on average .

Unreasonable strategy example - Always pull arm 3

Multi Armed Bandit Problem

Suppose $1 \geq \mu_1 > \mu_2 \cdots \geq \mu_K$ Arm Gap $\Delta_i := \mu_1 - \mu_i$

<u>Lower Bound</u> [Lai and Robbins '85]

Under any "reasonable" strategy, any sub-optimal arm j , will be played at-least $\mathbb{E}[N_k(T)] \geq \dfrac{\log(T)}{\mathrm{kl}(\mu_j, \mu_1)}$ times, on average .

Unreasonable strategy example - Always pull arm 3

<u>Corollary</u> $R_T \geq \Omega\left(\sum_{k=2}^{K} \dfrac{\log(T)}{\Delta_k}\right)$ Logarithmic Regret is unavoidable.

<u>Proof</u> $R_T = \sum_{k=2}^{K} \Delta_k \mathbb{E}[N_k(T)], \quad KL(\mu_j, \mu_1) \geq 2\Delta_j^2$

Multi Armed Bandit Problem

UCB Algorithm [Auer et.al. '02]

At time t, choose arm

$$I_t \in \arg\max_k \left(\widehat{\mu}_k(t-1) + \sqrt{\frac{4\alpha \log(t)}{N_k(t-1)}} \right)$$

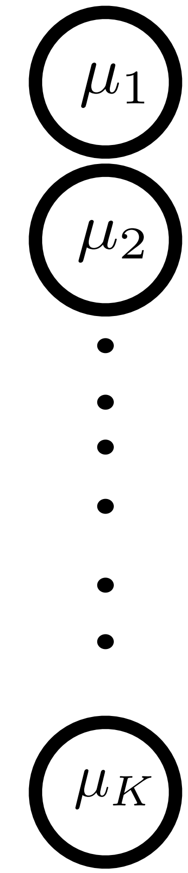
The right explore-exploit tradeoff !

$\widehat{\mu}_k(t-1)$ Empirical Observed Mean of arm k at time t-1

Theorem $R_T \leq \left(\sum_{k=2}^{K} \frac{4\alpha}{\Delta_k} \right) \log(T) + K \frac{\pi^2}{3}$

Matches lower bound unto constants.

Multi Agent Setup



What if *multiple agents* play the *same* MAB instance ?

Can they collaborate and jointly reduce their individual regret ?

Multi Agent Setup - Motivation

One server is serving ads for a *fixed* keyword

At each search request, server can choose to display one ad

Choice of an arm to pull

Ad 1 μ_1

Ad 2 μ_2

⋮

Ad K μ_K

At the end, receives a stochastic reward

Goal is to maximize revenue (minimize regret)

Multi Agent Setup - Motivation

<u>Multiple</u> servers serving ads for a *<u>fixed</u>* keyword

Each search request, routed to a server

Each server chooses to display one ad when routed to it.

Ad 1	μ_1
Ad 2	μ_2
⋮	
Ad K	μ_K

Multi Agent Setup - Motivation

Multiple servers serving ads for a *fixed* keyword

Each search request, routed to a server

Each server chooses to display one ad when routed to it.

Ad 1	μ_1
Ad 2	μ_2
⋮	
Ad K	μ_K

Servers can potentially collaborate and learn from each other's experience.

Managed by the same company

Multi Agent Setup - Motivation

At each time, every server makes a decision from K alternatives

Large volume of search queries

μ_1

μ_2

$\vdots$

μ_K

Multi Agent Setup - Motivation

At each time, every server makes a decision from K alternatives

Large volume of search queries

μ_1

μ_2

$\vdots$

μ_K

1. **No Communication** -

   Each server learns on its own from its own mistakes

   Individual Server Regret - $O\left(\frac{K}{\Delta}\log(T)\right)$

   Communication Resources - **0**

Multi Agent Setup - Motivation

At each time, every server makes a decision from K alternatives

μ_1

μ_2

$\vdots$

μ_K

Large volume of search queries

1. **No Communication** -

   Each server learns on its own from its own mistakes

   Individual Server Regret - $O\left(\frac{K}{\Delta}\log(T)\right)$

   Communication Resources - **0**

2. **Full Interaction** -

   Each server broadcasts its action and reward to all servers after every decision

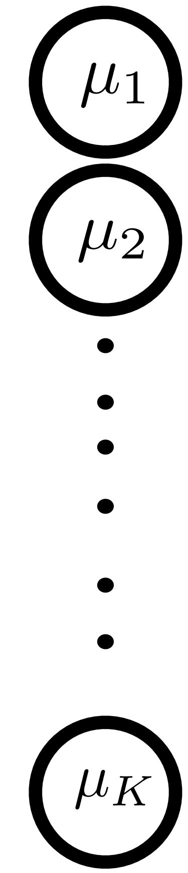
   Individual Server Regret - $O\left(\frac{1}{N}.\frac{K}{\Delta}\log(T)\right)$

   Overall system can be abstracted as a single agent

   Communication Resources - **T broadcasts per agent !**

Multi Agent Setup - Motivation

At each time, every server makes a decision from K alternatives



Large volume of search queries

1. **No Communication** -

   Each server learns on its own from its own mistakes

   Individual Server Regret - $O\left(\frac{K}{\Delta}\log(T)\right)$

   Communication Resources - **0**

2. **Full Interaction** -

   Each server broadcasts its action and reward to all servers after every decision

   Individual Server Regret - $O\left(\frac{1}{N}.\frac{K}{\Delta}\log(T)\right)$

   Overall system can be abstracted as a single agent

   Communication Resources - **T broadcasts per agent !**

How to effectively trade-off ? Best of both situations ??

Multi Agent Problem

N Agents $\quad G = (V, E)$ Network among agents $|V| = N$

K Arms

At each time $t \in \{1, \cdots, T\}$, every agent $j \in \{1, \cdots, N\}$

1. Play an arm $I_j(t) \in \{1, \cdots, K\}$ and receives reward $X_j(t) \in \{0, 1\}$
2. Can <u>**choose**</u> to pull information from any neighbor

$\mathbb{P}[X_j(t) = 1 | I_j(t)] = \mu_{I_j(t)}$ *Independent rewards across agents*

Multi Agent Problem

N Agents $G = (V, E)$ Network among agents $|V| = N$

K Arms

At each time $t \in \{1, \cdots, T\}$, every agent $j \in \{1, \cdots, N\}$

1. Play an arm $I_j(t) \in \{1, \cdots, K\}$ and receives reward $X_j(t) \in \{0, 1\}$
2. Can **choose** to pull information from any neighbor

$\mathbb{P}[X_j(t) = 1 | I_j(t)] = \mu_{I_j(t)}$ *Independent rewards across agents*

Agents have a communication budget $(B_t)_{t=1}^{T}$

At all times t, number of information pulls must be lesser than B_t

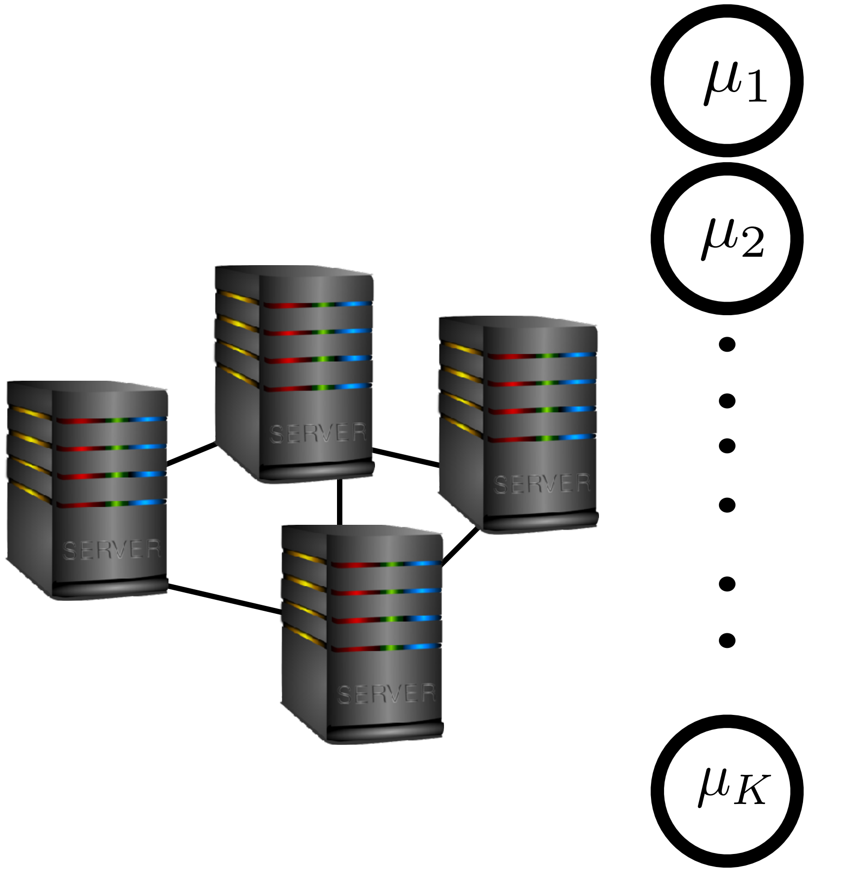
Example - $B_t = \sqrt{t}$ *Captures communication constraints*

Algorithm Design Considerations

Decentralized Algorithms -

All decisions only a function of the observed history at the agent.

Decisions - Choice of arm pull, information pull and message to communicate if asked



Algorithm Design Considerations

Decentralized Algorithms -

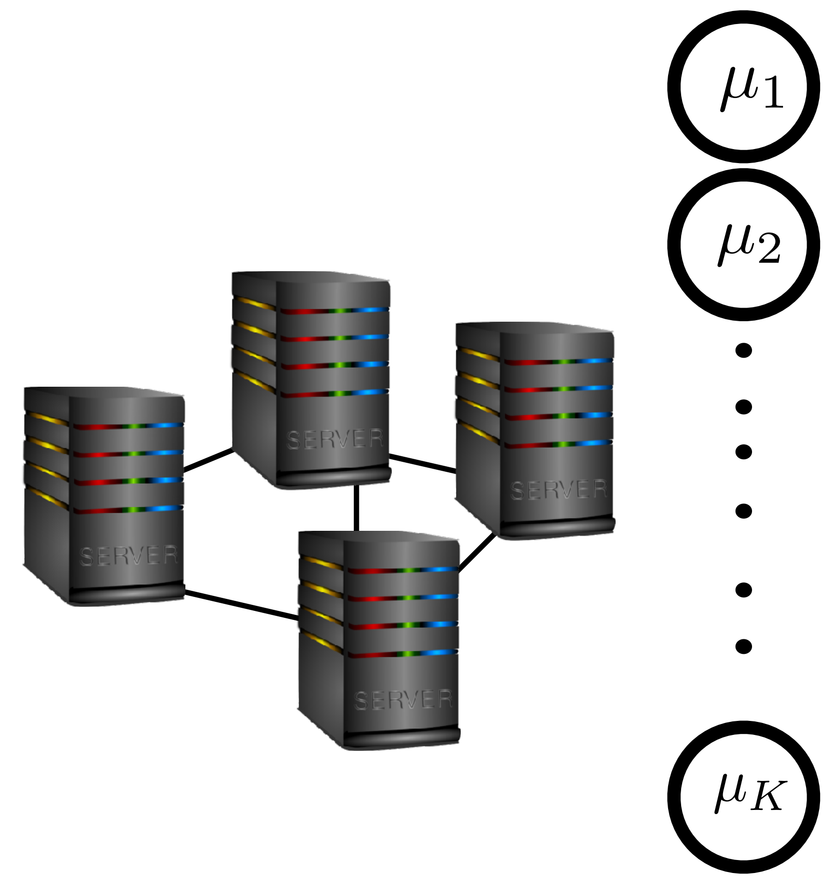
All decisions only a function of the observed history at the agent.

Decisions - Choice of arm pull, information pull and message to communicate if asked

What to communicate ?

How to incorporate the received messages ?

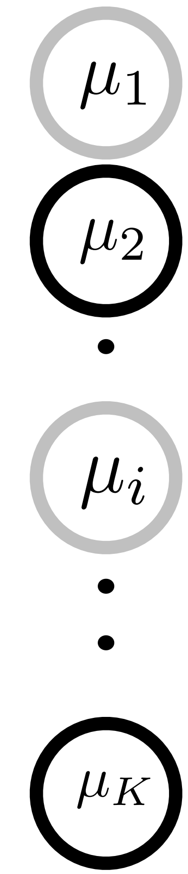
How to use communication budget ?



Key Ideas in the Algorithm

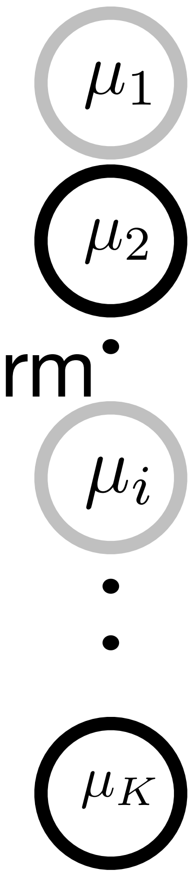
Key Ideas in the Algorithm

1. At all times, each agent only chooses from a *subset* of size $\left\lceil \frac{K}{N} \right\rceil + 2$ of possible arms



Social Learning Algorithm - Key Ideas

1. At all times, each agent only chooses from a *__subset__* of size $\left\lceil \frac{K}{N} \right\rceil + 2$ of possible arms



2. When asked for information, recommend your estimated best arm

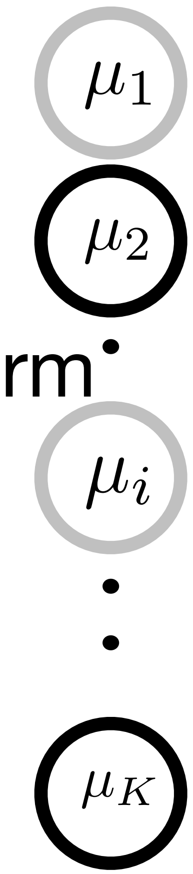
   On receiving information,

   1. Throw out the worst arm and
   2. Replace by the recommended arm

The "active" set of arms at each agent is dynamically evolving

Key Ideas in the Algorithm

1. At all times, each agent only chooses from a *<u>subset</u>* of size $\left\lceil \frac{K}{N} \right\rceil + 2$ of possible arms



2. When asked for information, recommend your estimated best arm

   On receiving information,

   1. Throw out the worst arm and
   2. Replace by the recommended arm

   The "active" set of arms at each agent is dynamically evolving

3. Frequency of communication ?

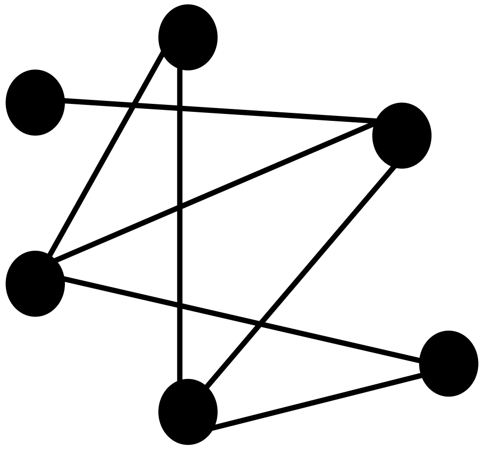
   <u>High</u> initially when unsure of having the best arm

   <u>Low</u> in late stage when confident of having the best arm.

Algorithm - Details

For simplicity assume there are N agents and N arms.

1. *Initialization* - each agents chose active set of 3 arms arbitrarily

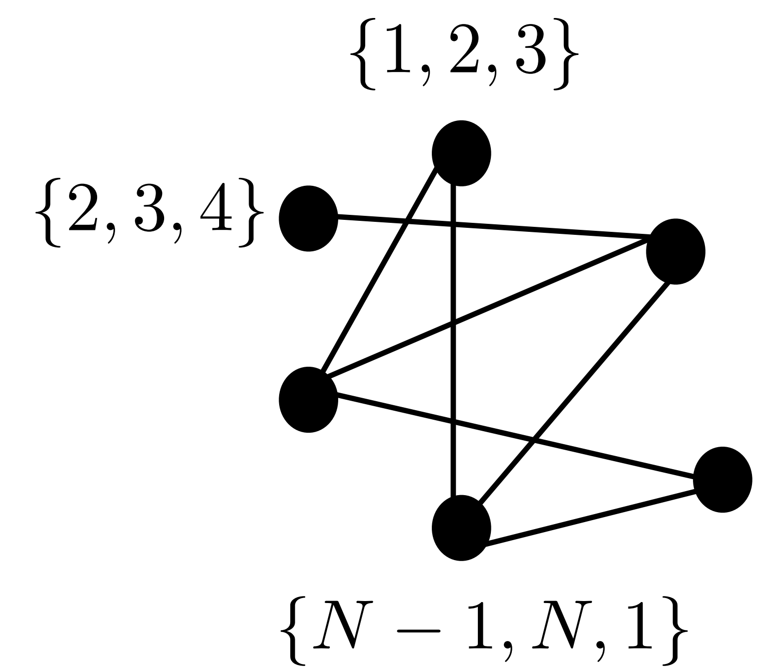


Algorithm - Details

For simplicity assume there are N agents and N arms.

1. *Initialization* - each agents chose active set of 3 arms arbitrarily

Every arm is being played by some agent in the beginning.



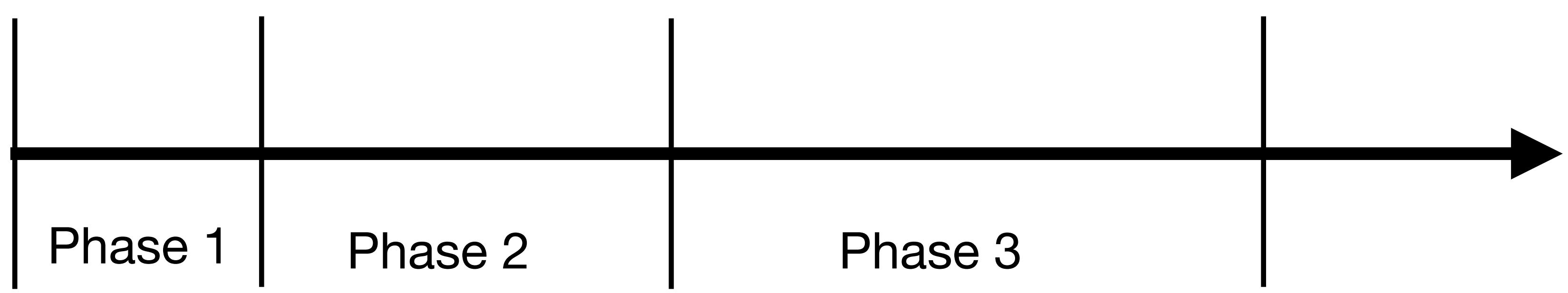
Gossiping Insert/Eliminate Algorithm

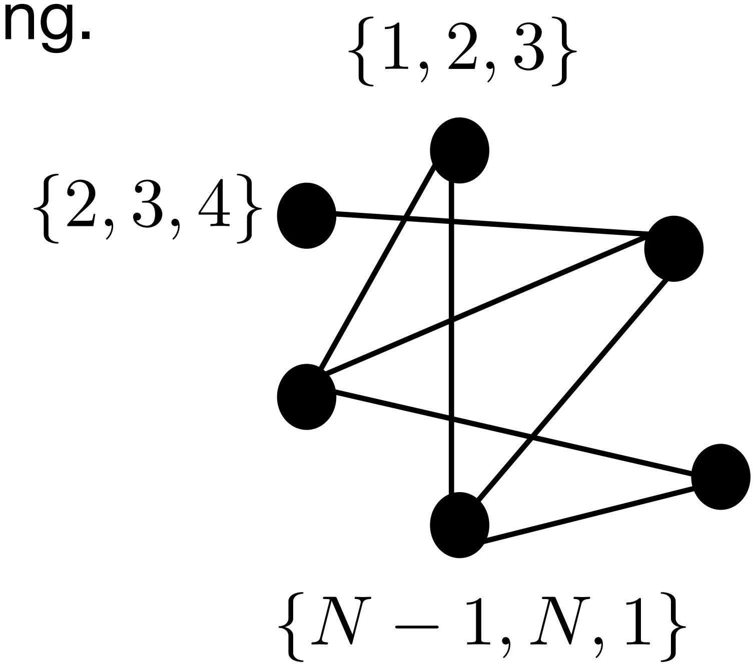
For simplicity assume there are N agents and N arms.

1. *Initialization* - each agent choses active set of 3 arms arbitrarily

Every arm is being played by some agent in the beginning.

Phases of increasing duration



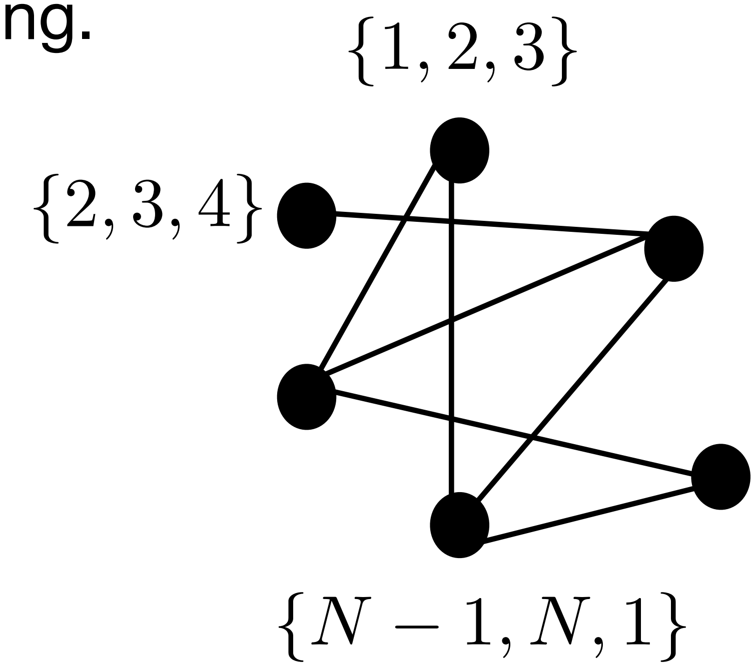


Gossiping Insert/Eliminate Algorithm

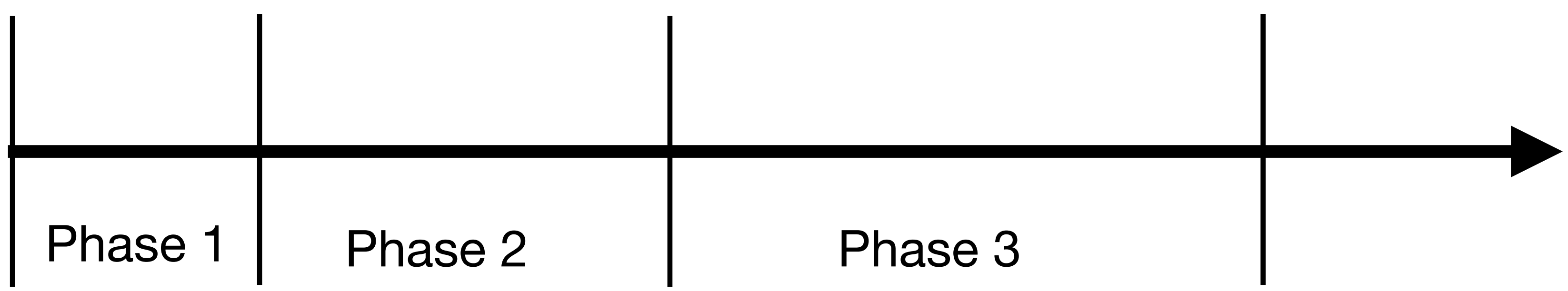
For simplicity assume there are N agents and N arms.

1. *<u>Initialization</u>* - each agent choses active set of 3 arms arbitrarily

Every arm is being played by some agent in the beginning.



Phases of increasing duration

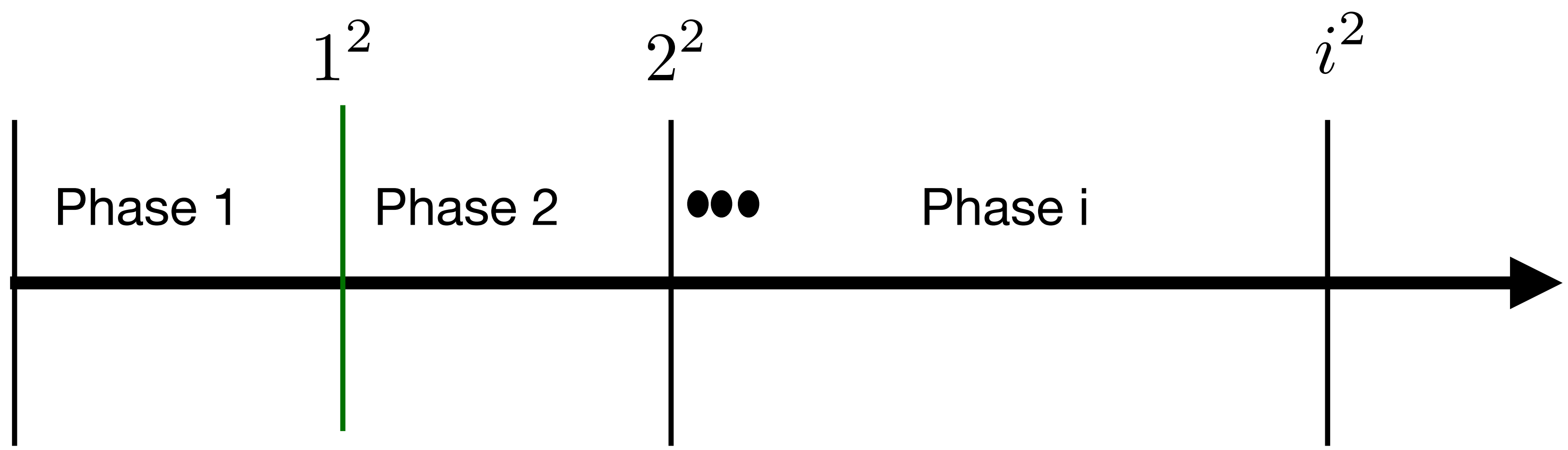


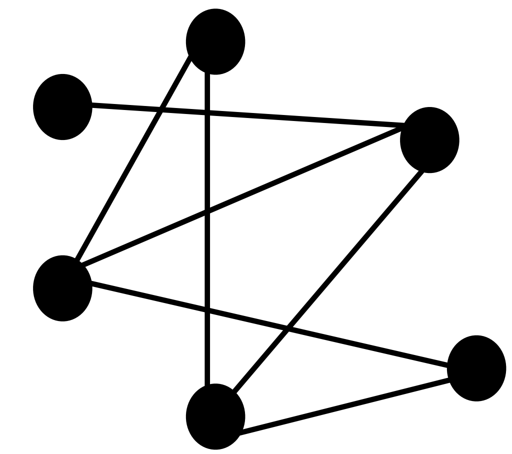
2. *<u>UCB among active arms in a phase. At the end of a phase</u>*

- Ask a random neighbor for a recommendation *Gossip*
- When asked, recommend the *most played arm in the previous phase*
- *Throw your least played arm and accept recommendation* *Insert/Eliminate*

Gossiping Insert/Eliminate Algorithm

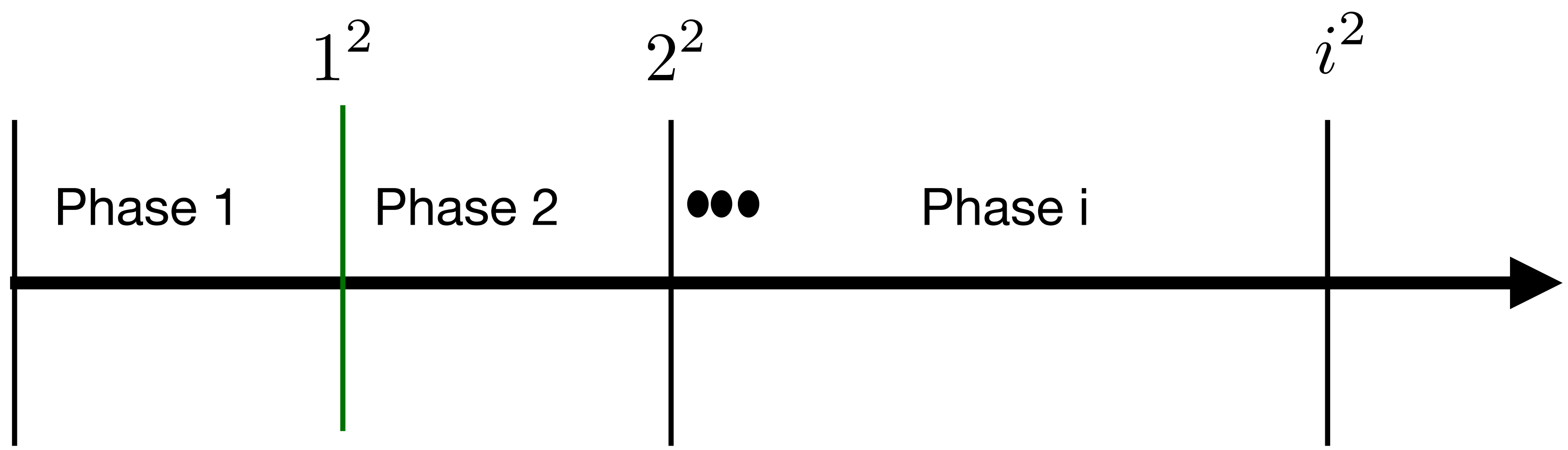
For ex. $B_t = \sqrt{t}$ as communication budget

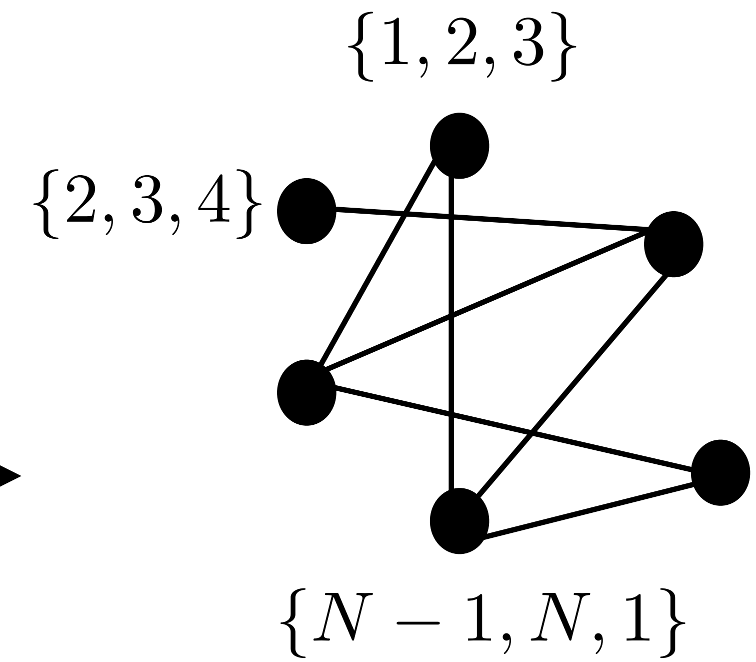




Gossiping Insert/Eliminate Algorithm

For ex. $B_t = \sqrt{t}$ as communication budget





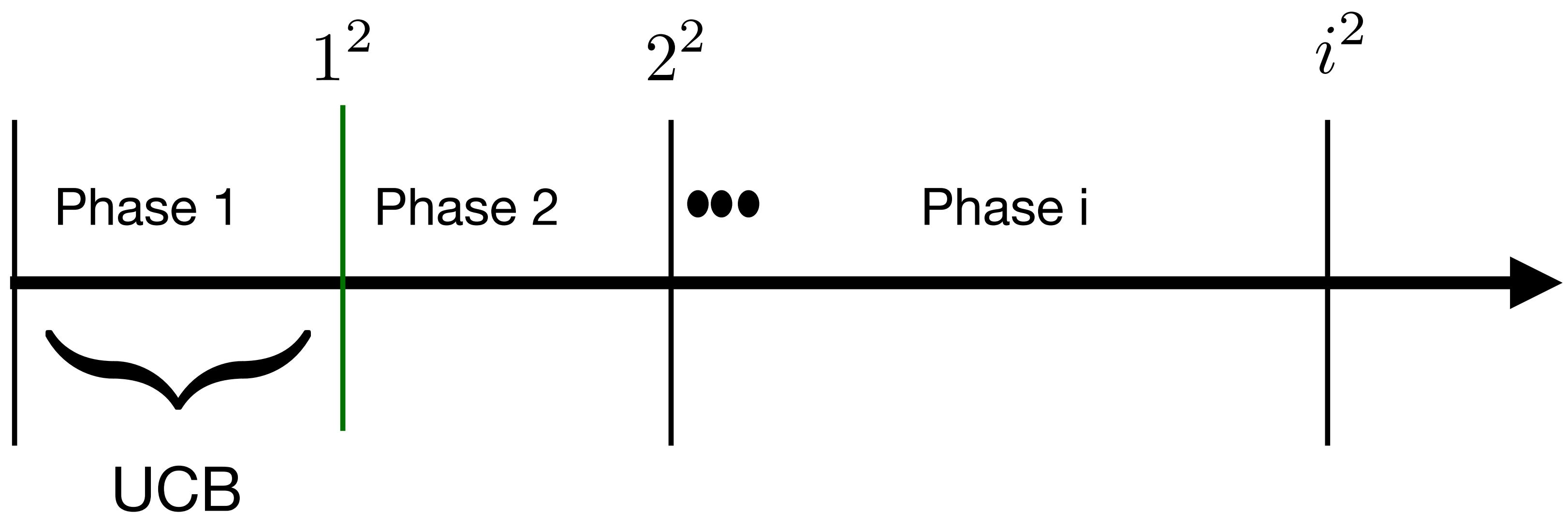
Agent 1 $\{1, 2, 3\}$

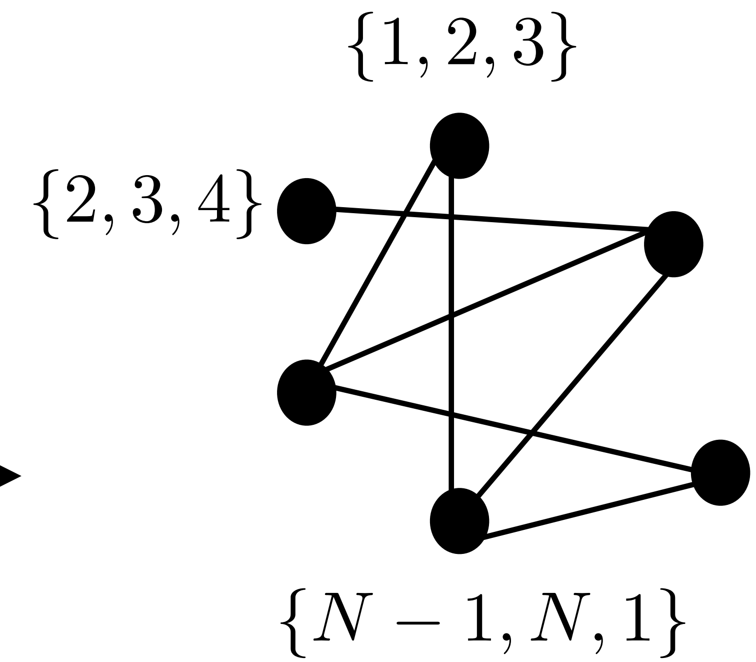
Agent 2 $\{2, 3, 4\}$

Agent N $\{N-1, N, 1\}$

Gossiping Insert/Eliminate Algorithm

For ex. $B_t = \sqrt{t}$ as communication budget





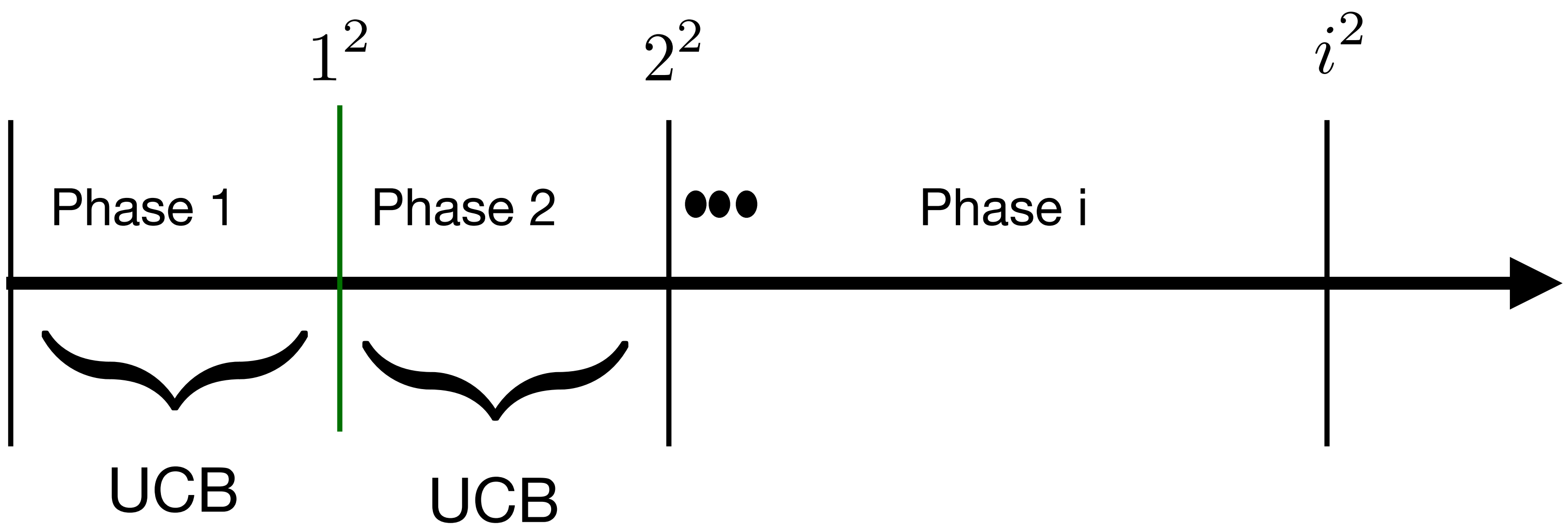
Agent 1 $\{1,2,3\}$

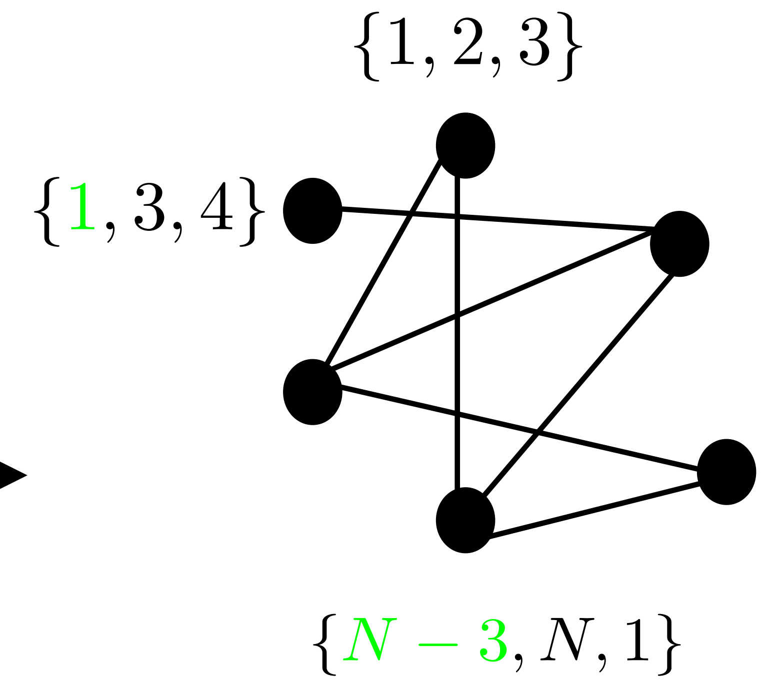
Agent 2 $\{2,3,4\}$

Agent N $\{N-1,N,1\}$

Gossiping Insert/Eliminate Algorithm

For ex. $B_t = \sqrt{t}$ as communication budget





Agent 1	$\{1, 2, 3\}$	$\{1, 2, 3\}$
Agent 2	$\{2, 3, 4\}$	$\{1, 3, 4\}$
$\vdots$	$\vdots$	$\vdots$
Agent N	$\{N-1, N, 1\}$	$\{N-3, N, 1\}$

Ask a random neighbor for recommendation

Suggest the most played arm in previous phase

Replace your worst arm with recommendation

Size of the active arms is fixed.

Algorithm - Why does it work ?

Best arm “eventually spreads” to all agents

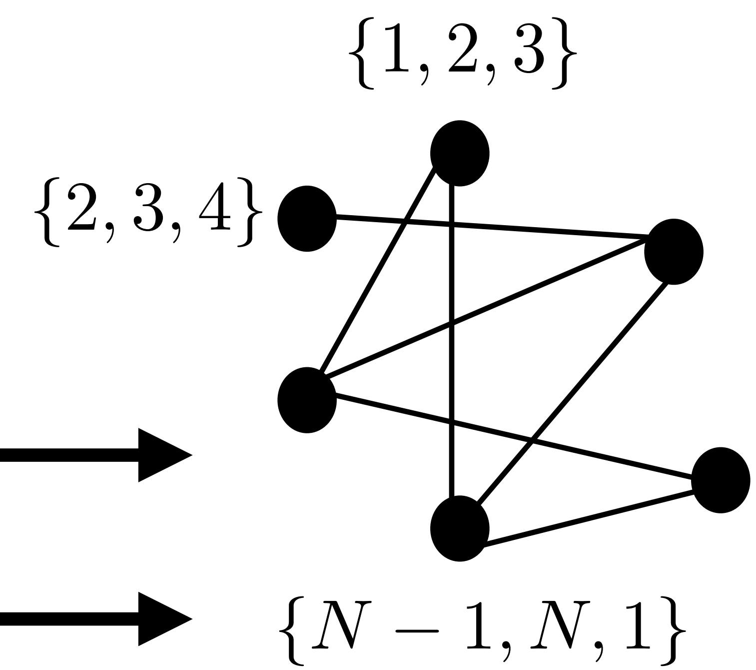
The bad arms never get recommended often and hence don’t spread.

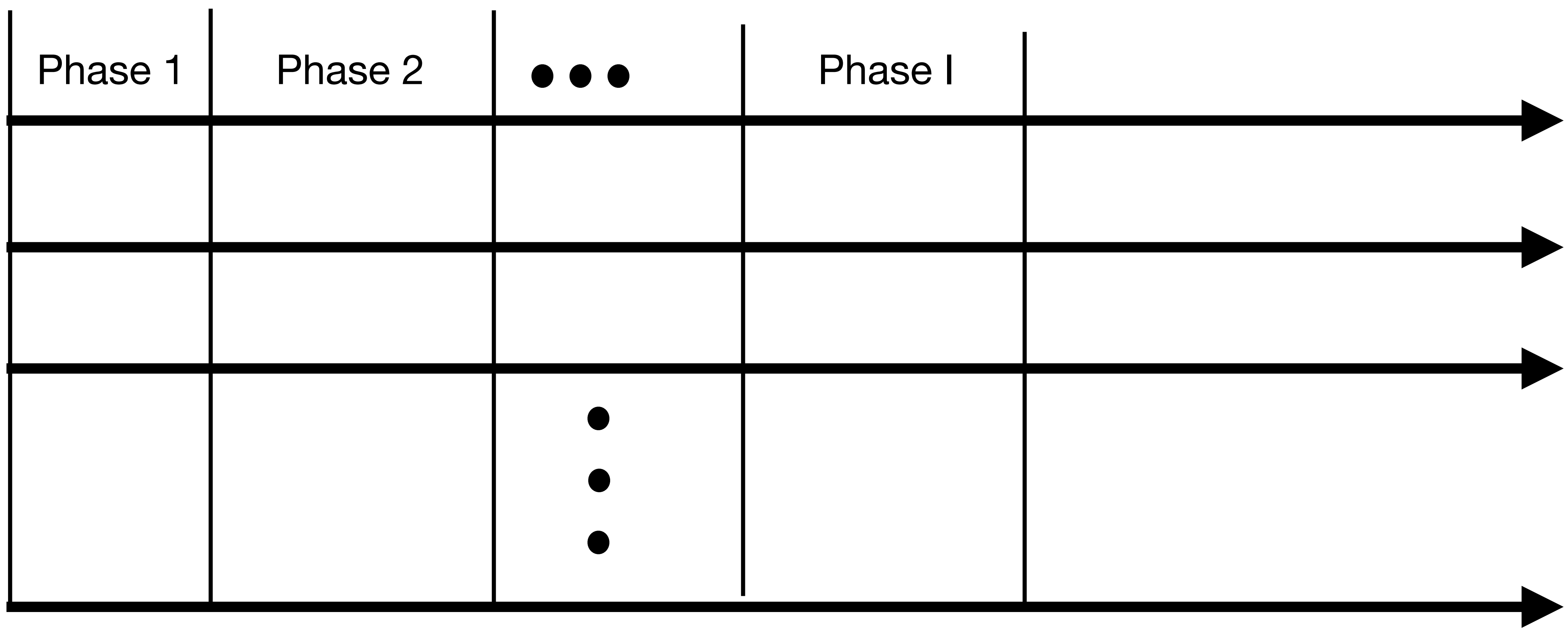
Algorithm - Why does it work ?

Best arm “eventually spreads” to all agents

The bad arms never get recommended often and hence don’t spread.

$$1 > \mu_1 > \mu_2 \geq \cdots \geq \mu_N > 0$$





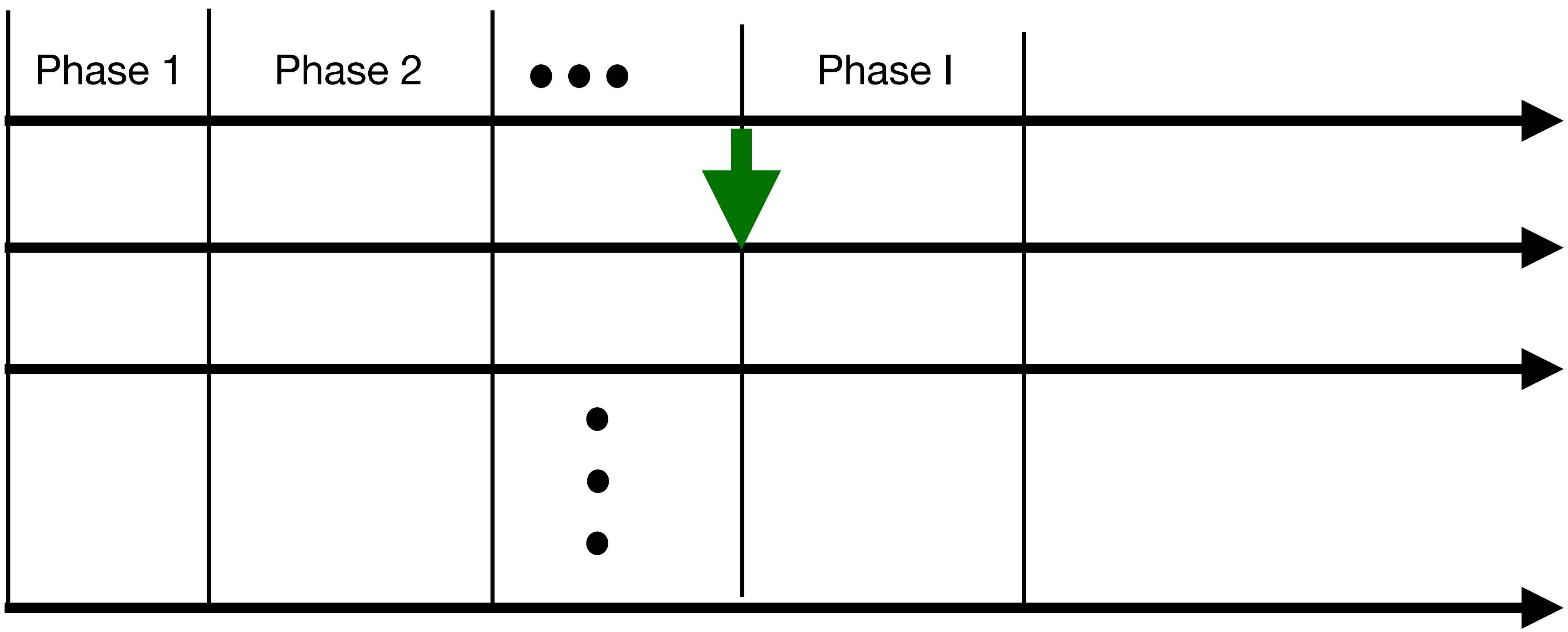
Algorithm - Why does it work ?

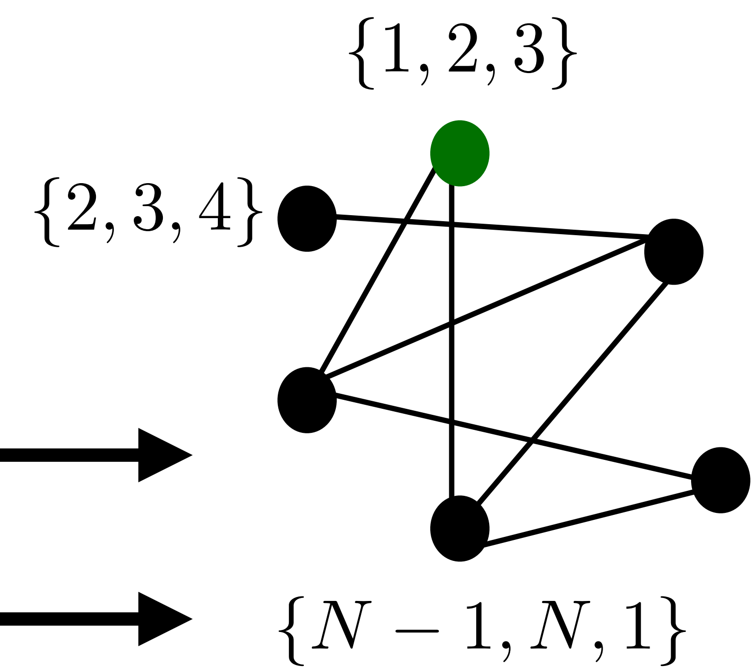
Best arm “eventually spreads” to all agents

The bad arms never get recommended often and hence don’t spread.

$$1 > \mu_1 > \mu_2 \geq \cdots \geq \mu_N > 0$$

Agent 1 has figured out the best arm





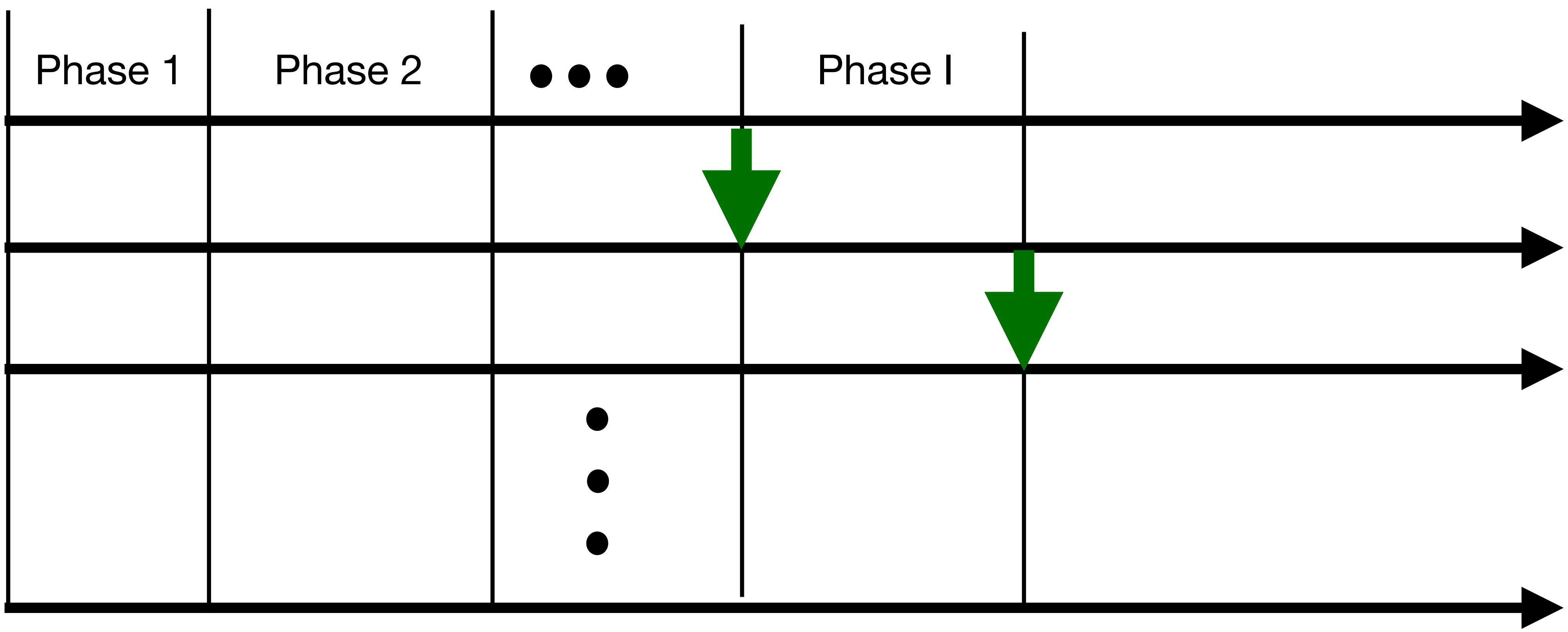
Algorithm - Why does it work ?

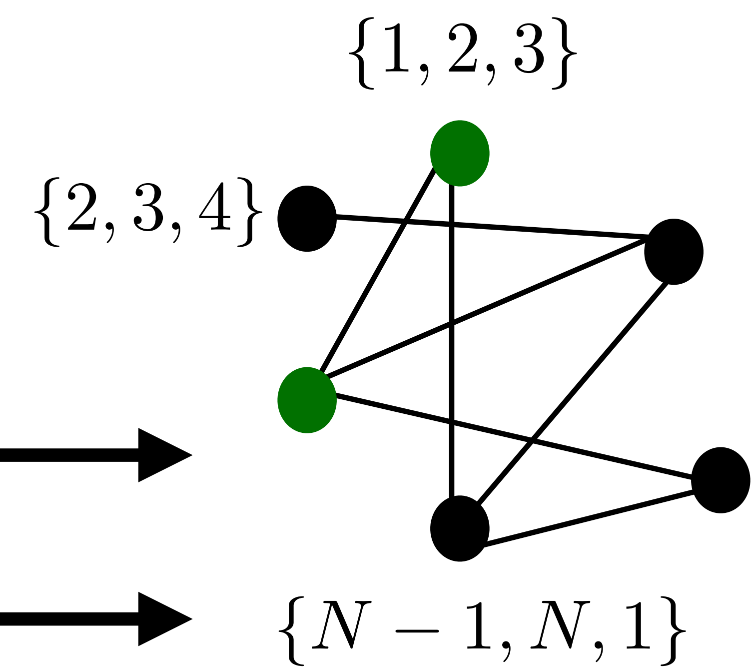
Best arm “eventually spreads” to all agents

The bad arms never get recommended often and hence don’t spread.

$$1 > \mu_1 > \mu_2 \geq \cdots \geq \mu_N > 0$$

Agents 1,2 have figured out the best arm



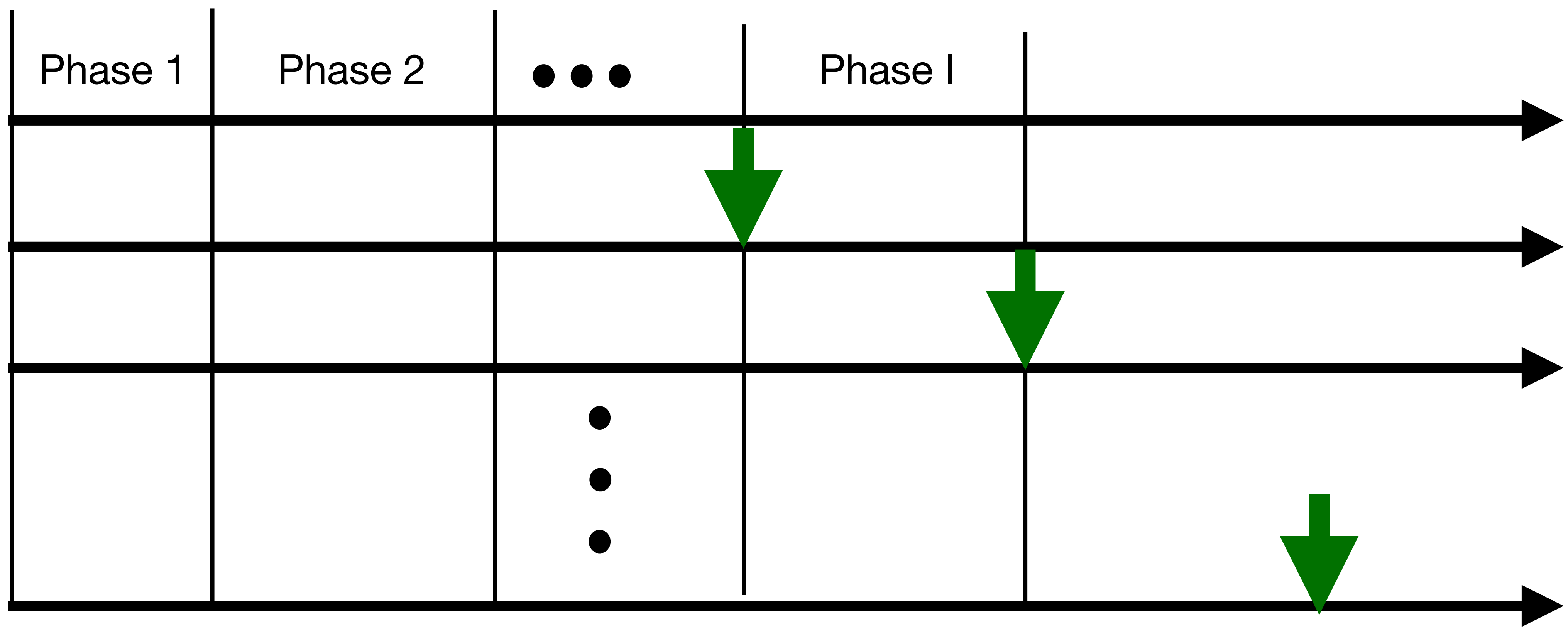


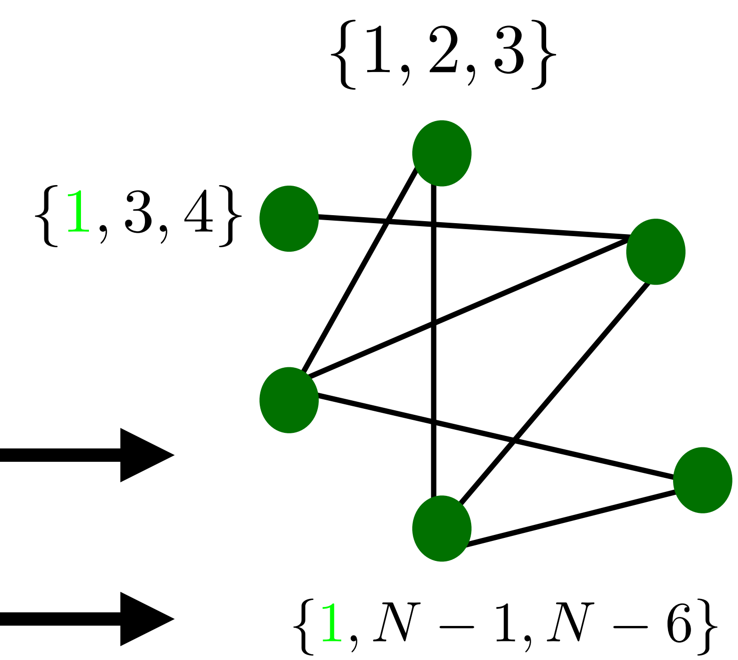
Algorithm - Why does it work ?

Best arm “eventually spreads” to all agents

The bad arms never get recommended often and hence don’t spread.

$$1 > \mu_1 > \mu_2 \geq \cdots \geq \mu_N > 0$$





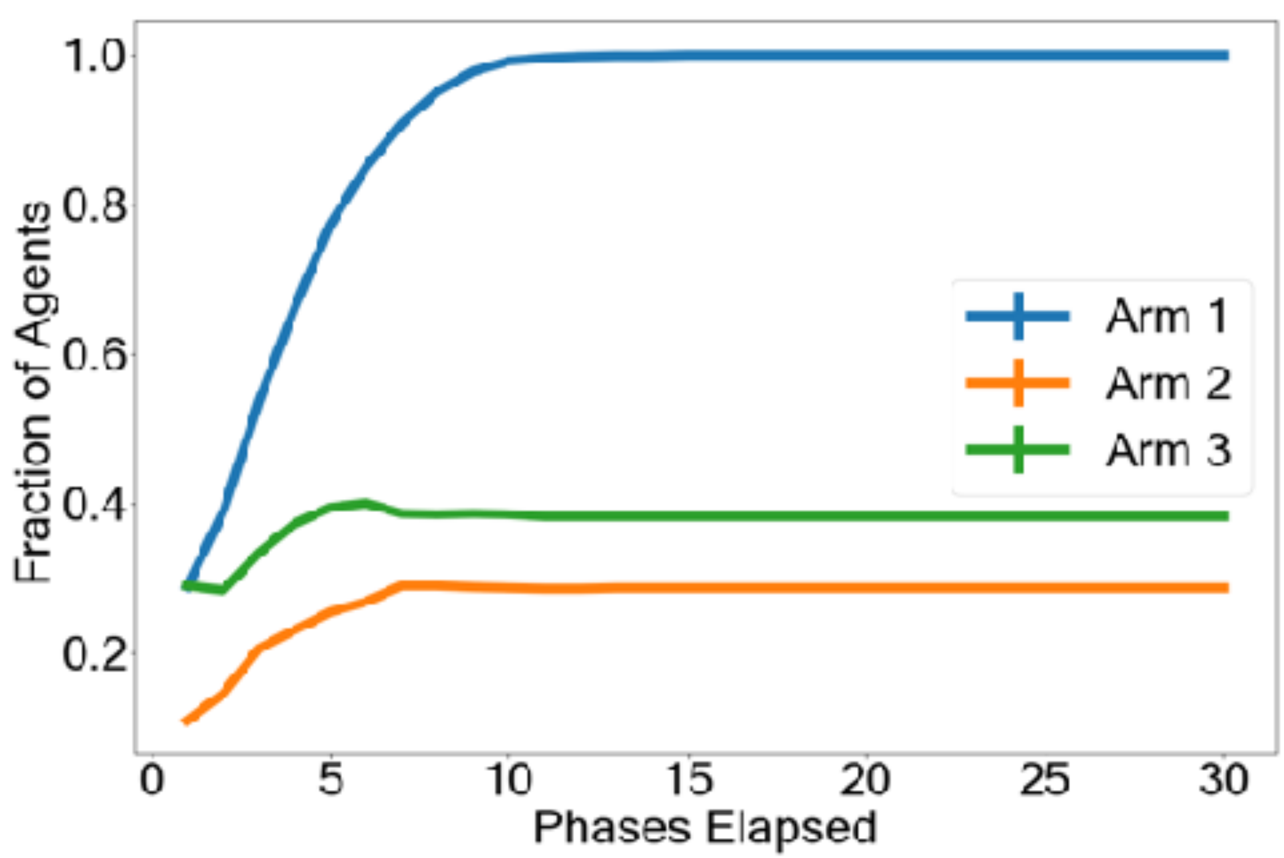
All agents have the best arm

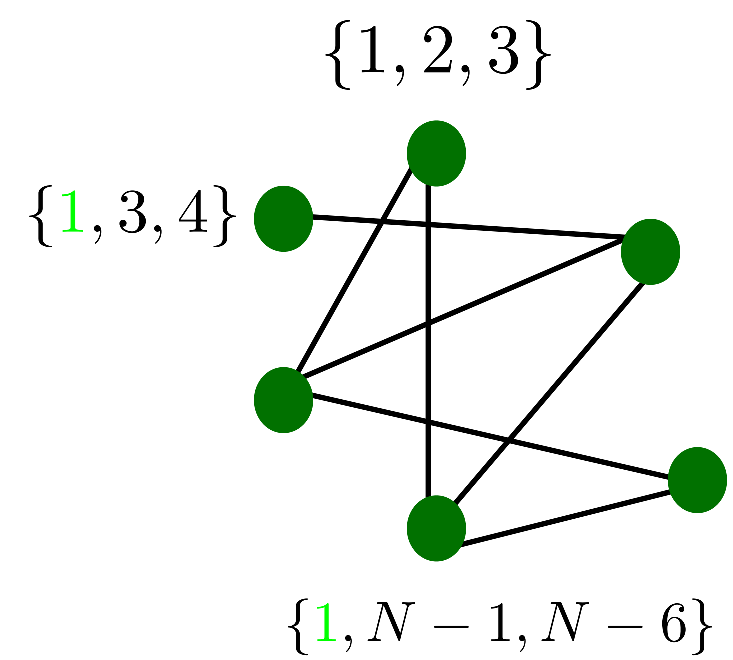
Algorithm - Why does it work ?

Best arm “eventually spreads” to all agents

The bad arms never get recommended often and hence don’t spread.

$$1 > \mu_1 > \mu_2 \geq \cdots \geq \mu_N > 0$$

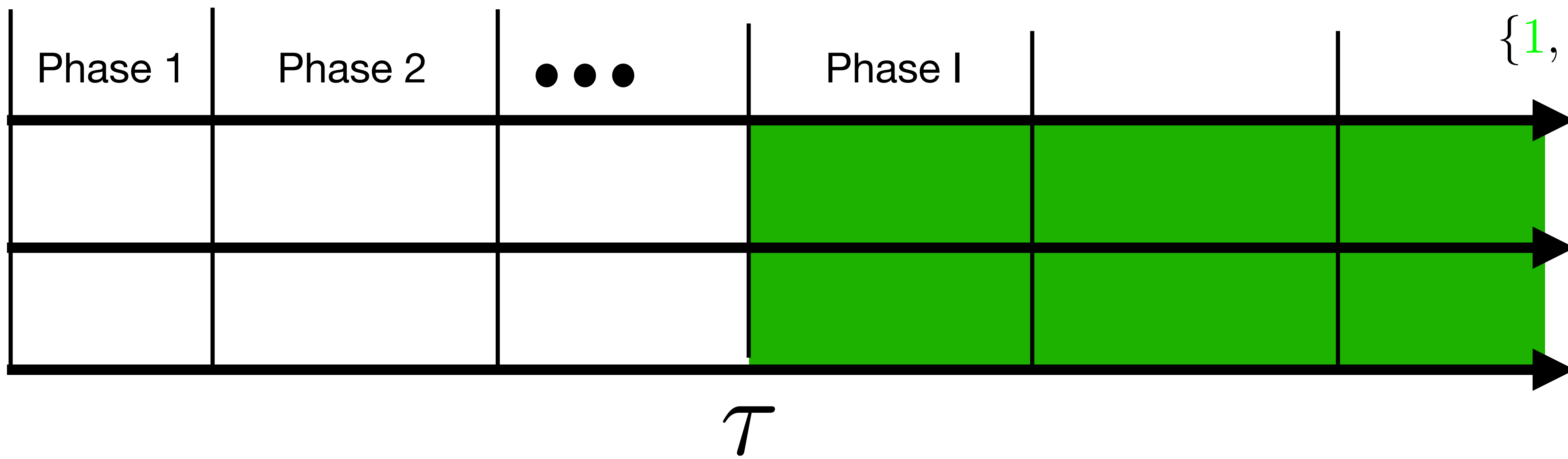


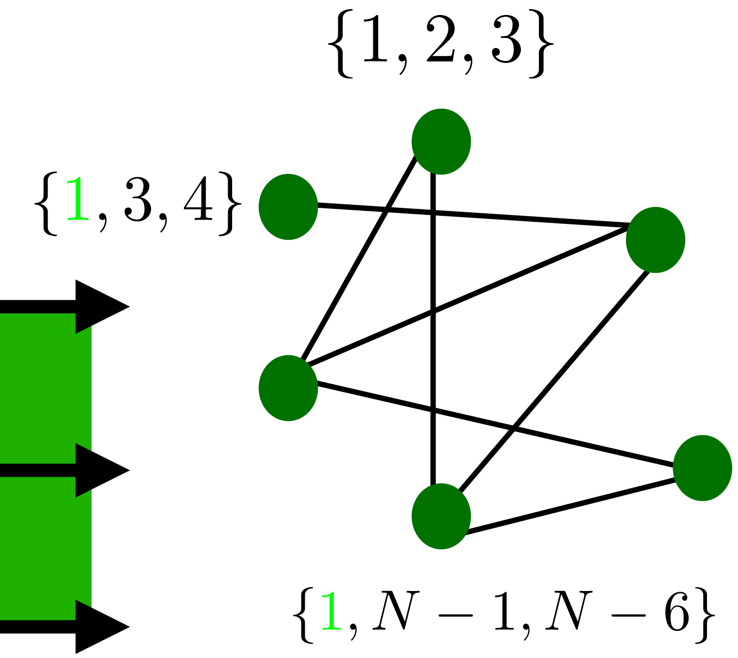


N=K=10, arm means randomly chosen in [0,1]

Algorithm - Why does it work ?

$$1 > \mu_1 > \mu_2 \geq \cdots \geq \mu_N > 0$$





Formally, we prove that, for all phases, after a random phase index τ

1. Best arm is in all agent's active arm set

2. Agents always recommend the best-arm

⇒ Regret of any agent - $R_T \leq \mathbb{E}[\tau] + \frac{8}{\Delta}\log(T) + O(1)$

Main Theorem

In a system with K arms and N agents, connected over **any connected graph** G, with communication budget scaling $B_t = \Omega(\log(t))$, the regret of any agent after T time steps is

$$R_T \leq O\left(\frac{1}{N}\frac{K}{\Delta}\log(T)\right) + f(G, (B_t)_{t=1}^{T})$$

Same scaling as full communication *Constant independent of time*

Main Theorem

In a system with K arms and N agents, connected over **any connected graph** G, with communication budget scaling $B_t = \Omega(\log(t))$, the regret of any agent after T time steps is

$$R_T \leq O\left(\frac{1}{N}\frac{K}{\Delta}\log(T)\right) + f(G, (B_t)_{t=1}^{T})$$

Same scaling as full communication *Constant independent of time*

Similar performance to full interaction, despite communication constraints

Communication constraints have only second order impact !

Regret/Communication Trade-Off

In a system with K arms and N agents, connected over **a regular graph with conductance** ϕ, with communication budget scaling $B_t = \lfloor t^{1\beta} \rfloor$ with $\beta > 1$, the regret of any agent after T time steps is

$$R_T \leq 12 \frac{\left\lceil \frac{K}{N} \right\rceil + 2}{\Delta} \log(T) + \left(\frac{2C \log(N)}{\phi} \right)^{\beta} + O(1)$$

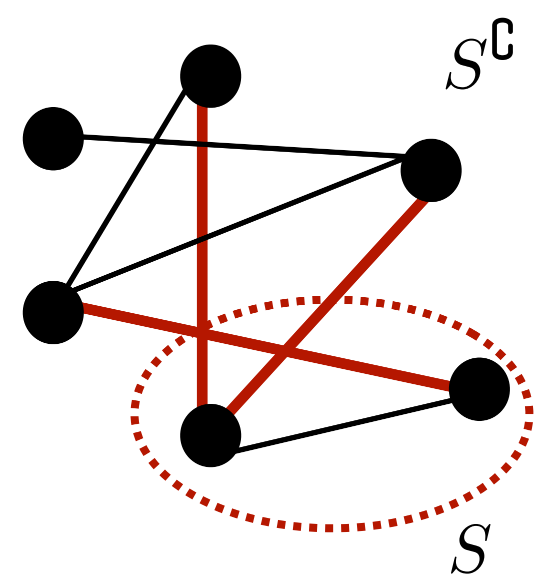
where C is an universal constant.

$$\phi := \min_{S \subset V, |S| = N/2} \frac{\mathrm{Cut}(S, S^\complement)}{\mathrm{Vol}(S)}$$

Conductance - Measure of connectivity

$\phi = 1/N$ Cycle Graph

$\phi = 1/2$ Complete Graph



Regret/Communication Trade-Off

In a system with K arms and N agents, connected over **a regular graph with conductance** ϕ, with communication budget scaling $B_t = \lfloor t^{1\beta} \rfloor$ with $\beta > 1$, the regret of any agent after T time steps is

$$R_T \leq 12 \frac{\left\lceil \frac{K}{N} \right\rceil + 2}{\Delta} \log(T) + \left(\frac{2C \log(N)}{\phi} \right)^{\beta} + O(1)$$

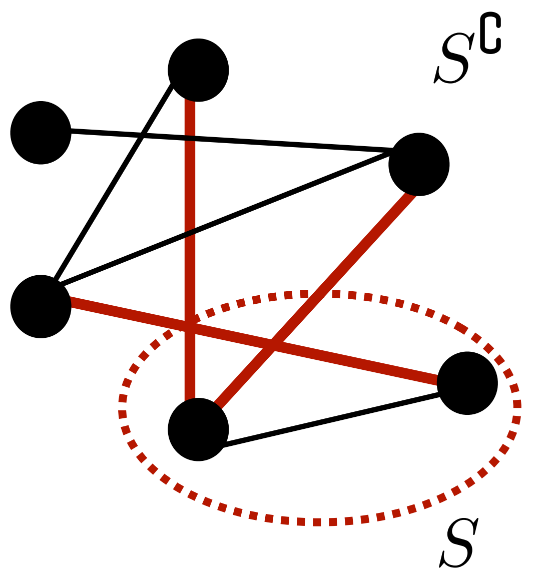
where C is an universal constant.

$$\phi := \min_{S \subset V, |S| = N/2} \frac{\mathrm{Cut}(S, S^{\complement})}{\mathrm{Vol}(S)}$$

Conductance - Measure of connectivity

$\phi = 1/N$ Cycle Graph

$\phi = 1/2$ Complete Graph

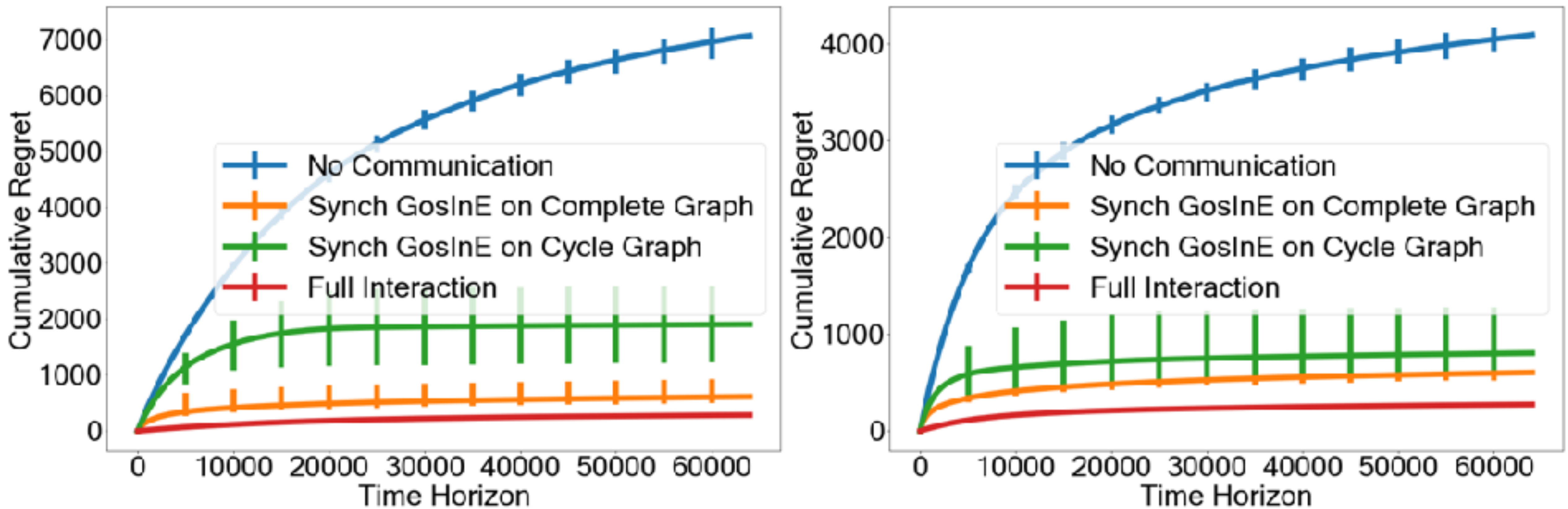


Examining the formula we get

For a fixed budget, smaller conductance => larger regret

For a fixed graph, smaller communication budget => larger regret

Impact of Network Structure

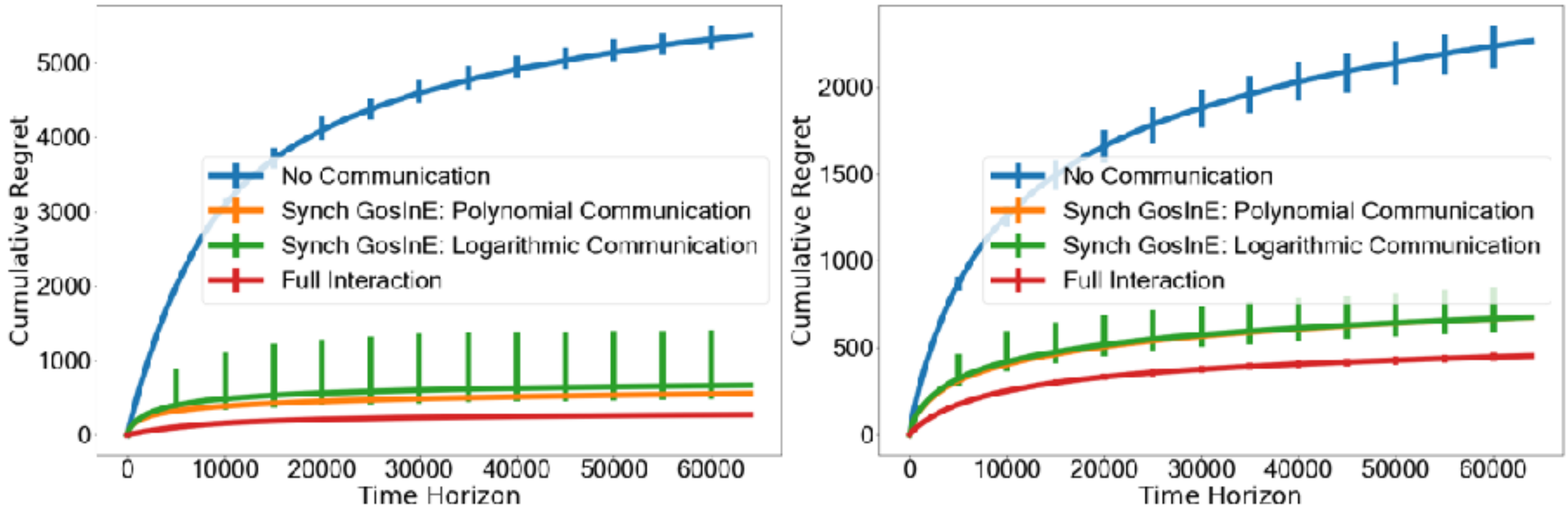


(N,K) are (25,75) and (15,50) respectively. Communication budget $B_t = t^{1/3}$

Performs nearly as well as full interaction with significantly sparse network !

Only sparse pairwise communication

Impact of Communication Budget



(N,K) are (20,70) and (5,50) respectively on Complete Graph.
Communication budgets $B_t = t^{1/3}$ and $B_t = \log_2(t)$ respectively

Performs nearly as well as full interaction with significantly small communication !

Conclusions

Formulated a collaborative multi armed bandit problem

A novel algorithmic paradigm based on social learning

"Only a fool learns from his mistakes. A wise man learns from the mistake of others"
Otto Van Bismarck

Future Work - Contextual Bandits, Heterogeneous arm means

Thank You For your Time

Reference

Gossiping Insert Eliminate Algorithm for Multi Agent Bandits - AISTATS 2020

https://arxiv.org/abs/2001.05452