Bandit networks

Charles AUGUSTE
École des Ponts
charles.auguste@ponts.org

Louis TREZZINI
École des Ponts
louis.trezzini@ponts.org

Abstract

A single-agent multi-armed bandit (MAB) problem is a problem in which a gambler, being faced with several slot machines, has to decide which machines to play, how many times and in which order to play each machine. In this project, we consider a multi-agent MAB scenario in which a group of agents connected through a social network are engaged in playing a stochastic MAB game and focus on minimizing their regret. Every time a player takes an action, the reward is observed by both the agent and its neighbors in the network. The goal of this project is to understand different collaborative policies (NAIC-type and FYL) described in [1], and to compare their performance over different network structures.

1 Introduction

This project is a review of paper [1]. As such, we will only succinctly present their model and their main results, and refer to additional results in [1] when necessary.

The interesting part of bandit networks is that we consider multiple agents playing the same multi-armed bandit (MAB) problem. Instead of playing independently, they can be connected through a graph and share their knowledge of the game (actions taken and resulting rewards) with their neighbors. Moreover, they can decide to play very diverse policies, which can be much more than simple extensions of UCB or Thomson-Sampling to the network-setting. These collaborative policies can really profit of the structure of the graph considered and even adapt to the locally available information.

2 The model

We use the definitions and notations from [1]. Let's restate them quickly.

2.1 Single-agent stochastic multi-armed bandit (MAB) problem

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of arms available to the agent. Each arm is associated with a distribution $\mathcal{P}_k$, and let μ_k be the corresponding mean, unknown to the agent. Let n be the time horizon or the total number of rounds. In each round t, the agent chooses an arm, for which he receives a reward, an i.i.d. sample drawn from the chosen arm's distribution. The agent can use the knowledge of the chosen arms and the corresponding rewards up to round $(t-1)$ to select an arm in round t. The goal of the agent is to maximize the cumulative expected reward up to round n.

2.2 Multi-agent stochastic multi-armed bandit (MAB) problem

We consider a set of users V connected by an undirected fixed network $G = (V, E)$, with $|V| = m$. Assume that each user is learning the same stochastic MAB problem i.e., has to choose from the same set of arms $\mathcal{K}$ at each time step. In the tth round, each user v chooses an arm, denoted by $a^v(t) \in \mathcal{K}$,

and receives a reward, denoted by $X_{a^v(t)}^v(t)$, an i.i.d. sample drawn from $\mathcal{P}_{a^v(t)}$. In the stochastic MAB problem setup, for a given user v, the rewards from arm i, denoted by $\{X_i^v(t) : t = 1, 2, \ldots\}$, are i.i.d. across rounds. Moreover, the rewards from distinct arms i and j, $X_i^v(t)$, $X_j^v(s)$, are independent. If multiple users choose the same action in a certain round, then each of them gets an independent reward sample drawn from the chosen arm's distribution. We use the subscripts i, v and t for arms, nodes and time respectively. The information structure available to each user is as follows. A user v can observe the actions and the respective rewards of itself and its one hop neighbors in round t, before deciding the action for round $(t+1)$.

The policy Φ^v followed by a user prescribes actions at each time t, $\Phi^v(t) : H^v(t) \to \mathcal{K}$, where $H^v(t)$ is the information available with the user until round t. A policy of the network G, denoted by Φ, comprises of the policies pertaining to all users in G. The performance of a policy is quantified by a real-valued random variable, called *regret*, defined as follows. The regret incurred by user v for using the policy Φ^v up to round n is defined as,

$$R_\Phi^v(n) = \sum_{t=1}^n \left(\mu^* - \mu_{a^v(t)} \right) = n\mu^* - \sum_{t=1}^n \mu_{a^v(t)},$$

where $a^v(t)$ is the action chosen by the policy Φ^v at time t, and $\mu^* = \max_{1 \le i \le K} \mu_i$. We refer to the arm with the highest expected reward as the optimal arm. The regret of the entire network G under the policy Φ is denoted by $R_\Phi^G(n)$, and is defined as the sum of the regrets of all users in G. The expected regret of the network is given by:

$$\mathbb{E}[R_\Phi^G(n)] = \sum_{v \in V} \sum_{i=1}^K \Delta_i \mathbb{E}[T_i^v(n)], \tag{1}$$

where $\Delta_i = \mu^* - \mu_i$, and $T_i^v(n)$ is the number of times arm i has been chosen by Φ^v up to round n. We omit Φ from the regret notation, whenever the policy can be understood from the context. Our goal is to devise learning policies in order to minimise the expected regret of the network.

Let $\mathcal{N}(v)$ denote the set consisting of the node v and its one-hop neighbours. Let $m_i^v(t)$ be the number of times arm i has been chosen by node v and its one-hop neighbors until round t, and $\hat{\mu}_{m_i^v(t)}$ be the average of the corresponding reward samples. These are given as:

$$m_i^v(t) = \sum_{u \in \mathcal{N}(v)} T_i^u(t)$$

$$\hat{\mu}_{m_i^v(t)} = \frac{1}{m_i^v(t)} \sum_{u \in \mathcal{N}(v)} \sum_{k=1}^t X_{a^u(k)}^u(k) \mathbb{I}\{a^u(k) = i\},$$

where $\mathbb{I}$ denotes the indicator function. We use $m_i^G(t)$ to denote the number of times arm i has been chosen by all nodes in the network until round t.

3 Policies

3.1 UCB-Network

[1] proposes the natural extension of the well-known single agent policy UCB to the network-setting, that they call "UCB-user". When each user in the network follows the UCB-user policy, the network policy is called UCB-Network and is outlined in Algorithm 1.

The following theorem presents an upper bound on the expected regret of a generic network, under the UCB-Network policy.

Theorem 1. *Assume that the network G follows the UCB-Network policy to learn a stochastic MAB problem with K arms. Further, assume that the rewards lie in $[0, 1]$. Then,*

(i) The expected total regret of G is upper bounded as:

$$\mathbb{E}\left[R^G(n)\right] \le b + \sum_{i : \mu_i < \mu^*} C_G \frac{8 \ln n}{\Delta_i},$$

2

Algorithm 1 Upper-Confidence-Bound-Network (UCB-Network)

Each user in G follows UCB-user policy
UCB-user policy for a user v:
Initialization: For $1 \leq t \leq K$
- play arm t
Loop: For $K \leq t \leq n$
- $a^v(t+1) = \underset{j}{\arg\max} \ \hat{\mu}_{m_j^v(t)} + \sqrt{\frac{2 \ln t}{m_j^v(t)}}$

---

where $\Delta_i = \mu^* - \mu_i$, $\beta \in (0.25, 1)$, $b = m \left[\frac{2}{4\beta - 1} + \frac{2}{(4\beta - 1)^2 \ln(1/\beta)} \right] \left(\sum_{j=1}^{K} \Delta_j \right)$, and

C_G is a network dependent parameter, defined as follows.

(ii) Let $\gamma_k = \min\{ t \in \{1, \ldots, n\} : |\{ v \in V : m_i^v(t) \geq l_i = \frac{8 \ln n}{\Delta_i^2} \}| \geq k \}$ denote the smallest time index when at least k nodes have access to at least l_i samples of arm i. Let η_k be the index of the 'latest' node to acquire l_i samples of arm i at γ_k, such that $\eta_k \neq \eta_{k'}$ for $1 \leq k, k' \leq m$. Define $z_k = T_i(\gamma_k) := \left(T_i^1(\gamma_k), \ldots, T_i^m(\gamma_k) \right)$, which contains the arm i counts of all nodes at time γ_k. Then, $C_G l_i$ is the solution of the following optimisation problem:

$$
\begin{aligned}
\max \quad & \|z_m\|_1 \\
s.t \ & \exists \text{ a sequence } \{z_k\}_{k=1}^m \\
& z_j(\eta_k) = z_k(\eta_k) \ \ \forall j \geq k \\
& \langle z_k, A(\eta_k, :) \rangle \geq l_i, \ \ 1 \leq k \leq m
\end{aligned}
\tag{2}
$$

Remark 1. *We propose the result in red, which is a correction of the result from [1]. After checking the proof, we think that we can deduce this tighter upper bound on the expected total regret of G.*

Corollary 1. *For an m-node star network:*

$$
\mathbb{E}[R^G(n)] \leq b + (m - 1) \sum_{i : \mu_i < \mu^*} \frac{8 \ln n}{\Delta_i}
\tag{3}
$$

3.2 Follow Your Leader (FYL)

They also present a second network policy called Follow Your Leader (FYL) for a generic m-node network. The policy is based on exploiting high-degree hubs in the graph; for this purpose, we need to define notions of dominating sets and dominating set partitions.

Definition 3 [Dominating set of a graph] A *dominating set* D of a graph $G = (V, E)$ is a subset of V such that every node in $V \setminus D$ is adjacent to atleast one of the nodes in D.

Definition 4 [Dominating set partition of a graph] Let D be a dominating set of G. A dominating set partition based on D is obtained by partitioning V into $|D|$ components such that each component contains a node in D and a subset of its one hop neighbors.

The FYL policy for an m-node generic network is outlined in Algorithm 2. Under the FYL policy, all nodes in the dominating set are called *leaders* and all other nodes as *followers*; the follower nodes follow their leaders while choosing an action in a round. The following theorem presents an upper bound on the expected regret of an m-node star network which employs the FYL policy.

Theorem 2 (FYL regret bound, star networks)**.** *Suppose the star network G with a dominating set as the center node, follows the FYL policy to learn a stochastic MAB problem with K arms. Assume that the rewards lie in $[0, 1]$. Then,*

$$
\mathbb{E}[R^G(n)] \leq d + \sum_{i : \mu_i < \mu^*}^{K} \frac{8 \ln n}{\Delta_i},
$$

where $d = \left[2m - 1 + \frac{2m}{4\beta - 1} \left(1 + \frac{1}{(4\beta - 1) \ln(1/\beta)} \right) \right] \left(\sum_{j=1}^{K} \Delta_j \right)$, $\Delta_i = \mu^* - \mu_i$ and $\beta \in (0.25, 1)$.

Algorithm 2 Follow Your Leader (FYL) Policy

Input: Graph G, a dominating set D and a dominating set partition
Leader - Each node in D:
Follows the UCB-user policy by using the samples of itself and its neighbors
Follower - Each node in $V \setminus D$:
In round $t = 1$:
    - Chooses an action randomly from $\mathcal{K}$
In round $t > 1$:
    - Chooses the action taken by the leader in its component, in the previous round $(t - 1)$

A key insight obtained from Theorem 2 is that an m-node star network with the FYL policy incurs an expected regret that is lower by a factor $(m - 1)$, as compared to UCB-Network.

3.3 Lower bounds on the expected regret

We decided not to focus on the lower bounds on the expected regret during this review, but it's important to notice that they prove in [1] that the regret upper bound for star networks under the FYL policy meets a lower bound universal to all policies. Hence, they conclude that the FYL policy is order optimal for star networks.

4 Our theoretical contributions

During our review of paper [1], we thought it would be interesting to elaborate on some proofs, and that we could improve the proposed policies. In this section, we present these theoretical contributions to the multi-agent stochastic multi-armed bandit (MAB) problem.

4.1 Upper bound of the UCB-network policy on a star graph

We want to prove Corollary 1 of [1] using Theorem 1. Therefore we need to prove that for a m-node star network G, $C_G = m - 1$. In order to do that we set $l_i \in \mathbb{N}$ and we need to solve the optimization problem (2). As said in [1], the maximum of the optimization program happens when every leaf plays the sub-optimal arm i l_i times, and the central node never plays it.

To prove it we can see that z_m being equal to l_i for every leaf and 0 for the center node is admissible, and in that case the permutation η_k doesn't matter. Moreover, if z_m for the central node is equal to $L > 0$, then z_m can never be more than $l_i - L$ for the leaves, because when an node has l_i samples of the sub-optimal arm, it stops playing it. Therefore, we would have,

$$\|z_m\|_1 = L + (m - 1)(l_i - L) < (m - 1)l_i$$

4.2 Upper bound on the expected regret for UCB-Network policy on a multiple stars graph

Suppose that you have the following m-nodes graph, that we call multiple stars graph. There are S stars on the graph. Each star j is made of a hub and its m_j children. The only connections between stars are made through their hub (central node). These connections can be can be anything ranging from full disconnection to full connection of the hubs.

We want to study this type of graph, because we think that it is an interesting generalization of star graphs. Moreover, we will use this type of graph to compare FYL and a new policy, FBI, that we propose in the next section.

Proposition 1 (Upper bound on the expected regret for UCB-Network policy on a multiple stars graph). *Assume that a multiple stars graph G as defined above, follows a UCB-Network policy to learn a stochastic MAB problem with K arms. Further, assume that rewards are in $[0, 1]$. Then, regardless of the connections between the central nodes of the stars, the expected total regret of G is upper bounded as :*

$$\mathbb{E}(R^G(n)) \leq (m - S) \sum_{i:\mu_i < \mu^*} \frac{8 \ln(n)}{\Delta_i} + b$$

4

where $\Delta_i = \mu^* - \mu_i$, $\beta \in (0.25, 1)$, $b = m \left[\frac{2}{4\beta - 1} + \frac{2}{(4\beta - 1)^2 \ln(1/\beta)} \right] \left(\sum_{j=1}^{K} \Delta_j \right)$.

Remark 2. *This result can be seen as an extension of Corollary 1 when $S = 1$.*

Proof. We use Theorem 1 of [1] to get the general result, and we only need to prove that for our multiple stars network G, we have that $C_G = \sum_{j=1}^{m} m_j = m - S$. For that, we need to solve the optimization program (2) of Theorem 1 of [1]. For our network, the formulation is that $C_G l_i$ is the solution of the following optimisation problem where A is the adjacency matrix of our graph.

$$
\begin{aligned}
\underset{x}{\text{maximize}} \quad & \|z_m\|_1 \\
\text{subject to} \quad & \exists \text{ a sequence } \{z_k\}_{k=1}^{m} \in \{0, l_i\}^m \\
& \exists \text{ a permutation } \{\eta_k\}_{k=1}^{m} \text{ of } (1, ..., m) \\
& z_l(\eta_k) = z_k(\eta_k) \qquad\qquad\qquad \forall l \geq k \\
& \langle z_k, A(\eta_k, :) \rangle \geq l_i \qquad\qquad\quad\; \forall 1 \leq k \leq m
\end{aligned}
$$

For a multiple-stars network, the solution of this program is $l_i \sum_{j=1}^{n} m_i$. This corresponds to the scenario where none of the stars centers ever chooses the sub-optimal arm i, and every leaf of every star chooses it l_i times. It is clearly feasible because, we can have z_m be 0 for all the centers of the stars, and l_i for all the leaves. Then the order of the permutation η_k doesn't matter.

Moreover, if the centers of the stars are fully disconnected, we know that $l_i \sum_{j=1}^{S} m_j$ is indeed the maximum of the program because we can solve it star by star as they don't interact. When there is a connection between the centers, there is more information going through the graph, and therefore, the maximum can only be lower than $l_i \sum_{j=1}^{S} m_j$.

Therefore, the maximum is indeed $l_i \sum_{j=1}^{S} m_j$, and we have $C_G = \sum_{j=1}^{S} m_j = m - S$.

$\square$

4.3 Follow Best Informed (FBI) policy

For most types of graphs, the FYL policy is an improvement of the UCB-network policy. For example, for a star graph, the FYL policy makes perfect sense. However, as the graphs become more and more complex, FYL suffers from the issue that a node may want to follow another node that is not directly its neighbor because its leader may not be very informed, and another node not "too far" may have much better information that could be conveyed to the original node. Moreover, finding the optimal leaders set on a graph may not be easy. To solve these issue, we introduce a new policy that we call the Follow Best Informed (FBI) policy.

4.3.1 Description of the FBI policy

When a node performs an action, it will output the action and the reward just like in FYL, but it will also output the number of samples the action was decided upon. For example, if the node chooses an action at random, then its action was based of 0 samples. If it uses an UCB-user policy, then the action was based on the time passed times the number of its neighbors plus 1 (itself), because it has access and uses that number of samples. Finally, if the node follows an action of one of its neighbors, then this action is based on as many samples as the original action from the neighbor node was based on.

At each step, a node of a network following the FBI policy will look only at its neighbors. It will find the one that performed an action during the previous round based on the most information, i.e. based on the most samples. If the action it performed is based on more samples than what the current node has access to, then it will replicate the action. If it was based on the same number of samples, or less samples, the current node will perform an action using a UCB policy based on its own samples it collected previously. It is better to decide by oneself what action to perform when one has the same amount of information as its neighbors, because that will reduce the variance of the overall reward.

Note that the policy of a follower node is not individually consistent. Therefore, we actually need to make sure that some nodes perform the UCB policy, which will be propagated through the graph.That will always happen because there is at least one node in a graph that has access to equal or higher information than any other node in the graph.

4.3.2 FBI answers the same problem as FYL, and doesn't need further hypotheses

When the FBI policy is described as above, it seems that the nodes need to do additional work than in the FYL policy, because nodes have to output an additional parameter when they performed an action. However, the additional parameter they output is in fact deterministic for all runs, and only depends on the structure of the graph. Whatever randomness happens, a node will always choose an action the same way (random choice, UCB, or follow a neighbor) because it always has access to the same number of information, and so do its neighbors. Therefore, the only thing required to apply our policy is to know the structure of the graph. This is also needed in the FYL policy because we need to determine the leader set beforehand. Therefore we don't need to make further hypotheses to implement the FBI policy.

4.3.3 Usefulness of the FBI policy

The first improvement of **the FBI policy is that it doesn't need a set of leaders as input**. As said in [1], finding a leaders set is a NP-hard problem and the solution output by the FYL policy greatly depends on this leaders set. That means that the FBI policy will be much easier to implement on a complex graph.

Moreover, on a star graph FYL with optimal leaders set and FBI will perform the same way. On more complicated graphs like multiple stars network, or a star network with a chain; the FBI policy will perform better than FLY policy, sometimes significantly. The experimental results we got are shown in section 5.2.

4.3.4 Further improvement of FBI

The FBI policy only looks at its most informed neighbor. However, it could happen that a node has multiple neighbors informed about in the same way. If the best informed plays an action different from the rest, then there may be a high probability that the most informed neighbor is wrong. Therefore, an improvement of FBI could be to choose the action of a neighbor with a certain probability depending on the number of sample it based its last action on. Actually, the probability would probably depend on the square root of the number of samples because confidence bounds work with square roots of number of samples. That way, we think we could again improve the FBI policy.

5 Experimental results

5.1 Reproducing experiments from [1]

Considering this is a review project, we decided to first reproduce the experimental results from the paper.

5.1.1 Experiment 1

Performance comparison of UCB-Network policy on various 10 node networks: 2 arms, Bernoulli rewards with means 0.5 and 0.7.
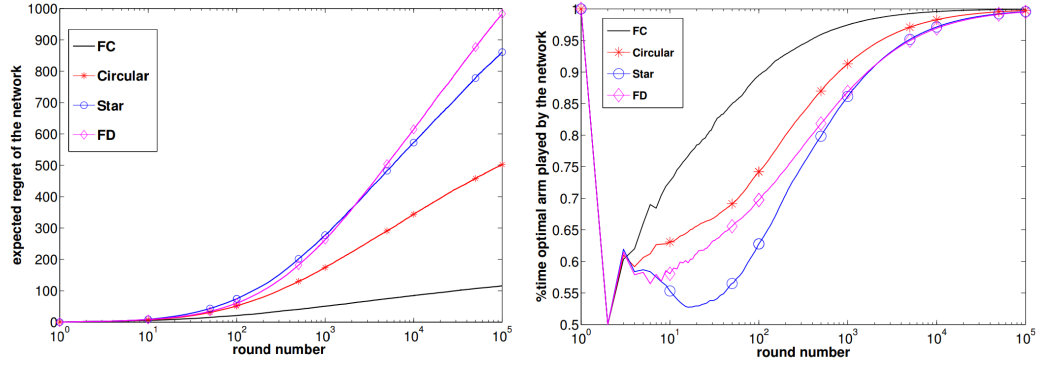
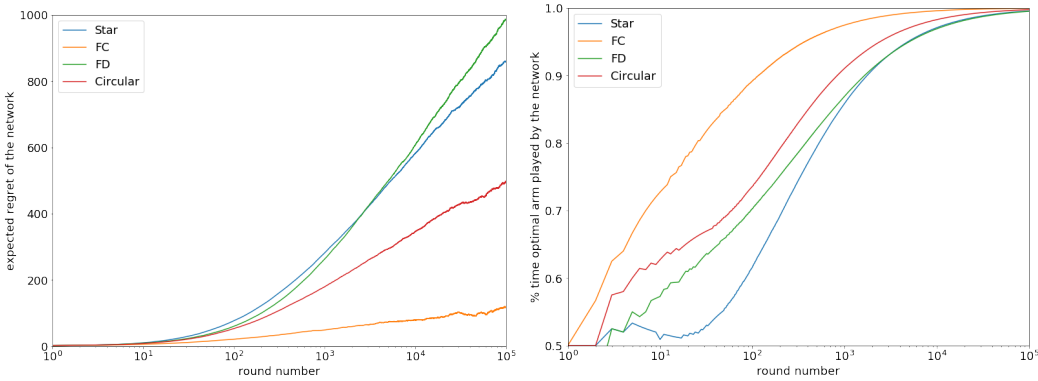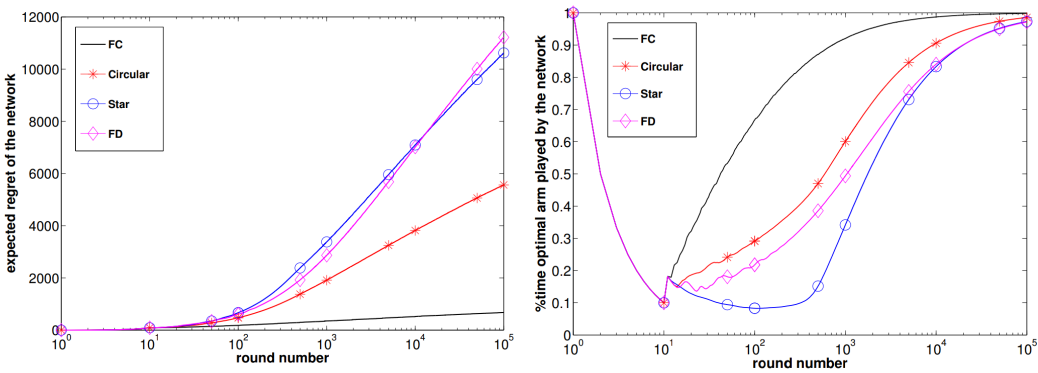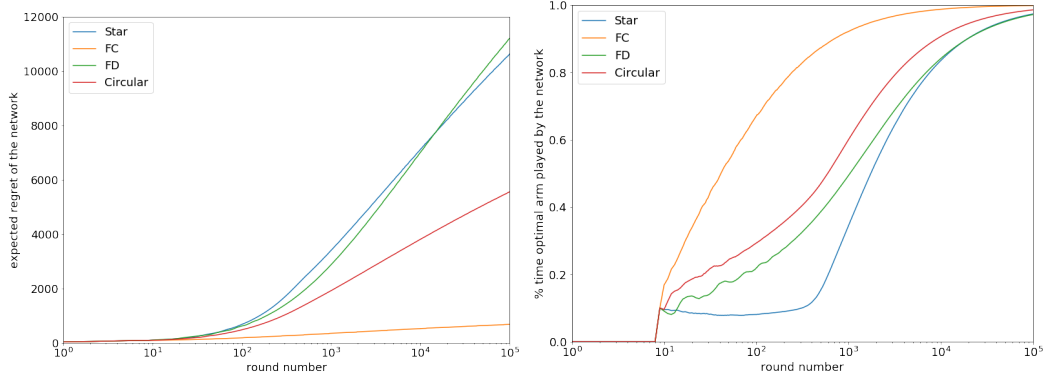Figure 1: Results for experiment 1 from [1] (100 sample paths).

Figure 2: Our results for experiment 1 (1000 sample paths).

We observe that we obtain exactly the same results as in [1], except for the expected regret of the network at rounds > 10000. We get a noisier regret, even though we increased the number of simulations up to 1000 sample paths. We think that the authors of [1] decided to smooth their curves.

5.1.2 Experiment 2

Performance comparison of UCB-Network policy on various 20 node networks: 10 arms, Bernoulli rewards with means $0.1, 0.2, \dots, 1$.

Figure 3: Results for experiment 2 from [1] (100 sample paths).

Figure 4: Our results for experiment 2 (1000 sample paths).

Here, we obtain the exact same results as in [1].

5.1.3 Experiment 3

Performance comparison of UCB-Network and FYL policies on various star networks: 2 arms, Bernoulli rewards with means 0.5 and 0.7.
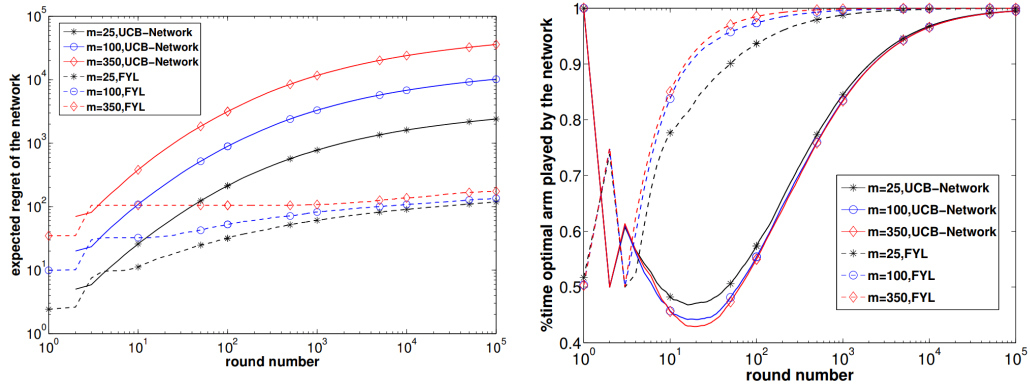

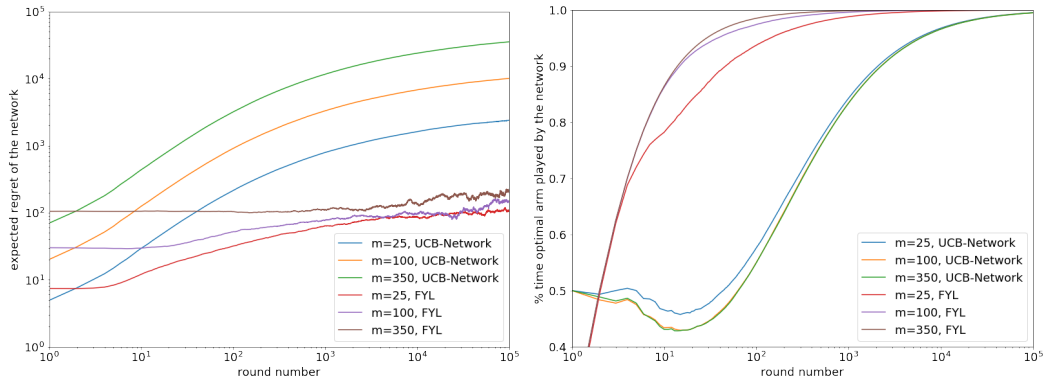Figure 5: Results for experiment 3 from [1] (100 sample paths).

Figure 6: Our results for experiment 3 (1000 sample paths).

Same observation here as for experiment 1: we notice that we obtain exactly the same results as in [1], except for the expected regret of the network at rounds > 10000. We get a noisier regret, even

8

though we increased the number of simulations up to 1000 sample paths. We think that the authors of [1] decided to smooth their curves.

5.2   Evaluating Follow Best Informed (FBI) policy

5.2.1   Comparison between star network and multiple stars network

Finally, we provide experimental results to demonstrate the efficiency our Follow Best Informed (FBI) policy.

We show that it provides similar results as FYL for star networks, and improves the performance for general networks (here, multiple fully connected stars network, see Figure 7).
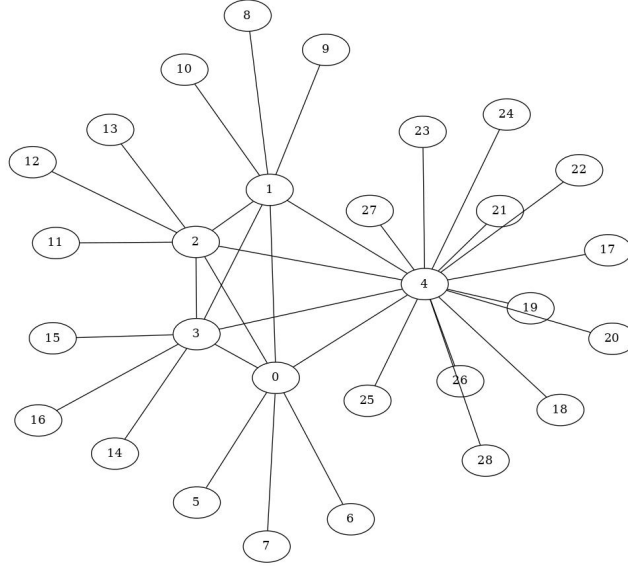

Figure 7: A 29-nodes 5-FC-stars network. It can be seen as a star network where the central hub got split into several sub-hubs.
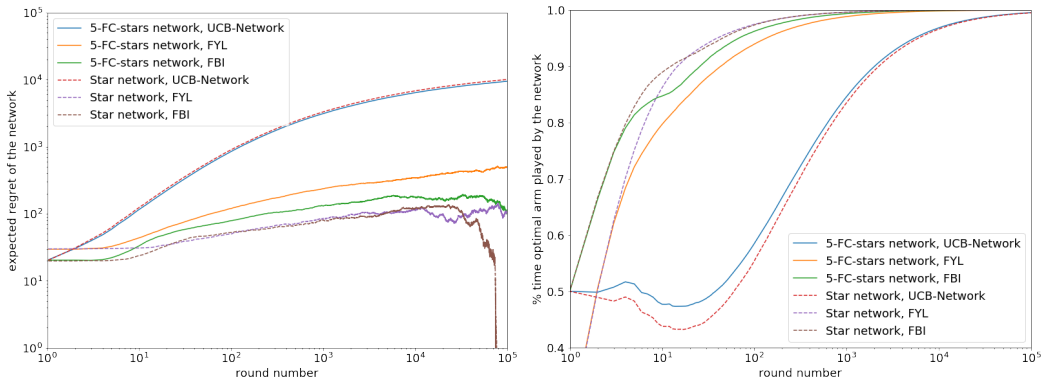

Figure 8: Performance comparison of UCB-Network, FYL, and FBI policies on a 100-nodes star network and on the 100-nodes 5-FC-stars network: 2 arms, Bernoulli rewards with means 0.5 and 0.7 (1000 sample paths).

As expected, FYL and FBI policies perform about the same on star and multiple stars networks. But the great improvement of FBI over FYL lies in its ability to naturally adapt to the graph structure. It greatly decreases expected regret for the multiple stars network, compared to FYL.

9

5.2.2 FYL can be really expensive on general graphs

This gap is even larger on more pathological graph structure. Here, we compare FYL and FBI on this kind of graphs:
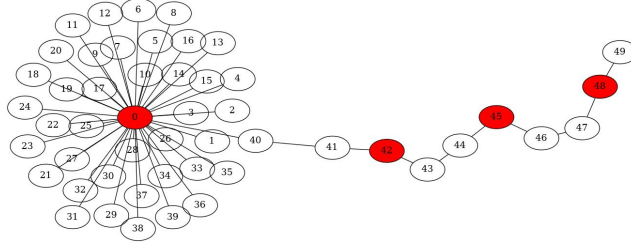

Figure 9: A pathological graph structure. The minimum dominating set is indicated in red.
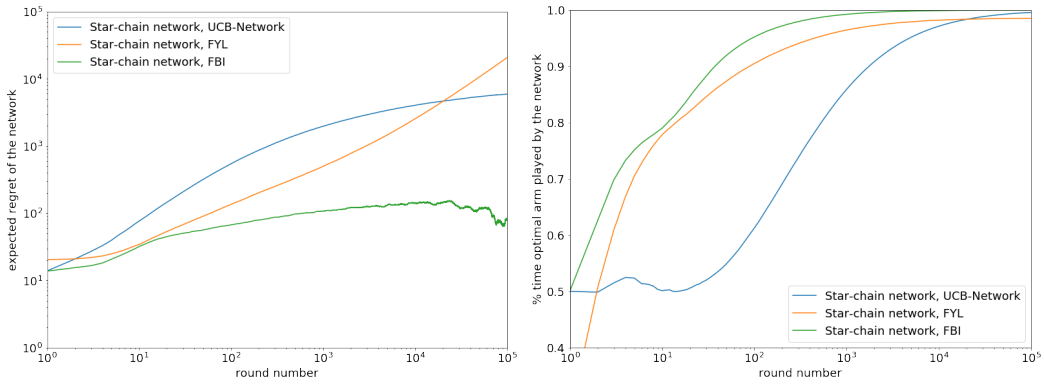

Figure 10: Performance comparison of FYL and FBI policies on the pathological graph structure (star graph with 70 nodes, among which a 20-nodes long chain): 2 arms, Bernoulli rewards with means 0.5 and 0.7 (1000 sample paths).

One issue of FYL is that it doesn't really take into account the full structure of the graph, and can only benefit from star structures. The more general the graph, the worst the policy. This comes from the fact that FYL policy relies on the dominating set (see upper bound of Theorem 6 from [1]). In our example, the dominating set doesn't allow children nodes to collect samples efficiently, so FBI performs much better than FYL.

6 Conclusion

In this report we reviewed article [1]. We understood the proofs and reproduced there experimental results. We indeed came to the same conclusions such as the UCB-Network policy not taking advantage of the structure of the graph, and being largely beaten by the FYL policy on star graphs.

Moreover, we added our own theoretical contribution. First, we computed the upper bound for UCB-Network policies on multiple stars graphs. Second, we identified weak spots of the FYL policy and imagined a new policy, the FBI policy that fills the gaps. We then experimented and found out that the FBI policy was indeed better than the FYL policy on complex graphs such as multiple stars graphs or a star graph with a chain. Besides, the FBI policy does not need a leaders set as input, which is a big improvement compared to the FYL policy as finding the optimal leaders set is a NP-hard problem.

References

[1] Ravi Kumar Kolla, Krishna P. Jagannathan, and Aditya Gopalan. Stochastic bandits on a social network: Collaborative learning with local information sharing. *CoRR*, abs/1602.08886, 2016.