

# Bandits on stochastic block-model graphs

Arthur Mensch, Michaël Weiss

January 8, 2015

## 1 Introduction

We consider an adversarial multi-arm bandit problem with *side information*. Such a problem, located between *full-information* problem (at each round, the player observe the loss of every arm), and the *bandit problem* (at each round, the player observe only the loss for the arm he choose). It can model several real-life situations, an important one being preference observation on social networks : feeding a user with a given content, we can observe its feedback (*retweet* on Twitter, *like* on Facebook), along with the feedback of some of his friends virally fed with the same content.

The *bandit* problem can be addressed most efficiently using the well-know EXP3 algorithm, described in [2]. Very recently, [1] has proposed to extend this algorithm to the *side-information problem*, where feedback distribution is modeled using Erdos-Renyi random graphs. Most importantly, this article focuses on ways to estimate loss distribution over arms, not knowing the underlying edge-revelation probability. It draws upper-bounds on maximum expected regret, using methods inspired from geometrical resampling, present in [3].

Erdos-Renyi graph are too simple to successfully model social networks, as population clusters leads to non-uniform connection probabilities. More complex models has been proposed for this purpose, including stochastic block models (SBM), which extends Erdos-Renyi to graphs with different clusters. We propose an algorithm that extends DUPLEXP3 from [1] to stochastic block models, where bandit cluster label is known. We empirically show that our algorithm outperform [1] algorithm on any stochastic blockmodel graph with more than 2 clusters.

## 2 Exp3 on Erdős-Rényi Graph

We first recall [1] settings, and describe the algorithm adapted to Erdős-Renyi graphs, and its underlying principles. The problem is reduced to the situation where all non-chosen arms reveal their loss with unknown probability  $r_{11} = r$ . This can be modeled by Erdős-Rényi random graphs with parameter  $r$ : at each step  $t$ , for all  $(i, j)$  node pairs, we construct an edge between  $i$  and  $j$  with the probability  $r$ ; we observe loss from the neighbors of chosen arm  $I_t$ .

Erdős-Rényi graph models interaction between a fully-connected group of people, where content is shared with probability  $r$ . Although it seems very simple, it already requires subtle adaptation of vanilla EXP3 algorithm for the player to benefit from side-observation.

### 2.1 Problem Definition

We consider a sequential set on interactions with the multi-armed bandit we assume to have  $N$  arms, for each step  $t = 1, \dots, T$  these are the actions performed by the environment :

1. The environment chooses losses for every arm noted  $l_{t,i}$  for the arm  $i$  at the step  $t$ .
2. Following the algorithm we hope would minimize as much as possible the regret the player draws an arm  $I_t$ .
3. The player receives the loss  $l_{t,I_t}$ .
4. We define  $(O_t)_{i \in [N]}$  as the indicative function of observed loss at step  $t$ . We have:

$$O_{t,I_t} = 1 \quad \forall i \neq I_t, O_{t,i} \sim B(r)$$

$(O_t)_{i \in [N]}$  corresponds to the value of the logic expression *i is neighbor of  $I_t$*  in the Erdős-Rényi random graph drawn at step  $t$ .

5. For all  $i$  such that  $O_{t,i} = 1$  the player can observe the loss  $l_{t,i}$ .

We write  $p_{t,i} = \mathbb{P}[I_t = i | \mathcal{F}_{t-1}]$  where  $\mathcal{F}_{t-1}$  corresponds to all the actions and observations the player had until the step  $t$ . Then intuitively the probability of observing the loss of the arm  $i$  at the step  $t$  would be  $q_{t,i} = p_{t,i} + (1 - p_{t,i})r$  and to use the EXP3. Then the loss estimate would be :

$$\hat{l}_{t,i} = \frac{O_{t,i} l_{t,i}}{q_{t,i}}.$$

But the main problem resides in the fact that  $r$  is unknown so the algorithms presented use tricks to obtain loss estimates such that we keep the property :

$$\mathbb{E}[\hat{l}_{t,i} | \mathcal{F}_{t-1}] = l_{t,i}$$

The principal idea is to have access to two **independent** geometrically distributed random variables  $M_t^*$  and  $K_{t,i}$  with respective parameters  $r$  and  $p_{t,i}$ , then the variable  $G_{t,i}^* = \min\{K_{t,i}, M_t^*\}$  is also geometrically distributed with the parameter  $q_{t,i}$  previously defined. Then if we have  $G_{t,i}^*$  **independent** of  $O_{t,i}$  we can replace in the definition of  $\hat{l}_{t,i}$ ,  $\frac{1}{q_{t,i}}$  by  $G_{t,i}^*$ .

## 2.2 DuplExp3 for large values of $r$

We assume  $r \geq \frac{\log(T)}{2N}$ , which implies that the probability of having no additional observations in round  $t$  is bounded by  $\frac{1}{\sqrt{T}}$ .

This algorithm needs two EXP3 sub-algorithms, with learning rates  $(\eta_t)$ . one for the round when  $t$  is even and the other one for the rest so that we can construct independent  $M_t^*$  and  $K_{t,i}$  and independent  $G_{t,i}^*$  and  $O_{t,i}$ . For each  $t$ , the algorithm draws:

$$p_{t+2,i} \propto w_{t+2,i} = \frac{1}{N} \exp\left(-\eta_{t+2} \hat{L}_{t,i}\right)$$

Where  $\hat{L}_{t,i} = \sum_{k=0}^{t/2} \hat{l}_{t-2k,i}$  the cumulative sum of the loss estimates for the arm  $i$  for one of the EXP3 sub-algorithms.

$M_t^*$ , truncated geometrical variable of parameters  $r$  is constructed as such : For all  $i = 1, \dots, N-1$ , we define  $O'_{t,i}$  as:

$$\begin{aligned} \forall i < I_t \quad O'_{t,i} &= O_{t,i} & \forall N \geq i > I_t \quad O'_{t,i-1} &= O_{t,i} \\ M_t^* &= \min\{1 \leq i < N : O'_{t-1,i} = 1\} \cup \{N\} \end{aligned}$$

We also define  $K_{t,i}$  as a geometric random variable with parameter  $p_{t,i}$  computed at the step  $t-2$ . The since  $M_t$  depends of  $O_{t-1}$  and  $p_{t,i}$  of  $(O_k)_{k \leq t-2}$  they are obviously independent. That's why we can consider :

$$G_{t,i} = \min(K_{t,i}, M_t)$$

with  $G_{t,i}$  independent of  $O_{t,i}$ .  $G_{t,i}$  follows a geometrical law of parameter  $p_{t,i} + (1 - p_{t,i})r$ .

We set the loss estimate as:

$$\hat{l}_{t,i} = G_{t,i} O_{t,i} l_{t,i}$$

which, taking the expectation, yields:

$$\begin{aligned} \mathbb{E} \hat{l}_{t,i} &= \mathbb{E} G_{t,i} \mathbb{E} O_{t,i} \mathbb{E} l_{t,i} \\ &= \frac{1}{p_{t,i} + (1 - p_{t,i})r} (p_{t,i} + (1 - p_{t,i})r) \mathbb{E} l_{t,i} \\ &= \mathbb{E} l_{t,i} \end{aligned}$$

which is unbiased estimator of  $l_{t,i}$ , as independence allows us to separate expectation. Setting  $\eta_t =$ , the following upper-bound on the regret can be drawn, using unbiased estimator  $\hat{l}$ :

$$R_T \leq 4 \sqrt{\left(\frac{T}{r} + N^2\right) \log N} + \sqrt{T}$$

### 2.3 Lower-bounding $r$

However, a problem remains in this algorithm ; since we don't know a priori what is the value of  $r$ , we can't ensure that  $r \geq \frac{\log(T)}{2N}$  as did the assumption previously. So we need to find a lower bound on  $r$  to know in which case we probably are. The algorithm ESTIMATE\_R returns the argument  $\underline{r}$  with the following properties :

$$\begin{aligned} \mathbb{P}[\underline{r} \leq r] &\geq 1 - \frac{1}{\sqrt{T}} \\ \mathbb{P}[\underline{r} = 0] &= 1 - \frac{1}{\sqrt{T}}, \quad \text{if } r \leq \frac{1}{N} \\ \mathbb{P}[\underline{r} = 0] &\leq \frac{1}{\sqrt{T}}, \quad \text{if } r \geq \frac{2}{N} \end{aligned}$$

Furthermore, let  $\tau$  be the index of the round when ESTIMATE\_R  $\underline{r}$  terminates, then we have :

$$\begin{aligned} \mathbb{E}[\tau] &= \frac{4 \log T}{N} + \sqrt{T} + 1, \quad \text{if } r \leq \frac{1}{N} \\ \mathbb{E}[\tau] &= \log T \left( \frac{4}{N} + \frac{e}{2} \right) + 2, \quad \text{if } r \geq \frac{2}{N} \\ \mathbb{E}[\tau] &= \log T \left( \frac{4}{N} + e \right) + 2, \quad \text{if } r \in [1/N, 2/N] \end{aligned}$$

These properties will allow us to run this algorithm in first before running a learning algorithm without increasing too much the expected regret. In this configuration, then we can decide with low probability error if :

$$\underline{r} \geq \frac{\log T}{N}$$

We then consider that  $r$  is large enough to run the previously algorithm DUPLEXP3 on the graph. We will now see how we should proceed in the other cases.

### 2.4 Generalized algorithm

We now are able to generate a safe lower bound of  $r$ ,  $\underline{r}$ . This is the first step of the algorithm. We already explained that we could resume to DUPLEXP3 at the current step  $\tau$  if we had  $\underline{r} \geq \frac{\log T}{N}$ . We now explain what are the solutions in the other cases.

If  $\underline{r} = 0$  :

We consider that  $r$  is really equal to 0 and run vanilla EXP3 with parameter  $\eta = \sqrt{(2 \log N) / (TN)}$  from the step  $\tau$ .

If  $0 < \underline{r} < \frac{\log T}{N}$  :

Then  $r$  might be too small so we can have side information and so have a biased estimation  $r$  during the steps. In this case, we use a simple trick to reattach to the case of DUPLEXP3 : The new algorithm groups multiple steps together so that it was as if the number of episodes became,

$$J = \left\lceil \frac{T}{A} \right\rceil$$

$$\text{where } A = \left\lceil \frac{\log T}{N\underline{r}} \right\rceil$$

With some adaptations, for instance  $I_t$  should be the same for  $A$  real rounds and we make sure none of the  $O_{t,I_t}$  are counted in the new computation of  $M_j$  - where  $A(j-1) < t \leq Aj$  which are the grouped steps - and leads more accurate estimations of  $\hat{l}_{j,i}$ .

### 3 Exp3 on Stochastic Block-Model Graphs

We consider a multi-armed bandit where the arms are divided up in several classes. Every time the learner chooses an arm, not only does he observe the loss of this arm, but he can also have information about losses of non-chosen arms. We model observation graphs by stochastic block models. We consider that when the learner choose a arm of the class  $i$ , the other arms of the class  $i$  have the probability  $r_{ii}$  to reveal their loss and the other arms of a class  $j$  have the probability  $r_{ij}$ . The problem is characterized by the matrix  $R$  which represents the probabilities of communication between two arms of different classes. In this problem, we assume that the classification of the arm as already been done, so we know the class of each arm and we define  $N_i$ , the size of the cluster  $i$ ,  $i = 1, \dots, n$  where  $n$  represents the number of clusters.

#### 3.1 Problem definition

As in Sec. 1, everything happens as if, at every time step  $t$ , we built a stochastic block-model graph, knowing bandit labels. Such graph can model a variety of group interaction, from dissociation to entanglement and layer organisation. We present an example of stochastic block model in Fig 1, with 5 clusters, cluster  $i$  being closely connected to cluster  $i-1$  and  $i+1$  only on the left, and cluster being closely connected from within on the right.

Notations and definition are mostly the same as in Sec. 1, and will be introduced as we adapt [1] algorithm.

#### 3.2 Algorithm

[1] algorithm aims at computing an unbiased estimator for each  $l_{i,t}$ , in order to yield satisfying upper bounds on maximum expected regret. We follow the same principle, making use of arms' cluster labels. If we had access to every values  $r_{ij}$ , we would be able to define, choosing arm  $I_t$  at round  $t$  :

$$\hat{l}_{t,i}^* = \frac{O_{t,i} l_{t,i}}{p_{t,i} + (1 - p_{t,i}) r_{C(I_t), C(i)}}$$

---

<sup>1</sup>[http://tuvalu.santafe.edu/~aaronc/courses/5352/fall2013/csci5352\\_2013\\_L16.pdf](http://tuvalu.santafe.edu/~aaronc/courses/5352/fall2013/csci5352_2013_L16.pdf)

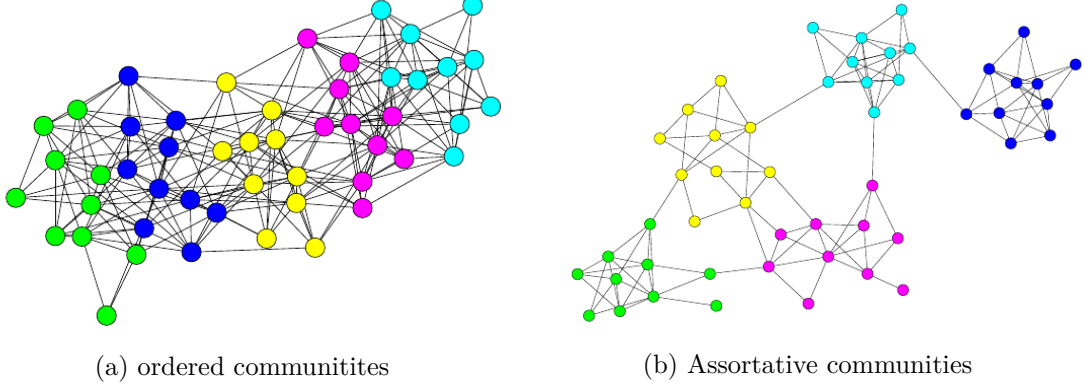


Figure 1: Stochastic block model random graph<sup>1</sup>

where operator  $C$  output the cluster number of an index. Keeping with the original algorithm, we thus want to build an estimator  $G_{i,t}$  that follows a geometrical law of parameter  $p_{t,i} + (1 - p_{t,i})r_{I_t,i}$ . We will rely on the sampling of a geometric variable  $M_t$  of parameter  $r_{C(I_t),C(i)}$ , making sure that it stays independent from  $O_{t,i}$ . That way, setting

$$\hat{l}_{t,i} = G_{t,i} O_{t,i} l_{t,i}$$

we obtain the desired unbiased estimator. The whole difficulty thus rely on the sampling of  $M_t$  so that it stays independent of  $O_t$ .

**$M_t$  sampling :** In order to sample  $M_t$ , we crucially runs two *Exp3* algorithm on even and odd time-step, in order to draw  $p_{t,i}$  from history from all  $t' < t$  of the same parity as  $t$ , and  $M_t$  from  $O_{t'+1}$ , i.e from observation of other parity algorithm. We thus ensure independence of  $M_t$  and  $O_t$ , as  $(O_o)_{o \equiv t[2]}$  and  $(O_e)_{e \equiv t+1[2]}$  are independent (output of two parallel algorithms). To draw a geometric distribution of parameter  $r_{C(I_t),C(i)}$ , we simply find the last  $O_{t'}$ , with  $t' \equiv t - 1[2]$  in which the observed  $I(t')$  was in the same cluster as  $I(t)$  and average the distance between two observation within cluster  $C(j)$ .

### 3.3 Lower-bounding $r_{i,j}$

We now want to provide lower-bounds for every  $r_{ij}$  to adapt the previous algorithm. For every  $r_{ij}$ , we compute  $C_i$  which depends of  $N_i$  the same way  $C$  depended of  $N$  in [1] in the algorithm `ESTIMATE_R`. For the following explanations, we consider variables  $x_{ij}$  which are computed when we draw  $I_t$  with uniform probability in the cluster  $i$  and look at the results in the cluster  $j$ , i.e. we look at the  $O_{t,k}$  where  $k$  is in the cluster  $j$ .

For each cluster  $i$  we compute  $c_{ij}$  associated for each  $j = 1, \dots, n$  allowing us to safely estimate in time  $t \in [0, \max_j(C_j)]$  if  $\underline{r}_{ij} = 0 \ \forall j = 1, \dots, n$ . We repeat this step for all  $i$ , which ensures that  $t \in [0, n \max_j(C_j)]$

The second step needs also to estimate  $\underline{r}_{ij}$  first step has not concluded  $r_{ij} = 0$ . We also perform it for each cluster  $i$  where we compute  $m_{ij}$  and  $M_{ij}[m_{ij}]$ ,  $m = 1, \dots, k, j = 1, \dots, k$  in parallel, the equivalents of  $j$  and  $M[j]$  in [1].

Finally when all end conditions are met we return  $\underline{R}$ . Adapating [1], we should be able to prove that the algorithm provides  $\underline{R}$  in time  $\tau$  such that :

$$\mathbb{E}(\tau) \leq \sum_{i,j \in [N]^2} \frac{4 \log T}{N_i} \mathbf{1}_{r_{ij} > \frac{1}{N_j}} + \sqrt{T} \mathbf{1}_{r_{ij} < \frac{1}{N_j}} + \text{Cste}$$

with the probabilities:

$$\begin{aligned}\mathbb{P}[\underline{r}_{ij} \leq r_{ij}] &\geq 1 - \frac{1}{\sqrt{T}} \\ \mathbb{P}[\underline{r}_{ij} = 0] &= 1 - \frac{1}{\sqrt{T}}, \quad \text{if } r_{ij} \leq \frac{1}{N_j} \\ \mathbb{P}[\underline{r}_{ij} = 0] &\leq \frac{1}{\sqrt{T}}, \quad \text{if } r_{ij} \geq \frac{2}{N_j}\end{aligned}$$

We thus obtain safe lower-bounds in reasonable time (if the number of cluster is kept reasonable).

Computation of  $\underline{R}$  allows us to compute the episode size  $A^*$  that we need in our algoirthm to yield unbiased estimates and allow proof of upper bounds. This can be done considering, at each time step  $t$ ,  $\underline{r}_N^*(t) = \min_{j, \underline{r}_{I(t),j} > 0} \underline{r}_{I(t),j} N_j$  and then using  $A^*(t) = \left\lceil \frac{\log T}{\underline{r}_N^*} \right\rceil$  to group the next round, but would make depend  $A^*$  of  $I_t$ . However, it forces us to update the number of steps we group at each episode, complicating the implementation. To obtain a general  $A$  which would not depend of  $I_j$  we chose at round  $j$  we consider :

$$\begin{aligned}\underline{r}_N^* &= \min_{i,j, \underline{r}_{i,j} > 0} \underline{r}_{i,j} N_j \\ A &= \left\lceil \frac{\log T}{\underline{r}_N^*} \right\rceil\end{aligned}$$

Then we can group  $A$  steps for each rounds and apply the previously described algorithm.

If some  $\underline{r}_{kl} = 0$ , we consider, when we choose  $I(t)$  in cluster  $k$ , that we do not observe any observation in cluster  $l$ . The geometric resampling part of our algorithm in cluster  $l$  is ignored, as we compute

$$\forall j \in \text{Clust}_l \quad \hat{l}_{t,j} = \begin{cases} \frac{1}{p_{t,I(t)}} & \text{if } j = I(t) \\ 0 & \text{otherwise} \end{cases}$$

We thus 'partially' apply the vanilla EXP3 algorithm on some cluster (which depends on the choice  $I(t)$ ). If the whole  $\underline{R}$  matrix is null, we have  $A = 1$  episode size and the last equation is applied in every cluster, which amount to simply apply the vanilla EXP3 algorithm on the multi-arm bandit problem.

## 4 Results

**Algorithm correctness** Though we did not prove the correctness of  $\underline{R}$  estimation, we empirically tested that we always had a lower bound for  $R$ . We computed maximal expected regret along time for different adversary strategy (from totally adversarial (i.e. random) to more structured ones, comparing performance of original DUPLEXP3 algorithm with our stochastic blockmodel tuned algorithm, and with vanilla EXP3 algorithm. We mostly often observed domination of the regret curve for our algorithm, with occasional deviation, which was expected, as *Duplexp3* is an adaption of the non-modified EXP3 algorithm, which gets mistaken with non-zero probability.

**Adversarial settings** We expose performance of our three algorithms in an adversarial settings in Fig. 2, averaged over 50 tries using the same observation set. We observe that our algorithm seem to be upper-bounded, though DUPLEXP3 and EXP3 seem to yield equivalent regret - even on complicated SBM graph.

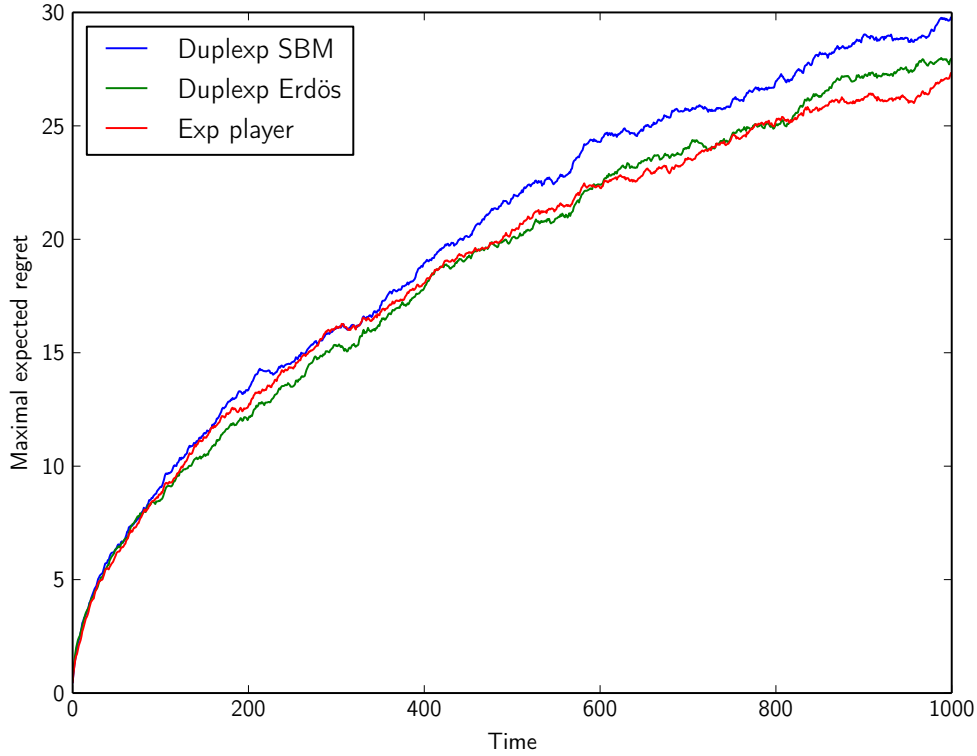


Figure 2: Adversarial setting performance

**Non-adversarial setting** Although our algorithm is designed to allow us to prove upper-bounds that cannot be proved on SBM graph using the original algorithm, we had difficulties observing considerable out-performance of our algorithm on SBM graphs. We observed original DUPLEXP3 seemed to perform as well, if not better, in simple cases (i.e. when the random graph is 'not far' from being an Erdős-Rényi graph). This is partially due to the fact that original DUPLEXP3 tends to find a  $\underline{r}$  larger than  $\min_{\underline{r}_{ij}>0} \underline{r}_{ij}$  and thus a smaller episode size  $A$ . Original algorithm thus learns faster, though theoretically less accurately. However, its accuracy appears to be sufficient on simple problems. We present an example of simple-problem results in Fig. 3, with 4 assortative clusters.

In order to 'trick' the original algorithm, we designed a disassociative graph with 2 clusters, of size 100 and 10, and with rate matrix:

$$\begin{pmatrix} 0.1 & 0.02 \\ 0.02 & 0.08 \end{pmatrix}$$

That way we were able to make original ALGORITHM overestimate the loss in cluster 2, and perform below EXP3. Setting the adversary to yield lesser loss within cluster 2, we observed that our algorithm outperformed the original, as shown in Fig. 4

We should add that, although original algorithm seems to work better in some case, the regret that we observe for SBM algorithm seems to be asymptotically equivalent to the original algorithm regret in any case.

## 5 Conclusion

Adapting [1], we proposed an algorithm that produce satisfying maximal expected regret on labelled stochastic blockmodel random graph, for any connection matrix  $R$ . Its design

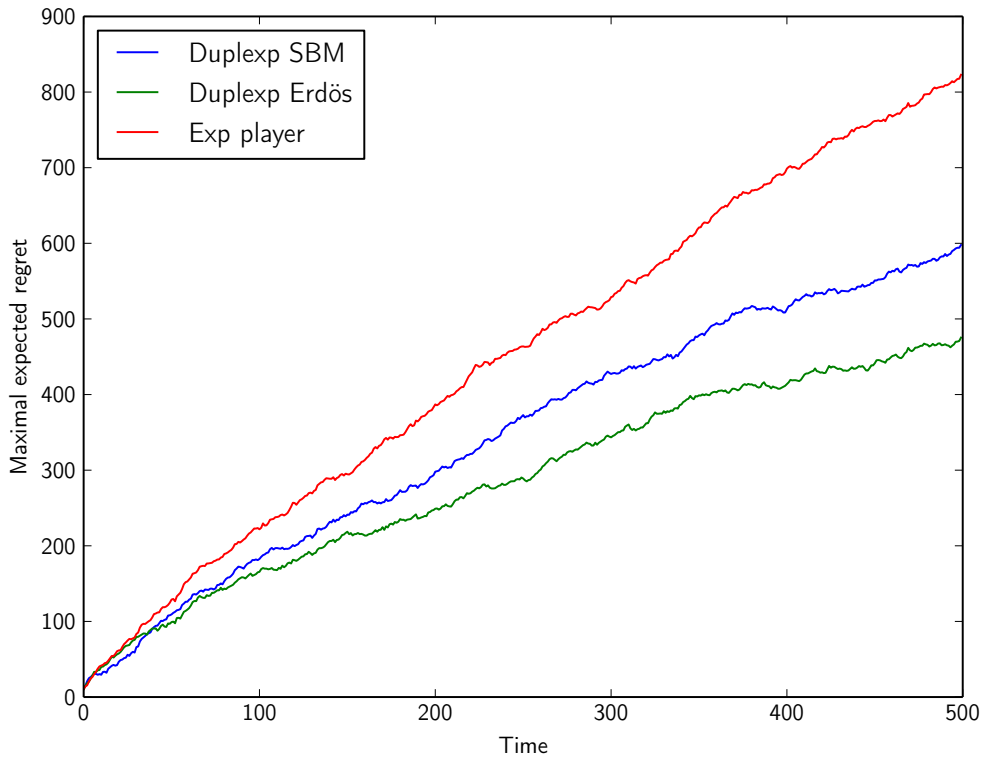


Figure 3: Original algorithm outperforms our algorithm

should allow us to prove the existence of upper-bounds, simply adding cluster indices to [1] proofs. Futur work could include reflexion on a different algorithm for non labelled random graphs : the major issue there would be to find a correct estimation of labels in reasonable time.

## References

- [1] Anonymized. Well-informed bandits: Learning with stochastic side-observations. In *AISTATS*, 2015.
- [2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The non-stochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
- [3] Gergely Neu and Gábor Bartók. An efficient algorithm for learning with semi-bandit feedback. In *Algorithmic Learning Theory*, pages 234–248. Springer, 2013.



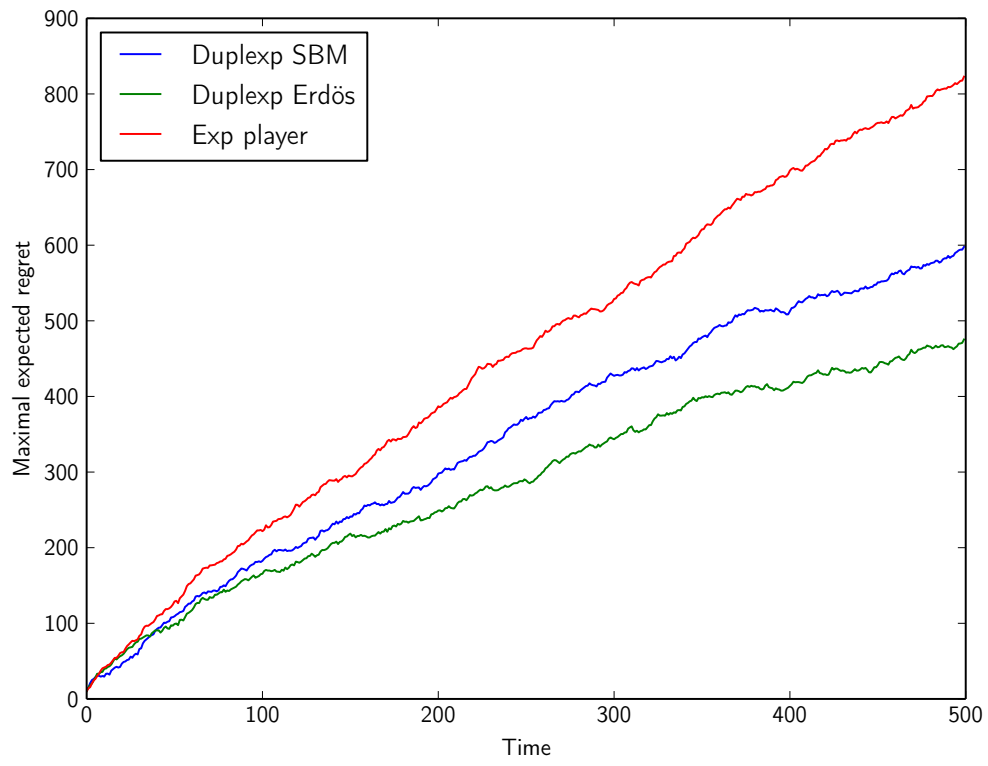


Figure 4: Outperformance of our algorithm