Allocation strategies for the Dark Pool Problem

Hugues Tavenard
hugues.tavenard@eleves.enpc.fr

Sylvain Truong
struong@ens-paris-saclay.fr

Abstract

Dark pools is a particular type of financial market which was introduced to address an issue faced by large-volume traders: buying large volumes of shares might cause prices to rise against the buyer, in traditional (or *light*) markets. Difficulty to conceal trades in such setups might refrain financial markets' dynamism. On the contrary, in *dark pool* setups, traders (buyers or sellers) can submit their orders, with prices only determined by factors other than offer and demand. What can be noted is that no information is given to the trader about the amount of shares available in the pool, thus resulting in *censored* feedback information: if v shares are submitted, the only information is the number $s \leq v$ of executed shares. The maximum possible number of executed shares is not available. When a trader is given an amount V of shares to submit and to split between a certain number of *pools*, he or she is confronted to an *allocation problem*. This *allocation problem* is closely related to the one of Multi-Armed Bandits (MAB), with the difference that, for the basic case of MABs, the player chooses one arm at each time-step, whereas, in the described case, *allocation* to *dark pools* could be considered as continuous weighted arm choices over all arms. The aim of this project is to understand how the *Dark pool* problem can be formalised and how state-of-the-art algorithms address this problem. This project also involves implementation and evaluation of these algorithms. In this report, we investigate two types of algorithms which are referred to as *Kaplan-Meier* and Adapted EXP3. We show, with the support of experiments, that these methods consistently out-beat a bandit-like allocation strategy. We also investigate the reaction of these methods to *pool* parameter changing.

1 Introduction

In the common *dark pool* context, at time t a trader is given a number $V^t \in \{1, \dots, V\}$ sampled from a fixed distribution, of shares (or *units*) that he has to allocate to a fixed number of K *pools* (or *venues*). The reward at time t is the total number of shares that are effectively executed, among the ones that were allocated, over all venues.

1.1 Notations

- t is the current time step
- V is the maximum number of shares to allocate
- V^t is the number of shares to allocate at time step t, drawn at each t from a distribution on $\{1, \dots, V\}$
- $(v_k(t))_{k \in [\![1,K]\!]}$ is the vector of allocation at time t, for all K venues
  (Note that $\sum_{k=1}^{K} v_k(t) = V^t$)
- K is the number of venues
- $s_k(t)$ is the available number of shares at venue k and at time t

Master MVA 2016/17, Semester 1, Reinforcement Learning class project

- $T_k^{(t)}(s) = Pr(s_k(t) \geq s)$ designates the tail probability of available resources (shares) at venue k, at time t

With these notations, the expression of the cumulative reward at time T, for a given allocation strategy $(v_k(t))_{t=0,\ldots,T}$ becomes:

$$R(T) = \sum_{t=0}^{T} \sum_{k=1}^{K} min(v_k(t), s_k(t))$$

1.2 Problem statement

The goal of this class project is to explore and understand different allocation strategies for the *dark pool* problem, as well as implement and test them on toy *dark pool* simulators. This problem is particular because it involves *censored* observations: at each time step t, the only information the player observes is the amount of resources that were effectively consumed at each of the K venues (the number shares that were executed at each pool):

$$\text{Observations } \forall k \in [\llbracket 1, K \rrbracket \,, min(s_k(t), v_k(t))$$

instead of $\forall k \in [1, K]$, $s_k(t)$ in a more conventional context. As in many reinforcement learning problems, the goal of *dark pool* allocation algorithms is to maximise $R(T)$ for a given time horizon T, according to the allocation strategy.

2 State of the art

In this section, we explore the state-of-the-art allocation algorithms, highlight their corresponding assumptions and the way they work.

2.1 Bandit-style allocation strategy

We chose to describe a bandit-style allocation strategy which will be used as a baseline for the other, more sophisticated strategies. The description corresponds to Algorithm 1. This method proposes to set a weight for each of the venues. The allocation is done in accordance with these weights. The weights are updated at each time step, according to the censored observation. The weight of the venue of maximum reward is multiplied by a parameter α. In [Ganchev], α is set to 0.05. We chose to keep this parameter for our tests.

Algorithm 1 Bandit style allocation strategy

```
Input: time horizon $T$, $K$ venues, $\alpha$ parameter
Uniformly initialize venue weights $\forall k \in \llbracket 1, K \rrbracket, w_k = \frac{1}{K}$
for $0 \leq t \leq T$ do
    $V^t \leq V$ is drawn
    Allocate $V^t$ to venues, accordingly with weights
    $\forall k \in \llbracket 1, K \rrbracket, v_k(t) = w_k \cdot V^t$
    Collect reward and observe censored information, per venue $k$
    $r_k(t) = \min(s_k(t), v_k(t))$
    Update venue weights accordingly
    $k_{max} = \arg\max(r_k(t))$
    $w_{k_{max}} \leftarrow \alpha * w_{k_max}$
    Normalize venue weights
end for
```

2.2 Kaplan-Meier strategies

The Kaplan-Meier strategies refer to allocation strategies which are based on estimating the tail probabilities for each of the venues, at each time step. Once the tail probabilities are estimated, the allocation is done *greedily* (Algorithm 2). The description of the Kaplan-Meier method is in Algorithm 3. This approach is rather intuitive. It can proved that it corresponds to the optimal allocation in expectation.

The estimators we will describe for tail probabilities, under the assumption of censored observations are referred to as *Kaplan-Meier* estimators.

For a given venue k, the *Kaplan-Meier* estimators are based on estimating the probability $Pr(s = s^*|s \geq s^*)$ of a demand equal to s^* given that it is at least s^*.

$1 - Pr(s = s^*|s \geq s^*)$ can be interpreted as the probability of there being a demand s of at least $s^* + 1$, given that it was at least s^* ($Pr(s \geq s^* + 1|s \geq s^*)$).

Using the chain-rule, we infer an expression of the tail probability $T(s)$ (with $T(0) = Pr(s' \geq 0) = 1$):

$$T(s) = \prod_{s^*=0}^{s-1} Pr(s' \geq s^* + 1|s \geq s^*)$$
$$T(s) = \prod_{s^*=0}^{s-1} (1 - Pr(s' = s^*|s \geq s^*))$$

Estimates of direct observation probabilities $Pr(s' = s^*|s \geq s^*)$, $\hat{z}_{t,k}(s^*)$ can be easily derived with counting variables $D_{t,k}$ and $N_{t,k}$, at a given time t.

$$D_{t,k}(s) = \sum_{\tau=0}^{t} I(r_k(t) = s, v_k(t) > s)$$
$$N_{t,k}(s) = \sum_{\tau=0}^{t} I(r_k(t) \geq s, v_k(t) > s)$$
$$\hat{z}_{t,k}(s) = \frac{D_{t,k}(s)}{N_{t,k}(s)} \text{ if } N_{t,k}(s) \neq 0\text{, 0 else}$$

The tail probability estimates:

$$\hat{T}_{t,k}(s) = \prod_{s^*=0}^{s-1} (1 - \hat{z}_{t,k}(s^*))$$

The algorithm is described in Algorithm 4.

In [Ganchev], a concern it raised about the possible lack of exploration of the current algorithm. An optimistic version of the *Kaplan-Meier* strategy is described in Algorithm 5.

This optimistic version proposes to increase the tail probability estimates of venues, past *cut-off points*, below which, *Kaplan-Meier* estimates are "accurate", in order to promote exploration.

Algorithm 2 GreedyAllocation

```
Input: number of resources to allocate $V$, tail probability estimates at each venue $\hat{T}_k$
Output: allocation vector $(v_k)$
initialize $v_k = 0, \forall k \in [\![1, K]\!]$
Allocate one by one
for $1 \leq l \leq V$ do
    $k' \leftarrow \arg\max_k(\hat{T}_k(v_k + 1))$
    $v_{k'} \leftarrow v_{k'} + 1$
end for
return $(v_k)$
```

Algorithm 3 Kaplan-Meier strategy

```
  Input: time horizon $T$, $K$ venues
  Arbitrarily initialize tail probabilities estimates at each venue $\hat{T}_k, \forall k \in \llbracket 1, K \rrbracket$
  for $0 \leq t \leq T$ do
      $V^t \leq V$ is drawn
      Allocate $(v_k(t))_{k \in \llbracket 1,K \rrbracket} =$ GreedyAllocation$(V^t, \{\hat{T}_k\}_{k \in \llbracket 1,K \rrbracket})$
      Collect reward and observe censored information, per venue $k$
      $r_k(t) = \min(s_k(t), v_k(t))$
      Update tail probability estimates
      $\forall k \in \llbracket 1, K \rrbracket, \hat{T}_k \leftarrow$ KM-Estimate$(\{(s_k(\tau), v_k(\tau))\}_{\tau=0}^{t})$
  end for
```

Algorithm 4 KM-Normal, at time t

```
  Input: Observations $\forall k \in \llbracket 1, K \rrbracket, \{(s_k(\tau), v_k(\tau))\}_{\tau=0}^{t})$
  Output: $(\hat{T}_k)$, tail probability estimates
  Compute $\forall k \in \llbracket 1, K \rrbracket, \forall s \in \llbracket 0, V+1 \rrbracket$
  $D_{t,k}(s) = \sum_{\tau=0}^{t} I(r_k(t) = s, v_k(t) > s)$ number of direct observations on venue $k$, up to time $t$
  $N_{t,k}(s) = \sum_{\tau=0}^{t} I(r_k(t) \geq s, v_k(t) > s)$
  Compute $\forall k \in \llbracket 1, K \rrbracket, \forall s \in \llbracket 0, V+1 \rrbracket$, probability estimates
  $\hat{z}_{t,k} = \frac{D_{t,k}(s)}{N_{t,k}(s)}$, if $N_{t,k}(s) > 0$
  $\hat{z}_{t,k} = 0$, else
  Kaplan-Meier estimator, $\forall k \in \llbracket 1, K \rrbracket, \forall s \in \llbracket 0, V+1 \rrbracket$
  $\hat{T}_k(s) = \prod_{s'=0}^{s-1}(1 - \hat{z}_{t,k})$
  return $(\hat{T}_k)$
```

2.3 Adapted EXP3 strategy

The EXP3 algorithm presented here was designed in [Agarwal], adapting ideas from the EXP3 algorithm in the bandit problem to the *dark pools* setting.

The algorithm is described in Algorithm 7. The idea is to first compute fractional allocations $v_k(t)$ for each pool k using the weight re-estimation of the original EXP3 algorithm : $x_k^v(t+1) \propto x_k^v(t) exp(\eta \bar{g}_k^v(t))$ where $\bar{g}(t)$ is described in Equation 1 and is (as shown in [Agarwal]) an unbiased estimator of the gradient at $(v_1(t), ..., v_K(t))$.

Once these fractional allocations $v_k(t)$ are computed, we have to compute the integer valued allocations we are actually going to play, since we are only allowed to play integer allocations. Let $f_k(t) = \lfloor v_k(t) \rfloor$ and $d_k(t) = v_k(t) - f_k(t)$. Now, if we allocate $u_k(t) = f_k(t)$ units to venue k with probability $d_k(t)$ and $u_k(t) = f_k(t) + 1$ units with probability $1 - d_k(t)$. It can be shown that $\mathbb{E}\, min(u_k(t), s_k(t)) = min(v_k(t), s_k(t))$ which means that the expected reward that we get when

Algorithm 5 KM-Optimistic, at time t

```
  Input: Observations $\forall k \in \llbracket 1, K \rrbracket, \{(s_k(\tau), v_k(\tau))\}_{\tau=0}^{t})$, parameters $\epsilon > 0, \delta > 0$
  Output: $(\hat{T}_k)$, tail probability estimates
  Kaplan-Meier estimator
  $(\hat{T}_k(s))_{k,s} \leftarrow$ KM-Normal$(\{(s_k(\tau), v_k(\tau))\}_{\tau=0}^{t})\})$

  for $1 \leq k \leq K$ do
      Calculate cut-off
      $c \leftarrow \max\{s | s = 0$ or $N_{t,k}(s-1) \geq 128(sV/\epsilon)^2 ln(2V/\delta)\}$
      Modify optimistically $\hat{T}_k(c+1) \leftarrow \hat{T}_k(c)$
  end for
  return $(\hat{T}_k)$
```

playing these integer valued allocations is the same as the reward we would have obtained when playing the fractional allocations (if it were allowed). In other words, playing these allocations $u_k(t)$ would be unbiased in expectation.

However, the computation of the integral allocations from the fractional allocations is not as straightforward as it may seem. In order to achieve this, we have to be able to sample $m = \sum_{k=1}^{K} d_k(t)$ elements from $\{1, ..., K\}$ such that the probability of observing k in the subset sampled is $d_k(t)$. Although [Agarwal] proves that such a distribution exists, it does not provide the explicit algorithm to compute this distribution, it gives ideas on how to achieve this. Following one of the ideas given in [Agarwal], Algorithm 6 explains how we have implemented this sampling using non-negative least squares.

In order to define the gradient $\bar{g}(t)$ used in the re-estimation step, we first need to define $\tilde{V}_k(t)$ the largest index v_0 such that $\sum_{v=0}^{v_0} x_k^v(t) \leq f_k(t)$:

$$\bar{g}_k^v(t) = \begin{cases} \mathbb{1}(s_k(t) \geq f_k(t)) - \frac{\mathbb{1}(s_k(t)=f_k(t))\mathbb{1}(u(t)_k=\lceil v_k(t)\rceil)}{\bar{d}_k(t)} & \text{if} \qquad v \leq \tilde{V}_k(t) \\ \frac{\mathbb{1}(s_k(t)\geq f_k(t))\mathbb{1}(u_k(t)=\lceil v_k(t)\rceil)}{\bar{d}_k(t)} & \tilde{V}_k(t)+1 \leq v \leq V(t) \end{cases} \tag{1}$$

Algorithm 6 Sampling algorithm of m elements in a set of size K, given the probabilities of appearance using non-negative least squares.

```
Input: $0 \leq d_k < 1$, $\sum_{k=1}^{K} d_k = m$ where $m \geq 1$ is an integer
The goal is to find a distribution over subsets of $\{1, ..., K\}$ of size m such that $k$ is sampled with probability $d_k$.
The proof of the existence of such a distribution is provided in [Agarwal].
Output: $p(s)$, the distribution over the subsets $s$ of $\{1, ..., K\}$
Compute the inputs of the non-negative least squares:
Compute the matrix $M$ of size $(m, \binom{K}{m} - 1)$ that will, in the end verify: $Mp = d$
($M$ is found doing a loop over the subsets of $\{1, ..., K\}$ and checking what elements are in each subset.)
Compute the non-negative least squares on $M$ and $d$ to find p such that $Mp = d$.
return p
```

Algorithm 7 EXP3 algorithm for the dark pool problem

```
Input: learning rate $\eta$, threshold $\gamma$, bound on volumes $V$
Initialize $\forall v \in [\![1, V]\!]$, $x_{1,k}^v = \frac{1}{K}$
for $t = 1...T$ do
    Set $v_k(t) = \sum_{v=1}^{V(t)} x_k^v(t)$
    Set $f_k(t) = \lfloor v_k(t) \rfloor$
    Set $d_k(t) = v_k(t) - f_k(t)$
    Set $m = \sum_{k=1}^{K} d_k(t)$ ($m$ is an integer)
    Set $\bar{d}_k(t) = (1 - \gamma)d_k(t) + \frac{\gamma m}{K}$
    Sample a subset of size $m$ from $[\![1, K]\!]$ according to the distribution in Algorithm 6
    For each venue $k$, play $u_k(t) = f_k(t) + 1$ if $k$ is in the subset sampled and play $u_k(t) = f_k(t)$ otherwise.
    Receive $r_k(t) = min(u_k(t), s_k(t))$
    Set $\bar{g}_k^v(t)$ as defined in Equation 1
    Update $x_k^v(t + 1) \propto x_k^v(t + 1) exp(\eta \bar{g}_k^v(t))$
end for
```

3 Experimental Setup

3.1 On dark pool datasets

Some *dark pool* data sets are available freely online. These data consist of tracebacks of censored observations $\{(s_k(\tau), v_k(\tau))\}_{\tau=0}^{t}$, of a given trader. Because the observations are censored, simulating allocation algorithms directly from these data is quite difficult. For instance, if the algorithm was willing to allocate 100 units to a venue while the actual trader only allocated 10, the actual number of consumed units remains unknown. Thus, to evaluate the performance of the aforementioned strategy, we must derive simulated *dark pools*.

3.2 Simulated dark pools

In [Ganchev], several probability models were fitted to actual *dark pool* data, according to a Maximum Likelihood principle. One main observation from the actual data is that, for a venue, a demand equal to zero is quite frequent. This consideration motivated making candidate demand distribution model combinations with zero-bins. Among the candidate models (ZB + Uniform, ZB + Power Law, ZB + Poisson, ZB + Exponential), ZB + Power Law proved to achieve the best likelihood. For this model, the demand probability model is as follows:

$$\begin{aligned} p(s) &\propto (1-\alpha) * \frac{1}{s^{\beta}}, && \text{if } s > 0 \\ p(s=0) &= \alpha, && \text{else} \end{aligned}$$

3.3 Regret calculation

For regret calculation, an optimal strategy needs to be set. As we have seen in class (Lecture on MABs), the definition of this optimal strategy has to be tailored to its application and does not necessarily correspond to the sequence of decisions that achieve the best reward. In our context, given that the *dark pool* and simulated and that demands at venues are sampled from models, we chose to set the optimal policy as the one that corresponds to performing greedy allocations (Algorithm 2) upon the known tail probabilities.

3.4 Parameter study

The aim of our study is to measure the performance of the different allocation strategies, for different venue (pool) parameters.

The *dark pool* problem parameters are the following:

- K, the number of venues
- V, the maximum allocation size
- α and β, the *zero-bin* and β distribution parameters per venue

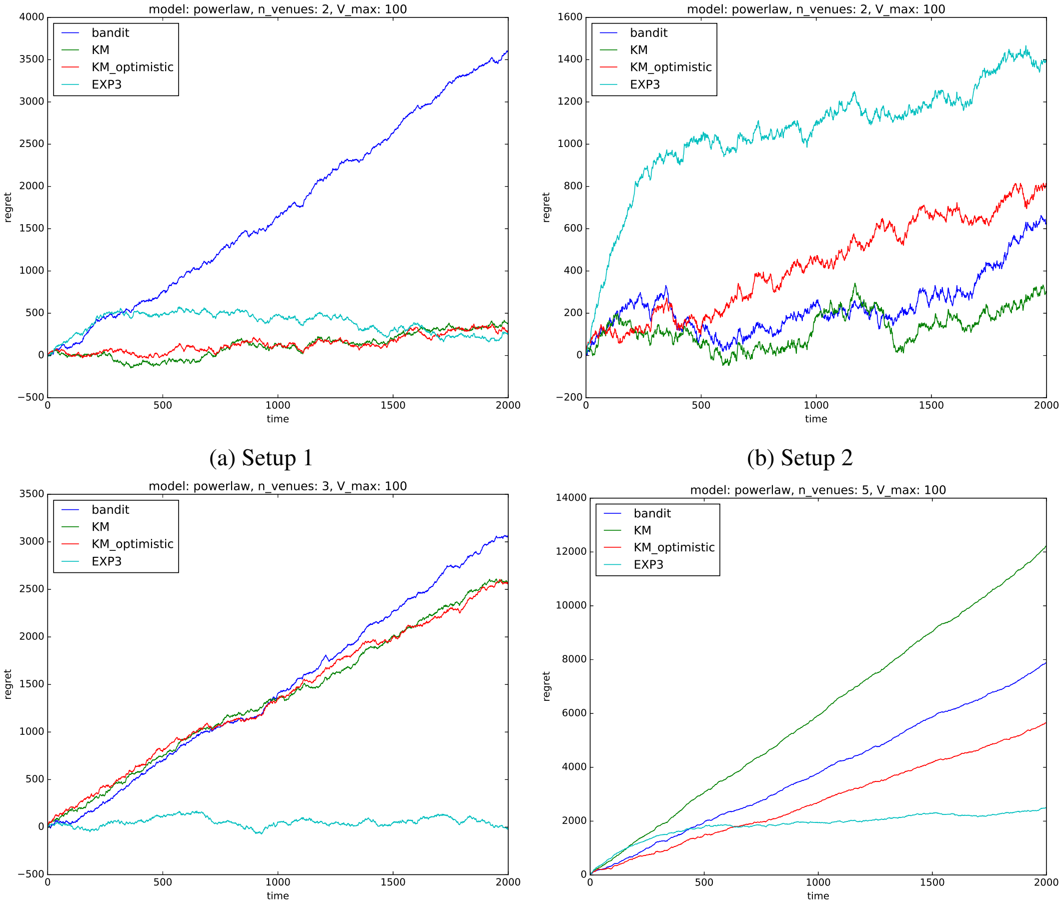
For the purpose of our experiments, we chose to set $\forall t, V^t = 100$ as a constant, but drawing V^t from a distribution is possible. The algorithms we presented make no assumption over this aspect.

4 Results and Discussions

4.1 Sensitivity to problem complexity

We explored the performance of the algorithms on 4 pools of different complexities. Being able to quantify a *dark pool* problem's complexity is also a challenge. For instance, for MABs, one can refer to the Lai and Robbins' measure of complexity. This question will not be addressed here and complexity will be assessed qualitatively.

We established four setups, for which the powerlaw parameters α and β were inspired from [Ganchev]:



(a) Setup 1

(b) Setup 2

(c) Setup 3

(d) Setup 4

Figure 1: Regret evolution

- **Setup 1**: 2 venues, $(\alpha = 0.6, \beta = 0.1)$ and $(\alpha = 0.7, \beta = 1.2)$
- **Setup 2**: 2 venues, $(\alpha = 0.9, \beta = 1.5)$ and $(\alpha = 0.5, \beta = 0.2)$
- **Setup 3**: 3 venues, $(\alpha = 0.7, \beta = 0.5)$, $(\alpha = 0.6, \beta = 1.2)$ and $(\alpha = 0.8, \beta = 0.3)$
- **Setup 4**: 5 venues, $(\alpha = 0.5, \beta = 0.4)$, $(\alpha = 0.9, \beta = 1.2)$, $(\alpha = 0.8, \beta = 1.2)$, $(\alpha = 0.5, \beta = 0.4)$ and $(\alpha = 0.7, \beta = 0.8)$

The intuition would be that more venues, means more complexity in general. Between setups 1 and 2, it is clear that it is easier for an algorithm to learn on setup 2 since between the two venues, the first one has both a much higher probability of yielding zero and a much higher beta coefficient whereas on setup 1, the two venues have a similar ZB parameter which is the reason why the simple bandit algorithm does not manage to learn.

To assess the performance of the algorithms, we plot regret over a time horizon of 2000 (Figure 1).

Figure 1 shows that exp3 is a generally stronger algorithm than the *Kaplan-Meier* algorithms. Exp3 does seem to under-perform in setup 2 compared to other algorithms, we can see that it still manages to learn and reach a normal regime after having troubles adjusting its parameters. But on the three other settings it performs as well or better than the other algorithms. In setup 3 and 4 (pools with 3 and 5 venues), it is the only algorithm that manages to actually learn.

4.2 Adaptation to venue swapping

In this subsection, we analyse the reaction of the algorithms when two venues are swapped. For this analysis, we restricted ourselves to setup 1. Over time the time horizon T, we swap the two venues 4 times and look at the evolution of the allocations (Figure 2).

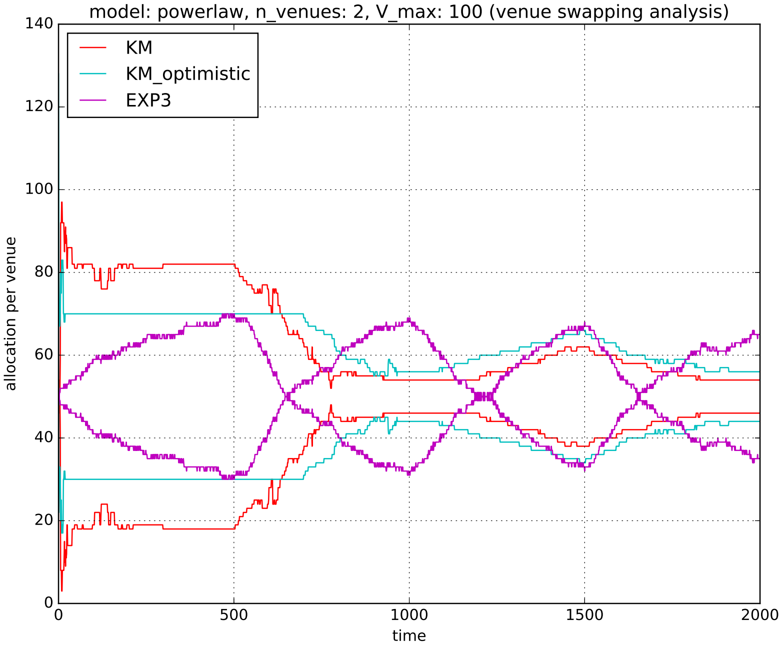


Figure 2: Allocation evolution when venues are swapped

Our observation is that EXP3 adapts very quickly to venue swapping while the *Kaplan-Meier* estimators, because of their tail probability estimators that are refined after each iteration, are slower to adapt.

Acknowledgments

We would like to thank Pr. Alessandro Lazaric and TA Emilie Kaufmann for their teaching in the Reinforcement Learning class at ENS Paris Saclay.

References

[1] Ganchev, K., Nevmyvaka, Y., Kearns, M., & Vaughan, J. W. (2010). Censored exploration and the dark pool problem. Communications of the ACM, 53(5), 99-107.

[2] Agarwal, A., Bartlett, P. L., & Dama, M. (2010, May). Optimal Allocation Strategies for the Dark Pool Problem. In AISTATS (Vol. 9, pp. 9-16).