Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Special Track on AI for Good

Sequential Vaccine Allocation with Delayed Feedback

Yichen Xiao[1] , **Han-Ching Ou**[2] , **Haipeng Chen**[2] , **Van Thieu Nguyen**[3] and **Long Tran-Thanh**[1,3]
[1]University of Warwick
[2]Harvard University
[3]FSOFT QAI
{chen.xiao, long.tran-thanh}@warwick.ac.uk, hou@g.harvard.edu, hpchen@seas.harvard.edu, thieunv4@fsoft.com.vn

Abstract

In this work we consider the problem of how to best allocate a limited supply of vaccines in the aftermath of an infectious disease outbreak by viewing the problem as a sequential game between a learner and an environment (specifically, a bandit problem). The difficulty of this problem lies in the fact that the payoff of vaccination cannot be directly observed, making it difficult to compare the relative effectiveness of vaccination on different population groups. Currently used vaccination policies make recommendations based on mathematical modelling and ethical considerations. These policies are static, and do not adapt as conditions change. Our aim is to design and evaluate an algorithm which can make use of routine surveillance data to dynamically adjust its recommendation. We evaluate the performance of our approach by applying it to a simulated epidemic of a disease based on real-world COVID-19 data, and show that our vaccination policy was able to perform better than existing vaccine allocation policies. In particular, we show that with our allocation method, we can reduce the number of required vaccination by at least 50% in order to keep the peak number of hospitalised patients below a certain threshold. Also, when the same batch sizes are used, our method can reduce the peak number of hospitalisation by up to 20%. We also demonstrate that our vaccine allocation does not vary the number of batches per group much, making it socially more acceptable (as it reduces uncertainty, hence results in better and more interpretable communication).

1 Introduction

Vaccination is a crucial tool in the fight against infectious disease. The advent of vaccination has all but eliminated diseases which have devastated human populations in the past, such as smallpox. The eradication of smallpox alone has been estimated to have prevented 40 million deaths [Ehreth, 2003]. More recently, the effectiveness of the COVID-19 vaccines [Pritchard *et al.*, 2021] demonstrates that vaccination remains one of our most important weapons against infectious disease.

There are good reasons to believe that pandemics like COVID-19 will become more common due to factors such as global travel, urbanisation, increased human-animal contact, and climate change [Houghton, 2019; Bloomfield, 2020; McMichael *et al.*, 1996]. Since vaccines are specific to a single pathogen, any serious novel outbreak will likely require the development of a new vaccine. Due to the time it takes to develop and produce new vaccines, it is likely that public health agencies will have to contend with shortages like the one we saw during the COVID-19 pandemic [Carpetta *et al.*, 2021]. This leads to an important question: How can we maximise impact of a limited supply of vaccines?

Determining who should be vaccinated first is not trivial. There are a variety of possibly conflicting goals, chiefly revolving around limiting dissemination, mortality, and morbidity. For example, there is a trade-off between prioritising those who suffer the most from a disease (usually the elderly) and those who transmit the most (usually children and young adults) [Matrajt *et al.*, 2020]. Even if our goal was to minimise the number of hospitalisations, targeting the group that transmits the most could still be more effective than targeting the group that is most likely to be hospitalised due to the fact that there would be fewer people infected in total. Mathematical models usually show that the optimal strategy for this goal is a mixed strategy (i.e. split the coverage) over the two groups [Foy *et al.*, 2021; Shim, 2021], though such strategies are not typically employed in real life.

During the COVID-19 pandemic, major public health agencies devised a prioritisation list which defined the order in which different population groups should be vaccinated [Noh *et al.*, 2021]. For instance, the WHO recommended that healthcare workers and older adults should be in the highest priority group [Noh *et al.*, 2021]. Although these recommendations are useful and much better than a random allocation, it is far from clear that this approach will lead to the optimal allocation for any goal. Most previous works on vaccine allocation have focused on developing strategies preemptively before the start of the epidemic. While very useful to provide insights into which baseline policies can best control an infection, they may not be ideal to make real-time decisions as the infection is progressing. Instead of a static, non-adaptive prioritisation list, we argue that an adaptive allocation algorithm that dynamically changes its recommendation using up-to-date data is needed.

As such, we view the vaccine allocation problem through the lens of a sequential decision making problem. A sequential decision making problem is a game with discrete rounds. In each round, an agent has a number of possible actions to choose from. Once an action has been chosen, the agent observes the consequences of their choice. The agent is able to use the history of actions and consequences to inform their future choices. Previous work in that aims to tackle similar decision making problems typically rely on MDP [Sutton and Barto, 2018], or PoMDP [Kaelbling *et al.*, 1998] models. However, the vaccine allocation problem is significantly more complicated than a the standard MDP/PoMDP models in at least two ways. First, the *effects of vaccination cannot be observed in a straightforward manner*: It can only be inferred based on changes in the number of positive tests, hospitalisations, or deaths over a period of time, *after a delay*. When observations can be made, they are likely the result of the combined effects of multiple days of vaccine distribution, and hence it's hard to attribute the effects to individual days. Second, the *number of possible ways to allocate some supply of vaccines to a population is enormous*. These challenges of combinatorial and delayed feedback makes any MDP (or PoMDP) based models in-trackable.

Against this background, in this paper we will provide a novel solution based on the bandit framework [Lattimore and Szepesvári, 2020], a simplified but more flexible version of MDPs. In addition, its solutions are typically sufficiently simple to be *easily interpretable* to the society (which we believe is a crucial criterion). In particular, we deal with the first challenge by expanding upon a similar work by Cesa-Bianchi, Gentile and Mansour [Cesa-Bianchi *et al.*, 2018], which extends the standard bandit algorithm to deal with exactly this type of feedback. For the second problem, we restrict our choices by dividing the available supply of vaccines into some number of equal sized batches, and we also divide the population into some number of non-overlapping groups (for example, by age, or by location). We will show that we can further impose structure on the set of possible actions by recasting the vaccine allocation problem as a path finding problem, which allows us to use results given by [Vu *et al.*, 2020] to efficiently search the strategy space.

In order to evaluate the algorithm, we used a version of a standard epidemiological model: the age-structured deterministic SEIR (susceptible-exposed-infectious-recovered) compartmental model. The parameters of our model are based upon on the parameters of a SEIR model for COVID-19, using real data collected in the UK. Our model accounts for the varying severity that COVID-19 seems to have on different age-groups [Davies *et al.*, 2020], and we represent social interactions through age group to age group contact matrices built by survey data.

1.1 Our Contribution

In contrast to deriving prioritisation from mathematical models (more details about the existing methods can be found in Section 1.2), our approach requires very few assumptions about the nature of the disease. The measurements that we need (such as number of hospitalisations) are simple to collect and unlikely to be incorrect. Bandit algorithms are also innately able to adapt to changing conditions (such as the emergence of a new variant) and are more practical for real-time decision making as the infection progresses. Mathematical models can usually agree on which groups are the most important, but they cannot agree on an optimal strategy due to differences in construction. This leads to strict prioritisation lists which our algorithm avoids. In more detail, we show that using our bandit algorithm for vaccine allocation reduced the peak number of hospitalisations by 20% compared to a 'static' vaccination policy where the eldest were always prioritised. We also show that with our allocation method, we can reduce the number of required vaccination by at least 50% in order to keep the peak number of hospitalised patients below a certain threshold. We also demonstrate that our vaccine allocation does not vary the number of batches per group much, making it socially more acceptable (as it reduces uncertainty, hence results in better and more interpretable communication).

1.2 Existing Approaches to Vaccine Allocation

The Framework for Equitable Allocation of COVID-19 Vaccine [National Academies of Sciences *et al.*, 2020] invokes the principle of maximum benefit as one of their foundational principles in establishing their plan for vaccine allocation. For them, maximum benefit means minimising severe morbidity and mortality caused by COVID-19. It is important to note that this is not the only principle, for allocating a limited supply of life-saving vaccines is also a difficult moral problem. Concerns about fairness, health inequities and other ethical problems are a major component in any allocation framework. However, ethical concerns are largely outside the scope of this work. We will concentrate on the more measurable goals like minimising incidence, morbidity and mortality.

2 Algorithm Design

In this section we describe our vaccine allocation algorithm in more detail, which builds on top of path planning with side-observation and bandit with delayed feedback models. Given this, we first briefly discuss the latter two in Sections 2.1 and 2.2, respectively. We then turn to investigate a simplified version of our problem, where the feedback is not delayed (Section 2.3). Finally, we discuss how to incorporate feedback delay into this framework (Section 2.4).

2.1 Path Planning Problems with Side-Observations

We consider the following problem, of path planning with side-observations feedback (SOPPP), introduced by Vu et al. [Vu *et al.*, 2020], Consider an acyclic graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges of the graph. Two special vertices, the source and destination, are denoted s and d respectively. Let $\mathcal{P}$ denote the set of paths from starting from s and ending at d. Each path $\mathbf{p} \in \mathcal{P}$ corresponds to a vector in $\{0,1\}^{|\mathcal{E}|}$, where $\mathbf{p}(e) = 1$ if and only if $e \in \mathcal{E}$ is in $\mathbf{p}$. Let $E = |\mathcal{E}|$ and $V = |\mathcal{V}|$ for convenience.

Given a time horizon $T \in \mathbb{N}$, at each (discrete) stage $t \in \{1, 2, \ldots, T\}$, a learner chooses a path $\mathbf{p} \in \mathcal{P}$. Then, a loss vector $\boldsymbol{\ell} \in [0,1]^E$ is secretly and adversarially chosen, i.e. it can be an arbitrary function of the learner's history. Each element $\boldsymbol{\ell}_t(e)$ corresponds to the scalar loss embedded on the edge $e \in \mathcal{E}$. The learner's incurred loss is $L_t(\tilde{\boldsymbol{p}}_t) = (\tilde{\boldsymbol{p}}_t)^\top \boldsymbol{\ell}_t = \sum_{e \in \tilde{\boldsymbol{p}}_t} \boldsymbol{\ell}_t(e)$, i.e., the sum of the

Algorithm 1 EXP3-OE Algorithm for SOPPP.

1: **Input:** T, $\eta, \beta > 0$, graph G.
2: Initialize $w_1(e) := 1, \forall e \in \mathcal{E}$.
3: **for** $t = 1$ **to** T **do**
4: Loss vector $\boldsymbol{\ell}_t$ is chosen adversarially (unobserved).
5: Use WP Algorithm (see Algorithm 4 in the appendix) to sample a path $\tilde{\boldsymbol{p}}_t$ according to $x_t(\tilde{\boldsymbol{p}}_t)$ (defined in (1)).
6: Suffer the loss $L_t(\tilde{\boldsymbol{p}}_t) = \sum_{e \in \tilde{\boldsymbol{p}}_t} \boldsymbol{\ell}_t(e)$.
7: Observation graph G_t^O is generated and $\boldsymbol{\ell}_t(e)$, $\forall e \in \mathbb{O}_t(\tilde{\boldsymbol{p}}_t)$ are observed.
8: $\hat{\boldsymbol{\ell}}_t(e) := \boldsymbol{\ell}_t(e)\mathbb{I}_{\{e \in \mathbb{O}_t(\tilde{\boldsymbol{p}}_t)\}} / (q_t(e) + \beta),\ \forall e \in \mathcal{E}$, where $q_t(e) := \sum_{\boldsymbol{p} \in \mathbb{O}_t(e)} x_t(\boldsymbol{p})$ is computed by Algorithm 3 (see Appendix B.2).
9: Update weights $w_{t+1}(e) := w_t(e) \cdot \exp(-\eta \hat{\boldsymbol{\ell}}_t(e))$.
10: **end for**

losses from the edges belonging to $\tilde{\boldsymbol{p}}_t$. The learner's feedback at stage t after choosing $\tilde{\boldsymbol{p}}_t$ is as follows. First, they observe the edges' losses $\boldsymbol{\ell}_t(e)$ for any e belonging to the chosen path $\tilde{\boldsymbol{p}}_t$. Additionally, each edge $e \in \tilde{\boldsymbol{p}}_t$ may reveal the losses on several other edges. To represent these *side-observations* at time t, we consider a graph, denoted G_t^O, containing E vertices. Each vertex v_e of G_t^O corresponds to an edge $e \in \mathcal{E}$ of the graph G. There exists a directed edge from a vertex v_e to a vertex $v_{e'}$ in G_t^O if, by observing the edge loss $\boldsymbol{\ell}_t(e)$, the learner can also deduce the edge loss $\boldsymbol{\ell}_t(e')$; we also denote this by $e \to e'$ and say that the edge e reveals the edge e'. The objective of the learner is to minimize the cumulative expected regret, defined as $R_T := \mathbb{E}\left[\sum_{t \in [T]} L\left(\tilde{\boldsymbol{p}}_t\right)\right] - \min_{\boldsymbol{p}^* \in \mathcal{P}} \sum_{t \in [T]} L\left(\boldsymbol{p}^*\right)$.

We also define

$$\mathbb{O}_t(e) := \{\boldsymbol{p} \in \mathcal{P} : \exists e' \in \boldsymbol{p}, e' \to e\}, \forall e \in \mathcal{E},$$
$$\mathbb{O}_t(\boldsymbol{p}) := \{e \in \mathcal{E} : \exists e' \in \boldsymbol{p}, e' \to e\}, \forall \boldsymbol{p} \in \mathcal{P}.$$

The two main innovations from Vu et al. [Vu *et al.*, 2020] relevant to the vaccine allocation problem is EXP3-OE, the algorithm for SOPPP, and the graphical representation of the action set, which will be formally stated later. Here we will present the main algorithm from EXP3-OE: Algorithm 1. The first parameter is T, the time horizon of the game. Informally, the parameter η controls how aggressively the algorithm will exploit the best known choice. A smaller value of η leads to greater exploration. Finally, β is the implicit exploration parameter, which is used to "pretend to explore" when the observation graph G_t^O is not known.

The algorithm works by assigning a weight to each edge in the graph G. The weight is related to the loss that the algorithm observes based on the update rule in line 9. Each path $\mathbf{p} \in \mathcal{P}$ is also assigned a weight by summing up the weights of each edge $e \in \mathbf{p}$. The probability that a particular path will be chosen is given by $x_t(\mathbf{p})$, given in equation (1), and used in line 5.

$$x_t(\boldsymbol{p}) := \frac{\prod_{e \in \boldsymbol{p}} w_t(e)}{\sum_{\boldsymbol{p}' \in \mathcal{P}} \prod_{e' \in \boldsymbol{p}'} w_t(e')} = \frac{w_t(\boldsymbol{p})}{\sum_{\boldsymbol{p}' \in \mathcal{P}} w_t(\boldsymbol{p}')}, \forall \boldsymbol{p} \in \mathcal{P}. \quad (1)$$

Line 8 estimates edge losses in light of the observation graph. The value $q_t(e)$ referenced on line 8 refers (intuitively) to the probability that the loss on edge e is revealed from the chosen path at t. In summary, in each round t the algorithm proceeds in the following steps:

1. The adversarial environment chooses a loss vector $\boldsymbol{\ell}_t$ for each edge in G.
2. A path $\mathbf{p}_t \in \mathcal{P}$ is sampled with probability $x_t(\mathbf{p})$ using current weights $\mathbf{w}_t$.
3. Losses from edges in $\mathbb{O}_t(\mathbf{p}_t)$ are observed. The loss from path $\mathbf{p}_t$ is suffered.
4. Losses for all edges are estimated by
$$\hat{\boldsymbol{\ell}}_t(e) := \boldsymbol{\ell}_t(e)\mathbb{I}_{\{e \in \mathbb{O}_t(\tilde{\boldsymbol{p}}_t)\}} / (q_t(e) + \beta), \forall e \in \mathcal{E},$$
where $q_t(e)$ is computed by Algorithm 3 (written in Appendix B.2).
5. Weights are updated by $w_{t+1}(e) := w_t(e) \cdot \exp(-\eta \hat{\boldsymbol{\ell}}_t(e))$.

Three other algorithms are presented in Vu et al. [Vu *et al.*, 2020], and can be found in Appendix B.2. These algorithms concern weight pushing, which is a technique used to sample a path in $O(E)$ time instead of $O(|\mathcal{P}|)$. This is extremely useful as $|\mathcal{P}| \gg E$, but the details of weight pushing can be omitted in this work. The proof of the regret bounds (and hence correctness) for EXP3-OE is also omitted, but can also be found in Vu et al. [Vu *et al.*, 2020].

2.2 Multi-Armed Bandits with Delay

Multi-armed bandits (MAB) are decision making models in which at each time step, a player chooses from a set of actions (arms) and play (pull) it to receive a reward drawn from an unknown distribution (each arm may have a different reward distribution). The goal of the player is then to learn which is the best arm and play it as many time as possible to maximise the total expected reward (due to space limitations, we defer the detailed description of the MAB model to Appendix A).

The idea that a MAB will not reveal feedback on the round that an action is chosen has been explored in some detail [Vernade *et al.*, 2017; Cesa-Bianchi *et al.*, 2016; Gergely Neu *et al.*, 2010; Joulani *et al.*, 2013; Bistritz *et al.*, 2019; Cesa-Bianchi *et al.*, 2018] in a variety of settings, including both stochastic and nonstochastic bandits. We consider the case of a nonstochastic bandit with *composite anonymous* feedback, which is studied in detail by Cesa-Bianchi et al. [Cesa-Bianchi *et al.*, 2018], another primary source for this work. Composite feedback refers to a scenario where the loss associated with choosing an action is not observed all at once at a single moment in time. More precisely, the loss for choosing an action at time t is adversarially spread over at most $d \in \mathbb{N}$ consecutive time steps $t, t+1, \ldots, t+d-1$. As a result, the player will observe at time t a composite loss, which is a sum of components of losses associated with the last $d-1$ actions. The feedback is anonymous in the sense that the observed loss at any time t is the sum of an unknown subset of losses from past actions.

Cesa-Bianchi et al. [Cesa-Bianchi *et al.*, 2018] provides a general reduction technique turning a base nonstochastic bandit algorithm into one operating within the composite anonymous feedback setting. Specifically, it provides a wrapper

algorithm (which will be more precisely described later) that takes a MAB algorithm as input. They also prove that the regret of the transformed algorithm is bounded in terms of the regret of the original algorithm.

The general idea of the algorithm is that any given action will be played at least $2d-2$ times by the wrapper algorithm. The losses observed are then aggregated and treated as a single loss, which is then fed into the base algorithm. This approach is simple: we compensate for anonymous feedback by playing the same action a lot of times so as to guarantee feedback is the result of that action. We compensate for composite feedback by taking the aggregate loss and dividing by $2d$. This approach does not make any assumptions about the base algorithm, and is therefore easy to generalise.

2.3 Vaccine Allocation Without Delay

The general blueprint for our vaccine allocation algorithm involves combining the EXP3-OE algorithm with a modified version of the wrapper algorithm. We can observe that given $n > m$, allocating n batches of vaccines to some group is at least as good as allocating m batches to that group. This is a type of side observation that exists in the vaccine allocation problem. We will first work on the version of the vaccine allocation problem where feedback is not delayed. In this case, we show how the vaccine allocation problem can be formulated as a SOPPP. Afterwards, we will incorporate composite anonymous feedback by extending the wrapper algorithm to work on EXP3-OE instead of regular MAB algorithms.

We start with the simplified version of the vaccine allocation problem without any delay. We maintain the framework from problem formulation, except in each round t, choosing action $\mathbf{a}_t$ will immediately yield a loss vector $\boldsymbol{\ell}_t = \mathbf{f}_t$.

Graphical Representation

We define a graphical representation of our action set, an idea inspired by the work of Vu et al. [Vu *et al.*, 2020]. Let graph $G_{b,K}$ be a directed acyclic graph that contains:

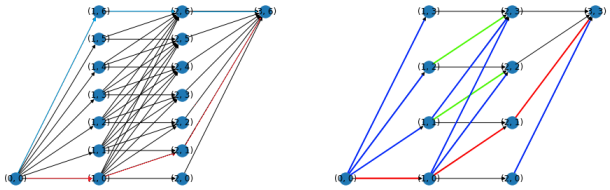
(i) $N := 2 + (b+1)(K-1)$ vertices arranged into $K+1$ layers. Layer 0 and Layer K, each contains only one vertex, respectively labeled $s := (0,0)$–the source vertex and $d := (K,b)$–the destination vertex. Each Layer $i \in [K-1]$ contains $b+1$ vertices whose labels are ordered from left to right by $(i,0),(i,1),\ldots,(i,k)$.

(ii) There are directed edges from vertex $(0,0)$ to every vertex in Layer 1 and edges from every vertex in Layer $K-1$ to vertex (K,b). For $i \in \{1,2,\ldots,K-2\}$, there exists an edge connecting vertex (i,j_1) (of Layer i) to vertex $(i+1,j_2)$ (of Layer $(i+1)$) if $b \geq j_2 \geq j_1 \geq 0$.

Each path from s to d in $G_{b,K}$ corresponds to an allocation of b batches of vaccines for K groups. In particular, consider a path $(0,0),(1,j_1),\ldots,(k,j_k),\ldots,(K,b)$ with $0 \leq j_1 \leq j_2 \cdots \leq b$. This represents an allocation with $j_k - j_{k-1}$ batches allocated to group k for each $k = \{1,2,\ldots,K\}$. See Figure 1a for a visual representation of an example.

Note that each component of the action vector $\mathbf{a}_t(i)$ corresponds to the edge

$$((i-1,\sum_{k=0}^{i-1}\mathbf{a}_t(k)),(i,\sum_{k=0}^{i}\mathbf{a}_t(k))),$$



(a) (b)

Figure 1: (a) $G_{3,6}$ - each path from $(0,0)$ to $(3,6)$ corresponds to an vaccine allocation. The red path represents giving zero batches to group 1, one batch to group 2, and five batches to group 3. The blue path represents giving all six batches to group 1.
(b) This figure depicts $G_{3,3}$. The red path represents giving zero batches to group 1, one batch to group 2, and two batches to group 3. The edges in green are all the direct side-observations described by (4.1) given the path in red. Note that there cannot be direct side-observations in the first and last layers. The edges in blue are all indirect side-observations described by (4.2) given the path in red. Note that the indirect side-observations in the last layer are 'below' the edge being explored, whereas they are 'above' the edge being directly explored in every other layer.

and all K edges in order form a path in G. We can verify this by first noting that the starting vertex is $(0,0)$ and the ending vertex is $(K,\sum_{k=0}^{K}\mathbf{a}_t(k)) = (K,b)$. We can verify by inspection that the ending vertex from the ith edge and the starting vertex from the $i+1$th edges are the same.

Side Observations

EXP3-OE is able to make use of observation graphs, which indicate that from a chosen vaccine allocation, how we can infer about the feedback of another, not chosen (and therefore not observed) allocation. This graph is useful in reducing the computational complexity of the allocation algorithm (as it infers about far more possible allocations by just trying out much less combinations).

This observation graph can be described as follows: Let G be a graph representing the action set of the vaccine allocation problem. Let d be an integer, with $0 \leq d < b$. The first thing to notice is that there is sometimes more than one edge representing the allocation of d batches to a group. More precisely, if $e = ((i,j),(i{+}1,k))$ and $e' = ((i,j'),(i{+}1,k'))$ are valid edges in G, then we say

$$k - j = k' - j' \implies e' \to e. \tag{2}$$

That is, by observing the feedback of edge e', we can also infer the feedback we would get if we have chosen edge e. The explanation of this is as follows: Note that $k - j = k' - j' = d$, which the number of doses allocated to group $i+1$ when a path includes e. That is, the allocation that contains edge e and the allocation that contains edge e' allocates the same amount of batches to that group, hence the observation should be the same. We call this a direct side-observation.

We now turn to the indirect observations: As mentioned before, vaccinating n people is at least as good vaccinating $m < n$ people. The loss observed from allocating $d \geq 0$ batches to a particular group gives us an upper bound on the loss of allocating d' batches to that group, where $d < d' \leq b$. For valid edges $e = ((i,j),(i+1,k))$ and $e' = ((i,j'),(i+1,k'))$, we can formulate this as

$$k - j > k' - j' \implies e' \to e. \tag{3}$$

Note that $k - j = d$ and $k' - j' = d'$. We call these indirect side-observations and we distinguish edges observed this way from direct side-observation. A simple example of side-observations can be found in Figure 1b.

The reason for this separation is due to the fact that EXP3-OE requires observed losses to be exact, and there is no straightforward way to incorporate bounding. Therefore we will have to approximate the losses from indirect side-observations in some way, whereas this is unnecessary for direct side-observation. We propose a very simple function based on the assumption that the loss that we suffer from choosing an action is inversely proportional to the proportion of the population group that is vaccinated.

More formally, let the number of vaccines in each batch be x. Suppose the edge $e' = ((i,j),(i+1,j+k))$ was chosen at time t, with $k \in \mathbb{N}$, $j + k < b$, with observed loss $\boldsymbol{\ell}_t(e') = \ell$. From the problem statement, the proportion of people vaccinated in that group is $\frac{\mathbf{V}_t(i+1)}{N_{i+1}}$. We define $p := \ell(\frac{N_{i+1}}{\mathbf{V}_t(i+1)})$. If $e' \to e$, then $e = ((i,j),(i+1,j'+k'))$ for some $j' \geq j$, $k' \geq k$. For $k' > k$ (i.e. indirect side-observations), we approximate the loss $\tilde{\boldsymbol{\ell}}_t(e)$ given the direct observation $\boldsymbol{\ell}_t(e')$ by $\tilde{\boldsymbol{\ell}}_t(e) = \frac{(\mathbf{V}_t(i+1)+(k'-k)x)p}{N_{i+1}}$. The combination of (2) and (3) fully describe the observation graph G_t^O for the vaccine allocation problem, and we substitute true losses for those edges described by (3) with our approximation $\tilde{\boldsymbol{\ell}}_t$ when building the observation graph (see line 7 in Algorithm 1 from the appendix). Note that the $\tilde{\boldsymbol{\ell}}_t$ we use here is not the same as the $\hat{\boldsymbol{\ell}}_t$ in Algorithm 1.

Modifying EXP3-OE

With both graphical representation and side observations in place, the vaccine problem without delay is now a SOPPP. We only need a few modifications to Algorithm 1. We lift the restriction that $\boldsymbol{\ell}_t \in [0,1]^E$, instead allowing $\boldsymbol{\ell}_t \in [0,\infty)^E$. Choosing an edge is expected to reduce future losses from that edge, so we need to increase the weight of an edge when we see a loss on that edge, which requires a sign change of $-\eta$ to η in the weight update equation (line 9 of Algorithm 1). The algorithms involved in weight pushing do not need any alteration. In preparation for the wrapper algorithm, we will implement Algorithm 1 with two separate functions: one which performs a weight update given observed losses and current weights; and one which samples a new path based given a set of edge weights.

2.4 Incorporating Delayed Feedback

With the undelayed vaccine allocation problem formulated as a SOPPP, we will incorporate delayed feedback using a wrapper algorithm. Before defining the wrapper algorithm, we need to introduce some notation which is heavily based on work by Cesa-Bianchi et al. [Cesa-Bianchi *et al.*, 2018]. Instead of charging the loss of each action to the player immediately, as we have done thus far, we now break each loss $\boldsymbol{\ell}_t(e)$ into the sum $\boldsymbol{\ell}_t(e) = \sum_{s=0}^{d-1} \boldsymbol{\ell}_t^{(s)}(e)$ of d-many components $\boldsymbol{\ell}_t^{(s)}(e) \geq 0$ for $s = 0, \ldots, d-1$. If a player chooses a path $\mathbf{p}_t \in \mathcal{P}$ at time t, then for each edge $e \in \mathbf{p}_t$, the player incurs loss $\boldsymbol{\ell}_t^0(e)$ at time t, loss $\boldsymbol{\ell}_t^1(e)$ at time $t+1$, and so on until time $t+d-1$. However, the player can only observe the combined loss for all the edges in some layer m at any time t. Let

$$H_m = \{e = ((i,j),(i+1,k)) \in \mathcal{E} : i+1 = m, i \in [K-1]\}$$

be the set of edges connecting a vertex from layer i to layer $i+1$. H_m is the set of all edges which corresponds to allocating vaccines to group m. Then the feedback we obtain from group m is given by

$$\mathbf{f}_t(m) = \sum_{e \in H_m} \sum_{s=0}^{d-1} \mathbb{I}(e \in \mathbf{p}_{t-s}) \boldsymbol{\ell}_{t-s}^{(s)}(e).$$

Assuming $d \geq 2$, and for any sequence of paths $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_d \in \mathcal{P}$, we define the d-delayed composite loss function (with vector-valued output) for each $m \in [K]$ by

$$\boldsymbol{\ell}_t^O(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_d)(m) = \sum_{e \in H_m} \sum_{s=0}^{d-1} \mathbb{I}(e \in \mathbf{q}_{d-s}) \boldsymbol{\ell}_{t-s}^{(s)}(e),$$

with $\boldsymbol{\ell}_t^{(s)}(e) = 0$ for all e and s when $t \leq 0$. With this notation, the player observes at the end of each round t the composite reward $\boldsymbol{\ell}_t^O(\mathbf{p}_{t-d+1}, \mathbf{p}_{t-d+2}, \ldots, \mathbf{p}_t)$.

Algorithm 2 Composite Reward Wrapper for Path Planning

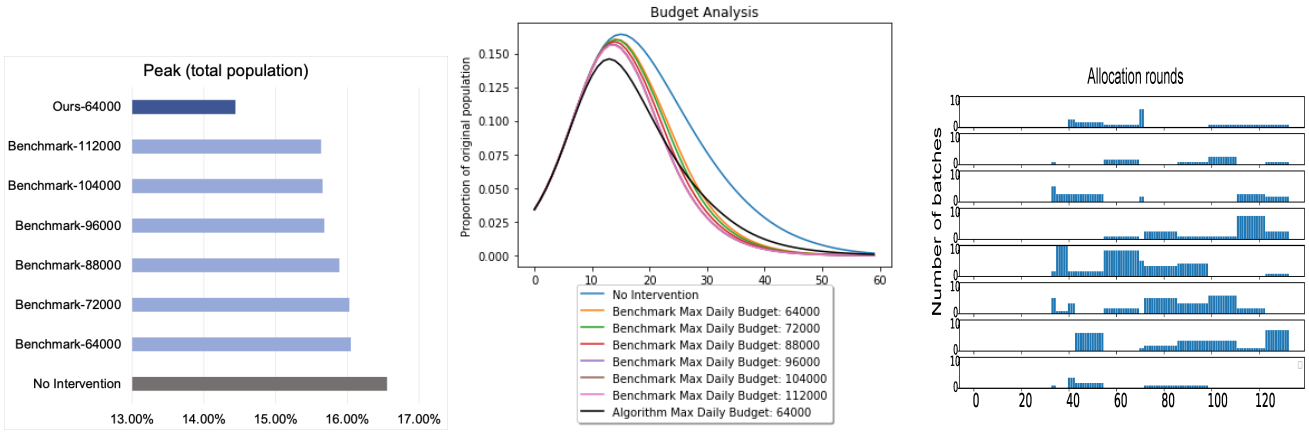
Input: Base PPP Algorithm A (e.g., the EXP3-OE)
Initialization:
Draw $\mathbf{p}_0$ from the uniform distribution P_1 over $\mathcal{P}$.
If $B_0 = 1$ then $t = 0$ is an update round.
for $t = 1, 2, \ldots$ **do**
1. If $t-1$ was update round, draw $\mathbf{p}_t \sim P_t$ and play it without updating P_t (i.e., $P_{t+1} = P_t$);
2. Else if update round was in the interval $\{t - 2d + 1, \ldots, t-2\}$, play $\mathbf{p}_t = \mathbf{p}_{t-1}$ without updating $\mathbf{p}_t$ (i.e., $P_{t+1} = P_t$);
3. Else play $\mathbf{p}_t = \mathbf{p}_{t-1}$ (stay round), and if $B_t = 1$ then the stay round becomes an update round. In such a case:
3.1 Find average composite loss vector

$$\hat{\boldsymbol{\ell}}_t = \frac{1}{2d} \sum_{s=t-d+1}^{t} \boldsymbol{\ell}_s^O(P_{s-d+1}, \ldots, P_s)$$

3.2 Feed Base PPP A with observed rewards from explored edges.
3.3 Use the update rule $p_t \to p_{t+1}$ of Base PPP to obtain new distribution p_{t+1} over $\mathbb{P}$.
end

Wrapper Algorithm

Algorithm 2 is the main wrapper algorithm we will use for the vaccine allocation problem, which is a modified version of the main wrapper algorithm in Cesa-Bianchi et al. [Cesa-Bianchi *et al.*, 2018]. This algorithm takes as input an algorithm called Base PPP which operates on a path planning problem. We assume Base PPP has an update rule which produces probability distributions P_t over the possible paths $\mathcal{P}$ after observing rewards from previous rounds. The Base PPP operates on a path planning problem without delay, where a new probability distribution is drawn in every round and



(a) (b) (c)

Figure 2: (a) Peak value of hospitalised patients with different vaccine batch values. Our algorithm used daily batch of $64,000$, while thewe run the benchmark with varying batches (from $64,000$ to $112,000$.) (b) Number of hospitalised patients over time. (c) Vaccine allocation per group over time.

losses are suffered in every round. The wrapper has access to a sequence $B_0, B_1, \ldots$ of identical and independently drawn Bernoulli random variables, and consists of draw, update and stay rounds:

1. A draw round (number 1 in the for loop) always follows an update round. In a draw round, we simply draw a path from the distribution P_t and play it, where 'playing' refers to suffering the loss of the chosen path.
2. In a stay round (number 2 in the for loop), the algorithm simply plays the same path as the previous round. No change is made to the distribution we draw from.
3. In an update round (number 3 in the for loop), we play the same path as the previous round. We calculate a composite loss which is roughly the average loss suffered in all the rounds since the last update round. We also use the update rule from Base PPP to generate a new distribution given the composite loss.

It is important to notice that the same path is always played between any two update rounds, and there are at least $2d - 1$ rounds between any two update rounds. Intuitively, this algorithms works by compressing about $2d$ rounds in the delayed setting into a single round in the non-delayed setting (recall that d is the max delay).

3 Experimental Validation

We set population $N = 3,000,000$ for all of our simulations. The epidemic begins with 10 random infected individuals aged from 20 to 50. Vaccination begins once 10% of the population has either been exposed, infected, or have recovered from the disease. We compare the performance of our algorithm with a benchmark algorithm that mimics common vaccine allocation policies for COVID-19 in many countries by allocating its daily budget to the eldest age group for which there are susceptible individuals.

We first investigate the peak value of hospitalised patients within the investigated regions (Figures 2a and 2b). This is an important measure as a high number of hospitalised patients at its peak could exceed the total capacity of the region's healthcare system and therefore would cause the system's collapse. From Figure 2a we can see that by using a daily batch of $64,000$, our algorithm can already reduce the peak value from 16.5% of the population to 14.5% (approx. 60 thousand patients), while even with a batch as twice as large ($112,000$ daily batch), the benchmark method would not be able to reduce the peak value down to 15.5%. Note that if our algorithm also runs with $112,000$ baatch size, the peak can be further reduced to less than 12.5% from 15.5% (see Figure 2a), which is a 20% reduction. From Figure 2b we can also see that our algorithm can contain the disease spread down to 10% of the population approx. 5 days earlier than the benchmark (when used the same amount of vaccinations). Both these improvements could save numerous lives and costs spent on the fight against the spread of the disease.

Finally, we investigate whether the vaccine allocation calculated by our algorithm can be easily communicated to the wider audience. Our main concern was if the vaccine allocation per group/region would vary too much, that would generate some uncertainty in the society, and therefore would introduce some sort of lack of trust towards the allocation. Thanks to the nature of the wrapper algorithm (Algorithm 2) which consistently chooses the same allocation for multiple rounds, this is not the case (see Figure 2c). That is, allocations come in (a small number of) large-sized bursts, minimising the number of transitions between allocated/not allocated.

4 Conclusion

We have designed and implemented a novel online learning algorithm with the aim of improving vaccine allocation. This algorithm is useful in that it avoids mathematical modelling of the underlying disease, which is difficult and often fails to yield a precise vaccination strategy. Our algorithm is able to integrate up-to-date data to generate vaccination strategies which are both actionable and sophisticated. We tested our algorithm in a simulation of an epidemic loosely based on COVID-19, which showed that our algorithm is able to make better use of a limited supply vaccines than existing vaccine allocation policies.

References

[Bistritz *et al.*, 2019] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Exp3 learning in adversarial bandits with delayed feedback. *Advances in neural information processing systems*, 2019.

[Bloomfield, 2020] Laura S. P. Bloomfield. Global mapping of landscape fragmentation, human-animal interactions, and livelihood behaviors to prevent the next pandemic. *Agric Human Values*, pages 1–2, May 2020.

[Carpetta *et al.*, 2021] Stefano Carpetta, Mark Pearson, Francesca Colombo, Ruth Lopert, Guillaume Dedet, and Martin Wenzl. Access to covid-19 vaccines: Global approaches in a global crisis, Mar 2021.

[Cesa-Bianchi *et al.*, 2016] Nicolò Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In *Conference on Learning Theory*, pages 605–622. PMLR, 2016.

[Cesa-Bianchi *et al.*, 2018] Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 750–773. PMLR, 06–09 Jul 2018.

[Davies *et al.*, 2020] Nicholas G Davies, Petra Klepac, Yang Liu, Kiesha Prem, Mark Jit, and Rosalind M Eggo. Age-dependent effects in the transmission and control of covid-19 epidemics. *Nature medicine*, 26(8):1205–1211, 2020.

[Ehreth, 2003] J. Ehreth. The global value of vaccination. *Vaccine*, 21(7-8):596–600, Jan 2003.

[Foy *et al.*, 2021] Brody H. Foy, Brian Wahl, Kayur Mehta, Anita Shet, Gautam I. Menon, and Carl Britto. Comparing covid-19 vaccine allocation strategies in india: A mathematical modelling study. *International Journal of Infectious Diseases*, 103:431–438, 2021.

[Gergely Neu *et al.*, 2010] András György Gergely Neu, Csaba Szepesvári, and András Antos. Online markov decision processes under bandit feedback. In *Proceedings of the Twenty-Fourth Annual Conference on Neural Information Processing Systems*, 2010.

[Houghton, 2019] Frank Houghton. Geography, global pandemics & air travel: faster, fuller, further & more frequent. *Journal of infection and public health*, 12(3):448, 2019.

[Joulani *et al.*, 2013] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461. PMLR, 2013.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[Lattimore and Szepesvári, 2020] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[Matrajt *et al.*, 2020] Laura Matrajt, Julia Eaton, Tiffany Leung, and Elizabeth R. Brown. Vaccine optimization for covid-19: Who to vaccinate first? *Science Advances*, 7(6), 2020.

[McMichael *et al.*, 1996] Anthony J McMichael, J. A Haines, Rudolf Slooff, R Sari Kovats, World Health Organization. Office of Global, Integrated Environmental Health, and WHO/WMO/UNEP Task Group on Health Impact Assessement of Climate Change. Climate change and human health : an assessment / prepared by a task group on behalf of the world health organization, the world meteorological association and the united nations environment programme ; editors : A. j. mcmichael ... [et al.], 1996.

[National Academies of Sciences *et al.*, 2020] E.M. National Academies of Sciences, H.M. Division, B.P.H.P.H. Practice, B.H.S. Policy, C.E.A.V.N. Coronavirus, B. Kahn, L. Brown, W. Foege, and H. Gayle. *Framework for Equitable Allocation of COVID-19 Vaccine*. National Academies Press, 2020.

[Noh *et al.*, 2021] E. B. Noh, H. K. Nam, and H. Lee. Which Group Should be Vaccinated First?: A Systematic Review. *Infect Chemother*, 53(2):261–270, Jun 2021.

[Pritchard *et al.*, 2021] Emma Pritchard, Philippa C Matthews, Nicole Stoesser, David W Eyre, Owen Gethings, Karina-Doris Vihta, Joel Jones, Thomas House, Harper VanSteenHouse, Iain Bell, et al. Impact of vaccination on new sars-cov-2 infections in the united kingdom. *Nature medicine*, 27(8):1370–1378, 2021.

[Shim, 2021] Eunha Shim. Optimal allocation of the limited covid-19 vaccine supply in south korea. *Journal of Clinical Medicine*, 10(4), 2021.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Vernade *et al.*, 2017] Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. *arXiv preprint arXiv:1706.09186*, 2017.

[Vu *et al.*, 2020] Dong Quan Vu, Patrick Loiseau, Alonso Silva, and Long Tran-Thanh. Path planning problems with side observations—when colonels play hide-and-seek. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):2252–2259, Apr. 2020.