

Optimizing the order of actions in a model of contact tracing

Michela Meister * and Jon Kleinberg 

Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

*To whom correspondence should be addressed: Email: meister@cs.cornell.edu

Edited By: Yannis Yortsos

Abstract

Contact tracing is a key tool for managing epidemic diseases like HIV, tuberculosis, COVID-19, and monkeypox. Manual investigations by human-contact tracers remain a dominant way in which this is carried out. This process is limited by the number of contact tracers available, who are often overburdened during an outbreak or epidemic. As a result, a crucial decision in any contact tracing strategy is, given a set of contacts, which person should a tracer trace next? In this work, we develop a formal model that articulates these questions and provides a framework for comparing contact tracing strategies. Through analyzing our model, we give provably optimal prioritization policies via a clean connection to a tool from operations research called a “branching bandit”. Examining these policies gives qualitative insight into trade-offs in contact tracing applications.

Keywords: contact tracing, bandits, index policy, epidemiology

Significance Statement

Mathematical models have long played a crucial role in epidemiology, however far fewer models for contact tracing exist. During an epidemic or outbreak, contact tracers often overwhelmed with cases to investigate, and as a result, must strategically choose which cases to investigate first. We develop a mathematical model of contact tracing which articulates this decision-making process, and using this model, we are able to analyze trade-offs in prioritization policies for contact tracing.

Introduction

Mathematical models have provided many useful frameworks to epidemiology. For example, branching processes inspired the R_0 metric used to describe the growth of a disease, and compartmental models help to describe the process through which disease spreads (1). These stylized models complement empirical research in public health by providing a framework for modeling the spread of disease. While there is an abundance of mathematical models for the spread of infection, far fewer models exist for contact tracing. Our goal in this paper is to provide an initial framework for formalizing the contact tracing process.

Contact tracing, the process of identifying individuals exposed to an infected case of a disease, is a key tool for managing epidemic diseases like HIV, tuberculosis, COVID-19, and monkeypox (2–7). Once a contact is identified, they may undergo testing, quarantine, or medical treatment. Identifying infected contacts soon after exposure mitigates the impact of disease on multiple fronts: quarantining cases is crucial to limiting the disease’s spread (8–12), and early initiation of treatment is associated with positive health outcomes such as shorter hospital stays and lower mortality (13–19).

Manual investigations by human contact tracers remain a dominant way in which contact tracing is carried out. Typically, a team of contact tracers work together to interview infected cases and follow up with contacts, tasks which are often arduous and time-consuming (20, 3). To simplify things, we consider the workflow of a single tracer tasked with investigating a set of contacts. The tracer iteratively chooses an individual from the set to query. When an individual is queried, they are tested for infection; if they are infected, they receive medical treatment and list their contacts, which are then added to the set. Since the contact tracing process is limited by the number of tracers available, during an outbreak or epidemic there may not be resources to query each contact immediately. Therefore, an important strategic decision is, which contact should the tracer query next? For example, for an easily transmissible respiratory disease like COVID-19, a tracer may prioritize querying a clerk at a grocery store over a writer working from home, as the clerk likely interacts with many more people each day than the writer. As different diseases have different vectors of transmission, these trade-offs change in other settings.

Deciding which contact to query next quickly becomes complex, since each query a tracer makes has downstream effects.

Competing Interest: The authors declare no competing interest.

Received: May 6, 2022. **Accepted:** December 20, 2022

© The Author(s) 2023. Published by Oxford University Press on behalf of National Academy of Sciences. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

As illustrated in Fig. 1, suppose an infected case A exposed contacts B and C. If B became infected, they exposed D and E, and if C became infected they exposed F, G, and H. Upon querying A, the tracer only has access to first-generation contacts B and C and must decide which one to query next. Querying B opens up the possibility of querying second-generation contacts D and E, whereas querying C gives access to F, G, and H. Thus, the decision whether to query B or C first affects the options available going forward.

Many other factors, such as the probability a contact is infected, their risk of infecting others, and the recency of their exposure affect these choices, and tracers often trade-off between these different factors when deciding which contact to query next. To guide this decision-making, groups like the WHO and the CDC develop detailed recommendations for conducting contact tracing investigations (3, 2, 12, 21–24). A primary aim of these guidelines is to synthesize these trade-offs into a decision-making protocol for tracers to follow. The complexity of this process is emphasized in the CDC’s *Guidelines for Investigations of Contacts with Infectious Tuberculosis*.

Contact investigations are complicated undertakings that typically require hundreds of interdependent decisions, the majority of which are made on the basis of incomplete data, and dozens of time-consuming interventions. (3)

Protocols for prioritizing contacts typically take the form of a flow-chart or matrix that assigns contacts to different groups and then dictates the order in which these groups ought to be queried. Often multiple criteria are considered when categorizing contacts. In many cases, two of the most important factors are the probability that a contact is infected and the recency of their exposure.

Despite the fact that the importance of these factors is well understood, agencies still have difficulty managing these trade-offs, which is not without cost. During a 2017 HIV outbreak in West Virginia, contact tracers in the West Virginia Department of Health and Human Resources were overwhelmed by the surge

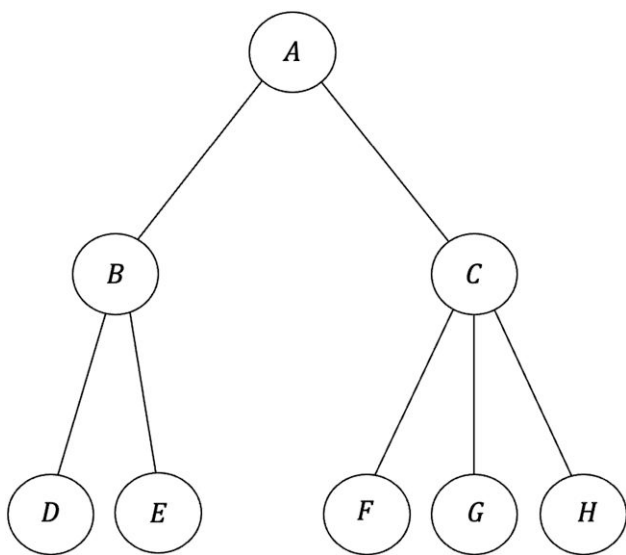


Fig. 1. Manual contact tracing begins with an index case (A), after which further contacts are revealed. Contact tracing proceeds recursively, where the tracer must decide the order in which to investigate contacts, with different orderings yielding different rewards.

in cases. In such a situation, CDC guidelines recommend interviewing contacts associated with clusters of infections, who may be at high risk of infection (2). However, the contact tracers had no means of adjusting the order in which they investigated cases to respond to the outbreak.

[T]here was no supervisory triage system to respond to a cluster of HIV infections. As a result, [contact tracers] would investigate cases linearly—prioritizing index case investigations over investigating contacts within clusters—and did not have the flexibility to shift their priorities based on the identification of an ongoing HIV cluster. (25)

Given the importance of identifying infected cases quickly, both with respect to preventing future infections and initiating medical treatment early, delays like this may cause real harm.

Even in the numerous cases where an agency has an effective method of prioritization, there are subtleties to how these priorities are chosen (21, 23). For example, the protocol for COVID-19 contact tracing developed by the North Carolina Department of Health and Human Services prioritizes contacts by recency of exposure, with one exception: contacts associated with a cluster or outbreak of infections are prioritized first, regardless of when they were exposed (22). By clearly dictating when to shift priorities in an investigation, this protocol addresses the issues presented in the West Virginia case. Yet many questions still remain. How are these priorities chosen? How do these tools translate to different settings? How might priorities change with slightly different factors or as parameters change? Is there even a way to ask these sorts of questions?

In this work, we develop a formal model that articulates these questions and provides a framework for comparing contact tracing strategies. Through analyzing our model, we give provably optimal prioritization policies via a clean connection to a tool from operations research called a “branching bandit” (26). Examining these policies gives qualitative insight into trade-offs in contact tracing applications. The model we study has two phases: first an infection spreads within a population; then the infection process halts and a contact tracing intervention begins. Of course, contact tracing is an extremely complex process requiring nuance and domain expertise, and there are many factors which this stylized model cannot capture—for example, the dynamics of contact tracing while a disease is still spreading, which we explore in forthcoming work (27). Yet this two-phase model already exposes trade-offs and questions about prioritization, decision-making, and resource allocation, which not only seem important in their own right, but also seem prerequisite to understanding more complex settings.

Paper organization

First, in “Further related work,” we discuss relevant work from the contact tracing, operations research, and computer science literatures. Then in “Example” we present an example to illustrate the trade-offs at play in contact tracing. In “Model” we present our formal model, and in “Overview of results” we give an overview of our results for the four models we consider. The remaining “Basic model” and Sections 6–8 present each of the four models, their optimal policies, and the analysis of these policies. Finally, we discuss a few compelling directions for future work in “Discussion.”

Further related work

Work closest to ours focuses on comparing contact tracing policies and the question of “who to trace.” Specifically, Armbruster and Brandeau consider a network model where a tracer with limited capacity must decide at each step which contact to trace next. In (28) via simulation they evaluate three different policies for prioritizing contacts. Using the best of these three policies, in (29), they describe the trade-offs between investing in such a contact tracing effort versus directing funding toward other interventions. Tian et al. (30) also consider a network model where they evaluate a set of contact tracing strategies targeted at various subgroups within a population. Among compartmental models, Hethcote et al. evaluate sets of targeted contact tracing strategies in (31), and Eames considers targeted tracing for different population structures in (32). Our results differ from this prior work in that we provide provably optimal contact tracing policies, as opposed to evaluating a set of specified policies or strategies.

The importance of developing prioritization strategies for contact tracing under resource constraints is highlighted in recent surveys (33, 34). Kaplan et al. consider tracing under resource constraints as well, however in a somewhat different setting (35, 36). There have also been many studies evaluating under what conditions contact tracing is effective and when a disease can be controlled via tracing (37, 38, 11, 39, 5). In (40), Müller et al. analyze a branching process model and compare the fraction of contacts traced to the effective reproduction number.

Digital contact tracing is surveyed in (41). Our work focuses on contact tracing carried out by human tracers and is orthogonal to digital contact tracing apps. In the digital setting, Lunz et al. develop optimal policies for deciding how to quarantine contacts of infected individuals, where some fraction of contacts can be traced by digital means (42).

Within the operations research and computer science literatures, our work is most closely related to search problems on trees that do not have clear connections to contact tracing, but which use similar techniques. The problem closest to our work is the tree-constrained Pandora’s Box problem studied in (43), which also analyzes stochastic selection on a tree, but which involves a fairly different objective. Another related problem is stochastic probing with constraints (44). While our model of contact tracing involves a tracer operating on a tree, it is quite different from the minimum latency problem on weighted trees (45) in that the tracer does not physically travel to the individuals they trace. Finally, our model formulation can be viewed as falling under the general class of Markov decision processes, but the key is that it falls under the specific class of branching bandit problems, which leads to an efficient solution.

Example

We begin with an example that illustrates the trade-offs at play in prioritizing contacts. Suppose an infection spreads in a population over the course of two days,^a after which the infection halts and contact tracing begins. Specifically, over the course of two days each individual in a population meets one new contact each day with probability $q \in (0, 1]$, and infected individuals probabilistically infect each new person they meet. The following events take place:

- On day $t = -2$, w is infected with probability p_w .
- On day $t = -1$, with probability q , w meets x and if w is infected they infect x with probability p_x .

- On day $t = 0$, with probability q , w meets y and if w is infected they infect y with probability p_y .
- On day $t = 0$, with probability q , x meets z and if x is infected they infect z with probability p_z .
- After day $t = 0$, the community goes into lockdown and no new infections occur.

Now consider a contact tracer working to discover infected cases. On day $t = 0$, the tracer finds that w is infected and learns that they met x on day $t = -1$ and also met y on day $t = 0$.

Going forward, on each day $t \geq 1$ the tracer chooses one individual to query. When an infected individual is queried, they receive medical treatment and their contacts become available to query in the future. Querying an individual infected for τ days returns benefit $2^{-\tau+1}$, which represents the probability they respond to treatment.^b Querying an uninfected individual returns benefit 0. Each individual may be queried at most once. The tracer’s objective is to maximize the total benefit accumulated. Since x was exposed to w on day $t_x = -1$, the tracer knows that x may have met another contact (arbitrarily called z) on day $t = 0$, however they have no way of accessing z (or even knowing if z exists) before querying x . Since y was exposed on day $t_y = 0$, the tracer knows that y met no further contacts.

The question is, which contact should the tracer query first, x or y ? Querying x potentially grants access to z , however y was exposed more recently than x , so if y is infected it returns a higher benefit. Fig. 2 shows the benefit of different ways in which the tracer can operate, numerically evaluated for two different parameter settings.

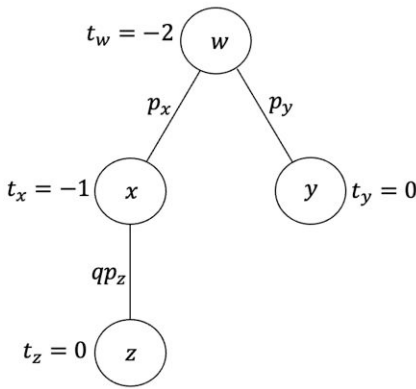
This simple example already highlights a few crucial points we explore in this work. For one, querying the node with the higher expected immediate benefit is not always optimal. This lack of a simple rule might at first seem to imply that a contact tracer working in this synthetic model would need to calculate the expected benefit of each possible ordering in order to identify the optimal choice.

In fact this is not the case. Via a dynamic programming approach we provide optimal policies for querying individuals that are computed before contact tracing begins and which are straightforward to implement: a “priority index” is computed for each individual, and at each step the contact tracer queries the individual with the highest priority index.

Model

Modeling a contact tracing process involves a few different ingredients. People need to meet and make new contacts, through these interactions the infection needs to spread to some individuals and not others, and finally, we need a way of identifying infected individuals and their contacts. Similar to previous models of infection, such as branching processes, a tree forms the basis of our model.

We develop a model with two phases. Phase 1 spans steps $-T \leq t \leq 0$ and involves a contact process, which describes how people meet new contacts, and an infection process, which describes how the infection spreads through these interactions. At the end of Phase 1, the contact and infection processes halt, and from then on no new infections occur. Phase 2 begins on step $t = 0$ and continues indefinitely. At the start of Phase 2, a set of index cases are identified. We define an index case as an individual that is exposed to infection and becomes infected according to a probability function p . We are agnostic to the origin of the index



Set $p_x = 1/2$, $p_z = 3/4$, and $q = 2/3$. There are three possible orderings: xyz , xzy , and yxz . If $p_y = 1/2$, the optimal ordering is yxz , but if $p_y = 5/16$, the optimal ordering is xzy .

p_y	$E[b(xyz)]$	$E[b(xzy)]$	$E[b(yxz)]$
1/2	108/192	112/192	132/192
5/16	90/192	97/192	96/192

After querying w and discovering that x and y exist, the tracer knows the conditional probabilities of infection p_x and p_y for x and y , respectively. The tracer does not know whether z exists, so the probability that z exists and is infected is qp_z . We evaluate the three possible orderings below.

xyz: We query x at $t = 1$. With probability p_x , x is infected, so we receive benefit $1/2$; then we query y at $t = 2$, receiving benefit $1/2$ with probability p_y , and z at $t = 3$ (if it exists), receiving benefit $1/4$ with probability qp_z . If x is not infected, we query y at $t = 2$ and skip z .

$$E[b(xyz)] = p_x(1/2 + p_y/2 + qp_z/4) + (1 - p_x)(p_y/2)$$

xzy: We query x at $t = 1$. With probability p_x , x is infected, so we receive benefit $1/2$. With probability q , z exists, so we query z at $t = 2$, receiving benefit $1/2$ with probability p_z , and we query y at $t = 3$, receiving benefit $1/4$ with probability p_y . If x is not infected or z does not exist, we skip z and query y at $t = 2$, receiving benefit $1/2$ with probability p_y .

$$E[b(xzy)] = p_x(1/2 + q(p_z/2 + p_y/4)) + (1 - qp_x)(p_y/2)$$

yxz: We query y at $t = 1$, receiving benefit 1 with probability p_y . Then we query x at $t = 2$. With probability p_x , x is infected, so we receive benefit $1/4$ and query z at $t = 3$ (if it exists), receiving benefit $1/4$ with probability qp_z . If x is not infected, we skip z .

$$E[b(yxz)] = p_y + p_x(1/4 + qp_z/4)$$

Fig. 2. In a mathematical model of contact tracing, the tracer knows the probability of infection for each contact and needs to choose an ordering for investigating contacts. As we show in this paper’s main results, there is an efficient algorithm to decide an optimal ordering.

cases; for example, an index case may have been identified via surveillance testing, another contact tracing effort, or random chance. Starting with the index cases, on each step $t \geq 0$ a contact tracer, simply called a *tracer*, selects one individual to *query*. Querying an individual models the traditional test-and-trace process: it reveals the individual’s infection status, and if they are infected it reveals their contacts; these contacts may then be queried on future steps. When an infected individual is queried, they also receive medical treatment for the disease. For the remainder of the second phase, the tracer iteratively queries individuals with the goal of identifying infected cases as efficiently as possible.

We show that this two-phase model already requires delicate analysis, and we leave open the problem of analyzing concurrent infection and tracing processes. We hope that our framework can serve as a first step toward more complex models of contact tracing.

Phase 1

Phase 1 spans steps $t = -T$ to $t = 0$. During this phase, individuals meet new contacts and the infection spreads through these interactions.

Let D be an arbitrary distribution on $\{0, 1, 2, \dots\}$. On each step each individual meets a random number of contacts $Z \sim D$, where Z is drawn independently for each individual at each step. If an individual is infected, they infect each new contact they meet independently according to a probability function p defined separately for each model we consider. We call these contacts *exposed*. Exposed individuals are labeled by the recency of their exposure: an individual exposed at time $t = -h$ has *recency* h . Since Phase 1 spans steps $t = -T$ to $t = 0$, exposed individuals have recencies in $\{0, 1, \dots, T\}$. We assume that an individual’s recency can be observed but their infection status is hidden.

Step $t = 0$:

At the end of step $t = 0$ the contact and infection processes halt, and from then on no new infections occur.

To understand the system on step $t = 0$, consider an individual v exposed on step $t = -h$ in Phase 1. If v becomes infected, then for each step $h - 1 \leq t \leq 0$ in the remainder of Phase 1, v exposes $Z_t \sim D$ individuals. Then by the end of step $t = 0$, v has exposed a multiset of contacts $Z(h) = (Z_0, Z_1, \dots, Z_{h-1}) \sim D^h$ where Z_j indicates the number of contacts of recency j . Thus, we can model v as the root of a tree, where the nodes in the first layer represent the contacts v met after being exposed, the nodes in the second layer represent contacts individuals in the first layer met after meeting v , and so on. We call this a *tree of potential exposures* because there is a path of contacts from v to each individual in the tree along which the infection could potentially travel. Since the distribution D on contacts is fixed, v ’s recency $h(v)$ determines the distribution on the tree of potential exposures. We often refer to the nodes in the tree of exposures as v ’s *descendants*.

We call the probability that an exposed node is infected the *probability of infection*. In the first model, we examine, the probability of infection is constant. In the second model, the probability of infection depends on the node’s recency as defined by a function $p(h)$. Either way, for both models a node’s recency contains all the information needed to determine its probability of infection and the distribution on its tree of exposures. In the third model, we consider, the probability of infection depends both on the node’s recency and the recency of its parent. Going forward, we use the terms “node” and “individual” interchangeably.

Phase 2

Phase 2 begins on step $t = 0$ and continues indefinitely. Throughout Phase 2, an individual’s infection status is fixed but hidden. During this phase, a contact tracing effort proceeds.

Contact tracing begins on step $t=0$ when a set of index cases are identified. From then on, at each step $t \geq 0$ the tracer selects one node to query. Querying a node reveals its infection status, and if it is infected reveals the node's children along with the recency of each child. These children may then be queried on future steps. We assume that nodes of the same recency are indistinguishable until they are queried.

On step $t=0$, the index cases are the only individuals available to query. The contact tracer observes the recency of each index case but has no information about any other individuals in the population. For the remainder of Phase 2, the tracer may only query a node that is an index case or the child of an infected node queried on an earlier step. Equivalently, we can view each index case as the root of a tree of potential exposures, which together form a forest. Through this query process, the tracer maintains a sub-forest where any leaf not already queried is either a root or the child of an infected node. These leaves form the *frontier*, and each step the tracer selects one node to query from the frontier. Observe that, by definition, each node in the frontier was exposed to infection in Phase 1.

To understand this process further, consider the options available to the tracer each step. Since nodes of the same recency are indistinguishable until they are queried, the system on any step $t \geq 0$ is defined by a multiset $S_t = (X_0, X_1, \dots, X_T)$, where X_j indicates the number of nodes of recency j present in the frontier. We call S_t the *state* on step t . We use both notions of a multiset when referring to states; i.e. S_t can be viewed as a collection of elements or as a vector of counts.

Querying an infected node v returns a *benefit*, which represents the probability that v responds to medical treatment. The benefit of querying an infected node decays relative to the duration of the infection and depends on the node's recency $h(v)$ and the step t it is queried, as defined by a function $b(h(v), t)$. The benefit of querying an uninfected node is 0, and each node may be queried at most once. The tracer's objective is to maximize the total expected benefit returned over the course of Phase 2.

Defining the objective

On each step $t \geq 0$, the tracer selects a node v_t to query. Since the tracer only selects nodes from the frontier, by definition the node v_t was exposed to infection in Phase 1. Let $\mathbb{1}(v_t)$ indicate whether v_t is infected (1) or uninfected (0). If v_t is infected, then benefit $b(h(v_t), t)$ is returned. Thus the total benefit the tracer accumulates over the course of Phase 2 is

$$\sum_{t \geq 0} \mathbb{1}(v_t) \cdot b(h(v_t), t).$$

The main objective is to develop a policy for querying nodes that maximizes the total expected benefit, where the expectation is taken over all realizations $\{\mathbb{1}(v_0), \mathbb{1}(v_1), \mathbb{1}(v_2), \dots\}$. Such a policy is called an *optimal policy*. In order to make this problem tractable, like many other stochastic models we assume exponential discounting. Specifically, the benefit of querying a node infected for τ steps is $e^{-\beta\tau}$ for a fixed parameter $\beta > 0$.^c

Overview of results

Our results can be summarized in three main contributions.

1) Constructing optimal policies.

We show how to construct an optimal policy for any instance of our model. These policies have a special property: for any instance

of our model, the optimal policy can be described by an algorithm that assigns each node a "type", computes an index based on the type, and chooses the node of the highest index. Such a policy is called an *index policy*.^d An index policy is efficiently computable if each individual index can be computed in polynomial time. For any instance of our model, we show how to compute an optimal policy that is an efficiently computable index policy. We prove this result in [Section 8](#).

We can interpret this result in the context of the contact tracing protocols discussed in the Introduction. Recall that many contact tracing protocols assign individuals to groups and then dictate the order in which groups ought to be queried. Since our construction computes indices from types, the resulting index policy induces a fixed priority ordering on types. Mapping individuals to nodes and groups to types, our results imply that, for our model, (1) the optimal policy overall is defined by a priority ordering on groups, and (2) this priority ordering has an explicit, efficient construction. It is important to note that, taken on its own, this result does not explicitly describe any policies, and it makes no guarantees about the structure of an optimal policy beyond the promise that it is an index policy. Describing the structure of optimal policies any further requires analyzing the construction itself.

2) Analyzing optimal policies for different models of infection.

We examine three different versions of our model, each corresponding to a different model of infection. Analyzing the construction of optimal policies in each model gives qualitative insight into questions about prioritizing contacts from the Introduction, such as how to trade-off between an individual's probability of infection, the recency of their exposure, or the number of other contacts they may have exposed. The three versions we examine are identical except for a function defining the probability of infection. First we examine a basic model, where the probability of infection is constant, then we examine a univariate model, where the probability of infection decays absolutely with time, and finally we examine a bivariate model where a node's probability of infection decays relative to the incubation period of its parent.

3) Connecting contact tracing and the branching bandit problem.

Our key finding is a clean connection between contact tracing and the branching bandit problem (26). The branching bandit problem broadly belongs to a large class of online decision problems called "bandit" problems. The general motivation for the branching bandit problem is scheduling projects, where each project returns a reward and begets new projects, which must then be scheduled. While bandit problems in general have numerous applications (46–48), most applications of the branching bandit model are similar scheduling problems. We therefore find this connection between contact tracing and the branching bandit problem especially striking, since it extends the branching bandit problem to a domain to which it previously has not been applied. We formalize this connection in [Section 8](#).

Section organization.

The remainder of this section reviews our results in the order in which they appear. First we discuss the three models of infection we examine: a basic model (Basic model), a univariate model ([Section 6](#)), and a bivariate model ([Section 7](#)). Then we discuss a general model of infection which encompasses the previous three

models and demonstrates the connection to the branching bandit problem (Section 8). Finally, we give an overview of techniques. Throughout the paper, for simplicity we refer to “the” optimal policy when we are discussing a specific optimal policy we are constructing, however models may have multiple optimal policies.

Basic model

The basic model describes a standard model of infection where the probability of infection is constant.

THEOREM 4.1 In the basic model, it is optimal to query nodes in order of recency.

This result provides insight into some of the most vexing questions from the Introduction. Namely, in practice tracers seem to have these two opposing priorities—whether to query more recent cases or cases that may have exposed many other contacts. In practice, it seems that tracers lean towards querying in order of recency. In our model, this result shows that—even taking into account the downstream effects of accessing an individual’s contacts—the optimal policy is indeed to query the most recent case first.

Univariate model

In the univariate model, the probability of infection decays absolutely with time according to an exponential functional form. The rate of decay is parameterized by a constant $\alpha \geq 0$, where for small values of α , the probability of infection is close to constant, and the rate of decay accelerates as α increases. While in the basic model querying nodes in order of recency is optimal, this is not always the case in the univariate model.

For small values of α , when the probability of infection is close to constant and the setting resembles that of the basic model, the optimal policy queries nodes in order of recency. However, once α reaches a certain threshold, the optimal policy no longer queries nodes in order of recency, and one may wonder whether any structure remains at all. In fact, we find that there is still structure to the optimal policy: the policy always queries either the most recent or least recent node available. We say that such a policy is defined by an *interleaved* priority ordering.

DEFINITION 6.1: Interleaving property

An ordering σ on $\{0, 1, \dots, T\}$ is *interleaved* if for all $0 \leq j \leq T$, σ_j is either the maximum or minimum element of the suffix $\sigma_j, \dots, \sigma_T$.

Observe that many different priority orderings satisfy the interleaving property. For example, an ordering that prioritizes nodes by recency is interleaved, as is an ordering that prioritizes nodes by reverse recency. Our main result shows that the interleaving property holds for optimal policies in the univariate model.

THEOREM 6.1: In the univariate model, there is an optimal policy defined by an interleaved priority ordering.

This result implies that the optimal policy always queries either the most recent node or least recent node in the frontier, which exposes an interesting trade-off between the probability that a node is infected, the recency of its exposure, and its expected number of children. Since a less recent node was exposed earlier in time, it has a larger number of children in expectation. Additionally, since the probability of infection decays with time, a less recent node also has a higher probability of infection. However, a more recent node, should it be infected, returns a

higher immediate benefit. This result implies that the optimal policy pursues the extremes: it either queries the least recent node with the highest probability of infection and the most children in expectation, or it queries the most recent node, which is associated with the highest benefit. It is particularly interesting that the optimal policy never tends toward a node of intermediate recency, which seems to imply that compromising between these two extremes is suboptimal.

Bivariate model

In the bivariate model, the probability that a node is infected decays relative to the incubation period of its parent. As a result, defining a node in this model requires examining two parameters, the recency of the node and the recency of its parent. For a node of recency h with a parent of recency h' , we call $\Delta = h' - h$ the *span*, representing the span of time the parent has been infected upon meeting the child. The probability that a node is infected decays exponentially as a function of Δ . In this model, each node in the frontier is defined by a *type* (h, Δ) , and we examine policies on the set of types $\{0, 1, \dots, T\}^2$. Thus, the bivariate model demonstrates that we can analyze policies that take into account multiple parameters.

Analyzing optimal policies in the bivariate model reveals monotonic structure in the ordering of nodes with the same recency.

THEOREM 8.1: In the bivariate model, there is an optimal policy that queries nodes with the same recency in order of increasing span.

We also explore monotonicity with respect to recency, in a more restricted model. Here we find that, for recencies in $\{0, 1, 2\}$ and within certain constraints, nodes of the same span are queried in order of recency.

THEOREM 8.2: In the bivariate model, for any Bernoulli distribution D , a large enough constant $\beta > 0$, and restricted to types with recencies in $\{0, 1, 2\}$, it is optimal to query nodes of the same span in order of recency.

While there are examples to show that a restriction on β is necessary, the restriction to Bernoulli distributions and to recencies in $\{0, 1, 2\}$ are functions of our proof technique.

General model

The general model provides a broad framework that allows for a variety of factors to affect how individuals interact and how the infection spreads. As we saw in the Introduction, in practice individuals are often categorized according to multiple attributes, such as their profession or role within a community, their age, or the recency of their exposure. The general model captures this complexity, by assigning each node a type representing a set of attributes. Whereas types in the bivariate model are defined by two parameters, types in the general model can be defined by an arbitrary number of parameters. As a result, the general model encompasses the previous three models as well: in the basic and univariate models a node’s type is its recency, and, as before, in the bivariate model a node’s type is the pair (h, Δ) .

Our main result shows that optimal policies in the general model are index policies on types.

THEOREM 8.1: For any instance of the general model, there is an optimal policy that is an index policy on the set of types. Moreover, this index policy has an efficient construction.

This result implies that any instance of the general model has an optimal policy defined by a priority ordering on types. In order to prove this result, we show that any instance of the general model maps to an instance of the branching bandit model.

THEOREM 8.2: The general model reduces to the branching bandit model.

This reduction formalizes the connection between contact tracing in our model and the branching bandit problem, by showing that finding an optimal policy for any instance of the general model requires analyzing an optimal policy for a corresponding instance of the branching bandit model.

Summary of techniques

Here we formally define the branching bandit problem and describe how the reduction from the general model to the branching bandit problem lays the foundation for our results in the basic, univariate, and bivariate models.

The branching bandit model involves arms belonging to classes $\{1, \dots, L\}$, where when an arm of class i is pulled it yields a non-negative reward $R(i)$, occupies $\mu(i)$ steps, and is replaced by a set of new arms N_{i1}, \dots, N_{iL} . Each class i has an arbitrary, known, joint distribution on the random variables $R(i)$, $\mu(i)$, and N_{i1}, \dots, N_{iL} . At each step t , the system is defined by a vector $n(t) = (N_1, \dots, N_L)$, where N_i is the total number of arms of class i available, and a reward received at step t is discounted by $e^{-\eta t}$ for a fixed parameter $\eta > 0$. The objective is to find a policy for pulling arms that maximizes the total discounted reward accumulated. As we describe in “Summarizing Weiss’s Model,” the optimal policy in the branching bandit model is an index policy on the set of classes $\{1, \dots, L\}$ with an efficient construction.

In Section 8, we reduce the general model to the branching bandit model, under a mapping where nodes map to arms, types map to classes, benefit maps to reward, and new children revealed by querying a node map to new arms acquired by pulling an arm. This reduction implies that the optimal policy for any instance of the general model is an index policy on types. Since the basic, univariate, and bivariate models are all versions of the general model, this guarantee extends to these three models as well. In particular, for each instance the reduction provides a construction for a priority ordering that defines the optimal policy. However, the construction on its own does not describe the policy explicitly or define any properties of the policy beyond the promise that it is an index policy. Revealing the structure of these policies requires analyzing the construction of optimal policies for each model, which is the main focus of the following sections.

Basic model

The basic model describes a standard model of infection where the probability of infection is a constant $p_T \in (0, 1]$. Since the benefit of querying an individual infected for τ steps is $e^{-\beta\tau}$, the benefit of querying an infected individual of recency h on step t is $b(h, t) = e^{-\beta(h+t)}$. The basic model is a special case of both the univariate and bivariate models.

Understanding the trade-offs at play.

As described in “Model,” we can view each node in the frontier as the root of a tree of exposures, where less recent nodes have more children in expectation. Recall that if a node is queried and found to be infected, its children are added to the frontier. Therefore, querying a less recent node provides an opportunity to significantly expand the frontier. On the other hand, since the benefit of querying an infected node decays with time, querying a more recent node returns a higher expected benefit. When selecting a node to query, how should the tracer trade-off between these two factors—on the one hand, the expected benefit associated with querying a node, and on the other hand, the opportunity to access its descendants in the future?

Our main result shows that—even taking into account these downstream effects—the optimal policy queries nodes in order of recency.

THEOREM 4.1: In the basic model, it is optimal to query nodes in order of recency.

One might assume that such a straightforward policy has a correspondingly straightforward proof. After all, since querying the most recent node returns the highest expected benefit, perhaps an elementary exchange argument is sufficient.

An attempt at an elementary exchange argument.

Following this line of reasoning, suppose on step t a node u is the most recent node in the frontier, but the tracer selects a less recent node v and queries u on some later step. Now consider exchanging the order of u and v so that u is queried on step t instead. If u is infected, then its children are added to the frontier. Since any child is more recent than its parent, and since u is the most recent node in the frontier on step t , on step $t + 1$ the children of u are more recent than all other nodes in the frontier. Therefore, a commitment to prioritizing nodes by recency requires querying the descendants of u recursively in order of recency.

Exchanging u and v then becomes tricky. As a thought experiment, consider querying either u or v in isolation and then recursively querying any descendants in order of recency. Since a node is always less recent than its descendants, querying u only ever leads to querying nodes more recent than u . However, since v is less recent than u , querying v could lead to querying nodes more recent than v but less recent than u . From this standpoint, it is unclear how to compare these two processes or go about an exchange. One observation is that we could potentially compare the two processes if we measured the process catalyzed by v only up until the first step where a node less recent than u is queried. In fact, such a scheme already exists; continuing with this argument (which is now far from elementary) essentially requires reinventing machinery developed by Weiss for the branching bandit model.

Summarizing Weiss’s model

Here we summarize Weiss’s branching bandit model and the optimal policy for pulling arms, with a slight departure from the original notation in (26). Then in “Defining the optimal policy” we map the basic model to the branching bandit model by mapping each node to an arm and each recency to a class of arms.

Recall that the branching bandit model is a general framework involving arms of classes $\{1, 2, \dots, L\}$, where each time an arm of class i is pulled it returns a nonnegative reward $R(i)$, occupies $\mu(i)$ steps, and is replaced by a set of new arms N_{i1}, \dots, N_{iL} .

Weiss's key idea is the notion of a *period*. A period is defined with respect to any arbitrary priority ordering σ on classes $\{1, 2, \dots, L\}$. For any class $i \in \{1, 2, \dots, L\}$, an (i, σ_j) -period is defined as follows. Initially only a single arm of class i exists in the system. On step $t=0$ the arm of class i is pulled, and from then on at each step an arm is pulled according to the priority ordering σ until all classes $i' \preceq \sigma_j$ are exhausted. Therefore, an (i, σ_j) -period is really defined with respect to the prefix $\sigma_0, \sigma_1, \dots, \sigma_j$, since the ordering of the later elements is irrelevant.

A few random variables describe an (i, σ_j) -period. Let $r(i, \sigma_j)$ be the total discounted reward accumulated during the period, and let $\tau(i, \sigma_j)$ be the duration. Observe that following a period of duration $\tau(i, \sigma_j)$, the reward of the next query is premultiplied by the discount factor $e^{-\tau(i, \sigma_j)}$. Call $\gamma(h, \sigma_j) = \mathbb{E}[e^{-\tau(i, \sigma_j)}]$ the expected premultiplier.

Defining the optimal policy in Weiss's model.

Weiss leverages this notion of a period to inductively construct the optimal priority ordering. He then proves that the index policy defined by this optimal priority ordering is in fact an optimal policy outright.

The optimal priority ordering is constructed via a dynamic program which maintains an optimal prefix that lengthens in each round. In round 0, the highest priority element σ_0 is selected. Entering any round $k > 0$, the prefix $\sigma_0, \sigma_1, \dots, \sigma_{k-1}$ is fixed, and σ_k is selected by comparing (i, σ_{k-1}) -periods over all classes i not in the prefix.

Describing the base case requires one more definition. For any $i \in \{1, 2, \dots, L\}$ an (i, \emptyset) -period simply pulls an arm of class i exactly once. The reward $r(i, \emptyset)$, duration $\tau(i, \emptyset)$, and premultiplier $\gamma(i, \emptyset)$ are defined analogously, so by definition $\tau(i, \emptyset) = 1$ and $\gamma(i, \emptyset) = e^{-\tau}$.

The highest priority is assigned to the element with the highest expected immediate reward.

$$\sigma_0 = \arg \max_{i \in \{1, \dots, L\}} \mathbb{E}[r(i, \emptyset)]$$

In any round $k > 0$, the prefix $\sigma_0, \sigma_1, \dots, \sigma_{k-1}$ is fixed, and σ_k is selected from the elements not already in the prefix.

$$\sigma_k = \arg \max_{i \in \{1, \dots, L\} \setminus \{\sigma_0, \dots, \sigma_{k-1}\}} \frac{\mathbb{E}[r(i, \sigma_{k-1})]}{1 - \gamma(i, \sigma_{k-1})}$$

Section 3 of (26) proves that the priority ordering σ constructed via this dynamic program is the optimal priority ordering, and moreover, that the index policy defined by σ is the optimal policy overall.

An overview of Weiss's proof idea.

For a high-level overview of the proof idea, first observe that σ_0 is the element associated with the maximal expected immediate reward. To understand the intuition behind this selection, imagine choosing between a node u which returns the maximal expected immediate reward and some other node v . Since the expected immediate reward of any descendant of v is at most that of u , there is no reason to delay querying u in order to access the descendants of v .

Moving ahead to any round $k > 0$, committing to the prefix $\sigma_0, \sigma_1, \dots, \sigma_{k-1}$ implies that if we query a node, we are committing to recursively querying its descendants according to the ordering defined by the prefix. Therefore, we are no longer comparing individual queries but instead (h, σ_{k-1}) -periods. To compare periods, we need to trade-off between the expected reward $\mathbb{E}[r(h, \sigma_{k-1})]$ returned and the expected premultiplier $\gamma(h, \sigma_{k-1})$ imposed on the

queries that follow. In this sense, we can think of σ_k as selecting, from the elements not already in the prefix, the element h whose (h, σ_{k-1}) -period has the highest expected "rate" of reward in this time-discounted setting.

Defining the optimal policy

Here we map the basic model of contact tracing to the branching bandit model by mapping nodes to arms, recencies to classes, benefit to reward, and the new children revealed by querying a node to new arms acquired by pulling an arm. We prove this reduction formally in Section 8. We now restate the above dynamic program applied to the basic model, beginning with the definition of a period.

A period is defined with respect to any arbitrary priority ordering σ on recencies $\{0, 1, \dots, T\}$. For any $h \in \{0, 1, \dots, T\}$, an (h, σ_j) -period is defined as follows. Initially only a single root v of recency $h(v) = h$ exists in the frontier. On step $t=0$ the node v is queried, and from then on at each step a descendant is queried according to σ until all nodes v' with recency $h(v') \preceq \sigma_j$ are exhausted. Let $b(h, \sigma_j)$ be the total discounted benefit accumulated during the period, let $\tau(h, \sigma_j)$ be the duration, and let $\gamma(h, \sigma_j) = \mathbb{E}[e^{-\beta \tau(h, \sigma_j)}]$ be the expected premultiplier. For any $h \in \{0, 1, \dots, T\}$, an (h, \emptyset) -period simply queries a single root v of recency $h(v) = h$ exactly once. The benefit $b(h, \emptyset)$, duration $\tau(h, \emptyset)$, and premultiplier $\gamma(h, \emptyset)$ are defined analogously, so by definition $\tau(h, \emptyset) = 1$ and $\gamma(h, \emptyset) = e^{-\beta}$. Note that the benefit, duration, and premultiplier associated with a period are by definition nonnegative.

Then σ_0 is assigned to the element with the highest expected immediate benefit.

$$\sigma_0 = \arg \max_{h \in \{0, \dots, T\}} \mathbb{E}[b(h, \emptyset)] \quad (1)$$

In any round $k > 0$, the prefix $\sigma_0, \sigma_1, \dots, \sigma_{k-1}$ is fixed, and σ_k is selected from the elements not already in the prefix.

$$\sigma_k = \arg \max_{h \in \{0, \dots, T\} \setminus \{\sigma_0, \dots, \sigma_{k-1}\}} \frac{\mathbb{E}[b(h, \sigma_{k-1})]}{1 - \gamma(h, \sigma_{k-1})} \quad (2)$$

Analyzing the optimal policy

While the above dynamic program produces a priority ordering σ that defines an optimal policy, this in no way implies that the resulting optimal policy queries nodes in order of recency. Indeed, in Section 6, we explore other regimes in which optimal policies exhibit very different structure. The main challenge in the following proof is to show that $\sigma = (0, 1, \dots, T)$, which implies that the optimal policy queries nodes in order of recency.

An observation about periods.

The following proof, in addition to many of the proofs in later sections, involves analyzing periods. For the purpose of analysis, it will be helpful to separate the immediate benefit returned by querying a single root from the benefit returned through recursively querying its descendants. Specifically, for a root v with recency $h(v) = h$, we can think of an (h, σ_j) -period as the concatenation of two sub-periods, where v is queried in the first sub-period and descendants of v are queried in the second sub-period. Thus the first sub-period is equivalent to an (h, \emptyset) -period. If v is infected, then at the start of the second sub-period the children of v make up the frontier. Since $h(v) = h$, the children of v have recencies defined by a random multiset $Z(h) = (Z_0, Z_1, \dots, Z_{h-1}) \sim D^h$,

where Z_j indicates the number of children of recency j . To describe the second sub-period, we first need to define a generalization of a period, called an epoch.

For any state $S = (X_0, X_1, \dots, X_T)$, define an (S, σ_j) -epoch as follows. At the start of step $t=0$, the state S defines the recencies of nodes present in the frontier. On each step, $t \geq 0$ nodes are queried according to the ordering defined by the prefix $\sigma_0, \sigma_1, \dots, \sigma_j$ until all nodes v' with recency $h(v') \leq \sigma_j$ are exhausted. Let $b(S, \sigma_j)$ be the total discounted benefit accumulated over the period, let $\tau(S, \sigma_j)$ be the duration, and call $\gamma(S, \sigma_j) = \mathbb{E}[e^{-\beta\tau(S, \sigma_j)}]$ the expected premultiplier.

Therefore, an (h, σ_j) -period can be thought of as an (h, \emptyset) -period, which in the event that the root is infected, is followed by a $(Z(h), \sigma_j)$ -epoch for $Z(h) \sim D^h$. If the root is infected, then a benefit of $e^{-\beta h}$ plus the benefit accumulated during the following epoch is returned. If the root is not infected, then no benefit is returned. Therefore, the expected benefit accumulated over the (h, σ_j) -period is

$$\mathbb{E}[b(h, \sigma_j)] = p_T \left(e^{-\beta h} + e^{-\beta} \mathbb{E}_{Z(h) \sim D^h} [b(Z(h), \sigma_j)] \right) \quad (3)$$

Likewise $\gamma(h, \sigma_j)$ has a similar decomposition. If the root is infected, then $\tau(h, \sigma_j) = 1 + \tau(Z(h), \sigma_j)$ where $Z(h) \sim D^h$. If the root is not infected, then $\tau(h, \sigma_j) = 1$. Therefore,

$$\begin{aligned} \gamma(h, \sigma_j) &= \mathbb{E}[e^{-\beta\tau(h, \sigma_j)}] \\ &= p_T \mathbb{E}_{Z(h) \sim D^h} [e^{-\beta(1+\tau(Z(h), \sigma_j))}] + (1 - p_T)e^{-\beta} \\ &= e^{-\beta} \left(p_T \mathbb{E}_{Z(h) \sim D^h} [e^{-\beta\tau(Z(h), \sigma_j)}] + 1 - p_T \right) \end{aligned} \quad (4)$$

For both decompositions, the expectation is over the randomness of a node's descendants and their infection statuses.

Restating the optimal policy.

The probability of infection is p_T , and the benefit of querying an infected node of recency h is $e^{-\beta h}$, so $\mathbb{E}[b(h, \emptyset)] = p_T e^{-\beta h}$. Applying this definition to Eq. 1, σ_0 is assigned the value

$$\arg \max_{h \in \{0, 1, \dots, T\}} p_T e^{-\beta h} \quad (5)$$

In any round $k > 0$, the prefix $\sigma_0, \sigma_1, \dots, \sigma_{k-1}$ is fixed, and σ_k is selected from the elements not already in the prefix. Let $H' = \{0, \dots, T\} \setminus \{\sigma_0, \dots, \sigma_{k-1}\}$. Then applying Eqs. 3 and 4 to 2, σ_k is assigned the value

$$\arg \max_{h \in H'} \frac{p_T (e^{-\beta h} + e^{-\beta} \mathbb{E}_{Z(h) \sim D^h} [b(Z(h), \sigma_{k-1})])}{1 - e^{-\beta} (p_T \mathbb{E}_{Z(h) \sim D^h} [e^{-\beta\tau(Z(h), \sigma_j)}] + 1 - p_T)} \quad (6)$$

With these decompositions, we are now equipped to prove Theorem 4.1.

THEOREM 4.1: In the basic model, it is optimal to query nodes in order of recency.

Proof. Fix $T \in \mathbb{N}$, $p_T \in (0, 1]$, and $\beta > 0$. Let σ be the optimal priority ordering constructed via the dynamic program in ‘‘Defining the optimal policy.’’ It suffices to show that for all $j \in \{0, 1, \dots, T\}$, $\sigma_j = j$.

Proof by induction on j . By Eq. 5,

$$\sigma_0 = \arg \max_{h \in \{0, 1, \dots, T\}} p_T e^{-\beta h} = 0$$

Fix $0 < k < T$. Assume that for all $0 \leq j \leq k-1$, $\sigma_j = j$, resulting in the prefix is $(0, 1, \dots, k-1)$. Fix $h \geq k$. For $Z(h) = (Z_0, Z_1, \dots, Z_{h-1}) \sim D^h$, consider a $(Z(h), \sigma_{k-1})$ -epoch. The epoch begins in state $Z(h)$, and since

this particular prefix is in order of recency, nodes are then queried recursively in order of recency until all nodes v' of recency $h(v') \leq k-1$ are exhausted. Thus a $(Z(h), \sigma_{k-1})$ -epoch only queries nodes with recencies in $\{0, 1, \dots, k-1\}$. Let $\Pi_k(Z(h)) = (Z_0, Z_1, \dots, Z_{k-1})$ be the projection of $Z(h)$ onto the first k coordinates, and note that $\Pi_k(Z(h)) \sim D^k$. Thus the benefit and duration of a $(Z(h), \sigma_{k-1})$ -epoch are identically distributed as the benefit and duration of a $(\Pi_k(Z(h)), \sigma_{k-1})$ -epoch. Call the total expected benefit b_k and observe that it does not depend on h .

$$\begin{aligned} b_k &= \mathbb{E}_{Z(h) \sim D^h} [b(Z(h), \sigma_{k-1})] \\ &= \mathbb{E}_{Z(h) \sim D^h} [b(\Pi_k(Z(h)), \sigma_{k-1})] \\ &= \mathbb{E}_{Z(k) \sim D^k} [b(Z(k), \sigma_{k-1})] \end{aligned} \quad (7)$$

Since the durations are also identically distributed, the expected premultipliers are equal. Call this premultiplier γ_k and observe that it does not depend on h .

$$\begin{aligned} \gamma_k &= \mathbb{E}_{Z(h) \sim D^h} [\gamma(Z(h), \sigma_{k-1})] \\ &= \mathbb{E}_{Z(h) \sim D^h} [\gamma(\Pi_k(Z(h)), \sigma_{k-1})] \\ &= \mathbb{E}_{Z(k) \sim D^k} [\gamma(Z(k), \sigma_{k-1})] \end{aligned} \quad (8)$$

Combining Eqs. 3 and 7, the expected benefit of an (h, σ_{k-1}) -period is

$$\begin{aligned} \mathbb{E}[b(h, \sigma_{k-1})] &= p_T \left(e^{-\beta h} + e^{-\beta} \mathbb{E}_{Z(h) \sim D^h} [b(Z(h), \sigma_{k-1})] \right) \\ &= p_T (e^{-\beta h} + e^{-\beta} b_k) \end{aligned}$$

Likewise, combining Eqs. 4 and 8, the expected premultiplier is

$$\begin{aligned} \gamma(h, \sigma_{k-1}) &= e^{-\beta} \left(p_T \mathbb{E}_{Z(h) \sim D^h} [e^{-\beta\tau(Z(h), \sigma_{k-1})}] + 1 - p_T \right) \\ &= e^{-\beta} (p_T \gamma_k + 1 - p_T) \end{aligned}$$

By Eq. 6, given the prefix $(0, 1, \dots, k-1)$, σ_k is selected by comparing (h, σ_{k-1}) -periods for $h \in \{k, \dots, T\}$. Therefore for all h in consideration, the above identities hold. Then

$$\begin{aligned} \sigma_k &= \arg \max_{h \in \{k, \dots, T\}} \frac{\mathbb{E}[b(h, \sigma_{k-1})]}{1 - \gamma(h, \sigma_{k-1})} \\ &= \arg \max_{h \in \{k, \dots, T\}} \frac{p_T (e^{-\beta h} + e^{-\beta} b_k)}{1 - e^{-\beta} (p_T \gamma_k + 1 - p_T)} \\ &= k \end{aligned}$$

To understand the final equality, note that $e^{-\beta h}$ decreases with h , and all other terms are constant with respect to h . Thus k , the smallest value of h in consideration, attains the maximum. Since $\sigma_k = k$, for all $j \in \{0, 1, \dots, T\}$, $\sigma_j = j$.

Thus, despite the fact that the optimal policy in the basic setting is simply querying nodes in order of recency, the proof of optimality requires balancing trade-offs between recency and benefit.

Discussion

We present contact tracing as an algorithm design problem where the objective is to develop a policy that prioritizes contacts to trace. Through a clean connection to the branching bandit model (26), we develop provably optimal policies for a variety of infection models. Analyzing the structure of these policies leads to

qualitative insights about trade-offs in contact tracing applications. In the previous section, we examined the basic model, where the probability of transmission is constant, and the optimal policy reflects the prioritization of contacts by recency that we see in practice. In the supplementary section, we study more complex models of infection, where the optimal policy depends on the specific parameters of the model yet still exhibits clear structure. Finally, we conclude with a general model, which has the capacity to model arbitrary interactions between individuals based on factors like an individual's profession, role within a community, or their risk of infection.

There are many compelling questions to consider going forward. One interesting question is how to analyze optimal policies in a dynamic setting, where contact tracing proceeds while the infection is spreading. We are currently exploring this setting in ongoing work (27). A key question here is how to choose the objective function. On the one hand, the algorithm ought to be rewarded for identifying infected cases. On the other hand, the algorithm ought to prevent the spread of new cases, which in a formal sense works against the goal of identifying infected cases. This makes defining the objective function somewhat subtle. Another interesting question to consider is backward contact tracing. In this paper, we examined forward contact tracing, which traces any infections due to an individual, while in backward contact tracing the goal is to identify the source of the infection. Finally, it is an interesting question to ask how theoretical recommendations in general can be integrated into manual contact tracing.

Notes

- We say "day" for simplicity; a day represents an arbitrary unit of time.
- We will have more to say about how benefit depends on time, but roughly speaking this says that treating someone sooner is better than later.
- That is, discounting begins at $t=0$. For the sake of simplicity, for the example in "Example" discounting begins at $t=1$.
- The terms "index" and "index policy" are from the operations research literature and have no relation to the term "index case."

Supplementary material

Supplementary material is available at PNAS Nexus online.

Funding

Michela Meister was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. Jon Kleinberg was supported in part by a Simons Investigator Award, a Vannevar Bush Faculty Fellowship, MURI grant W911NF-19-0217, AFOSR grant FA9550-19-1-0183, ARO grant W911NF19-1-0057, and a grant from the MacArthur Foundation.

Authors' contribution

M.M. and J.K. designed research, performed research, contributed new reagents/analytic tools, and analyzed data. M.M. wrote the paper with guidance and feedback from J.K.

Data availability

Because this paper is focused on mathematical analysis, no experiments on humans or laboratory animals were performed, and there are no datasets associated with this paper.

References

- Martcheva M. 2015. *An introduction to mathematical epidemiology*. Vol. 61. New York: Springer.
- STD National Center for HIV/AIDS, Viral Hepatitis and TB Prevention (CDC). 2008. Recommendations for partner services programs for HIV infection, syphilis, gonorrhea, and chlamydial infection. *MMWR Recomm Rep*. 57(RR-9):1–92.
- National Tuberculosis Controllers Association, Centers for Disease Control, Prevention (CDC, and others). 2005. Guidelines for the investigation of contacts of persons with infectious tuberculosis. Recommendations from the national tuberculosis controllers association and CDC. *MMWR Recomm Rep*. 54(RR-15):1–47.
- Vecino-Ortiz AI, Congote JV, Bedoya SZ, Cucunuba ZM. 2021. Impact of contact tracing on COVID-19 mortality: an impact evaluation using surveillance data from Colombia. *PLoS ONE*. 16(3):e0246987.
- Keeling MJ, Hollingsworth TD, Read JM. 2020. Efficacy of contact tracing for the containment of the 2019 novel coronavirus (COVID-19). *J Epidemiol Community Health*. 74(10):861–866.
- World Health Organization, et al. 2022. Surveillance, case investigation and contact tracing for monkeypox: interim guidance, 25 August 2022. Technical report. World Health Organization.
- New Jersey Department of Health. Interim monkeypox case contact monitoring guidelines for local health departments [accessed 2022 Oct. 5]. <https://www.nj.gov/health/cd/documents/topics/Monkeypox/monkeypox-contact-monitoring.pdf>.
- MacIntyre CR, et al. 1995. High rate of transmission of tuberculosis in an office: impact of delayed diagnosis. *Clin Infect Dis*. 21(5):1170–1174.
- Riley RL, Moodie AS. 1974. Infectivity of patients with pulmonary tuberculosis in inner city homes. *Am Rev Respir Dis*. 110(6P1):810–812.
- Kretzschmar ME, et al. 2020. Impact of delays on effectiveness of contact tracing strategies for COVID-19: a modelling study. *Lancet Public Health*. 5(8):e452–e459.
- Klinkenberg D, Fraser C, Heesterbeek H. 2006. The effectiveness of contact tracing in emerging epidemics. *PLoS ONE*. 1(1):e12.
- World Health Organization Regional Office for the Eastern Mediterranean. 2006. Diagnostic and treatment delay in tuberculosis.
- May M, et al. 2011. Impact of late diagnosis and treatment on life expectancy in people with HIV-1: UK Collaborative HIV Cohort (UK CHIC) Study. *BMJ*. 343:d6016.
- Krawczyk CS, Funkhouser E, Kilby JM, Vermund SH. 2006. Delayed access to HIV diagnosis and care: Special concerns for the southern united states. *AIDS Care*. 18(sup1):35–44.
- Sobrino-Vegas P, et al. 2009. Delayed diagnosis of HIV infection in a multicenter cohort: prevalence, risk factors, response to HAART and impact on mortality. *Curr HIV Res*. 7(2):224–230.
- Nakagawa F, et al. 2012. Projected life expectancy of people with HIV according to timing of diagnosis. *Aids*. 26(3):335–343.
- Greenaway C, et al. 2002. Delay in diagnosis among hospitalized patients with active tuberculosis-predictors and outcomes. *Am J Respir Crit Care Med*. 165(7):927–933.

- 18 Álvarez-Lerma F, et al. 2016. Delay in diagnosis of influenza A (H1N1)pdm09 virus infection in critically ill patients and impact on clinical outcome. *Crit Care*. 20(1):1–11.
- 19 Grinsztejn B, et al. 2014. Effects of early versus delayed initiation of antiretroviral treatment on clinical outcomes of HIV-1 infection: results from the phase 3 HPTN 052 randomised controlled trial. *Lancet Infect Dis*. 14(4):281–290.
- 20 Spencer KD, et al. 2021. COVID-19 case investigation and contact tracing efforts from health departments – United States, June 25–July 24, 2020. *Morb Mortal Wkly Rep*. 70(3):83.
- 21 TB Virginia Department of Health and Newcomer Health. Determining the need for a contact investigation and prioritizing public health response [accessed 2021 Jun 25]. <https://www.vdh.virginia.gov/content/uploads/sites/112/2018/11/Determining-the-Need-for-a-Contact-Investigation-and-Prioritizing-Public-Health-Response.pdf>.
- 22 Moore Z. Contact tracing prioritization [accessed 2021 Jun 25]. https://epi.dph.ncdhhs.gov/cd/lhds/manuals/cd/coronavirus/Contact*20Prioritization*20Guidance'12182020.pdf.
- 23 Centers for Disease Control and Prevention. Prioritizing case investigations and contact tracing for COVID-19 in high burden jurisdictions [accessed 2021 Jun 25]. <https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/prioritization.html>.
- 24 Centers for Disease Control and Prevention. Key information to collect during a case interview [accessed 2021 Jun 25]. <https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/keyinfo.html>.
- 25 Quilter LAS, et al. 2021. Establishing best practices in a response to an HIV cluster: an example from a surge response in West Virginia. *Sex Transm Dis*. 48(3):e35–e40.
- 26 Weiss G. 1988. Branching bandit processes. *Probab Eng Inf Sci*. 2(3): 269–278.
- 27 Meister M, Kleinberg J. 2022. Containing the spread of a contagion on a tree, arXiv, arXiv:2210.13247, preprint: not peer reviewed.
- 28 Armbruster B, Brandeau ML. 2007. Who do you know? A simulation study of infectious disease control through contact tracing. In: Proceedings of the 2007 Western Multiconference on Computer Simulation. San Diego, CA: Society for Modeling and Simulation International. p. 79–85.
- 29 Armbruster B, Brandeau ML. 2007. Contact tracing to control infectious disease: when enough is enough. *Health Care Manag Sci*. 10(4):341–355.
- 30 Tian Y, Osgood ND, Al-Azem A, Hoepfner VH. 2013. Evaluating the effectiveness of contact tracing on tuberculosis outcomes in saskatchewan using individual-based modeling. *Health Educ Behav*. 40(suppl_1):98S–110S.
- 31 Hethcote HW, Yorke JA, Nold A. 1982. Gonorrhea modeling: a comparison of control methods. *Math Biosci*. 58(1):93–109.
- 32 Eames KTD. 2007. Contact tracing strategies in heterogeneous populations. *Epidemiol Infect*. 135(3):443–454.
- 33 Müller J, Kretzschmar M. 2021. Contact tracing—old models and new challenges. *Infect Dis Model*. 6:222–231.
- 34 Kwok KO, et al. 2019. Epidemic models of contact tracing: systematic review of transmission studies of severe acute respiratory syndrome and middle east respiratory syndrome. *Comput Struct Biotechnol J*. 17:186–194.
- 35 Kaplan EH, Craft DL, Wein LM. 2002. Emergency response to a smallpox attack: the case for mass vaccination. *Proc Natl Acad Sci USA*. 99(16):10935–10940.
- 36 Kaplan EH, Craft DL, Wein LM. 2003. Analyzing bioterror response logistics: the case of smallpox. *Math Biosci*. 185(1):33–72.
- 37 Fraser C, Riley S, Anderson RM, Ferguson NM. 2004. Factors that make an infectious disease outbreak controllable. *Proc Natl Acad Sci USA*. 101(16):6146–6151.
- 38 Hellewell J, et al. 2020. Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. *Lancet Glob Health*. 8(4):e488–e496.
- 39 Eames KTD, Keeling MJ. 2003. Contact tracing and disease control. *Proc R Soc Lond B: Biol Sci*. 270(1533):2565–2571.
- 40 Müller J, Kretzschmar M, Dietz K. 2000. Contact tracing in stochastic and deterministic epidemic models. *Math Biosci*. 164(1): 39–64.
- 41 Ahmed N, Michelin RA, Xue W, Ruj S, Malaney R, et al. 2020. A survey of covid-19 contact tracing apps. *IEEE Access*. 8: 134577–134601.
- 42 Lunz D, Batt G, Ruess J. 2021. To quarantine, or not to quarantine: a theoretical framework for disease control via contact tracing. *Epidemics*. 34:100428.
- 43 Boodaghians S, Fusco F, Lazos P, Leonardi S. 2020. Pandora’s box problem with order constraints. In: Proceedings of the 21st ACM Conference on Economics and Computation. New York, NY: Association for Computing Machinery. p. 439–458.
- 44 Gupta A, Nagarajan V, Singla S. 2016. Algorithms and adaptivity gaps for stochastic probing. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM. p. 1731–1747.
- 45 Sitters R. 2002. The minimum latency problem is NP-hard for weighted trees. In: International Conference on Integer Programming and Combinatorial Optimization. Springer. p. 230–239.
- 46 Gittins J, Glazebrook K, Weber R. 2011. *Multi-armed bandit allocation indices*. West Sussex, UK: John Wiley & Sons.
- 47 Slivkins A. 2019. Introduction to multi-armed bandits. *Found Trends Mach Learn* 12(1-2):1–286.
- 48 Lattimore T, Szepesvári C. 2020. *Bandit algorithms*. Cambridge: Cambridge University Press.