Multi-Objective Multi-Cluster Optimization of Non-Pharmaceutical Interventions for Infectious Disease With Resource Constraints

Xueqiao Peng
The Ohio State University
Columbus, USA
peng.969@osu.edu

Yi Mao
The Ohio State University
Columbus, USA
mao.496@osu.edu

Xi Chen
The Ohio State University
Columbus, USA
chen.10183@osu.edu

Dinh Song An Nguyen
The Ohio State University
Columbus, USA
nguyen.2687@osu.edu

Andrew Perrault
The Ohio State University
Columbus, USA
perrault.17@osu.edu

ABSTRACT

In the early stages of an infectious disease crisis, non-pharmaceutical interventions (NPIs) such as quarantines and testing can play an important role. Optimizing the delivery of NPIs is challenging as they can impose substantial direct costs (e.g., test costs) and human impacts (e.g., quarantine of uninfected individuals) and can be especially difficult to target for infections that may spread pre- or asymptomatically. In addition, superspreading, a common characteristic of many infectious diseases, induces informational dependencies across a cluster (group of individuals exposed by the same seed case). We formulate NPI optimization as a partially observable Markov decision process (POMDP), which we aim to solve with reinforcement learning (RL). We find RL provides a promising technical foundation that is difficult to achieve even with modern methods. We propose a novel RL approach that leverages a supervised learning encoder as well as permutation invariant, fixed-size observation representations. Through extensive experimentation and evaluation, we show that our optimized policy can outperform all benchmarks by up to 27%. We also show that the policies discovered by RL can be distilled into decision trees to simplify deployment while still achieving strong performance. Additionally, we explore the possibility of applying the Restless Multi-Armed Bandit to our present setting, which introduces the coordination of limited resources across clusters.

KEYWORDS

reinforcement learning, machine learning, contact tracing, public health

ACM Reference Format:
Xueqiao Peng, Yi Mao, Xi Chen, Dinh Song An Nguyen, and Andrew Perrault. 2024. Multi-Objective Multi-Cluster Optimization of Non-Pharmaceutical Interventions for Infectious Disease With Resource Constraints. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

1 INTRODUCTION

The COVID-19 pandemic has highlighted the crucial role of non-pharmaceutical interventions (NPIs) in effectively managing the spread of infectious diseases. Implementation of NPIs requires careful consideration of multiple objectives, including prevention of viral transmission and reduction of costs associated with quarantine measures. Contact tracing has been widely adopted and extensively studied in infectious disease crises, particularly in the context of COVID-19 [13, 14, 19, 33].

Nevertheless, optimizing NPIs among clusters remains a computationally challenging problem in many settings. First, the action space in each cluster is naturally combinatorially large because an action must be selected for each contact. Second, the problem is inherently multi-objective as interventions have costs associated with them. For example, sensing actions, such as testing, can provide valuable information, but require resources to deploy, and quarantining has human impacts. Additionally, with resource constraints, it is hard to decide how many sensing actions should be allocated to each cluster. Third, inferring the probability that an individual is infectious can be difficult for infections that can be transmissible without symptoms. Finally, the constraints of deployment make it desirable that NPI policies can be executed without the need for computation.

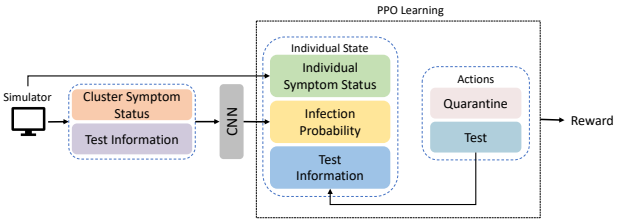

Figure 1: We combine an infection probability encoder that uses supervised learning with a reinforcement learning-based policy.

In this work, we aim to develop a generic approach for cluster-level optimization of NPIs based on reinforcement learning (RL) [32]. We find that modern RL approaches fail to outperform naive baselines such as quarantining all contacts or quarantining symptomatic

contacts. We augment RL in several ways. First, we observe that the high-dimensional state and action spaces exhibit substantial permutation *equivariance*. For example, the order of the contacts does not matter—permuting the contacts and actions should produce the same result. This observation and the combinatorial action space motivate the development of an *egocentric* fixed-size state for each contact. Second, we find that learning to predict the probability of infectiousness via RL training is inefficient and that this quantity has an important structural role in optimal policies in many settings. Thus, we develop a supervised learning module for the infectiousness inference task that leverages Convolutional Neural Networks [9], viewing the cluster state as if it were an image.

We summarize our approach in Fig. 1. Our vision is that, in an infectious disease crisis, an agent-based model simulating the infection would be developed based on observations and expert estimation (see, e.g., McAndrew et al. [21]). This model could be used to evaluate and optimize policies, e.g., using the methods of this paper, and could be refined using contact tracing data from the field. We thus develop a minimally complex agent-based model for SARS-CoV-2 using published research from the early stages of the pandemic and use it as a testbed.

In the real world, resources such as tests are often constrained. Initially, let's assume a situation where the budget for testing is unlimited, and each cluster functions independently. Under these conditions, the proposed NPI optimization approach is effective across multiple clusters without the need to consider testing limitations. However, when we introduce a testing budget, it's possible for one cluster to deplete its entire test allocation to prevent transmission. To address this, we form this problem as a Restless Multi-Armed Bandit problem for allocating tests across the clusters. For cluster level, the agent decides how many tests will be assigned to each cluster.

This paper makes the following contributions:

- We propose a novel RL approach for finding optimal contact tracing policies. Our approach combines RL with supervised learning and a permutation invariant, egocentric, state representation. The resulting agent can be trained and deployed simultaneously across all cluster sizes.
- To motivate the use of a supervised belief state encoder, we show the existence of a simple, yet optimal, *threshold* policy for contact tracing in the setting where no sensing actions are available.
- We develop a simple branching process-based model for SARS-CoV-2 and compare our policies with baselines. We show that we achieve better rewards across a range of objective parameters, even when distilled into decision trees that can be widely distributed.
- We form the sensing action constraint problem as a Restless multi-armed bandit problem and explore the possibility of solving it.

Related work. We identify two main thrusts of work that optimize contact tracing and NPIs: network and branching process. Network models represent connections between individuals as edges in a (possibly dynamic) contact graph [6, 16, 22, 26, 27]. These approaches can leverage network structure in their decisions, but make the strong assumption that the entire contact network is known at each time step. The closest existing approach to ours is RLGN [22], which formulates the problem as a sequential decision-making task within a temporal graph process. In contrast, we take a cluster-based, tree-structured view of contagion [17, 23], but add agent-based temporal elements. This approach has the advantage of aligning more closely with the information available to decision makers in many practical settings and requires less detailed information to construct.

2 PROBLEM DESCRIPTION

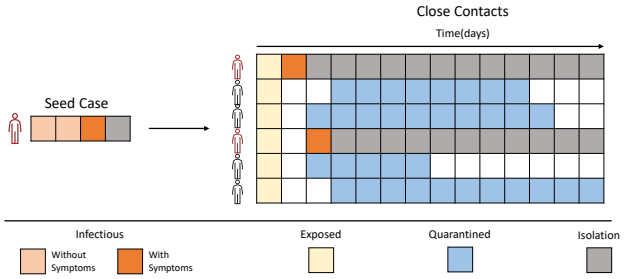

Figure 2: Cluster-based view of intervention planning.

We aim to create NPIs that operate on the cluster level. Fig. 2 shows a motivating example, taken from the cluster-level agent-based simulator we construct for SARS-CoV-2. A *seed case* exposes six *contacts* on the same day. Contacts 1 and 4 eventually become infected and show symptoms on day 2 and day 3, respectively. Contacts 2, 3, 5, and 6 never become infected. In this example, we must make a binary action for each contact on each day: quarantine or not. The goal of the NPI policy is to identify and quarantine (isolate) contacts that are infected and not quarantine uninfected contacts, but the infectious state is not directly observable. The optimal policy depends on trade-offs between different objectives: failing to isolate infected contacts, quarantining uninfected contacts, and direct policy costs (e.g., of tests). Formally, we define the objective we aim to maximize as:

$$(-S_1 - \alpha_2 \times S_2 - \alpha_3 \times S_3)/N, \tag{1}$$

where

- S_1 is the count of *transmission days* where an infected individual is not isolated,
- S_2 is the count of days where a quarantined individual is not infected, and α_2 (which we assume is in $[0, 1]$) is the weight for this term,
- S_3 is the sum of the action costs (e.g., test cost) and α_3 is the weight for this term, and
- N, which is the number of contacts, normalizing the objectives to a score per contact.

In summary, the objective function seeks to minimize the number of transmission days, minimize the number of days of non-effective quarantine, and minimize the cost associated with actions. Intuitively, $1/\alpha_2$ is the number of quarantine days of an uninfectious contact we are willing to accept in exchange for one additional day of isolation of an infectious contact.

Table 1: Parameters of the SARS-CoV-2 cluster infection trajectory generator and test action model.

Parameter	Assumed value	Details and references
Incubation time	Log-normal: Log mean 1.57 days and log std 0.65 days	Mean: 5.94 days. Bi et al. [1]
Duration of infectious period	7 days—2 days before and 5 days after onset if symptomatic	Bi et al. [1]
Probability of infection	0.03	Perrault et al. [29]
Probability that an infected individual shows symptoms	0.8	Buitrago-Garcia et al. [2]
Probability of symptoms without infectiousness	0.01 per day	Hinch et al. [10]
Probability of an asymptomatic infection	0.2	Buitrago-Garcia et al. [2]
Probability seed case is highly transmissive	0.109	Perrault et al. [29]
Infectiousness multiplier for highly transmissive individuals	24.4	Perrault et al. [29]
Test parameters	TP = 0.71, FN = 0.01 FP = 0.29, TN = 0.99	Caulley et al. [3]
Delays	Time to begin tracing a seed case = 3 days Test reporting delay = 1 day	Assumed—realism.
Cluster Size	Sample from uniform distribution on [2,40]	Assumed—we would like to find policies that perform well across cluster sizes.

We remark that the objective value for any NPI policy can be evaluated for a cluster as long as we have an "infection trajectory" for each contact, a record of if and when they become infectious and if and when they exhibit symptoms (which is needed if the policy execution depends on symptom status). This is because these infection timing events are unaffected by the NPI actions we consider.

Formally, we define an *infection trajectory* for contact $n \in N$ as the infectiousness state $i_n^{(t)} \in I \in \{0, 1\}$ and the symptom observation $o_n^{(t)} \in \{0, 1\}$, the true infectiousness of contact and observable symptom state, respectively, of contact n on day t, for all $t \in [T]$ (where $[T] = \{1, 2, \ldots, T\}$). We assume that each of these is binary for simplicity and that t is measured in days, but these are not requirements (e.g., $i_n^{(t)}$ could be a continuous viral load and S_1 could then represent risk-adjusted transmission days). We define an *cluster infection trajectory* as an infection trajectory for each contact in a cluster.

We require either a generator for cluster infection trajectories or a large library of them that we can sample from during training. As an example, we construct a generator for early SARS-CoV-2 using the parameters and sources shown in Tab. 1 Trajectories run from $t = 1$ to $t = 30$, and $t = 3$ is the first time actions are allowed to be taken (modeling a contact tracing delay). Many of the required components of such a generator are distributions that are often estimated in the early stages of an outbreak. Components that are not known can be filled in conservatively or as a belief distribution (e.g., by aggregating expert opinion).

We allow for any set of NPI actions as long they can be simulated on any infection trajectory and their impact on S_1, S_2 and S_3 is defined. For example, a quarantine action, when applied to contact n on day t, causes S_1 to be not incremented if $i_n^{(t)} = 1$, and increments S_2 if $i_n^{(t)} = 0$. A more complex quarantine action may have a failure rate (an individual may not quarantine if directed), incur an additional financial cost (which would be added to S_3), or may include a sensing component (see below). An action with a sensing component reveals information about the contact's infectiousness state $i_n^{(t)}$ according to some distribution, e.g., a test with a binary outcome according to the confusion matrix of test parameters in Table 1. More complex actions can combine sensing and quarantine, e.g., test and quarantine only if positive.

Our simulated environment has four actions: null action (S_3 cost of 0, no effect), quarantine (S_3 cost of 0), test (S_3 cost of 1, draw outcome according to Table 1), and test and quarantine only if results are positive (draw outcome according to Table 1, S_3 cost of 1).

3 APPROACH

The optimization problem from the previous section can be formulated as a partially observable Markov decision process (POMDP). However, solving this POMDP directly is intractable, even with modern RL techniques. Some hope arrives from the result that, under a simplified model that contains only quarantine actions, the POMDP can be solved optimally if the probability that an individual is infectious can be estimated—but this is itself a challenging problem due to the high dimensional observation space. Motivated by this observation, we formulate our solution approach: we use a Convolutional Neural Network (CNN) to estimate the probability of infectiousness for each individual in a cluster, and this output, along with an egocentric state representation for each contact, serves as the state for the RL agent.

3.1 POMDP Formulation

We define a POMDP [11] as $\langle S, A, R, P, \Omega, O, \gamma, S_0 \rangle$, where S and A are the state and action spaces, respectively, $R : S \times A \to \mathbb{R}$ is the reward function, $P : S \times A \to \Delta S$ is the transition function, Ω is the observation state, $O : S \times A \to \Delta\Omega$ is the observation probabilities, $\gamma \in [0, 1]$ is the discount factor, and $S_0 : \Delta S$ is the distribution of initial states.

We describe how to interpret the control problem of the previous section as a POMDP. The cluster infection trajectory and the current time t are contained in the state. The only aspect of the state that

changes when an action is taken is the time t. As this is a POMDP, the state is not observable by the agent directly—instead, the agent has to rely on action-dependent observations. The observation emitted contains all of the information that is always available regardless of action (the time t, the symptom $o_n^{(t)}$ for each contact). Additionally, if an action with a sensing component is taken, it contains the sensing return (e.g., positive or negative, PCR cycle count). The action set is combinatorially structured—we select one action for each contact in the cluster. If we have N contacts, we have an action space of size $|A_p|^N$, where $|A_p|$ is the number of actions available for each contact. The reward can be calculated for any policy from Obj. 1 for any cluster infection trajectory.

In principle, solving this POMDP results in the optimal control policy. In practice, solving it exactly is not possible due to the high computational complexity of the best-known algorithms. Peng et al. [28] shows how to solve this POMDP with a belief state. They show that, if the posterior probability of infection can be calculated exactly (i.e., the probability of infection of each contact given all observations so far), the optimal policy has a threshold-type form. We utilize a supervised learning encoder to estimate the posterior probability of infection.

3.2 Supervised Belief Encoder

Let $o_{[N]}^{[t]} = \{o_n^{(t')} : 0 \le t', n \in [N]\}$ represent all symptom observations for a cluster up to day t; $p_n^{(t)} = P\left(i_n^{(t)} = 1 \mid o_{[N]}^{[t]}\right)$ represent the posterior probability that contact n is infected given the symptom observations so far.

The generator for the library of cluster infection trajectories provides us with a large number of $\left(o_{[N]}^{[t]}, i_{[N]}^{(t)}\right)$ pairs (where $i_{[N]}^{(t)}$ is the infectiousness state for all contacts in a cluster). A natural question is whether we can produce useful estimates of $p_n^{(t)}$ from $o_{[N]}^{[t]}$ using a supervised learning approach. While it is possible for RL to produce strong policies without explicitly computing $p_n^{(t)}$, it is inefficiently positioned to do so because the information about $i_n^{(t)}$ must be inferred from the reward signal.

A key question for applying supervised learning is how to represent the observation space $o_{[N]}^{[t]}$. We have two desiderata. First, we would like the representation size to not vary with cluster size. We can also achieve this property in the RL agent, resulting in an agent that simultaneously be deployed across all cluster sizes, which makes both training and deployment simpler. Second, there is an advantage to using a representation that inherently accounts for the permutation equivariance that arises due to the ordering of individuals, i.e., if we permute the order of individuals in an observation, our supervised learning model would ideally predict $i_{[N]}^{(t)}$, but with the same permutation applied.

After testing several representations that satisfy these properties, we arrive at the $9 \times T$ matrix shown in Fig. 3 (recall T is the trajectory length). This is an egocentric representation of the observation—it is from the perspective of a particular contact and contains all information gathered so far. We train the supervised learning model f to produce output of dimension $[0, 1]^T$, i.e., given $o_{[N]}^{[t]}$ for some $t \le T$, predict $p_n^{(t')}$ for all $t' \in [T]$.

We show that this representation can achieve an AUC of 0.95 for the SARS-CoV-2 cluster infection trajectory generator if an appropriate architecture is selected. We experiment with a variety of supervised learning model architectures in Tab. 2 and find that Convolutional Neural Networks (CNNs) are generally the most effective. In single-layer CNN architectures, we find that larger 2D convolutions tend to achieve higher AUC, and that a single convolution layer followed by a linear layer performs just as well as deeper architectures—this setup of a (5, 2) 2D convolution followed by a linear layer is what we use in the experiments below.[1]

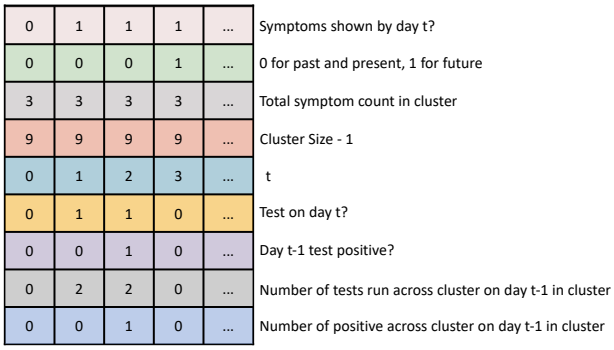

Figure 3: The observation representation used for supervised learning, shown on a cluster of size 10 after observing the outcome of t=2.

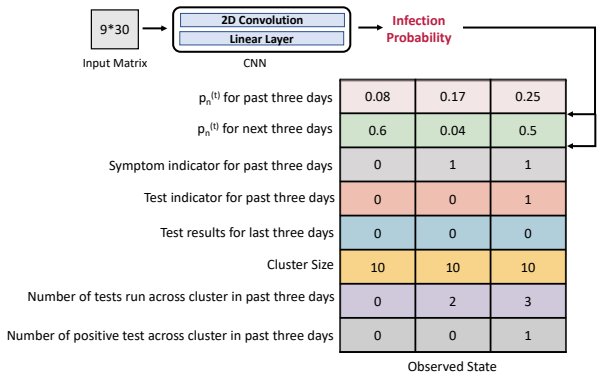

Figure 4: The supervised learning (CNN) output is used as input to the RL state which prioritizes immediately relevant information.

3.3 Reinforcement Learning

To make RL effective, we will develop a compact state representation that includes supervised learning outputs. As with supervised learning, we want the RL state representation to have the same size for all clusters and to naturally encode permutation invariance.

[1]These experiments were performed on an earlier representation, which only had five rows. In the following sections, we use (9, 2) 2D convolution followed by a linear layer.

Table 2: We find that two-layer architectures using a 2D convolution followed by a linear layer achieves performance on par with larger models.

		Cluster size = 4	8	16	32
1 Layer	Conv1d (5,2)	0.798	0.807	0.823	0.830
	Conv1d (5,3)	0.814	0.830	0.835	0.839
	Conv2d (5,2)	0.800	0.814	0.827	0.830
	Conv2d (5,3)	0.832	0.820	0.838	0.840
	Conv2d (5,4)	0.858	0.849	0.843	0.859
	Conv2d (5,5)	0.864	0.895	0.893	0.893
2 Layer	Conv1d (5,2) Conv1d (1,2)	0.824	0.830	0.833	0.840
	Conv2d (5,3) Conv2d (1,3)	0.883	0.903	0.898	0.897
	Conv2d(5,2) Linear Layer	0.955	0.960	0.947	0.961
	Conv2d (5,3) Linear Layer	0.951	0.960	0.940	0.964
3 Layer	Conv1d (5,3) Conv1d (1,3) Linear Layer	0.958	0.957	0.950	0.961
4 Layer	Conv1d (4,3) Conv1d (2,3) Conv1d (1,3) Linear Layer	0.958	0.958	0.953	0.965
xgboost		0.763	0.732	0.804	0.770

In doing so, we can also reduce the action space size from combinatorial by factorizing across the contacts, i.e., training a single policy which is applied separately to each contact—this is without loss of performance in the setting without sensing actions if $p_n^{(t)}$ is correct. The representation we use is a 8×3 matrix shown in Fig. 4. As with the supervised learning representation, it is egocentric and time-specific.

The following training procedure for the RL policy and supervised belief encoder is used. First, a fixed but stochastic *seed policy* generates 200 cluster infection trajectories and sensing actions, which are used to train the supervised belief encoder. These trajectories, along with the encoder outputs, are then used to train the RL policy. If performance is sufficient, terminate. If it is not (which happens when the current RL policy produces actions with a distribution that is too different from the seed policy), use the current RL policy to select sensing actions, continue training the encoder using these new actions, and then retrain the RL policy with the new supervised learning outputs. This process can be repeated any number of times, or the RL policy and the encoder can be trained in parallel.

For RL training, we use Proximal Policy Optimization (PPO) [31] and Deep Q Learning (DQN) [24, 25]. The RL policy is a multilayer perceptron with two layers and 128 hidden units (a standard architecture for PPO and DQN). In experiments, for each of six different policy initializations (seed), train for 80000 environment interactions, and pick the best based on 100 evaluation runs. All training is performed on an Intel Xeon E5-2680 v4 with 28 cores and 128 GB of RAM [5], and a single RL training run takes 3 hours on average.

Note that the optimal RL policy depends on α_2 and α_3 and thus a different policy must be trained for each different setting. However, the encoder depends on (α_2, α_3) only indirectly due to the sensing actions it sees in training. Thus, the same cluster infection trajectories and encoder can be reused across multiple RL training runs.

3.4 Restless Multi-Armed Bandit

Access to actions may be limited by available supply, i.e., not just cost. Thus, it is desirable to develop methods that can allocate a finite number of available resources across clusters. We provide a preliminary exploration of how Restless Multi-Armed Bandits (RMABs) [34] could be applied to this purpose.

RMABs define a sequential decision model where an agent aims to maximize rewards over a large group of independent Markov decision processes (MDPs) [30] with a shared budget. As true states of arms are not directly observable, each arm is a POMDP and we can rewrite it as a fully observable belief-state MDP [20], allowing for a direct representation as an RMAB with multiple actions [15]. RMABs have gained wide interest over decades in the field of resource allocation tasks [8, 12], which naturally inspires us that RMABs could be applied to allocate NPIs.

Each arm is a belief-state MDP and ith arm can be described as a tuple $\langle S', A, R, P, \gamma, S_0' \rangle$:

- S': state of the arm, which is a set containing all belief states of individuals in the cluster.
- A: actions. It is the number of tests allocated to this arm. The total number of tests among all arms is limited.
- R: reward function.
- P: state transition function.
- γ: reward discount factor.
- S_0': the initial state of S.

It is worth noting that these actions are applied to one arm on the cluster level. Inside the cluster, RL could be applied to learn a policy to maximize the total rewards under a certain budget.

Under a limited budget of total tests at time step t, the agent of RMAB is striving to maximize the sum of the following reward function: $(-S_1 - \alpha_2 \times S_2)/N$, where S_1 and S_2 are the same in the equation 1. Note that $\alpha_3 \times S_3$ is not involved here due to that the total cost itself is a parameter in the RMAB problem. We use a Lagrange relaxation approach to solve the RMAB, relaxing the budget constraint and add a subsidy or penalty λ to the Bellman objective function. Then, by decoupling the λ and the value function, our final goal is to solve:

$$J(s,\lambda) = \min_{V^i(s^i,\lambda),\lambda} \frac{\lambda B}{1-\gamma} + \sum_{i=0}^{N-1} \mu^i\left(s^i\right) V^i\left(s^i,\lambda\right)$$

$$\text{s.t. } V^i\left(s^i,\lambda\right) \geq r^i\left(s^i\right) - \lambda c_j + \gamma \sum_{s^{i\prime}} T\left(s^i, a_j^i, s^{i\prime}\right) V^i\left(s^{i\prime},\lambda\right)$$

$$\forall i \in \{0,\ldots,N-1\}, \quad \forall s^i \in \mathcal{S}, \quad \forall a_j \in \mathcal{A}, \text{ and } \lambda \geq 0$$

where B is the action budgets at each time step, γ is the discount factor, c_j is the action cost and $\mu^i(s^i) = 1$ if s^i is the starting state of arm i and 0 otherwise. By leveraging the state-of-art LP solution, the value function $V(s,\lambda)$ is known, and then we compute action-value function $Q(s,a,\lambda_{min})$. With the action-value function, a policy is generated without effort.

Table 3: Obj. 1 multiplied by 100 (higher is better). RLSL finds the best policy in all settings except $\alpha_2 = 0.05$ and $\alpha_3 = 0.2$, where RLSL, Threshold, Symptom-Based Quarantine and Always Quarantine are essentially tied—testing appears to provide no benefit here. The largest gaps between RLSL and others occur when α_2 is large and α_3 is small.

	$\alpha_2 = 0.05$ $\alpha_3 = 0.01$	$\alpha_2 = 0.05$ $\alpha_3 = 0.1$	$\alpha_2 = 0.05$ $\alpha_3 = 0.2$	$\alpha_2 = 0.1$ $\alpha_3 = 0.01$	$\alpha_2 = 0.1$ $\alpha_3 = 0.1$	$\alpha_2 = 0.1$ $\alpha_3 = 0.2$	$\alpha_2 = 0.2$ $\alpha_3 = 0.01$	$\alpha_2 = 0.2$ $\alpha_3 = 0.1$	$\alpha_2 = 0.2$ $\alpha_3 = 0.2$
RLSL	**−88.92 ± 1.68**	**−105.20 ± 1.03**	−112.78 ± 2.23	**−94.58 ± 2.34**	**−109.52 ± 2.54**	**−116.37 ± 3.17**	**−102.21 ± 2.39**	**−124.88 ± 2.82**	**−133.37 ± 3.35**
Threshold	−107.36 ± 7.67	−107.36 ± 7.67	**−107.36 ± 7.67**	−130.50 ± 4.76	−130.50 ± 4.76	−130.50 ± 4.76	−157.03 ± 4.37	−157.03 ± 4.37	−157.03 ± 4.37
Symptom-Based Quarantine	−113.58 ± 2.85	−113.58 ± 2.85	−113.58 ± 2.85	−134.32 ± 6.88	−134.32 ± 6.88	−134.32 ± 6.88	−158.42 ± 10.24	−158.42 ± 10.24	−158.42 ± 10.24
14-Day Quarantine	−141.50 ± 13.80	−141.50 ± 13.80	−141.50 ± 13.80	−175.50 ± 13.34	−175.50 ± 13.34	−175.50 ± 13.34	−276.50 ± 11.70	−276.50 ± 11.70	−276.50 ± 11.70
CDC 12/20	−203.67 ± 5.56	−213.40 ± 6.54	−228.80 ± 5.28	−235.37 ± 5.36	−246.30 ± 5.04	−277.23 ± 5.88	−310.77 ± 4.18	−330.67 ± 5.70	−344.10 ± 6.56
Always Quarantine	−110.50 ± 2.95	−110.50 ± 2.95	−110.50 ± 2.95	−215.44 ± 2.66	−215.44 ± 2.66	−215.44 ± 2.66	−425.91 ± 1.76	−425.91 ± 1.76	−425.91 ± 1.76
No Quarantine	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01	−249.33 ± 7.01

4 EXPERIMENTS

We compare different control policies in our SARS-CoV-2 cluster infection trajectory generator to evaluate our policy search procedure.

For α_2, we use three values of $0.05, 0.1$ and 0.2. For α_3, we use values of $0.01, 0.1$, and 0.2.

Table 4: S_1, S_2 and S_3 per contact across different cluster sizes (lower is better and - indicates 0), where RLSL and Threshold are set to $\alpha_2 = \alpha_3 = 0.1$. RLSL tests slightly more than CDC 12/20 (1.55 vs. 0.918 tests per contact) to dramatically decrease S_1 and S_2.

	S_1	S_2	S_3
RLSL	0.459 ± 0.017	4.195 ± 0.148	1.553 ± 0.035
RLSL ($N = 4$)	0.391 ± 0.017	3.907 ± 0.013	1.430 ± 0.007
RLSL ($N = 8$)	0.419 ± 0.014	4.600 ± 0.032	1.689 ± 0.010
RLSL ($N = 16$)	0.524 ± 0.071	3.821 ± 0.049	2.017 ± 0.056
RLSL ($N = 32$)	0.541 ± 0.043	4.031 ± 0.029	2.043 ± 0.019
Threshold	1.198 ± 0.040	1.751 ± 0.013	-
Threshold ($N = 4$)	0.973 ± 0.059	1.976 ± 0.057	-
Threshold ($N = 8$)	1.009 ± 0.044	1.659 ± 0.036	-
Threshold ($N = 16$)	1.321 ± 0.048	1.617 ± 0.018	-
Threshold ($N = 32$)	1.438 ± 0.056	1.762 ± 0.028	-
Symptom-Based Quarantine	1.413 ± 0.036	0.228 ± 0.005	-
14-Day Quarantine	0.753 ± 0.007	10.274 ± 0.040	-
CDC 12/20	1.273 ± 0.068	7.334 ± 0.084	0.918 ± 0.004
Always Quarantine	-	21.788 ± 0.085	-
No Quarantine	2.481 ± 0.046	-	-

4.1 Comparison Policies

The policies introduced by this paper are: **Threshold** is the threshold-type policy suggested in Sec. 3.1 (which does not use sensing actions); and **RLSL**, our primary contribution, combining RL with a supervised learning encoder.

We compare several benchmark policies. **Symptom-Based Quarantine** quarantines if an individual exhibits symptoms on the day before the observed day and otherwise does not. **14-day Quarantine** quarantines individuals from the initial day they exhibit symptoms until either 14 days have passed or until they no longer exhibit symptoms, whichever is later. **CDC 12/20** is a complex policy based on late 2020 (CDC) guidelines [4]. It quarantines symptomatic contacts for 10 days. Asymptomatic contacts are tested on day 5 and released on day 8 if the test is negative and they have no symptoms. If the test is positive, they are quarantined for 14 days after the exposure. **Always Quarantine** always performs the quarantine action. **No Quarantine** always performs the null action.

Our experimental results report the average objective value and standard error taken over 30 random clusters.

4.2 Analysis

We first show the performance among all policies in Tab. 3. We find that RLSL is able to find the strongest policy in all settings except $\alpha_2 = 0.05$ and $\alpha_3 = 0.2$, where RLSL, Threshold, Symptom-Based Quarantine and Always Quarantine are all competitive (with perhaps an edge to Threshold). Threshold is the second strongest performer in all other settings. RLSL can achieve large improvements over the benchmarks of up to 35%. We see improvements across all settings, but they are largest when α_2 is large and α_3 is small, i.e., where tests can be leveraged and the decision to quarantine or not is challenging.

The best benchmark policy is Symptom-Based Quarantine except when $\alpha_2 = 0.05$, where Always Quarantine is slightly better. Symptom-Based Quarantine is often competitive with Threshold, despite the presence of extensive asymptomatic and presymptomatic transmission, as well as symptoms without infection, in the generator.

We report objective values broken out by component and by cluster size as measured per contact, where $\alpha_2 = \alpha_3 = 0.1$ is used to train RLSL and set the parameters (Tab. 4). Here we can intuitively grasp the effects of the different policies. 14-Day Quarantine, CDC 12/20, and Always Quarantine quarantine widely, resulting in $S_2 \approx 10.3, 7.3, 21.8$ days of quarantine without infection per contact (respectively) and achieving $S_1 \approx 0.75, 1.27, 0.0$ as a result. Symptom-based quarantine takes a different approach, preventing only 57% of transmission days, but incurring minimal costs to do so. RLSL uses about 50% more tests than CDC 12/20, but reduces S_1 to about 40% lower than 14-day quarantine with 60% less S_2 cost. Threshold is simply more efficient than non-testing competitors at the trade-off between S_1 and S_2 by allowing S_1 to be larger to vastly reduce S_2.

Table 5: RLSL (PPO) and Threshold always achieve dramatically higher objective values than RL Only, which has no supervised learning component. For $\alpha_2 = 0.05$ and $\alpha_3 \in \{0.1, 0.2\}$, RLSL with no sensing action scores slightly better than standard RLSL. In many settings, we are able to find decision tree policies that perform similarly to the RLSL or Threshold policies, which are much more complex.

	$\alpha_2 = 0.05$ $\alpha_3 = 0.01$	$\alpha_2 = 0.05$ $\alpha_3 = 0.1$	$\alpha_2 = 0.05$ $\alpha_3 = 0.2$	$\alpha_2 = 0.1$ $\alpha_3 = 0.01$	$\alpha_2 = 0.1$ $\alpha_3 = 0.1$	$\alpha_2 = 0.1$ $\alpha_3 = 0.2$	$\alpha_2 = 0.2$ $\alpha_3 = 0.01$	$\alpha_2 = 0.2$ $\alpha_3 = 0.1$	$\alpha_2 = 0.2$ $\alpha_3 = 0.2$
RLSL	**−88.92 ± 1.68**	−105.20 ± 1.03	−112.78 ± 2.23	−94.58 ± 2.34	**−109.52 ± 2.54**	**−116.37 ± 3.17**	**−102.21 ± 2.39**	**−124.88 ± 2.82**	**−133.37 ± 3.35**
RLSL (always test)	−94.47 ± 0.97	−291.50 ± 3.97	−518.10 ± 3.38	−96.48 ± 3.14	−292.30 ± 2.86	−531.90 ± 4.48	−107.90 ± 4.23	−320.80 ± 4.70	−531.50 ± 3.97
RLSL (never test)	−98.67 ± 2.33	**−98.67 ± 2.33**	**−98.67 ± 2.33**	−128.25 ± 2.50	−128.25 ± 2.50	−128.25 ± 2.50	−148.41 ± 6.44	−148.41 ± 6.44	−148.41 ± 6.44
RL Only	−150.50 ± 5.19	−211.80 ± 7.81	−228.40 ± 6.84	−178.90 ± 7.63	−244.80 ± 8.06	−320.90 ± 9.70	−202.90 ± 13.32	−294.20 ± 13.34	−333.20 ± 7.55
Threshold	−107.36 ± 7.67	−107.36 ± 7.67	−107.36 ± 7.67	−130.50 ± 4.76	−130.50 ± 4.76	−130.50 ± 4.76	−157.03 ± 4.37	−157.03 ± 4.37	−157.03 ± 4.37
Decision Tree	−103.15 ± 4.20	−104.96 ± 3.12	**−97.46 ± 3.66**	**−91.10 ± 1.30**	−121.90 ± 2.66	−143.25 ± 3.80	−127.43 ± 2.52	−131.53 ± 2.89	−161.23 ± 3.03

Table 6: With DQN setting, standard RLSL always works better than RLSL with no sensing action and with always sensing action, except for $\alpha_2 = 0.05$ and $\alpha_3 \in 0.2$

.

	$\alpha_2 = 0.05$ $\alpha_3 = 0.01$	$\alpha_2 = 0.05$ $\alpha_3 = 0.1$	$\alpha_2 = 0.05$ $\alpha_3 = 0.2$	$\alpha_2 = 0.1$ $\alpha_3 = 0.01$	$\alpha_2 = 0.1$ $\alpha_3 = 0.1$	$\alpha_2 = 0.1$ $\alpha_3 = 0.2$	$\alpha_2 = 0.2$ $\alpha_3 = 0.01$	$\alpha_2 = 0.2$ $\alpha_3 = 0.1$	$\alpha_2 = 0.2$ $\alpha_3 = 0.2$
RLSL	**−47.57 ± 1.2**	**−69.66 ± 1.77**	−91.25 ± 1.33	−63.81 ± 1.03	**−77.06 ± 1.42**	**−101.03 ± 1.13**	**−89.49 ± 2.69**	**−100.61 ± 2.45**	**−116.54 ± 3.07**
RLSL (always test)	−56.98 ± 2.08	−268.08 ± 2.78	−487.88 ± 1.79	−65.04 ± 2.03	−277.02 ± 2.54	−508.18 ± 2.39	−85.87 ± 1.33	−284.95 ± 3.66	−511.73 ± 3.07
RLSL (never test)	−90.36 ± 1.37	−90.36 ± 1.37	**−90.36 ± 1.37**	−115.66 ± 1.29	−115.66 ± 1.29	−115.66 ± 1.29	−133.47 ± 1.57	−133.47 ± 1.57	−133.47 ± 1.57

Table 7: The generalized RLSL works not as well as the standard RLSL.

	$\alpha_3 = 0.01$	$\alpha_3 = 0.03$	$\alpha_3 = 0.05$	$\alpha_3 = 0.07$	$\alpha_3 = 0.1$	$\alpha_3 = 0.15$	$\alpha_3 = 0.02$
RLSL(generalised)	−99.31 ± 4.77	−111.16 ± 5.58	−118.80 ± 4.22	−126.77 ± 4.85	−135.16 ± 4.80	−137.42 ± 4.80	−138.51 ± 6.67
RLSL(fixed α_3)	−63.81 ± 1.03	−67.08 ± 1.58	−71.98 ± 1.77	−75.04 ± 2.03	−77.06 ± 1.42	−88.48 ± 2.09	−101.03 ± 1.13

In an ablation study (Tab. 5), we gain a more detailed view into the operation of the RLSL policy. We see that the introduction of the SL outputs to the RL state results in vastly improved performance in all tested scenarios compared to RL Only, which uses the state representation of Fig. 4 without the first two rows. RL Only performs worse than Symptom-Based Quarantine in all settings. RLSL (never test) and the decision tree policy (described below) sometimes outperform RLSL, indicating that the training procedure could still be improved.

We also use DQN for RLSL training. Compared to the results presented in Tab. 5 and Tab. 6, DQN outperformed PPO across all versions of RLSL. Similarly, as shown in Table 6, standard RLSL outperforms both the "always test" and "never test" strategies, with the exception of settings where $\alpha_2 = 0.05$ and $\alpha_3 = 0.2$.

Interpretable Policy. In contrast to the benchmarks, both RLSL and Threshold require neural network outputs, i.e., computation, to run. We experiment with a procedure to convert RLSL's policy into a decision tree that can be distributed on paper. We use nine interpretable features: days since exposure, days since positive test, days since symptom, yesterday's test result (0 if no test), whether tested yesterday, the number of symptomatic contacts in the cluster today, the number of positive tests in the cluster so far normalized by cluster size, and the cluster size. Using these features, we train a decision tree to predict RLSL's action. We consider five types of actions: (1) quarantine and no test, (2) quarantine and test, (3) no quarantine and test, (4) no quarantine and no test, (5) test and, if positive, quarantine; otherwise, no quarantine. The results are shown in the last row of Table 5. In three cases, the decision tree policy is at least as strong as the policy produced by RLSL. We believe that these policies can be further improved (see Discussion).

4.3 Sensing Action Constraints

In this section, we will show the experiment we did to explore the possibility of solving the limited sensing action problems.

We assume that we have a test budget and we need to allocate all tests to different clusters. From the reward function, we can know that when α_3 is larger, the agent is more inclined to choose actions 0 and 1 that are not tested. Based on this, we can find a specific $alpha_3$ that makes the total number of tests for all clusters meet the test budget.

Thus, we train a new generalized model with various α_3 and with fixed $\alpha_2 = 0.1$. This is a single-cluster version. It takes the original observation + α_3 as the input and the α_3 is sampled from a uniform distribution over $[0.01, 0.3]$. We compared the performance of the generalized RLSL with the original one Tab. 7. Note that we train generalized RLSL with different α_3 and evaluate it with specific α_3.

Additionally, we show the relationship between α_3 and the average daily test number in two clusters (cluster size = 20, 30) in Figure 5. Indeed, the amount of tests consumed decreases when α_3 is larger.

5 DISCUSSION AND CONCLUSION

This work aims to develop a generic multi-objective optimization approach for cluster-level optimization of NPIs. We formulate this

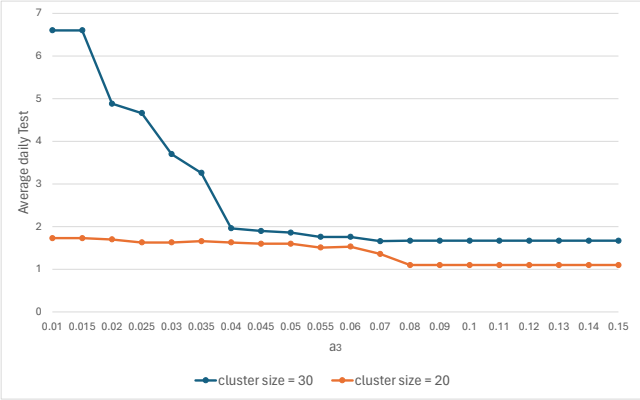

Figure 5: The number of average daily tests gets lower when α_3 is larger.

problem as a POMDP that we solve with RL, leveraging a supervised learning encoder, a permutation equivariant state representation, and a factorized action space. We demonstrate the potential of our approach—in a simple agent-based model of SARS-CoV-2, we can achieve substantially higher objective values than baseline policies. Our optimized policy can outperform all benchmarks by up to 27%. Moreover, the developed policies exhibit applicability across various cluster sizes and can be trained on consumer hardware, the fact that these policies can be implemented on consumer-grade hardware enhances their practicality and scalability, making them accessible for broader real-world application. In addition, our approach has shown promise in formulating strong and interpretable policies across multiple settings. This aspect is particularly important as it contributes to the transparency and understandability of the policies, which are crucial factors in public health interventions.

Our agent-based model represents a classic probabilistic framework for simulating disease dynamics, which can be applied to other epidemics. It is built using key disease parameters derived from various sources during a crisis, incorporating the inherent uncertainties in these estimates. In the early stages of a crisis, we emphasize the importance of focusing on superspreading dynamics, given their significant impact on the effectiveness of interventions, as demonstrated in our findings. Utilizing this model, we can create an environment based on a branching process, which is then optimized using the approach outlined in this paper.

Our approach combines RL and SL techniques. RL is a powerful optimization technique, but it has some drawbacks. One significant limitation is its inherent difficulty in exploiting problem structure. In this setting, the underlying POMDP has a substantial structure in the belief state that can be exploited to greatly simplify the learning task. We extract this information using a combination of manual insight and brute-force supervised learning. It is an open question as to whether RL techniques can learn to discover such structures through experimentation. Another challenge with RL is its well-known instability during the training phase [7]. We attempt to reduce this instability by using multiple initializations, but we still see evidence of it in the $\alpha_2 = 0.05$, $\alpha \geq 0.1$ settings, where reducing the action space produces higher objective values. Despite these challenges, we believe the advantages of using RL, especially in terms of its capability to provide high-quality solutions for complex optimization problems, outweigh its limitations.

While no existing work uses the same modeling framework or policy search space as ours, in some cases, we can compare our results. The model of Perrault et al. [29] is most similar to ours, and the risk-based quarantine (RBQ) policies they evaluate can be compared to Threshold and RLSL, but due to different assumptions, the amount of reduction in transmission they achieve relative to the status quo is much less. This is because they assume that individuals self-isolate even in the absence of an intervention and that some individuals drop out of quarantine. Threshold can be viewed as a policy that generalizes the RBQ approach they suggest, in that Threshold generates an infinite family of optimal risk-based families for different risk tolerance levels. However, Threshold's policies are less interpretable than RBQ.

Kucharski et al. [18] provides another point to compare the effectiveness of contact tracing. In their setting, combining self-isolation, household quarantine, and comprehensive manual contact tracing of all contacts resulted in a 64% reduction in disease transmission, which is equivalent to S_1. In our setting, we find a reduction of 69.95% for two-week quarantine, suggesting that interventions have a comparable impact in our (much simpler) model for COVID-19 transmission.

Our current work focuses on developing an effective policy for the coordination of a limited number of tests across clusters. We utilize the Restless Multi-Armed Bandit. For each time T, there is a test budget, and each arm allocates the test to each cluster. Our model can be viewed as having two levels of agents. For cluster level, the agent needs to determine how many tests are required for each cluster each day. For the individual level, all settings are the same as the current one. From our results so far, we know that different α_3 will cause each individual agent to make different choices. Therefore, α_3 can be grid searched, and if the total number of tests per cluster under the current α_3 exceeds the test budget, α_3 is increased, and if not, α_3 is reduced.

While our research provides valuable insights into the application of RMABs, there are several avenues for future exploration and improvement. One challenge would be how to train a generalized multi-cluster multi-agent model with α_3. Our preliminary results indicate that this generalized model has some promising characteristics, but we have not yet tested it in resource allocation experiments.

REFERENCES

[1] Qifang Bi, Yongsheng Wu, Shujiang Mei, Chenfei Ye, Xuan Zou, Zhen Zhang, Xiaojian Liu, Lan Wei, Shaun A Truelove, Tong Zhang, et al. 2020. Epidemiology and transmission of COVID-19 in 391 cases and 1286 of their close contacts in Shenzhen, China: a retrospective cohort study. *The Lancet infectious diseases* 20, 8 (2020), 911–919.

[2] Diana Buitrago-Garcia, Dianne Egli-Gany, Michel J Counotte, Stefanie Hossmann, Hira Imeri, Aziz Mert Ipekci, Georgia Salanti, and Nicola Low. 2020. Occurrence and transmission potential of asymptomatic and presymptomatic SARS-CoV-2 infections: A living systematic review and meta-analysis. *PLoS medicine* 17, 9 (2020), e1003346.

[3] Lisa Caulley, Martin Corsten, Libni Eapen, Jonathan Whelan, Jonathan B Angel, Kym Antonation, Nathalie Bastien, Guillaume Poliquin, and Stephanie Johnson-Obaseki. 2021. Salivary detection of COVID-19. *Annals of internal medicine* 174, 1 (2021), 131–133.

[4] CDC. 2020. COVID-19: When to Quarantine | CDC. https://web.archive.org/web/20201231011236/https://www.cdc.gov/coronavirus/2019-ncov/if-you-are-

sick/quarantine.html.

[5] Ohio Supercomputer Center. 1987. Ohio Supercomputer Center. http://osc.edu/ark:/19495/f5s1ph73

[6] Xingran Chen, Hesam Nikpey, Jungyeol Kim, Saswati Sarkar, and Shirin Saeedi-Bidokhti. 2023. Containing a spread through sequential learning: to exploit or to explore? *arXiv preprint arXiv:2303.00141* (2023).

[7] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning* 110, 9 (2021), 2419–2468.

[8] Pedro Cesar Lopes Gerum, Ayca Altay, and Melike Baykal-Gürsoy. 2019. Data-driven predictive maintenance scheduling policies for railways. *Transportation Research Part C: Emerging Technologies* 107 (2019), 137–154.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[10] Robert Hinch, Will Probert, Anel Nurtay, Michelle Kendall, Chris Wymant, Matthew Hall, Katrina Lythgoe, Ana Bulas Cruz, Lele Zhao, Andrea Stewart, et al. 2020. Effective configurations of a digital contact tracing app: a report to NHSX. *Retrieved July* 23 (2020), 2020.

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

[12] Kesav Kaza, Varun Mehta, Rahul Meshram, and S. N. Merchant. 2018. Restless bandits with cumulative feedback: Applications in wireless networks. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–6.

[13] Matt J Keeling, T Deirdre Hollingsworth, and Jonathan M Read. 2020. Efficacy of contact tracing for the containment of the 2019 novel coronavirus (COVID-19). *J Epidemiol Community Health* 74, 10 (2020), 861–866.

[14] Cliff C Kerr, Robyn M Stuart, Dina Mistry, Romesh G Abeysuriya, Katherine Rosenfeld, Gregory R Hart, Rafael C Núñez, Jamie A Cohen, Prashanth Selvaraj, Brittany Hagedorn, et al. 2021. Covasim: an agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology* 17, 7 (2021), e1009149.

[15] Jackson A Killian, Andrew Perrault, and Milind Tambe. 2021. Beyond" to act or not to act": Fast lagrangian approaches to general multi-action restless bandits. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 710–718.

[16] V Kompella, R Capobianco, S Jong, J Browne, S Fox, L Meyers, P Wurman, and P Stone. 2020. Reinforcement learning for optimization of COVID-19 mitigation policies. In *2020 AAAI Fall Symposium on AI for Social Good, AI4SG 2020*.

[17] Mirjam E Kretzschmar, Ganna Rozhnova, and Michiel Van Boven. 2021. Isolation and contact tracing can tip the scale to containment of COVID-19 in populations with social distancing. *Frontiers in Physics* (2021), 677.

[18] Adam J Kucharski, Petra Klepac, Andrew JK Conlan, Stephen M Kissler, Maria L Tang, Hannah Fry, Julia R Gog, W John Edmunds, Jon C Emery, Graham Medley, et al. 2020. Effectiveness of isolation, testing, contact tracing, and physical distancing on reducing transmission of SARS-CoV-2 in different settings: a mathematical modelling study. *The Lancet Infectious Diseases* 20, 10 (2020), 1151–1160.

[19] Shengjie Lai, Nick W Ruktanonchai, Liangcai Zhou, Olivia Prosper, Wei Luo, Jessica R Floyd, Amy Wesolowski, Mauricio Santillana, Chi Zhang, Xiangjun Du, et al. 2020. Effect of non-pharmaceutical interventions to contain COVID-19 in China. *nature* 585, 7825 (2020), 410–413.

[20] Aditya Mate, Jackson A. Killian, Haifeng Xu, Andrew Perrault, and Milind Tambe. 2020. Collapsing Bandits and Their Application to Public Health Interventions. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

[21] Thomas McAndrew, Allison Codi, Juan Cambeiro, Tamay Besiroglu, David Braun, Eva Chen, Luis Enrique Urtubey De Cèsaris, and Damon Luk. 2022. Chimeric forecasting: combining probabilistic predictions from computational models and human judgment. *BMC Infectious Diseases* 22, 1 (2022), 833.

[22] Eli Meirom, Haggai Maron, Shie Mannor, and Gal Chechik. 2021. Controlling graph dynamics with reinforcement learning and graph neural networks. In *International Conference on Machine Learning*. PMLR, 7565–7577.

[23] Michela Meister and Jon Kleinberg. 2023. Optimizing the order of actions in a model of contact tracing. *PNAS Nexus* (2023).

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[26] Han-Ching Ou, Haipeng Chen, Shahin Jabbari, and Milind Tambe. 2021. Active Screening for Recurrent Diseases: A Reinforcement Learning Approach. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 992–1000.

[27] Han-Ching Ou, Arunesh Sinha, Sze-Chuan Suen, Andrew Perrault, Alpan Raval, and Milind Tambe. 2020. Who and when to screen: Multi-round active screening for network recurrent infectious diseases under uncertainty. In *Proceedings of 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Auckland New Zealand, 2020 May 9-13*. ACM.

[28] Xueqiao Peng, Jiaqi Xu, Xi Chen, Dinh Song An Nguyen, and Andrew Perrault. 2023. Using Reinforcement Learning for Multi-Objective Cluster-Level NPI Optimization. In *epiDAMIK 6.0: The 6th International workshop on Epidemiology meets Data Mining and Knowledge Discovery at KDD 2023*. https://openreview.net/forum?id=Ql4CuaB3-D

[29] Andrew Perrault, Marie Charpignon, Jonathan Gruber, Milind Tambe, and Maimuna S Majumder. 2020. Designing Efficient Contact Tracing Through Risk-Based Quarantining. *medRxiv* (2020), 2020–11.

[30] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc., USA.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

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

[33] Xutong Wang, Zhanwei Du, Emily James, Spencer J Fox, Michael Lachmann, Lauren Ancel Meyers, and Darlene Bhavnani. 2022. The effectiveness of COVID-19 testing and contact tracing in a US city. *Proceedings of the National Academy of Sciences* 119, 34 (2022), e2200652119.

[34] P. Whittle. 1988. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability* 25, A (1988), 287–298.