IISE Transactions on Healthcare Systems Engineering

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uhse21

State discretization for continuous-state MDPs in infectious disease control

Suyanpeng Zhang & Sze-chuan Suen

To cite this article: Suyanpeng Zhang & Sze-chuan Suen (27 Nov 2024): State discretization for continuous-state MDPs in infectious disease control, IISE Transactions on Healthcare Systems Engineering, DOI: 10.1080/24725579.2024.2428953

To link to this article: https://doi.org/10.1080/24725579.2024.2428953

Published online: 27 Nov 2024.

Submit your article to this journal

Article views: 12

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=uhse21

IISE TRANSACTIONS ON HEALTHCARE SYSTEMS ENGINEERING
https://doi.org/10.1080/24725579.2024.2428953

Taylor & Francis
Taylor & Francis Group

Check for updates

State discretization for continuous-state MDPs in infectious disease control

Suyanpeng Zhang and Sze-chuan Suen

Daniel J. Epstein Department of Industrial and Systems Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

ABSTRACT

Repeated decision-making problems may arise in the health policy context, such as infectious disease control for COVID-19 and other epidemics. These problems may sometimes be effectively solved using Markov decision processes (MDPs). However, the continuous or large state space of such problems for capturing infectious disease prevalence renders it difficult to implement tractable MDPs to identify the optimal disease control policy over time. We therefore develop an algorithm for discretizing continuous states for approximate MDP solutions in this context. We benchmark performance against a uniform discretization using both a synthetic example and an example of COVID-19 in Los Angeles County.

KEYWORDS

State discretization; Markov decision processes; infectious disease control; COVID-19; continuous state

1. Introduction

Public health officials often need to determine the optimal population health intervention policy over time even as the state and trajectory of disease are driven by complex dynamics. Many of these problems require making policy decisions sequentially over time, where the state may be represented using a continuous measure (e.g., the proportion of the population that is infected). For instance, during the COVID-19 pandemic, decision-makers needed to repeatedly set the start and end times of lockdowns that limited travel and interactions between individuals to reduce transmission without fully understanding the exact transmissibility of COVID-19. This sequential decision-making problem appears repeatedly in infectious disease control problems, as evidenced by prior literature on similar problems (Blower *et al.*, 2002; Fu *et al.*, 2022; Kaplan *et al.*, 1996; Matrajt *et al.*, 2021; Talbot *et al.*, 2005; Zhang *et al.*, 2011). Such problems often take into account underlying disease dynamics, which are uncertain or depend on a variety of complex social and biological factors.

A difficulty in solving repeated decision-making problems for infectious disease control is the complexity of infectious disease dynamics, which are typically represented using compartmental models and simulation-based models (Brauer, 2008; Kopec *et al.*, 2010). Such models are difficult to use for repeated decision-making problems as one often needs to evaluate the model repeatedly to identify an optimal policy for disease control, which may require a significant investment of computational time, as there is no closed-form solution.

While there are sophisticated means to identify optimal policies, these techniques have their own challenges. For instance, the maximum principal approach (Goenka *et al.*, 2014; Piguillem & Shi, 2022; Pontryagin, 2018) offers a solution framework for optimal control issues under differential equation systems. However, its application becomes increasingly challenging with a large number of states or policies. Such expansion complicates both the Hamiltonian and the differential equations system, thereby rendering the process of deriving analytical or numerical solutions complicated and time-consuming. Moreover, it is difficult to find the optimal solution when the problem is non-convex. Simulation optimization, which can handle complex systems, has also been used in disease contexts (Carson & Maria, 1997). However, this can also be computationally expensive and time-consuming. Furthermore, the quality of the solution highly depends on the search space and the heuristic function chosen, presenting challenges to its practical application. This problem can be formulated as a dynamic programming problem (Calvia *et al.*, 2023), but the continuous or large state space can create difficulties. Furthermore, while the infectious disease control problem can be formulated as a mixed-integer programming problem, the inherently non-linear nature of most disease dynamics—such as the compartmental model—introduces non-linear constraints into the formulation. Consequently, the problem becomes extremely challenging to solve (Bertsimas *et al.*, 2022), which limits its generalizability, especially as the disease dynamics become more complex.

Markov decision processes (MDPs) are also a commonly used method for repeated decision-making problems. MDPs allow for state transitions, which can be used to describe changes in disease/health states over time and allow for repeated decisions over time. Given current computing innovations, many MDPs of useful size can be solved

CONTACT Suyanpeng Zhang suyanpen@usc.edu Daniel J. Epstein Department of Industrial and Systems Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA.

© 2024 "IISE"

effectively using algorithms such as backward induction, value iteration, policy iteration, etc. MDPs can also be efficiently solved with non-convex problems.

However, incorporating dynamics from compartmental models and simulations into an MDP framework is challenging because disease models often use a continuous or large number of possible states (as the state usually represents a proportion of the whole population in certain statuses like infected, recovered, and hospitalized). Having a continuous state space makes the MDP problem difficult to solve since traditional MDP solution methods may then not work even for a short time horizon due to state-space explosion issues. For example, backward induction needs $|S|^2|A||T-1|$ multiplications. In the case of value iteration, each iteration carries a complexity of $\mathcal{O}(|S|^2|A|)$. In the case of policy iteration, each iteration carries a complexity of $\mathcal{O}(|S|^3+|S|^2|A|)$, and modified policy iteration requires $\mathcal{O}(k|S|^2+|S|^2|A|)$ per iteration (Puterman, 1994). For this reason, many traditional MDP studies in the healthcare field focus on finite-state decision-making problems like monitoring, treatment initiation, and disease testing and diagnosis (Ahn & Hornberger, 1996; Alagoz *et al.*, 2004, 2007; 2013; Capan *et al.*, 2017; Chhatwal *et al.*, 2010; David & Yechiali, 1985; Denton *et al.*, 2009; Hu *et al.*, 1996; Kreke, 2007; Kurt *et al.*, 2011; Lefèvre, 1981; Liu *et al.*, 2017; Magni *et al.*, 2000; Maillart *et al.*, 2008; Mason *et al.*, 2014; Shechter *et al.*, 2008; Suen *et al.*, 2018; Zhang *et al.*, 2021). Therefore, finding a good state discretization method that translates infectious disease dynamics onto a limited number of states improves computational efficiency and potentially widens the scope of MDP applications, particularly in the infectious disease space.

Uniform discretization is a traditional way of addressing continuous state problems. However, this methodology is suboptimal for addressing infectious disease control challenges. The heterogeneity in state visit frequencies—wherein some states (with extremely high prevalence) may remain unvisited and others (with lower prevalence) might be visited more frequently—renders uniform discretization inefficient. This approach may result in the overuse of discretization regions toward states that are less likely to be visited and an inadequate number of discretization regions for those with higher probabilities of being reached. How can we find a better way of discretizing the state space to closely represent the changes in health systems/disease? While many works have used various discretization methods to reduce state spaces (Lovejoy, 1991; Sandıkçı *et al.*, 2013), we take a novel approach that treats the state discretization problem as an optimization problem. This allows us to find the discretization that will provide a smaller discretized region in more likely visited states for a more accurate description of the true dynamics.

We will explore the above state discretization in the context of a disease control problem where states are used to describe the disease dynamics over a population, actions are implemented to prevent disease spread (lockdown, social distancing, face masks, and so on). States are assumed to be fully observable at each time period. Under this framework, we find a better way of discretizing states such that the discretized state space serves as a good proxy of the original state space. This paper addresses the challenge of formulating infectious disease control problems as MDPs by proposing a new algorithm for non-uniform state discretization that enables the discrete representation of infinite state spaces.

1.1. Contributions

We make several contributions in this study. We provide a novel algorithm for defining a non-uniform, discrete state space for infectious disease control problems that well approximates the original continuous state dynamics. Our algorithm exploits the likelihood of each state being visited in the system to more efficiently capture the transitions between states. Defining a discrete set of states from an originally continuous system allows us to incorporate infectious disease dynamics within frameworks that are better suited for discrete state spaces, such as MDPs. Finally, we demonstrate that our state space discretization allows for more accurate MDP outcomes through two numerical examples, one using a classic SIR compartmental model and one using the COVID-19 model of Los Angeles County.

The remainder of this paper is organized as follows: we review the related literature in Section 2, present the problem setup in Section 3, and provide the algorithms in Section 4. The numerical example is shown in Section 5. In Section 6, we conclude.

2. Literature review

2.1. Markov decision processes in healthcare applications

MDPs have a rich history in the field of operations research, with wide range of applications such as inventory management (Giannoccaro & Pontrandolfo, 2002), portfolio management (Bäuerle & Rieder, 2009), production and storage optimization (Arruda & do Val, 2008), and various others. Extensive research has been conducted to solve and understand the structure of MDPs, with notable contributions from works such as Puterman (1994) and Topkis (2011).

MDPs also find widespread application in the field of healthcare. They offer valuable insights and solutions to various health-related issues, including scheduling (Agrawal *et al.*, 2023), screening (Alagoz *et al.*, 2013; Chhatwal *et al.*, 2010; Maillart *et al.*, 2008; McNealey *et al.*, 2023; Shen *et al.*, 2024), sequential disease testing (Arruda *et al.*, 2019; Singh *et al.*, 2020), treatment initiation (Liu *et al.*, 2017; Otero-Leon *et al.*, 2023; Shechter *et al.*, 2008), and organ transplantation (Sandıkçı *et al.*, 2008, 2013; Zhang *et al.*, 2021, 2024). For instance, patients in different age groups with risks of breast cancer may need personalized mammography exam frequencies (Alagoz *et al.*, 2013), or, in another example, a patient with organ failure may be presented at different states with organ transplant options that vary in their compatibility with the patient. The patient may face

the decision to either wait for a better match or accept an offered organ as their own survival probability decreases over time (Zhang *et al.*, 2021, 2024). For a more extensive exploration of MDPs in healthcare, refer to the comprehensive reviews by Schaefer *et al.* (2004), Alagoz *et al.* (2010), and Sonnenberg and Beck (1993). Although MDPs are widely used in healthcare applications, most of these consider finite-state decision-making. Constructing an MDP for infectious disease control problems with repeated decisions is challenging, especially when the state space for such problems is continuous.

2.2. Solving continuous state MDP

As previously discussed, an infinite or continuous state space is a major challenge when formulating MDPs. Several methods have been proposed to address this problem. In Li and Littman (2005), a discretization-free approach (modified value iteration with lazy approximation) is introduced, but it requires highly precise piece-wise constant approximations of both the value and transition functions. In Munos and Moore (2002), different criteria for discretizing state and time-space non-uniformly are discussed. These methods involve evaluating values or policies using dynamic programming; however, some of these methods raise computational concerns for problems with continuous or large numbers of states. Zhou *et al.* (2010) used Monte-Carlo simulation to approximate the belief state using a finite number of particles on a discretized grid mesh. However, the study does not provide guidance on the construction of the grid mesh. Brooks *et al.* (2006) proposed a parametric method to uniformly discretize a continuous state space over a lower dimensional parametric space. However, since prior knowledge of the distribution is required, MDPs for infectious disease control would be difficult to solve in this manner. One remedy is to solve the MDP formulation by truncation and discretization of the state (Boucherie & Van Dijk, 2017). Researchers have used various methods to achieve this. For example, Sandıkçı *et al.* (2013) used fixed-resolution, non-uniform grids to discretize the belief state and approximate the optimal policy for a partially observable MDP (POMDP) model. Lovejoy (1991) used fixed or uniform grids to approximate the solution of the POMDP. However, using uniform or pre-defined discretization regions (which requires domain knowledge) may not always be appropriate, particularly for infectious disease control problems where disease spread is subject to substantial changes across different policy scenarios. In such cases, a more effective discretization algorithm is needed to enable the computation of the optimal policy.

2.3. Modeling disease dynamics

To identify the optimal policy for an infectious disease control problem, it is necessary to have a model for describing the disease dynamics. For instance, during an emerging pandemic, how would disease transmission change if the government imposed a 1-month lockdown? How would it change if the government imposed a 3-month lockdown instead? Different policies may change the patterns of disease transmission and thus change the proportion of infections in total. To efficiently avert infections, these different possibilities need to be evaluated to understand the resultant health and cost outcomes. Multiple methods are available for assessing the impact of different policies on a specific population.

One common method to model disease dynamics is to use compartmental models based on differential equations (Brauer, 2008; Kermack & McKendrick, 1991a, 1991b, 1991c). A compartmental model uses a mathematical framework to provide insights into the mechanisms that affect the transmission and progression of disease. This framework partitions the population into different health or treatment states (compartments). For instance, each compartment represents a specific stage of the infectious disease (e.g., susceptible, infected, recovered), and proportions of the population move between compartments described by differential equations at certain rates. This model is fundamental in epidemiology for understanding the spread of diseases and evaluating the potential impact of public health interventions. For example, compartmental models can compare the effectiveness of wearing masks and social distancing during the COVID-19 pandemic (Grimm *et al.*, 2021; Kai *et al.*, 2020). Long *et al.* (2018) use a classical compartmental model to assist with the decision to allocate resources during the 2014 Ebola outbreak in Africa. In Section 5, we consider a classic Susceptible-Infected-Recovered (SIR) epidemic model, which has been extensively used in the epidemiological literature (Beckley *et al.*, 2013; Harko *et al.*, 2014; Kröger & Schlickeiser, 2020).

Another method of evaluating disease dynamics is to use simulation models, which can be used to track transmission, progression, and behavior as well as policy outcomes. For instance, simulation models can be employed to examine the cost-effectiveness of screening recommendations for positive-HIV men who have sex with men (MSM) (Tuite *et al.*, 2014), as well as to study the effectiveness of different disease control strategies for tuberculosis (TB) in India (Suen *et al.*, 2014). Although these methods indeed capture the dynamics of complicated diseases, they are unable to compute dynamic policies effectively as m^t evaluations are usually needed when there are m possible interventions and t decision epochs. Therefore, it is beneficial to find alternative effective ways of identifying the optimal policy for infectious disease control. In our paper, we consider a discrete-state MDP framework that takes advantage of its effective solution methods with underlying disease dynamics estimated from traditional disease models such as compartmental and simulation models.

To model this problem as a discrete-state MDP, we also need to define a transition function to describe the probability of transitioning between the states. Several existing techniques can be used to construct this function. For instance, Yaesoubi and Cohen (2011) proposed a way to compute transition probabilities given a system of ODEs. In another example, Mishalani and Madanat (2002) proposed a method

of developing transition probabilities from a stochastic duration model based on the hazard rate function. However, these methods are computationally intensive, which limits their usage to problems with small populations or disease models with special structures.

3. Problem setup

The notation used in this paper is as follows. We denote $\mathbf{X}_t \in \mathcal{X}$ as the state of the epidemic at time t. $\mathbf{X}_t = [X_{1t}, X_{2t}, ..., X_{nt}]$ has n components where each represents the proportion of the population in the compartment (e.g., for a SIR model, $n = 3$). For example, $\mathbf{X}_t = [X_{St}, X_{It}, X_{Rt}] \in [0,1]^3$ can describe the proportion of the population in susceptible (S), infected (I), and recovered (R) compartments at time t for an SIR model. We denote $\mathbf{X}_0$ as the initial state and we assume it follows an initial distribution Ω. We use $\{\mathbf{X}_t\} = (\mathbf{X}_0, ..., \mathbf{X}_N)$ to denote the disease trajectory.

In this paper, we focus on the finite horizon problem. Let $T = \{1, ..., N\}$ be the set of possible decision epochs for the problem. $A = \{1, ..., |A|\}$ is the set of possible policy interventions for the problem. We assume a small, finite number of actions/policies (e.g., lockdown versus no lockdown). We denote $\pi_t \in A$ as the policy intervention at time t.

We consider a model denoted by $f(\mathbf{X}_t, \pi_t) = \mathbf{X}_{t+1}$ that describes the disease dynamics across time epochs t. This function $f(\mathbf{X}_t, \pi_t)$ can consider disease progression, transmission over time, mortality, and interventions. Generally, $f(\mathbf{X}_t, \pi_t)$ takes the state of the system and policy intervention as an input and then returns the state in the next period. We assume that $f(\mathbf{X}_t, \pi_t)$ is time-homogeneous for simplicity (if time-inhomogeneous dynamics are desired, our methods can be easily extended).

The cost in state $\mathbf{X}_t \in \mathcal{X}$ and taking action $\pi_t \in A$ for $t \in T$ in the infectious disease control problem is denoted using $r(\mathbf{X}_t, \pi_t)$. This cost can be dependent on health outcomes (e.g., number of infected, total vaccinated population, etc.) as well as other factors (financial cost, economic burden, etc.). We let λ denote the discount factor.

Given the transition function $f(\mathbf{X}_t, \pi_t)$ and the cost function $r(\mathbf{X}_t, \pi_t)$, we have the following optimization formulation for our repeated decision-making disease control problem:

$$\min_{\pi_0, ..., \pi_{N-1}} \quad \sum_{t=0}^{N} \lambda^t r(\mathbf{X}_t, \pi_t) | \mathbf{X}_0 \tag{1}$$

$$\text{s.t.} \quad \mathbf{X}_t = f(\mathbf{X}_{t-1}, \pi_{t-1}) \tag{2}$$

In the above problem, the objective is to find a sequence of actions $\{\pi_0, ..., \pi_{N-1}\}$ that minimizes the total discounted cost function $r(\mathbf{X}_t, \pi_t)$ over states $\mathbf{X}_t$ for the whole N-period time horizon given a known initial state $\mathbf{X}_0$. For example, $\mathbf{X}_t$ can represent the proportion of individuals in each COVID-19-related health stage at time t, and let $r(\mathbf{X}_t, \pi_t)$ compute the proportion of people dead from COVID-19 at time t. If π_t denotes the policy intervention (lockdown or not) at time t, then $f(\mathbf{X}_{t-1}, \pi_{t-1})$ could be a system of different equations that describes the population flow across different health stages. Our objective in this problem then is to find the optimal policy intervention at each time t that minimizes the total cost within N periods.

There are challenges to solving the above formulation using traditional MDP solution methods (e.g., backward induction, value iteration, policy iteration, etc.) as this formulation usually contains constraints with non-linear dynamics on a continuous state space. These solution methods require a finite number of states for effective evaluation. Moreover, the function $f(\mathbf{X}_t, \pi_t)$ may not be expressed as transition probabilities from state to state, while many traditional MDP solution methods use transition probability matrices to allow for the modeling of uncertainty and variability in decision-making processes.

To discretize the continuous state space, we partition the state space $\mathcal{X}$ into a discrete set of states $\bar{\mathcal{X}}$. For each component d in $\mathcal{X}$, we use the discretization vector G_d to describe how the continuous state space is partitioned into discrete states. The discretization vector G_i contains the maximal and minimal values of the discretized regions for component d. We use G to represent the list of discretization vectors for all components in $\mathcal{X}$. For instance, for an SI model, if $G = \{[0, 0.6, 1], [0, 0.2, 1]\}$, we mean that group 1 (the susceptible proportion of the population) is partitioned into two regions $[0, 0.6)$ and $[0.6, 1]$, and the second group (infected proportion) is being partitioned into two regions $[0, 0.2)$ and $[0.2, 1]$. In this case, we have a total of $2 \times 2 = 4$ regions. These four regions are given by $(1): X_S \in [0, 0.6), X_I \in [0, 0.2); (2): X_S \in [0, 0.6), X_I \in [0.2, 1]; (3): X_S \in [0.6, 1], X_I \in [0, 0.2); (4): X_S \in [0.6, 1], X_I \in [0.2, 1]$ (shown in Fig. 1).

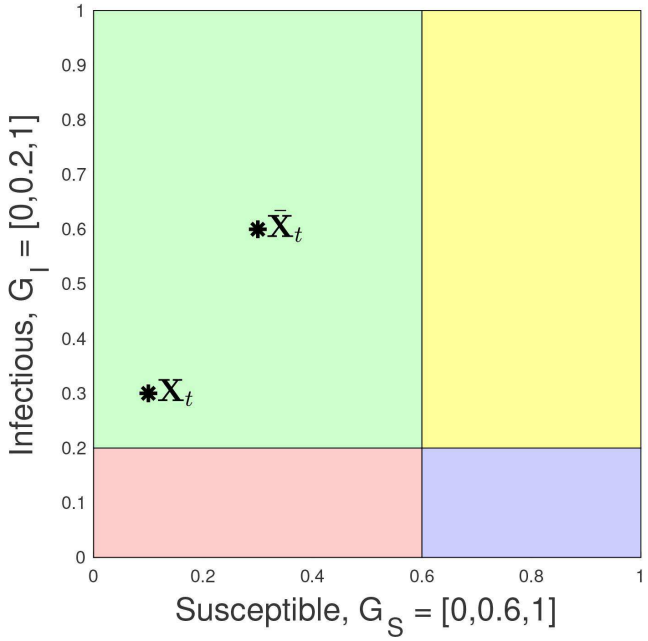

Figure 1. Four regions defined using $G = \{[0, 0.6, 1], [0, 0.2, 1]\}$ are shown in different colors. These correspond to four states: $(1): [\bar{X}_S, \bar{X}_I] = [0.3, 0.1]; (2): [\bar{X}_S, \bar{X}_I] = [0.3, 0.6]; (3): [\bar{X}_S, \bar{X}_I] = [0.8, 0.1]; (4): [\bar{X}_S, \bar{X}_I] = [0.8, 0.6]$. For example, $\mathbf{X}_t = [0.1, 0.3]$, the corresponding discretized state representation is $\bar{\mathbf{X}}_t = [0.3, 0.6]$.

From these regions, we capture the discretized state space in matrix $\bar{\mathcal{X}}$, which is comprised of the Euclidean centroids of each region. The dimension of $\bar{\mathcal{X}}$ is $B \times n$ where B is the number of discretization regions and n is the number of components. Thus, in the example above, we would have four states. $(1) : [\bar{X}_S, \bar{X}_I] = [0.3, 0.1]; (2)$: $[\bar{X}_S, \bar{X}_I] = [0.3, 0.6]; (3) : [\bar{X}_S, \bar{X}_I] = [0.8, 0.1]; (4) : [\bar{X}_S, \bar{X}_I] = [0.8, 0.6]$. In this case, $\bar{\mathcal{X}} = \begin{bmatrix} 0.3 & 0.1 \\ 0.3 & 0.6 \\ 0.8 & 0.1 \\ 0.8 & 0.6 \end{bmatrix}$. Similarly, we define $\bar{\mathbf{X}}_t \in \bar{\mathcal{X}}$ to be the discretized state at time t and $\{\bar{\mathbf{X}}_t\} = (\bar{\mathbf{X}}_0, ..., \bar{\mathbf{X}}_N)$ to be the trajectory for the discretized state.

With this new discretized state space, we can now define $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$, the disease dynamics on the discretized state space. Even though the true disease dynamics might be nonlinear, we approximate the transitions on the discretized state space using a linear transition matrix. This is a reasonably good approximation if the length of t is sufficiently small.

We denote this transition probability matrix as $P(\pi_t)$ for $\pi_t \in A$. $P(\pi_t)$ has the dimension of $|\bar{\mathcal{X}}| \times |\bar{\mathcal{X}}|$ where $|\bar{\mathcal{X}}|$ is the size of the state space. Then the probability of the system being in a state at time $t+1$, $\bar{\mathbf{X}}_{t+1}$, given it was in state $\bar{\mathbf{X}}_t$ at time t and policy intervention $\pi_t \in A$ is denoted as $P(\bar{\mathbf{X}}_{t+1}|\bar{\mathbf{X}}_t, \pi_t)$. It is important to note that the function $f(\mathbf{X}_t, \pi_t)$ is deterministic, whereas $P(\pi_t)$ is a stochastic matrix. The stochastic nature arises from the transition process within the discretized space. Initially, given the region of the initial state, the first transition is determined with 100% certainty, allowing us to precisely identify the subsequent region. However, after the first transition, within any given discretized region, predicting the next region becomes uncertain as the algorithm only records past regions, not the exact points. Consequently, a transition probability matrix is employed to approximate the transition function, thereby facilitating its application to discretized MDPs.

Let $V_t(\bar{\mathbf{X}}_t)$ denote the optimal value function of the discretized state $\bar{\mathbf{X}}_t \in \bar{\mathcal{X}}, t \in T$ for the discretized infectious disease control problem. At optimality, the following must hold:

$$V_t(\bar{\mathbf{X}}_t) = \max_{\pi_t \in A} \{ r(\bar{\mathbf{X}}_t, \pi_t) + \lambda \sum_{\bar{\mathbf{X}}_{t+1} \in \bar{\mathcal{X}}} P(\bar{\mathbf{X}}_{t+1}|\bar{\mathbf{X}}_t, \pi_t) V_t(\bar{\mathbf{X}}_{t+1}) \}$$

3.1. State space discretization problem

With the original system $f(\mathbf{X}_t, \pi_t)$ and state space $\mathcal{X}$, we aim to find the discretized state space $\bar{\mathcal{X}}$ and the transition matrices P that approximate well the original system in that it gives a similar objective value $V_t(\bar{\mathbf{X}}_t)$, trajectories $\{\bar{\mathbf{X}}_t\}$ given $\{\pi_0, ..., \pi_{N-1}\}$, and a small optimality gap. In order to do this, we need to find a suitable G and map from $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ to P.

We focus on approximating the original system by establishing an appropriate discretization approach. For a discretization method to be effective, it should provide accurate estimates for regions of the state space that are more frequently visited, where these frequencies are influenced by which interventions are implemented. To do this efficiently, the discretized states should be capable of providing higher precision in areas where the state space is more likely to be visited. This can lead to a better approximation of the true disease dynamics and can thus result in a more accurate MDP solution.

Given the function $f(\mathbf{X}_t, \pi_t)$, the initial state, the time horizon, and a sequence of policies $\{\pi_0, ..., \pi_{N-1}\}$, we can calculate a trajectory $\{\mathbf{X}_t\}$. Subsequently, we require a state discretization G that ensures the discretized trajectory $\{\bar{\mathbf{X}}_t\}$ closely approximates $\{\mathbf{X}_t\}$ for various initial states and policies. Therefore, our objective is to minimize the distance between the true trajectory and trajectory from the discretized model over all samples $\theta = (\mathbf{X}_0, \{\pi_0, ..., \pi_{N-1}\}) \in \Theta$, all policy intervention scenarios π_t, and all time, i.e., minimizing $\sum_{\theta \in \Theta} \sum_{t=1}^{N} ||\mathbf{X}_t - \bar{\mathbf{X}}_t||_2 |\theta$. Given a sequence of policy intervention $\{\pi_0, ..., \pi_{N-1}\}$ and an initial state $\mathbf{X}_0$, we compute the true trajectory using $f(\mathbf{X}_t, \pi_t)$. We use $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ to compute the trajectory from the discretization space matrix $\bar{\mathcal{X}}$.

We then map the transition function for discretized states $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ to transition probability matrix P. Various existing techniques help to construct transition probabilities given function $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$. We discuss how to find a generalizable and efficient way of computing transition probabilities from $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ given the state discretization in the next section. A complete list of notation used in this section is provided in Table 1.

4. Algorithms

In this section, we present a generalizable framework for discretizing a continuous state space for use in MDP frameworks and correspondingly constructing transition probability matrices.

4.1. Greedy algorithm for finding discretization regions (GreedyCut)

The main objective of discretization is to design an effective approach for approximating the disease dynamics with a high level of accuracy, making such problems tractable for

Table 1. Table of notation.

T	The set of all decision epochs
A	The set of all policy interventions
$\mathcal{X}$	The set of all state representations
$\bar{\mathcal{X}}$	The set of all discretized state representations
$\{\pi_t\}$	Policy intervention at decision epoch t
G_d	Discretization vector for component d
$\{\mathbf{X}_t\}$	Observed disease trajectory
$\{\bar{\mathbf{X}}_t\}$	Discretized disease trajectory
$\{\hat{\mathbf{X}}_t\}$	Markovian disease trajectory computed using transition probabilities
$f(\mathbf{X}_t, \pi_t)$	Transition function
$\bar{f}(\mathbf{X}_t, \pi_t, G)$	Discretized transition function
$P(\pi_t)$	Transition probability for policy π_t
λ	Discount factor
$r(\mathbf{X}_t, \pi_t)$	Reward function

Table 2. Table of additional notation used in Section 4.

Θ	The set of all samples used in GreedyCut
$\hat{\mathbf{X}}_0$	Samples generated in sample average approximation for generating transition probabilities
B	Total number of discretization regions generated
K	Computational costs for calculating the observed trajectory given a sample
$\bar{K}$	Computational costs for calculating the discretized trajectory given a sample
$\hat{K}$	Computational costs for computing the discretized state in the next transitions in Algorithm 2

conventional discrete-space MDP frameworks. However, it would not be advantageous if the process of finding discretization regions itself becomes excessively costly. Therefore, our motivation is to identify a low-cost method that can produce discretization regions capable of representing the disease dynamics effectively. In particular, we are interested in outperforming a uniform discretization, which can be considered a general default discretization appropriate across many domains.

We assume there is a budget B that represents the total number of discretization regions we can have in realize of computational considerations. We use simulated initial states and policy interventions $\theta = (\mathbf{X}_0, \{\pi_0, ..., \pi_{N-1}\}) \in \Theta$ to find the discretization regions.

The greedy approach has been widely applied to various optimization tasks, which is easy to implement and effective at finding solutions (Blanchard *et al.*, 2014; Wu *et al.*, 2018; Zhao *et al.*, 2022). We now propose Algorithm 1 (GreedyCut), a greedy-based iterative approach to finding a good discretization. A list of additional notation is listed in Table 2.

Algorithm 1 Iterative Discretization for Disease Control Problems

procedure COST($\{\mathbf{X}_t\}, \{\bar{\mathbf{X}}_t\}$) ▷ $\{\mathbf{X_t}\}$ is the true trajectory, $\{\bar{\mathbf{X}}_t\}$ is the trajectory from the discretization
  return $\sum_{t=1}^{N} ||\bar{\mathbf{X}}_t - \mathbf{X}_t||_2^2$
procedure CUT(d, i, G) ▷ d is the component we want to cut, and we want to cut the i-th interval in half, $i \in 1, 2, ..., G_d - 1$
  $G_d = [0, ..., G_{d,i}, (G_{d,i} + G_{d,i+1})/2, G_{d,i+1}, ..., 1]$
  return G
procedure GREEDY(B, G, $f(\mathbf{X}_t, \pi_t)$, $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$, θ) ▷ B is the budget, $f(\mathbf{X}_t, \pi_t)$ is the compartmental model dynamics, $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ calculates the trajectory using discretized states $\bar{\mathbf{X}}_t$, Θ is the pre-generated samples of initial states and policies, for each $\theta \in \Theta$, $\theta = (\mathbf{X}_0, \{\pi_0, ..., \pi_{N-1}\})$
  iter_per_sample $= B/|\Theta|$
  for $\theta \in \Theta$ **do**
    for iterations $= 1$: iter_per_sample **do**
      best cost $= \infty$
      worst cost $= -\infty$
      for Component d **do**
        for region $i \in G_d$ **do**
          Compute $\{\bar{\mathbf{X}}_t\}$ using $\bar{\mathbf{X}}_t = \bar{f}(\bar{\mathbf{X}}_{t-1}, \pi_{t-1}, \text{Cut}(d,i,G))$
          tmp cost $=$ Cost($\{\mathbf{X}_t\}$, $\{\bar{\mathbf{X}}_t\}$)
          if tmp cost $<$ best cost **then**
            best cost $=$ tmp cost
            $d^* = d$
            $i^* = i$
          if tmp cost $>$ worst cost **then**
            worst cost $=$ tmp cost
      if worst cost $=$ best cost **then**
        draw a point X_{dt} from $\{\mathbf{X}_t\}$
        Update (d^*, i^*) such that (d^*, i^*) satisfies $G_{d^*,i^*} \le X_{dt} \le G_{d^*,i^*+1}$
        update G=Cut(d^*, i^*, G)
      else
        update G=Cut(d^*, i^*, G)
  return G

In Algorithm 1, we have three functions. The cost function computes the sum of squared error between the trajectory from the discretized state space $\{\bar{\mathbf{X}}_t\}$ and true trajectory $\{\mathbf{X}_t\}$ from $f(X_t, \pi_t)$. Algorithm 1 can be adapted to any disease model without modifications to the algorithm. We compute the discretized trajectory $\{\bar{\mathbf{X}}_t\}$ using $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$, where the d-th component $\bar{\mathbf{X}}_{dt} = \sum_{j=0}^{|G_d|-1} \mathbf{1}_{G_{d,i} \le \bar{f}(\bar{\mathbf{X}}_{t-1}, \pi_{t-1})_d < G_{d,i+1}} \frac{G_{d,i}+G_{d,i+1}}{2}$ takes the average value of the region it belongs to after the discretization. The cost function can also be customized (e.g., introduce another penalty term to emphasize certain disease compartments).

The cut function divides the i-th region of the d-th component into two discretized regions, transforming a continuous range into discrete segments. For instance, consider running a single iteration of GreedyCut on $G = [0, 0.6, 1], [0, 0.2, 1]$ with the point $\mathbf{X}_t = (0.1, 0.3)$, as illustrated in Fig. 2(a). The initial cost is computed as $(0.3 - 0.1)^2 + (0.6 - 0.3)^2 = 0.11$. Four potential cuts are considered: $Cut(1, 1, G)$ with a cost of 0.0925, $Cut(1, 2, G)$ with a cost of 0.11, $Cut(2, 1, G)$ with a cost of 0.11, and $Cut(2, 2, G)$ with a cost of 0.05. Among these, the optimal cut is $Cut(2, 2, G)$. This operation results in a new discretization, G', defined as $G' = \{[0, 0.6, 1], [0, 0.2, 0.6, 1]\}$. Figure 2(b) displays this updated discretization. After the cut, there are now six discretized regions. Similarly, running another iteration of GreedyCut on G' updates the discretization to $G'' = [0, 0.3, 0.6, 1], [0, 0.2, 0.6, 1]$, with the optimal cut being $Cut(1, 1, G')$ (shown in Fig. 2(c)). After the cut, nine discretized are generated with G''. Consequently, the objective value decreases from 0.11 to 0.05 and then to 0.0125.

The greedy function then iteratively computes the cost of cutting one continuous range into two equal discretization regions along each component (dimension) and finds the best cut. We use the 'worst cost' and 'best cost' to monitor the least and most favorable cuts. Although a greedy algorithm typically does not need the 'worst cost', we use it to identify cases that all cuts have equivalent costs. If each cut has the same cost, a point ($X_{dt} \in \mathbf{X}_t$) from the sampled trajectory ($\{\mathbf{X}_t\}|\theta$) will be randomly drawn, and the region

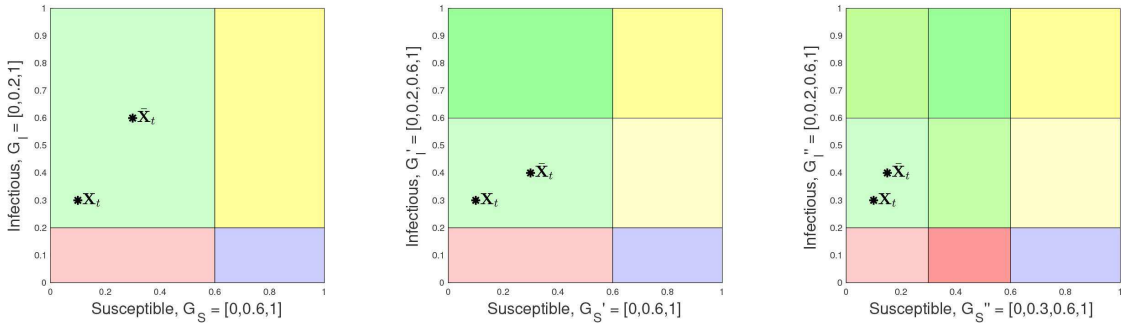

Figure 2. Apply Cut(2,2,G) where $G = \{[0, 0.6, 1], [0, 0.2, 1]\}$ gives new discretization regions $G' = \{[0, 0.6, 1], [0, 0.2, 0.6, 1]\}$. Then apply Cut(1,1, G') gives new discretization regions $G'' = \{[0, 0.3, 0.6, 1], [0, 0.2, 0.6, 1]\}$. In both G' and G'', the $\bar{\mathbf{X}}_t$ is changed as the Euclidean centroid where $\mathbf{X}_t$ belongs to has changed. For G, $||\mathbf{X}_t - \bar{\mathbf{X}}_t||^2 = 0.11$. For G', $||\mathbf{X}_t - \bar{\mathbf{X}}_t||^2 = 0.05$. For G'', $||\mathbf{X}_t - \bar{\mathbf{X}}_t||^2 = 0.0125$.

that this point belongs to (component d of the region i of G such that $G_{d,i} \leq X_{dt} \leq G_{d,i+1}$) will be cut into halves. When every cut incurs the same cost, we want to cut based on the data obtained through sampling. In general, it is unlikely that the costs for all cuts will be exactly the same; this might occur at the beginning of the algorithm when each discretized state encompasses a large range and the approximation will not improve if cut only once. An alternative solution is to generate a few random cuts at the beginning and then run GreedyCut without recording the 'worst cost'. Through this process, in total, $|G| = \sum_d |G_d| = B$ discretization regions will be generated. In Section 5.1.3, we will assess the effectiveness of this algorithm by comparing the sum of squared errors between the Markovian trajectory (the expected trajectory computed using probabilities and discretization regions) and the observed trajectory with different discretization methods. This comparison will help us evaluate how well GreedyCut estimates the actual disease trajectory using transition probability matrices.

4.1.1. Complexity analysis

In Algorithm 1, if we assume that computing $\{\mathbf{X}_t\}$ and $\{\bar{\mathbf{X}}_t\}$ using $f(\mathbf{X}_t, \pi_t)$ and $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ given one sample requires K and $\bar{K}$ operations respectively, we can analyze the total number of operations performed by the GreedyCut algorithm. Since the computational costs for generating new discretization regions, comparing costs, and computing costs are relatively small compared to computing $\{\mathbf{X}_t\}$ and $\{\bar{\mathbf{X}}_t\}$ in our problem, we assume these costs are negligible compared with other costs. At each iteration, the GreedyCut algorithm enumerates through each of b regions including all components d, where $1 \leq b \leq B$. For each discretization, the algorithm performs computations of $\{\mathbf{X}_t\}$ and $\{\bar{\mathbf{X}}_t\}$, which have time complexities of K and $\bar{K}$ operations respectively. The total number of iterations depends on B and $|\boldsymbol{\Theta}|$. It equals to $|\boldsymbol{\Theta}| \times \frac{B}{|\boldsymbol{\Theta}|} = B$.

The total number of operations for the GreedyCut algorithm can be estimated as the sum of operations over all iterations. This can be expressed as $\sum_{b=1}^{B} b(K + \bar{K}) = \frac{B(B+1)}{2}(K + \bar{K})$. Therefore, the complexity of the GreedyCut algorithm is on the order of $\mathcal{O}(B^2(K + \bar{K}))$. This implies that the complexity grows exponentially with the budget of discretization regions, given by B. The time complexity increases quadratically with B while linearly with the number of operations required for each region, represented by $(K + \bar{K})$.

As a result, the computational complexity of the algorithm grows rapidly as the number of discretization regions increases. This highlights the exponential relationship between the complexity and the desired level of granularity in the discretization process.

4.2. Constructing a corresponding transition matrix

Once a suitable discretization of a continuous state space has been constructed, we additionally need a transition matrix between these discretized states to capture the dynamics for use in an MDP framework. To do this, we use a well-known sample average approximation (SAA)-based algorithm (hereafter referred to as Algorithm 2), commonly utilized in various algorithms (Bertsimas *et al.*, 2018; Kim *et al.*, 2015; Zhang & Sen, 2024). In our approach, SAA draws samples from each discretized state to determine the frequency of transitions to subsequent states *via* the function $f(\mathbf{X}_t, \pi_t)$ and G (Hernández-Lerma & Lasserre, 2012). The detailed algorithm steps are provided in Appendix A.

In Algorithm 2, we draw c samples within each region in G and count the frequency of the transitions from the current region to other regions using policy π_t and $f(\mathbf{X}_t, \pi_t)$. By sampling and counting the transitions from the original system, we approximate the underlying transition probabilities directly. Creating approximated transition probability matrices in this way offers a practical approach to capturing the essential dynamics of the system and enables efficient decision-making at the population level.

4.2.1. Complexity analysis

In Algorithm 2, we generate c samples $B|A|$ times, where B denotes the number of discretization regions and $|A|$ represents the size of the action space. Assuming that computing and locating each transition in the appropriate region take $\hat{K}$ operations, we can analyze the time complexity of Algorithm 2. The number of operations performed by the algorithm then is $\mathcal{O}(c\hat{K}B|A|)$.

Furthermore, in most disease control problems, such as COVID-19 mitigation strategies like lockdown, social distancing, and face masks, the size of the action space $|A|$ is typically small. This implies that the algorithm's time complexity is primarily influenced by the number of samples c, the number of discretization regions B, and the operations $\hat{K}$ needed for computing and locating transitions.

5. Numerical examples

In this section, we first showcase our proposed framework for reformulating a SIR model to an infectious disease control MDP framework for supporting public health decisions around social distancing policy. We then demonstrate the utility of this framework with an example of COVID-19 in Los Angeles County, drawing from empirical data of case counts in 2020.

We benchmark the outcome of our method (we refer to the 'GreedyCut discretization method' hereafter) in both examples by comparing our model outcomes to those of several other frameworks: a uniform discretization framework, a discretization framework based on expert opinion (we refer to the 'expert discretization method' hereafter), and an InverseProportional discretization framework motivated by Zhou *et al.* (2010). In the uniform discretization framework, we discretize the entire state space uniformly using the same number of discretization regions as used in the GreedyCut discretization method. In the Expert discretization framework, the state spaces for susceptible and recovered individuals are discretized uniformly. However, the proportion of the population that is infected is only uniformly discretized between *[*0, 0.4*]* instead of [0, 1]. This is based on the observation that the proportion of the population infected at the same time does not exceed 40% in most infectious diseases (Biggerstaff *et al.*, 2014; Pei *et al.*, 2021). In the InverseProportional discretization framework, the state space is discretized by assigning higher resolution to the more frequently visited states based on the simulation results. The transition probabilities for all methods are generated using Algorithm 2. In the second example, we additionally compare our model outcomes to the empirical status-quo policy in Los Angeles in 2020 to demonstrate the improvement our method can achieve.

5.1. Example 1: a simple SIR model

The SIR model tracks the proportion of the population that is susceptible (S), infected (I), and recovered (R) at each time t. We use a discrete time model where the SIR model can be described using a system of difference equations (Allen, 1994):

$$\begin{aligned} S_{t+1} &= S_t - \beta S_t I_t \\ I_{t+1} &= I_t + \beta S_t I_t - \gamma I_t \\ R_{t+1} &= R_t + \gamma I_t \end{aligned}$$

The parameter β is the rate at which disease transmits from the infected to susceptible population proportions, and is dependent on the average contact rate and probability of transmission given a discordant contact. Similarly, γ is the recovery rate.

Typically, at the beginning of an epidemic, the exact proportion of the population that is infected may be unknown. We use $\mathbf{X}_0 = [X_{S0}, X_{I0}, X_{R0}] = [S_0, I_0, R_0]$ to denote the initial state at the first decision epoch. We assume that while the exact initial state is unknown, we know an upper and lower bound on each of the compartments. We use $\underline{S}, \bar{S}$, $\underline{I}, \bar{I}$, and $\underline{R}, \bar{R}$ to denote the upper and lower bound on initial states S_0, I_0, and R_0, respectively.

Suppose at each time t, the health department can choose to implement a social distancing policy (a "lockdown") until time epoch $t+1$ that reduces the transmission rate β. We assume there are a finite number of periods N.

The decision maker wishes to minimize the negative health outcomes and economic and social costs of implementing a lockdown policy. To capture this objective, at each decision epoch, we let the cost be $r(\mathbf{X}_t, \pi_t) = I_t + u(\pi_t)$, the proportion of the population infected (I_t) plus some time-invariant dis-utility value $u(\pi_t)$ that captures the economic and social costs that are only incurred when the intervention is in effect and zero otherwise.

Throughout this section, we refer to this discrete time system as the ground-truth system, and we will construct our discretized MDP framework based on this. We assume no discounting in the objective (λ=1). The objective of the MDP is therefore to minimize the total costs over the whole time horizon.

5.1.1. Inputs

To evaluate this example, we let the transmission rate (β) be 1.4 and the recovery rate (γ) be 0.49. The decision interval (Δt) is a week, and the time horizon (N) is ten weeks. Implementing a lockdown will incur an economic and social cost, but it is unclear how this dis-utility can be quantified in reality. For simplicity, we assume the dis-utility is 0.03 if lockdown was implemented and 0 otherwise ($u(\text{lockdown}) = 0.03$, $u(\text{nolockdown}) = 0$).

During the early stages of a pandemic, there is typically a large population in the susceptible category, while only a small population is infected. Therefore, we choose the initial states to be uniformly distributed within the upper and lower bounds for each compartment to be: $[\underline{S}, \bar{S}] = [0.7, 0.99]$, $[\underline{I}, \bar{I}] = [0.01, 0.1]$, and $[\underline{R}, \bar{R}] = [0, 0.29]$.

In the GreedyCut discretization method, for each sample (θ) generated, ten iterations are run to generate ten additional discretization regions (lines 9–24 in Algorithm 1). To generate samples θ, $\mathbf{X}_0$ are generated uniformly from the region above, and π is a vector with ten random binary variables to indicate the policy intervention (0—no lockdown, 1—lockdown). In Algorithm 2, we generate $c = 1000$ samples to compute two transition probability matrices to correspond to the no lockdown and lockdown policies, respectively. The reason for choosing $c = 1000$ is that we observed the matrix stabilizes once c exceeds 1000.

5.1.2. MDP solutions

We compare MDP solutions among the GreedyCut, InverseProportional, expert, and uniform discretization methods on 90, 150, 300, and 1200 discretization regions.

To evaluate our algorithm's performance, we create 300 samples (we denote the set of all samples as $\mathbf{X}_0$) by selecting the initial susceptible proportion from 0.7 to 0.99 with stepsize of 0.01 and initialize the infected proportion from 0.001 to 0.01 with 0.001 stepsize. We chose 3,000 state-time pairs because we observed that performance does not vary significantly when the number of state-time pairs is changed. By enumerating each pair of S and I, we can have a total of 300 different possible initial states (if $S + I > 1$, we will renormalize each compartment). We compute the following metrics for both GreedyCut and uniform discretization methods on 90, 150, 300, and 1200 discretization regions:

- ACC: accuracy in matching the percentage of optimal actions by comparing discretized MDP with brute force (ground-truth) solution over each state-time pair (a total of 3000 state-time pairs). ACC $= 1 - \frac{mismatch}{3000}$.
- MSE: mean squared error between the optimal value of the discretized MDP ($\bar{V}_0^*$) and the brute force solution (V_0^*) on the first decision epoch over all states. MSE $= \mathbb{E}_{\mathbf{X}_0 \in \check{\mathbf{X}}_0}[||\bar{V}_0^*(\mathbf{X}_0) - V_0^*(\mathbf{X}_0)||^2]$.
- E2: relative mean absolute error on the first decision epoch over all states. E2 $= \mathbb{E}_{\mathbf{X}_0 \in \check{\mathbf{X}}_0}[\frac{|\bar{V}_0^*(\mathbf{X}_0) - V_0^*(\mathbf{X}_0)|}{V_0^*(\mathbf{X}_0)}]$.
- Opt. Gap: average of the relative difference between the optimal value of brute force solution and the value of running optimal policy from discretized MDP on the true disease model ($\tilde{V}_0$) on the first decision epoch. Opt. Gap $= \mathbb{E}_{\mathbf{X}_0 \in \check{\mathbf{X}}_0}[\frac{|\tilde{V}_0(\mathbf{X}_0) - V_0^*(\mathbf{X}_0)|}{V_0^*(\mathbf{X}_0)}]$.

The GreedyCut discretized MDP is able to generate solutions with a higher accuracy than the uniform discretized MDP. We compare the solutions from GreedyCut and other discretized MDPs against the brute force solution in the fifth week ($t = 5$, shown in Fig. 3). We chose the fifth week to better illustrate the outcomes, as the decision boundary is less illustrative in earlier and later time periods. All state pairs recommend the same optimal actions. In the fifth

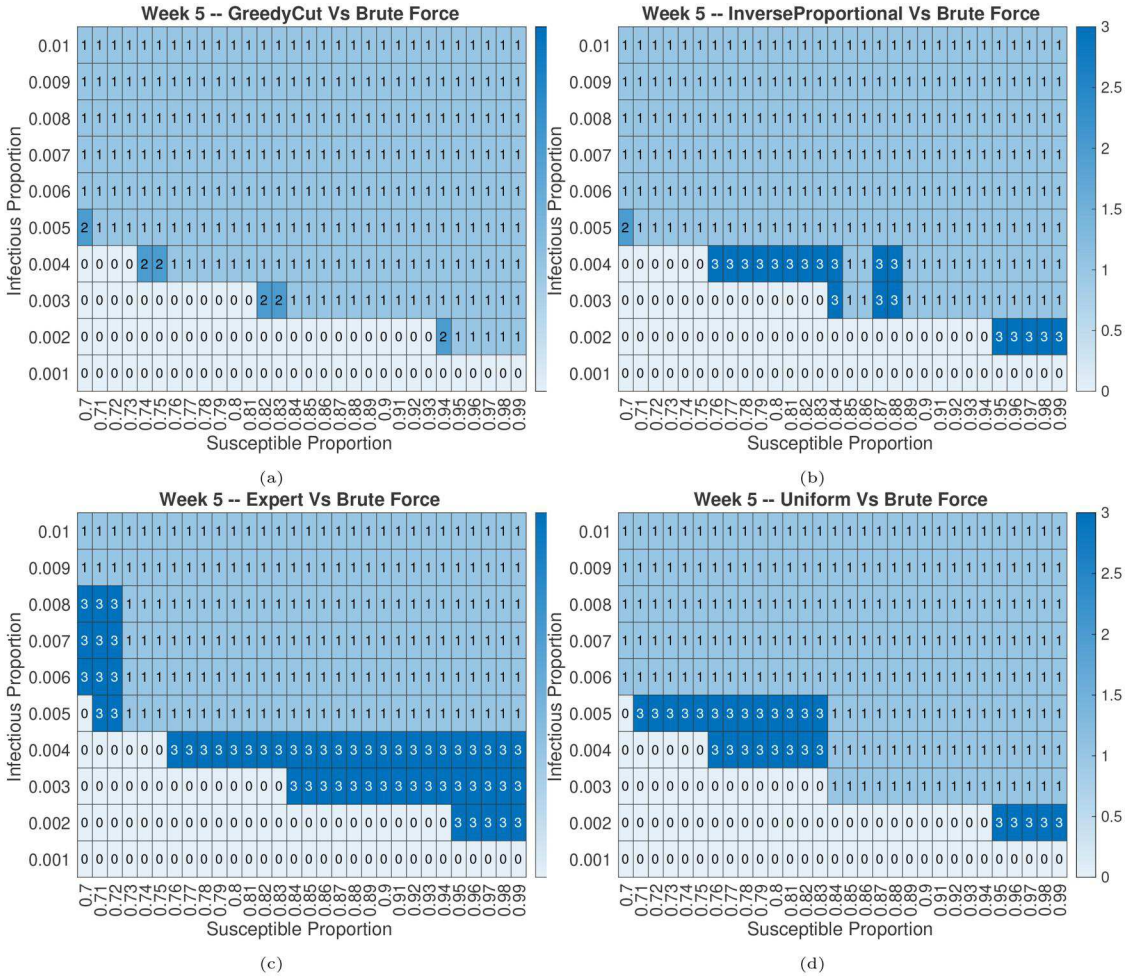

Figure 3. We compare the optimal solution at $t = 5$ across different states using the GreedyCut and uniform discretized MDPs against the ground truth optimal solution found using brute force methods. (a): Optimal solution from the GreedyCut discretized MDP compared to the brute force solution; (b): optimal solution from the uniform discretized MDP compared to the brute force solution. (0 – both models recommend not implementing lockdown; 1 – both models recommend implementing lockdown; 2 – the brute force method recommends not implementing lockdown while the other method recommends lockdown [type 2 error]; 3 – the brute force method recommends implementing lockdown while the other method recommends not implementing lockdown [type 1 error]).

Table 3. Comparison on MDP solutions.

	ACC				MSE			
B	GreedyCut	InverseProportional	Expert	Uniform	GreedyCut	InverseProportional	Expert	Uniform
90	0.9657	0.8687	0.8927	0.8120	4.3239e − 04	0.0040	0.0084	0.0580
150	0.9850	0.9073	0.8753	0.8820	1.7896e − 04	0.0012	0.0037	0.0169
300	0.9787	0.9187	0.8780	0.8613	6.5032e − 05	4.0714e − 04	0.0011	0.0054
1200	0.9893	0.9767	0.9423	0.9150	3.0731e − 06	4.8719e − 05	4.9485e-04	4.3529e-04
	E2				Opt. Gap			
B	GreedyCut	InverseProportional	Expert	Uniform	GreedyCut	InverseProportional	Expert	Uniform
90	0.0689	0.2134	0.3115	0.8846	0.0033	0.0194	0.0364	0.0954
150	0.0435	0.1180	0.2060	0.4946	0.0011	0.0093	0.0164	0.0583
300	0.0233	0.0537	0.1012	0.2487	0.0018	0.0071	0.0111	0.0251
1200	0.0048	0.0220	0.0775	0.0654	4.5576e − 04	8.4937e − 04	0.0023	0.0072

week, we compare optimal actions across 300 states. The results indicate that the GreedyCut discretized MDP has six mismatches, whereas the uniform discretized MDP has 26 mismatches, the expert discretized MDP has 56 mismatches, and the InverseProportional discretized MDP has 20 mismatches when compared with the brute force solution.

Moreover, the number of mismatches may be worse with the other discretized MDPs compared to GreedyCut. In the GreedyCut discretized MDP, all six mismatches belong to the case where the brute force solution recommends not to lockdown while the GreedyCut discretized MDP recommends implementing lockdown. However, in all other discretized MDPs, most mismatches belong to the case where the brute force solution recommends lockdown while the discretized MDP recommends not implementing lockdowns. In infectious disease control, failing to implement a lockdown when it is necessary can cause a rapid increase in the proportion of infected cases. Therefore, error in this direction may be practically worse than in the converse direction. We see this illustrated in the optimality gap among four discretized MDPs (last columns in Table 3), which measures the distance between solutions from discretized MDPs and the true optimal solution. Here we see that GreedyCut MDP achieves a much lower optimality gap for this reason.

The GreedyCut discretization method outperforms other discretization methods across different evaluation metrics across all time periods and different discretization budgets. Table 3 shows the comparison among three discretization methods. Also, the GreedyCut discretization method has higher accuracy (approximately 10% more compared to the uniform discretization method) in matching the optimal actions from the brute force (ground-truth) solution over all state-time pairs. The GreedyCut discretization method is able to generate accurate recommendations on policy interventions even with a small number of discretization regions. Additionally, this method is able to provide a closer approximation of the objective value in both MSE and E2 metrics across all discretization regions. When the number of discretization regions is small, the GreedyCut algorithm has an MSE that is under 1% of the MSE generated using the uniform discretization approach. Similarly, under these conditions, the GreedyCut algorithm's E2 remains below 10% of the E2 from the uniform discretization method. Moreover, the GreedyCut algorithm outperforms the uniform discretization method in reducing the optimality gap. The optimality gap ranges from 0.1% to 0.33% with different numbers of discretization regions in the GreedyCut algorithm, compared with its ranges from 0.72% to 9.54% in the uniform discretization method. We also implemented the discretization-free algorithm from (Li & Littman, 2005); however, it achieved only about 50% accuracy, which is significantly lower than all other methods. Therefore, we excluded this method from our comparisons. This poor performance might be due to the short time horizon of the problem and the need for accurate piece-wise constant approximations of the value and transition functions.

As expected, with a small number of discretization regions, the difference in performance between the GreedyCut and the uniform discretization methods is large. The performance gap shrinks when the number of discretization regions increases, as uniform discretization regions naturally benefit from smaller discretized regions—higher resolution. Additionally, to assess the robustness of GreedyCut against different cost functions, we implemented GreedyCut with the mean absolute error cost function. Although it performed slightly worse than the original GreedyCut, it still outperformed all benchmarking solutions across all numbers of discretization regions considered (results in the Appendix Table 2).

5.1.2.1. Interpretation of cuts generated. We compare the cuts generated by GreedyCut with those from other discretization methods. We found that the GreedyCut algorithm allocated most of its cuts to states that are more likely to be visited based on sample trajectories. Unlike the InverseProportional method, which focuses solely on frequently visited states, GreedyCut also made cuts in less frequently visited but still possible regions, potentially offering a more reliable and robust discretization (the distribution of cuts can be found in Appendix Figure 2).

5.1.2.2. Run time outcomes. To understand how much time is needed to construct an MDP using the GreedyCut discretization method and Algorithm 2, we compare the run time of Algorithm 1 and Algorithm 2 with different numbers of discretization regions using Matlab 2022b on a laptop with 16 GB memory and Apple M1 pro chip.

There is an exponential relationship between the number of discretization regions and the algorithm runtime (see Fig. 4), consistent with the time complexity analysis in Section 4.1.1.

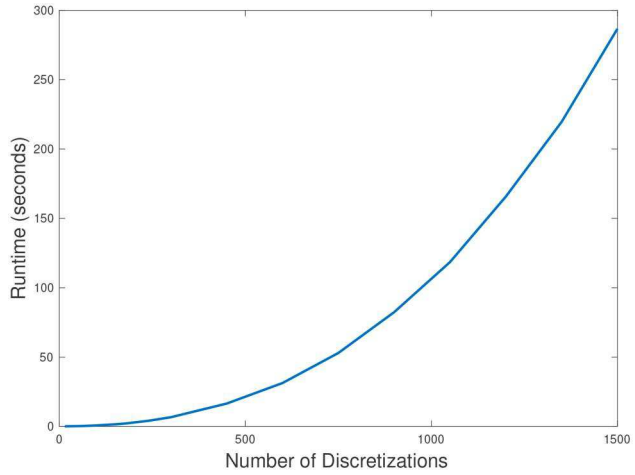

Figure 4. Runtime of algorithm 1.

Table 4. Runtime of Algorithm 2.

B	Runtime (hours):
90	0.12
150	0.34
>300	>1

The curvature of the exponential function will depend on the complexity of the disease model $f(\mathbf{X}_t, \pi_t)$; with more compartments or population stratifications, the total runtime may be larger for a similar number of discretization regions.

The runtime of generating transition matrices is much more costly compared with generating the discretization regions when number of discretization regions (B) is small for both GreedyCut and other discretization methods. Table 4 shows the runtime of generating transitions using Algorithm 2 for the GreedyCut discretization method (other discretization methods should have the same runtimes as there are same number of iterations needed). The runtime exceeds one hour with 300 discretization regions given $c = 1000$. Therefore, when B is small, the total runtime of constructing an approximate MDP is roughly the time for generating transitions using Algorithm 2. In this case, the time required to construct the MDP using the GreedyCut discretization method is similar to that of using the uniform discretization method. This is because the runtime of Algorithm 1, which is only needed for GreedyCut and not for the uniform discretization, is negligible compared to the runtime of Algorithm 2. The total runtime of each algorithm, including discretization, generation of the transition probability matrix, and solving the MDP can be found in the Appendix.

5.1.3. General algorithmic evaluations

In this section, we evaluate the GreedyCut discretization method's performance on generating discretization regions that approximate the disease dynamics $\{\mathbf{X}_t\}$ and Algorithm 2's performance on generating transition probability matrices to approximate the discretized trajectories $\{\bar{\mathbf{X}}_t\}$. We first examine the performance of Algorithm 2 to highlight its capability to generate precise transition probabilities. These

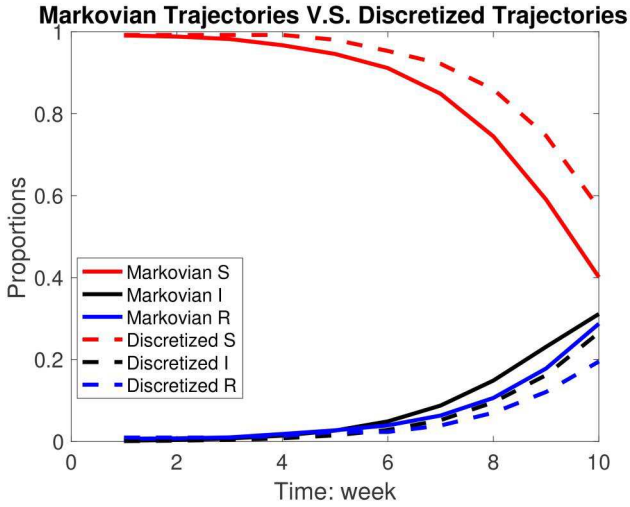

Figure 5. Comparison between trajectories generated from Algorithm 2 given discretization regions and trajectories generated from discretized states. For each compartment S, I, and R, both trajectories are close to each other.

probabilities are crucial for describing the discretized trajectory across various discretization settings. Subsequently, we assess the performance of the GreedyCut discretization method by comparing the Markovian trajectories (using transition probabilities from Algorithm 2) against the actual trajectory $\{\mathbf{X}_t\}$ between the GreedyCut algorithm and other discretization methods.

5.1.4. How accurate are the generated transition probabilities?

To evaluate the accuracy of generated transition probabilities from Algorithm 2, we draw samples and evaluate computed trajectories compared with trajectories $\{\bar{\mathbf{X}}_t\}$ from $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ to eliminate the influence of the quality of the discretization algorithm.

For evaluation, we uniformly draw 100 samples ($\hat{\theta} \in \hat{\Theta}$) that consist of initial states within the upper and lower bound of the proportions in each compartment and a sequence of policy interventions for each initial state. For each evaluation sample $\hat{\theta}$, we compute the discretized trajectory $\{\bar{\mathbf{X}}_t\}$ using $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$. To obtain the Markovian trajectories from the discretized Markov model, we use an initial belief $b_0 = \mathbf{e}_i$ where all entries of b_0 are zero except for i-th entry (corresponding to $\mathbf{X}_0$) which has value one. This indicates we know 100% the initial state of the discretized Markov model. Then we update the belief $b_t = P(\pi_t)b_{t-1}$ over time. To compute the expected proportion of people on each time t (Markovian trajectory at time t, $\tilde{\mathbf{X}}_t$), we use the weighted average over the belief vector at time t, e.g., $\tilde{\mathbf{X}}_t = b_t^T \bar{\mathcal{X}}$. We then compute the cost $\mathbb{E}_{\hat{\theta}\in\hat{\Theta}}[\sum_{t=1}^{N} ||\tilde{\mathbf{X}}_t - \bar{\mathbf{X}}_t||_2^2]$ to evaluate how close is Algorithm 2 able to generate reliable transition probability matrices.

The trajectories obtained from Algorithm 2 closely align with those generated from discretized states for each compartment. As shown in Fig. 5, we compared the trajectories obtained from Algorithm 2 using 300 discretization regions with the $\bar{f}(\bar{\mathbf{X}}_t, \pi_t, G)$ trajectories generated from the same 300 discretized states.

Table 5. Mean squared error for the trajectories given different numbers of discretization regions.

B	Mean squared error between discretized trajectory and Markovian trajectory [95% uncertainty interval]	
90	0.1598	[0.1411, 0.1785]
150	0.1173	[0.1020, 0.1326]
300	0.1086	[0.0940, 0.1232]
1200	0.1067	[0.0923, 0.1211]

We observed that as the number of discretization regions increases, Algorithm 2 is capable of generating transitions that closely resemble the dynamics of the disease, represented by $\{\bar{\mathbf{X}}_t\}$. In Table 5, we show a comparison of the sum of squared error between the Markovian trajectory and discretized trajectory ($\mathbb{E}[\sum_{t=1}^{N} ||\tilde{\mathbf{X}}_t - \bar{\mathbf{X}}_t||_2^2]$) for total budget of discretization regions (B) of 90, 150, 300, and 1,200 using the GreedyCut discretization method. We observe that the error consistently decreases as the number of discretization regions increases.

With fewer discretization regions, each individual discretization possesses a larger range, making it more difficult for samples drawn from these discretization regions to transition accurately between decision epochs, leading to more error in approximating $\{\bar{\mathbf{X}}_t\}$. On the other hand, when a larger number of discretization regions is employed, each discretization exhibits a smaller range. By drawing a sufficient number of samples, it becomes possible to provide a more precise description of $\{\bar{\mathbf{X}}_t\}$. These findings highlight the algorithm's reliability and accuracy in capturing the system's dynamics.

5.1.5. How accurate are the discretization regions generated from the GreedyCut discretization method?.

Next, to evaluate the quality of discretization regions generated from the GreedyCut discretization method, we draw samples and calculate the mean squared error across all samples. This error is measured between the actual trajectory ($\{\mathbf{X}_t\}$) and anticipated Markovian trajectory ($\{\tilde{\mathbf{X}}_t\}$), using the transition matrix created in Algorithm 2 using the discretization regions generated from Algorithm 1 based on the same 100 samples for evaluating Algorithm 2.

We use the same discretization levels (90, 150, 300, and 1200 discretization regions) generated in the previous section for evaluation. To benchmark our model, we also generated uniform and InverseProportional discretization regions with the same number of discretization regions. Then, for all discretization methods, transition probability matrices were generated using Algorithm 2.

We find that both the GreedyCut and InverseProportional discretization method is able to better approximate the disease dynamics over the uniform discretization method. As shown in Fig. 6, a comparison among the GreedyCut, the InverseProportional, the expert, and the uniform discretization methods based on 300 discretization regions shows that the Markovian trajectories for both the GreedyCut and the InverseProportional discretization methods are closely aligned with the actual trajectories. However, the uniform discretization method shows poor approximation, especially showing incorrect trends for the proportion of the population infected over time—where the proportion of the population infected over time starts to decline after week 8 in the Markovian trajectory, whereas the proportion of infected people over time increases in the entire time horizon in the actual trajectory. Additionally, the uniform overestimates the proportion of recovered populations by more than twice compared with the actual proportion of infected people.

For all comparison pairs, the GreedyCut discretization method outperforms both expert and uniform discretization methods in the squared error between Markovian and actual trajectories. GreedyCut and InverseProportional discretization methods demonstrate similar performance overall comparison pairs. In Table 6, we show the result of the comparison of $\mathbb{E}_{\hat{\theta}\in\hat{\Theta}}[||\{\tilde{\mathbf{X}}_t\} - \{\mathbf{X}_t\}||^2]$ over 100 samples and 10 time periods between the GreedyCut and uniform discretization methods. Both algorithms are able to improve the result of approximation when the number of discretization regions used increases, as expected. However, the improvement in approximations is small when the number of discretization regions is large, which suggests a high budget may not be necessary. When the number of allowable discretization regions is small due to the computational budget, the GreedyCut discretization method can provide a much better approximation than the uniform discretization method, and adding discretization regions may not add much accuracy.

Although GreedyCut and InverseProportional demonstrate similar MSE in estimating the actual trajectories, GreedyCut is still able to generate better MDP solutions. One possible reason could be that GreedyCut is finding a more accurate and relevant representation of the state space by minimizing $\mathbf{E}[||\{\bar{\mathbf{X}}_t\} - \mathbf{X}_t||^2]$, leading to better policy generation in the MDP. In contrast, the InverseProportional method primarily focuses on giving higher resolution to more frequently visited states, which results in good trajectory estimations but not necessarily in optimal MDP solutions. Additionally, the quality of InverseProportional discretization method is also sensitive to time horizon. For example, when the time horizon is long and the proportion of the population infected remains low for most of the time, the InverseProportional method will have an extremely high resolution in the region where the proportion of the population infected is low. However, this method will not provide sufficient resolution for other regions, potentially leading to suboptimal outcomes. This is also captured in the following example.

5.2. Example 2: COVID-19

COVID-19 led to a significant surge in infections within Los Angeles County (LAC). To mitigate the pandemic during its initial phases, LAC implemented a lockdown from the second week to the tenth week following March 1st, 2020, which marked the onset of the epidemic. In this example, we use an MDP with discretization regions to identify the optimal timing of imposing lockdowns in LAC to minimize the proportion of infected cases while considering the cost of a lockdown.

5.2.1. Model structure and inputs

To describe the disease dynamics of COVID-19 in LAC, we calibrated a SIR model that is stratified by health districts

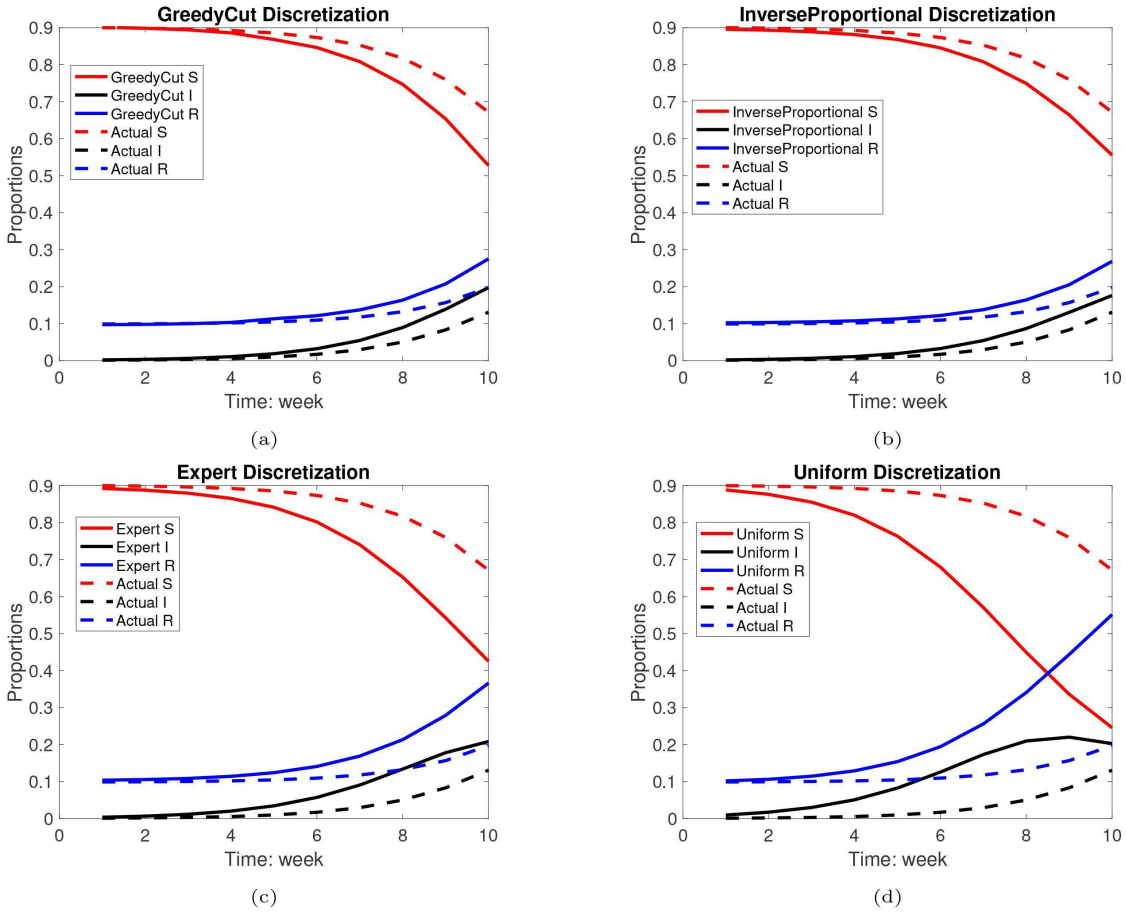

Figure 6. Comparison between trajectories generated from the GreedyCut discretization method against the uniform discretization method (using 300 discretization regions in total) given trajectories generated from SIR model. For each compartment S, I, and R, the GreedyCut discretization method can better capture the disease dynamics.

Table 6. Mean squared error for the trajectories given different numbers of discretization regions.

B	GreedyCut [95% CI]		InverseProportional [95% CI]		Expert [95% CI]		Uniform [95% CI]	
90	0.1261	[0.1100, 0.1423]	0.1344	[0.1160, 0.1528]	0.1437	[0.1173, 0.1700]	0.2399	[0.1627, 0.3170]
150	0.1165	[0.1013, 0.1318]	0.1117	[0.0960, 0.1274]	0.1301	[0.1143, 0.1459]	0.1716	[0.1218, 0.2215]
300	0.1088	[0.0943, 0.1232]	0.1070	[0.0927, 0.1214]	0.1214	[0.1083, 0.1346]	0.1326	[0.1118, 0.1535]
1200	0.1071	[0.0935, 0.1225]	0.1076	[0.0929, 0.1223]	0.1178	[0.1044, 0.1312]	0.1197	[0.1059, 0.1136]

(HD) (Redelings *et al.*, 2010), meaning that the model allows for heterogeneity in health outcomes across HDs. The transmission rates between HDs is also allowed to vary. The disease dynamics for HD i are then described as follows:

$$S_{t+1}^i = S_t^i - \sum_j \beta_{ji} S_t^i I_t^j$$
$$I_{t+1}^i = I_t^i + \sum_j \beta_{ji} S_t^i I_t^j - \gamma I_t^i$$
$$R_{t+1}^i = R_t^i + \gamma I_t^i$$

We use β_{ij} to represent the transmission rate from HD i to HD j, and all HDs are assumed to have the same clearance rate. We consider whether to implement a lockdown policy at each decision epoch, Δt, which has a duration of one week. Decisions need to be made over a total time horizon of 60 weeks ($N = 60$). If lockdown is implemented, transmission will be reduced (β decreases 80%).

We assume there were 1000 infections (0.01% of the total population) at the initial time epoch. This is consistent with the early stage of the COVID-19 epidemic in LAC where the proportion of the population infected remains a small proportion of the overall population. To calibrate the parameters of the stratified SIR model, we used empirical COVID-19 data of case counts to calibrate transmission rate β and recovery rate γ (City of Los Angeles Public Health, 2023). LAC mobility data are also used to help capture the heterogeneity in transmission rate among HDs (Caltrans, 2023; Yu *et al.*, 2024).

We let the stage costs be the proportion of the population infected plus the dis-utility if lockdown is implemented. We assume that the dis-utility without a policy intervention is zero. However, determining the dis-utility associated with a lockdown is challenging. If the dis-utility is excessively low, the optimal choice consistently leans toward implementing the lockdown, which ignores the potential economic and social burden brought by the lockdown. On the contrary, if the dis-utility proves to be excessively burdensome, it will never be enforced. To better reflect this tradeoff, we assume the dis-utility of implementing a one-week lockdown is

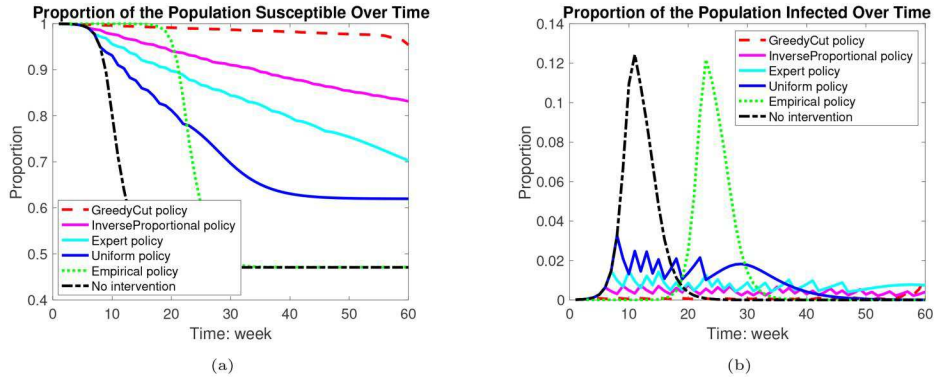

Figure 7. Proportions of the population that is susceptible/infected over time.

0.005, implying it equates to the dis-utility of 0.5% of the population infected per epoch.

We create three discretized MDPs with 150 discretization regions using the GreedyCut discretization method and compare outcomes against those of the InverseProportional, expert, and uniform discretization methods. This will guarantee the completion of model construction within an hour for all discretization methods. In this COVID-19 example, the brute force results cannot be generated within a reasonable time as there are 2^{60} different cases compared to only 2^{10} different cases in the previous example. Therefore, we directly compare the MDP objective values generated by using the identified policies on the true disease model to evaluate each algorithm's performance.

5.2.2. MDP results

We compared the optimal action recommended by the discretized state MDP from the GreedyCut, InverseProportional, expert, and uniform discretization methods. We find that our GreedyCut algorithm outperforms the other discretization methods by identifying a better MDP optimal solution with a smaller cumulative proportion of the population infected. Figure 7 shows a comparison of disease dynamics across different policies. Compared with no lockdown, the empirical policy in LAC (lockdown from week 2 to week 10) does not prevent but rather postpones infections (the total cumulative proportion of the population infected over time is 0.7504 in the empirical policy, and 0.7508 if no intervention is used). All uniform, InverseProportional, expert, and GreedyCut discretized MDPs are able to reduce the cumulative proportion of infections and the peak of infections. The GreedyCut discretization method outperforms the uniform discretization method in terms of the overall reduction in the proportion of the population infected by 0.4793 over the 60-week time horizon.

GreedyCut outperforms uniform, InverseProportional, expert, empirical, and 'do nothing' policies by achieving the lowest objective value (proportion of infected people over time plus lockdown disutility). Figure 8 compares the objective value among different models evaluated on the ground-truth disease (compartmental) model. The 8-week empirical lockdown policy LAC imposed has a lower objective value than doing nothing after considering the cost of lockdown, as it was not able to reduce infections while incurring lockdown disutility costs. Both uniform and GreedyCut MDPs

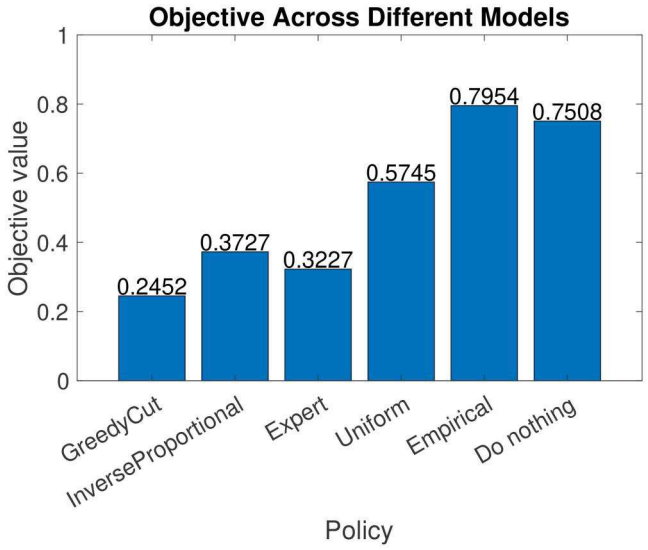

Figure 8. Comparison between objective values across policies from GreedyCut MDP, InverseProportional MDP, uniform MDP, empirical policy, and no policy.

provide a better solution. GreedyCut MDP is able to improve the objective value from doing nothing by 67%, from InverseProportional by 34%, and outperforms the uniform MDP outcome by 57%.

This example demonstrates that the GreedyCut algorithm is able to provide a solution that has a smaller total cost compared to the empirical lockdown policy in LAC and the policy generated by the other discretized MDPs. Even when the number of discretization regions is limited for each compartment (for example, a stratified compartmental model includes death, hospitalizations, exposed, etc.), the GreedyCut discretization method can generate quality solutions with low total discounted costs. Moreover, we found that the uniform discretization method is not able to generate a near-optimal solution as its optimal value exceeds twice the objective value of the GreedyCut discretized MDP.

5.2.3. Sensitivity analysis: Uncertainty in transmission rate

We calibrated the transmission rate of a COVID-19 compartmental model using empirical COVID-19 data and traffic data as a proxy for the contact matrix. However, these rates may be subject to measurement uncertainty. We, therefore, perform sensitivity analysis where the discretization is based on the original calibrated values, but the actual

transmission rate is (1) 30% lower and (2) 30% higher to understand the performance of the discretization methods when input parameters are erroneous. The MDP solutions are generated using the actual transmission rate, while the discretization uses the original calibrated value. This allows us to compare the robustness and reliability of different discretization regions. In each case, we found that GreedyCut was still able to generate the best policy with the lowest objective value (details shown in Appendix Figure 3). This demonstrates that discretization regions generated by GreedyCut are robust to variations in the disease variables.

5.2.4. Extension: MDP with at most two policy switches

In Section 5.2.2, all discretized MDPs recommended a policy with many policy switches where lockdown would be imposed for many short durations. For example, the policy from the uniform discretized MDP recommends lockdown every few weeks in weeks 9-23 (shown in Fig. 9). This is not practical, as inconsistency in policies can lead to poor adherence or even psychological issues (Pedrozo-Pupo *et al.*, 2020; Webster *et al.*, 2020). In this section, we consider the same problem with additional constraints where we allow the policy to switch at most twice (once from no lockdown to lockdown, and once from lockdown to no lockdown). We set up the COVID-19 dynamics in the same way as in Section 5.2.2.

With at most two policy switches (one lockdown duration), the GreedyCut discretized MDP recommends a shorter lockdown duration than the uniform discretized MDP and an earlier lockdown initiation date. Figure 9 shows the lockdown policy outcomes given the disease dynamics of the ground-truth model. The GreedyCut discretized MDP recommends starting the lockdown on week 7 for a duration of 42 weeks. The InverseProportional discretized MDP recommends starting the lockdown at the same time as GreedyCut but with a longer lockdown duration. The expert and uniform discretized MDP recommend initiating the lockdown in weeks 8 and 9, lasting for a duration of 50 weeks. Due to the highly transmissible nature of COVID-19, a lockdown of over 40 weeks is needed to reduce transmission. Figure 10 compares the trajectories using policies from the uniform discretized MDP and GreedyCut discretized MDP. The GreedyCut discretized MDP is able to generate a policy that reduces the cumulative proportion of the population infected by 0.0864 while using fewer weeks of lockdowns compared to the uniform discretized MDP. Additionally, we observed that most countries do not impose prolonged lockdowns. Therefore, we considered an 11-week lockdown using the different discretized MDPs. The GreedyCut discretized MDP recommends starting the lockdown from week 12, resulting in a cumulative infection proportion of 0.458. The uniform, expert, and InverseProportional discretized MDPs recommend starting

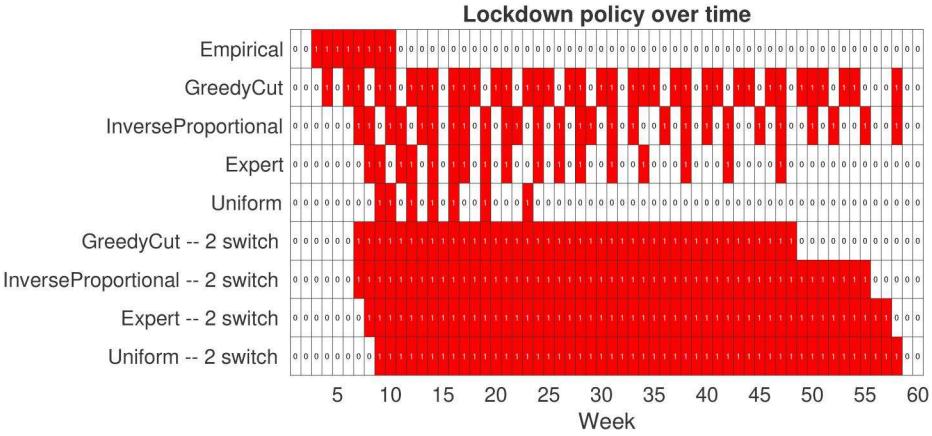

Figure 9. Lockdown policy. The GreedyCut discretized MDP recommends starting the lockdown on week 7 for 42 weeks. The InverseProportional discretized MDP recommends starting the lockdown on week 7 for 49 weeks. The expert discretized MDP recommends starting the lockdown on week 8 for 50 weeks. The uniform discretized MDP recommends starting the lockdown on week 9 for 50 weeks.

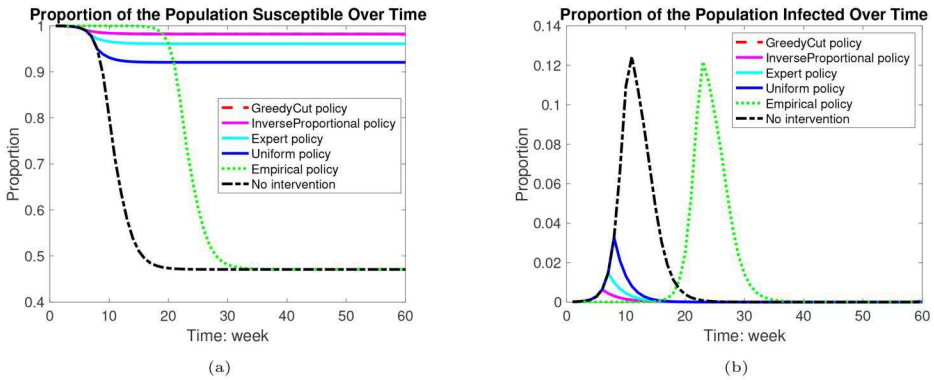

Figure 10. Proportion of the population susceptible/infected over time for different MDPs with policy constraints.

the lockdown in weeks 10, 13, and 16, respectively, resulting in cumulative infection proportions of 0.5834, 0.5399, and 0.7012. All these values are higher than those achieved by implementing longer lockdown policies, confirming the necessity of imposing longer lockdowns.

With the additional constraint on the policy switches, the GreedyCut discretized MDP consistently generates a better solution than other discretized MDPs and other policies we considered in this analysis. In addition, the GreedyCut discretized MDP recommends a shorter lockdown duration compared with other discretized MDPs which could reduce the economic and societal burden brought by the lockdown.

6. Conclusions

In this article, we introduce a novel algorithm for formulating an MDP framework tailored for continuous or large state-space problems in repeated decision-making for infected disease control. In our numerical analyses, we found that our algorithm provides better MDP solutions than the other discretized MDPs for the models we evaluated. Our approach better approximates the true value function than the uniform discretized MDP, therefore leading to a better policy with a lower optimality gap. Compared to other discretization regions, our method demonstrates better performance across different discretization budgets, particularly showing notable benefits when the number of discretization regions is small. This may be particularly pertinent for a compartmental model with many compartments, as the resultant number of discretization regions for each compartment may be extremely limited. In this case, a uniform discretization method may result in a poor estimation of disease dynamics.

We found that our algorithm is able to provide a better state discretization than the uniform discretization method in approximating disease dynamics for the examples considered. Our approach generates smaller regions for states with a higher likelihood of being visited, while also preserving some regions for states with a lower but still possible likelihood, leading to a robust and reliable discretization. Using the GreedyCut method would substantially improve the approximation quality, thus resulting in an improved decision-making process.

With a small number of discretization regions, the time spent discretizing the state using our approach is considerably smaller than the time required to produce transition probability matrices. In our examples, the time needed to formulate the discretized MDP using our approach is nearly equivalent to the time necessary for the construction of a uniform discretized MDP or other benchmark methods. Therefore, our approach could offer an improvement to the MDP solution without substantially increasing computational expense under limited budgets.

We provide numeric examples to demonstrate the efficiency and effectiveness of our algorithm in discretizing continuous-state decision-making problems. Benchmarking the performance with the different discretization methods, we demonstrate that our algorithm is able to generate preferable discretization regions with a limited budget that is a good proxy of the ground truth system. We also demonstrate that our algorithm can generate better policies in both synthetic examples and a COVID-19 example. In the synthetic example, our algorithm outperforms other discretization regions in all metrics for different discretization budgets. In the COVID-19 example, our algorithm improves the objective by nearly 100% from the uniform discretization.

Our numerical analysis also generated policy implications for social distancing policy during COVID-19 in LAC. The first policy implication is that the threshold of implementing the lockdown depends on the proportions of susceptible and infected (recommending implementing the lockdown if the proportions of infected and susceptible are above certain numbers). When the proportion of the population infected increases, the threshold of implementing the lockdown on the proportion of the susceptible population decreases. This is because a less susceptible population is needed to spread the disease as the infected population grows. Similarly, when the proportion of the susceptible population increases, the threshold of implementing the lockdown on the proportion of the infected population decreases. Secondly, a short lockdown interval may not effectively reduce the total number of cases, instead only delays the epidemic peak. To more effectively control cases, a prolonged lockdown period is needed.

We must acknowledge several limitations of this work. The GreedyCut algorithm may not find the discretization that globally minimizes the cost function. The performance gap between the GreedyCut and the uniform discretization methods is small with a large discretization budget. The GreedyCut algorithm may have computational difficulty if there is a large action space and does not consider continuous action spaces; this leaves an interesting optimization direction for future studies. The output of the GreedyCut algorithm may be sensitive to the choice of cost function; different choices may result in widely different discretization choices, thus requiring reevaluation of the objective function.

Despite these limitations, we believe that this work provides an effective and easy-to-handle scheme for dealing with decision-making problems in large or continuous state spaces. Our article provides insight into future work on improving the discretization of solving large-scale MDPs.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Financial support for this study was provided in part by a National Science Foundation (grant no. 2237959).

References

Agrawal, D., Pang, G., & Kumara, S. (2023). Preference based scheduling in a healthcare provider network. *European Journal of Operational Research*, *307*(3), 1318–1335. https://doi.org/10.1016/j.ejor.2022.09.027

Ahn, J.-H., & Hornberger, J. C. (1996). Involving patients in the cadaveric kidney transplant allocation process: A decision-theoretic perspective. *Management Science*, *42*(5), 629–641. https://doi.org/10.1287/mnsc.42.5.629

Alagoz, O., Chhatwal, J., & Burnside, E. S. (2013). Optimal policies for reducing unnecessary follow-up mammography exams in breast cancer diagnosis. *Decision Analysis: A Journal of the Institute for*

Operations Research and the Management Sciences, *10*(3), 200–224. https://doi.org/10.1287/deca.2013.0272

Alagoz, O., Hsu, H. E., Schaefer, A. J., & Roberts, M. S. (2010). Markov decision processes: A tool for sequential decision making under uncertainty. *Medical Decision Making: An International Journal of the Society for Medical Decision Making*, *30*(4), 474–483. https://doi.org/10.1177/0272989X09353194

Alagoz, O., Maillart, L. M., Schaefer, A. J., & Roberts, M. S. (2004). The optimal timing of living-donor liver transplantation. *Management Science*, *50*(10), 1420–1430. https://doi.org/10.1287/mnsc.1040.0287

Alagoz, O., Maillart, L. M., Schaefer, A. J., & Roberts, M. S. (2007). Choosing among living-donor and cadaveric livers. *Management Science*, *53*(11), 1702–1715. https://doi.org/10.1287/mnsc.1070.0726

Allen, L. J. (1994). Some discrete-time si, sir, and sis epidemic models. *Mathematical Biosciences*, *124*(1), 83–105. https://doi.org/10.1016/0025-5564(94)90025-6

Arruda, E. F., & do Val, J. B. R. (2008). Stability and optimality of a multi-product production and storage system under demand uncertainty. *European Journal of Operational Research*, *188*(2), 406–427. https://doi.org/10.1016/j.ejor.2007.04.028

Arruda, E. F., Pereira, B. B., Thiers, C. A., & Tura, B. R. (2019). Optimal testing policies for diagnosing patients with intermediary probability of disease. *Artificial Intelligence in Medicine*, *97*, 89–97. https://doi.org/10.1016/j.artmed.2018.11.005

Bäuerle, N., & Rieder, U. (2009). Mdp algorithms for portfolio optimization problems in pure jump markets. *Finance and Stochastics*, *13*(4), 591–611. https://doi.org/10.1007/s00780-009-0093-0

Beckley, R., Weatherspoon, C., Alexander, M., Chandler, M., Johnson, A., & Bhatt, G. S. (2013). *Modeling epidemics with differential equations*. Tennessee State University Internal Report.

Bertsimas, D., Digalakis, V., Jr, Jacquillat, A., Li, M. L., & Previero, A. (2022). Where to locate COVID-19 mass vaccination facilities? *Naval Research Logistics*, *69*(2), 179–200. https://doi.org/10.1002/nav.22007

Bertsimas, D., Gupta, V., & Kallus, N. (2018). Robust sample average approximation. *Mathematical Programming*, *171*(1–2), 217–282. https://doi.org/10.1007/s10107-017-1174-z

Biggerstaff, M., Cauchemez, S., Reed, C., Gambhir, M., & Finelli, L. (2014). Estimates of the reproduction number for seasonal, pandemic, and zoonotic influenza: A systematic review of the literature. *BMC Infectious Diseases*, *14*(1), 480. https://doi.org/10.1186/1471-2334-14-480

Blanchard, J. D., Cermak, M., Hanle, D., & Jing, Y. (2014). Greedy algorithms for joint sparse recovery. *IEEE Transactions on Signal Processing*, *62*(7), 1694–1704. https://doi.org/10.1109/TSP.2014.2301980

Blower, S., Koelle, K., & Mills, J. (2002). Health policy modeling: Epidemic control, HIV vaccines, and risky behavior. *Quantitative evaluation of HIV prevention programs* (pp. 260–289). Yale University Press.

Boucherie, R. J., & Van Dijk, N. M. (2017). *Markov decision processes in practice* (Vol. 248). Springer.

Brauer, F. (2008). Compartmental models in epidemiology. *Mathematical epidemiology* (pp. 19–79). Springer Berlin, Heidelberg.

Brooks, A., Makarenko, A., Williams, S., & Durrant-Whyte, H. (2006). Parametric pomdps for planning in continuous state spaces. *Robotics and Autonomous Systems*, *54*(11), 887–897. https://doi.org/10.1016/j.robot.2006.05.007

Caltrans. (2023). Pems data source performance measurement system (pems) data source. https://dot.ca.gov/programs/traffic-operations/mpr/pems-source

Calvia, A., Gozzi, F., Lippi, F., & Zanco, G. (2023). A simple planning problem for COVID-19 lockdown: A dynamic programming approach. *Economic Theory*, *77*, 1–28. https://doi.org/10.1007/s00199-023-01493-1

Capan, M., Ivy, J. S., Wilson, J. R., & Huddleston, J. M. (2017). A stochastic model of acute-care decisions based on patient and provider heterogeneity. *Health Care Management Science*, *20*(2), 187–206. https://doi.org/10.1007/s10729-015-9347-x

Carson, Y., & Maria, A. (1997). *Simulation optimization: Methods and applications* [Paper presentation]. Proceedings of the 29th Conference on Winter Simulation (pp. 118–126). https://doi.org/10.1145/268437.268460

Chhatwal, J., Alagoz, O., & Burnside, E. S. (2010). Optimal breast biopsy decision-making based on mammographic features and demographic factors. *Operations Research*, *58*(6), 1577–1591. https://doi.org/10.1287/opre.1100.0877

City of Los Angeles Public Health. (2023). LA county COVID-19 data. http://publichealth.lacounty.gov/media/Coronavirus/data/

David, I., & Yechiali, U. (1985). A time-dependent stopping problem with application to live organ transplants. *Operations Research*, *33*(3), 491–504. https://doi.org/10.1287/opre.33.3.491

Denton, B. T., Kurt, M., Shah, N. D., Bryant, S. C., & Smith, S. A. (2009). Optimizing the start time of statin therapy for patients with diabetes. *Medical Decision Making: An International Journal of the Society for Medical Decision Making*, *29*(3), 351–367. https://doi.org/10.1177/0272989X08329462

Fu, Y., Jin, H., Xiang, H., & Wang, N. (2022). Optimal lockdown policy for vaccination during COVID-19 pandemic. *Finance Research Letters*, *45*, 102123. https://doi.org/10.1016/j.frl.2021.102123

Giannoccaro, I., & Pontrandolfo, P. (2002). Inventory management in supply chains: A reinforcement learning approach. *International Journal of Production Economics*, *78*(2), 153–161. https://doi.org/10.1016/S0925-5273(00)00156-0

Goenka, A., Liu, L., & Nguyen, M.-H. (2014). Infectious diseases and economic growth. *Journal of Mathematical Economics*, *50*, 34–53. https://doi.org/10.1016/j.jmateco.2013.10.004

Grimm, V., Mengel, F., & Schmidt, M. (2021). Extensions of the seir model for the analysis of tailored social distancing and tracing approaches to cope with COVID-19. *Scientific Reports*, *11*(1), 4214. https://doi.org/10.1038/s41598-021-83540-2

Harko, T., Lobo, F. S., & Mak, M. (2014). Exact analytical solutions of the susceptible-infected-recovered (sir) epidemic model and of the sir model with equal death and birth rates. *Applied Mathematics and Computation*, *236*, 184–194. https://doi.org/10.1016/j.amc.2014.03.030

Hernández-Lerma, O., & Lasserre, J. B. (2012). *Discrete-time Markov control processes: Basic optimality criteria* (Vol. 30). Springer Science & Business Media.

Hu, C., Lovejoy, W. S., & Shafer, S. L. (1996). Comparison of some suboptimal control policies in medical drug therapy. *Operations Research*, *44*(5), 696–709. https://doi.org/10.1287/opre.44.5.696

Kai, D., Goldstein, G.-P., Morgunov, A., Nangalia, V., & Rotkirch, A. (2020). Universal masking is urgent in the COVID-19 pandemic: Seir and agent-based models, empirical validation, policy recommendations. arXiv preprint arXiv:2004.13553

Kaplan, R. M., & Anderson, J. P., *et al.* (1996). The general health policy model: An integrated approach. *Quality of Life and Pharmacoeconomics in Clinical Trials*, *2*, 302–322.

Kermack, W. O., & McKendrick, A. G. (1991b). Contributions to the mathematical theory of epidemics—i. *Bulletin of Mathematical Biology*, *53*(1–2), 33–55.

Kermack, W. O., & McKendrick, A. G. (1991b). Contributions to the mathematical theory of epidemics—ii. The problem of endemicity. *Bulletin of Mathematical Biology*, *53*(1–2), 57–87. https://doi.org/10.1016/S0092-8240(05)80041-2

Kermack, W., & McKendrick, A. (1991a). Contributions to the mathematical theory of epidemics—iii. Further studies of the problem of endemicity. *Bulletin of Mathematical Biology*, *53*(1–2), 89–118. https://doi.org/10.1016/S0092-8240(05)80042-4

Kim, S., Pasupathy, R., & Henderson, S. G. (2015). A guide to sample average approximation. *Handbook of simulation optimization* (pp. 207–243). Springer.

Kopec, J. A., Finès, P., Manuel, D. G., Buckeridge, D. L., Flanagan, W. M., Oderkirk, J., Abrahamowicz, M., Harper, S., Sharif, B., Okhmatovskaia, A., Sayre, E. C., Rahman, M. M., & Wolfson, M. C. (2010). Validation of population-based disease simulation models: A review of concepts and methods. *BMC Public Health*, *10*(1), 710. https://doi.org/10.1186/1471-2458-10-710

Kreke, J. E. (2007). Modeling disease management decisions for patients with pneumonia-related sepsis. [Doctoral Dissertation, University of Pittsburgh, Unpublished].

Kröger, M., & Schlickeiser, R. (2020). Analytical solution of the sir-model for the temporal evolution of epidemics. Part A: Time-independent

reproduction factor. *Journal of Physics A: Mathematical and Theoretical*, *53*(50), 505601. https://doi.org/10.1088/1751-8121/abc65d

Kurt, M., Denton, B. T., Schaefer, A. J., Shah, N. D., & Smith, S. A. (2011). The structure of optimal statin initiation policies for patients with type 2 diabetes. *IIE Transactions on Healthcare Systems Engineering*, *1*(1), 49–65. https://doi.org/10.1080/19488300.2010.550180

Lefèvre, C. (1981). Optimal control of a birth and death epidemic process. *Operations Research*, *29*(5), 971–982.

Li, L., & Littman, M. L. (2005). Lazy approximation for solving continuous finite-horizon mdps. *AAAI*, *5*, 1175–1180.

Liu, S., Brandeau, M. L., & Goldhaber-Fiebert, J. D. (2017). Optimizing patient treatment decisions in an era of rapid technological advances: The case of hepatitis c treatment. *Health Care Management Science*, *20*(1), 16–32. https://doi.org/10.1007/s10729-015-9330-6

Long, E. F., Nohdurft, E., & Spinler, S. (2018). Spatial resource allocation for emerging epidemics: A comparison of greedy, myopic, and dynamic policies. *Manufacturing & Service Operations Management*, *20*(2), 181–198. https://doi.org/10.1287/msom.2017.0681

Lovejoy, W. S. (1991). Computationally feasible bounds for partially observed Markov decision processes. *Operations Research*, *39*(1), 162–175. https://doi.org/10.1287/opre.39.1.162

Magni, P., Quaglini, S., Marchetti, M., & Barosi, G. (2000). Deciding when to intervene: A Markov decision process approach. *International Journal of Medical Informatics*, *60*(3), 237–253. https://doi.org/10.1016/s1386-5056(00)00099-x

Maillart, L. M., Ivy, J. S., Ransom, S., & Diehl, K. (2008). Assessing dynamic breast cancer screening policies. *Operations Research*, *56*(6), 1411–1427. https://doi.org/10.1287/opre.1080.0614

Mason, J., Denton, B., Shah, N., & Smith, S. (2014). Optimizing the simultaneous management of blood pressure and cholesterol for type 2 diabetes patients. *European Journal of Operational Research*, *233*(3), 727–738. https://doi.org/10.1016/j.ejor.2013.09.018

Matrajt, L., Eaton, J., Leung, T., Dimitrov, D., Schiffer, J. T., Swan, D. A., & Janes, H. (2021). Optimizing vaccine allocation for COVID-19 vaccines shows the potential role of single-dose vaccination. *Nature Communications*, *12*(1), 3449. https://doi.org/10.1038/s41467-021-23761-1

McNealey, A. K., Marrero, W. J., Steimle, L. N., & Garcia, G.-G. P. (2023). Optimizing interpretable treatment and screening policies in healthcare. In *Encyclopedia of optimization* (pp. 1–14). Springer.

Mishalani, R. G., & Madanat, S. M. (2002). Computation of infrastructure transition probabilities using stochastic duration models. *Journal of Infrastructure Systems*, *8*(4), 139–148. https://doi.org/10.1061/(ASCE)1076-0342(2002)8:4(139)

Munos, R., & Moore, A. (2002). Variable resolution discretization in optimal control. *Machine Learning*, *49*(2/3), 291–323. https://doi.org/10.1023/A:1017992615625

Otero-Leon, D. F., Lavieri, M. S., Denton, B. T., Sussman, J., & Hayward, R. A. (2023). Monitoring policy in the context of preventive treatment of cardiovascular disease. *Health Care Management Science*, *26*(1), 93–116. https://doi.org/10.1007/s10729-022-09621-4

Pedrozo-Pupo, J. C., Pedrozo-Cortés, M. J., & Campo-Arias, A. (2020). Perceived stress associated with COVID-19 epidemic in Colombia: An online survey. *Cadernos de Saude Publica*, *36*(5), e00090520. https://doi.org/10.1590/0102-311x00090520

Pei, S., Yamana, T. K., Kandula, S., Galanti, M., & Shaman, J. (2021). Burden and characteristics of COVID-19 in the United States during 2020. *Nature*, *598*(7880), 338–341. https://doi.org/10.1038/s41586-021-03914-4

Piguillem, F., & Shi, L. (2022). Optimal COVID-19 quarantine and testing policies. *The Economic Journal*, *132*(647), 2534–2562. https://doi.org/10.1093/ej/ueac026

Pontryagin, L. S. (2018). *Mathematical theory of optimal processes*. Routledge.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. (1st Edn.). John Wiley & Sons, Inc.

Redelings, M., Lieb, L., & Sorvillo, F. (2010). Years off your life? The effects of homicide on life expectancy by neighborhood and race/ethnicity in Los Angeles county. *Journal of Urban Health: Bulletin of the New York Academy of Medicine*, *87*(4), 670–676. https://doi.org/10.1007/s11524-010-9470-4

Sandıkçı, B., Maillart, L. M., Schaefer, A. J., Alagoz, O., & Roberts, M. S. (2008). Estimating the patient's price of privacy in liver transplantation. *Operations Research*, *56*(6), 1393–1410. https://doi.org/10.1287/opre.1080.0648

Sandıkçı, B., Maillart, L. M., Schaefer, A. J., & Roberts, M. S. (2013). Alleviating the patient's price of privacy through a partially observable waiting list. *Management Science*, *59*(8), 1836–1854. https://doi.org/10.1287/mnsc.1120.1671

Schaefer, A. J., Bailey, M. D., Shechter, S. M., & Roberts, M. S. (2004). *Modeling medical treatment using Markov decision processes* (pp. 593–612). Springer US.

Shechter, S. M., Bailey, M. D., Schaefer, A. J., & Roberts, M. S. (2008). The optimal time to initiate HIV therapy under ordered health states. *Operations Research*, *56*(1), 20–33. https://doi.org/10.1287/opre.1070.0480

Shen, X., Du, S.-C., Sun, Y.-N., Sun, P. Z., Law, R., & Wu, E. Q. (2024). Advance scheduling for chronic care under online or offline revisit uncertainty. *IEEE Transactions on Automation Science and Engineering*, *21*(4), 5297–5310. https://doi.org/10.1109/TASE.2023.3310116

Singh, R., Liu, F., & Shroff, N. B. (2020). A partially observable mdp approach for sequential testing for infectious diseases such as COVID-19. *arXiv preprint arXiv:2007.13023*

Sonnenberg, F. A., & Beck, J. R. (1993). Markov models in medical decision making: A practical guide. *Medical Decision Making: An International Journal of the Society for Medical Decision Making*, *13*(4), 322–338. https://doi.org/10.1177/0272989X9301300409

Suen, S-C., Bendavid, E., & Goldhaber-Fiebert, J. D. (2014). Disease control implications of India's changing multi-drug resistant tuberculosis epidemic. *PloS One*, *9*(3), e89822. https://doi.org/10.1371/journal.pone.0089822

Suen, S-C., Brandeau, M., & Goldhaber-Fiebert, J. G. (2018). Optimal timing of drug sensitivity testing for patients on first-line tuberculosis treatment. *Health Care Management Science*, *21*(4), 632–646. https://doi.org/10.1007/s10729-017-9416-4

Talbot, T. R., Bradley, S. F., Cosgrove, S. E., Ruef, C., Siegel, J. D., & Weber, D. J. (2005). Influenza vaccination of healthcare workers and vaccine allocation for healthcare workers during vaccine shortages. *Infection Control and Hospital Epidemiology*, *26*(11), 882–890. https://doi.org/10.1086/502512

Topkis, D. M. (2011). *Supermodularity and complementarity*. Princeton University Press.

Tuite, A. R., Burchell, A. N., & Fisman, D. N. (2014). Cost-effectiveness of enhanced syphilis screening among HIV-positive men who have sex with men: A microsimulation model. *PloS One*, *9*(7), e101240. https://doi.org/10.1371/journal.pone.0101240

Webster, R. K., Brooks, S. K., Smith, L. E., Woodland, L., Wessely, S., & Rubin, G. J. (2020). How to improve adherence with quarantine: Rapid review of the evidence. *Public Health*, *182*, 163–169. https://doi.org/10.1016/j.puhe.2020.03.007

Wu, C., Luo, C., Xiong, N., Zhang, W., & Kim, T.-H. (2018). A greedy deep learning method for medical disease analysis. *IEEE Access*, *6*, 20021–20030. https://doi.org/10.1109/ACCESS.2018.2823979

Yaesoubi, R., & Cohen, T. (2011). Generalized Markov models of infectious disease spread: A novel framework for developing dynamic health policies. *European Journal of Operational Research*, *215*(3), 679–687. https://doi.org/10.1016/j.ejor.2011.07.016

Yu, H., Zhang, S., Suen, S-C., Dessouky, M., & Ordonez, F. (2024). Extending dynamic origin-destination estimation to understand traffic patterns during COVID-19. *arXiv preprint arXiv:2401.10308*

Zhang, D., & Sen, S. (2024). A sampling-based progressive hedging algorithm for stochastic programming. *arXiv preprint arXiv:2407.20944*

Zhang, S., Suen, S.-C., Gong, C. L., Pham, J., Trebicka, J., Duvoux, C., Klein, A. S., Wu, T., Jalan, R., & Sundaram, V. (2021). Early transplantation maximizes survival in severe acute-on-chronic liver failure: Results of a Markov decision process model. *JHEP Reports: Innovation in Hepatology*, *3*(6), 100367. https://doi.org/10.1016/j.jhepr.2021.100367

Zhang, S., Suen, S.-C., Sundaram, V., & Gong, C. L. (2024). Quantifying the benefits of increasing decision-making frequency for health applications with regular decision epochs. *IISE Transactions*, 1–15. https://doi.org/10.1080/24725854.2024.2321492

Zhang, F., Wagner, A. K., & Ross-Degnan, D. (2011). Simulation-based power calculation for designing interrupted time series analyses of health policy interventions. *Journal of Clinical Epidemiology*, *64*(11), 1252–1261. https://doi.org/10.1016/j.jclinepi.2011.02.007

Zhao, Z., Zhou, M., & Liu, S. (2022). Iterated greedy algorithms for flow-shop scheduling problems: A tutorial. *IEEE Transactions on Automation Science and Engineering*, *19*(3), 1941–1959. https://doi.org/10.1109/TASE.2021.3062994

Zhou, E., Fu, M. C. and Marcus, S. I. (2010), Solving continuous-state pomdps via density projection, *IEEE Transactions on Automatic Control55*(5), 1101–1116. https://doi.org/10.1109/TAC.2010.2042005

Appendix A

1.1. SAA algorithm for constructing transition matrices

Algorithm 2 Generating Transition matrix

procedure GENERATE($f(\mathbf{X}_t, \pi_t), G$) ▷ $f(\mathbf{X}_t, \pi_t)$ is the ground-truth discrete time model, G is the discretization
  Let P be a $|\bar{\mathcal{X}}| \times |\bar{\mathcal{X}}| \times |A|$ transition matrix with all zeros and each state represents a discretized state from the discretized state space $\bar{\mathcal{X}}$ defined by G
  for each policy intervention $\pi_t \in A$ **do**
    for each discretized state i from $\bar{\mathcal{X}}$ **do**
      Uniformly draw c number of samples ($\hat{\mathbf{X}}_0$) within the region that contains i (including a centroid in this region)
      for each sample $\hat{\mathbf{X}}_0$ **do**
        Compute $\hat{\mathbf{X}}_1 = f(\hat{\mathbf{X}}_0, \pi_t)$
        Find the discretized state j such that the discretized region containing j also contains $\hat{\mathbf{X}}_1$
        $P(j|i, \pi_t) = P(j|i, \pi_t) + 1$
  Normalize P to make it a stochastic matrix
  return P

1.2 Total runtime by different algorithms

Table A1. Total runtime by different algorithms, including discretization, generation of the transition probability matrix, and solving the MDP (in seconds).

	Uniform	GreedyCut	InverseProportional	ExpertOpinion
90	434	436	436	434
150	1226	1236	1228	1226
300	3625	3660	3630	3625
1200	10,560	10,721	10,568	10,560

1.3. MDP solutions of GreedyCut using different cost functions

Table A2. Comparison on MDP solutions between sum of the squared error and mean absolute error.

	ACC		MSE	
B	GreedyCut (Sum of Squared Error)	GreedyCut (MAE)	GreedyCut (Sum of Squared Error)	GreedyCut (MAE)
90	0.9657	0.9457	4.3239e − 04	8.5366e − 04
150	0.9850	0.9357	1.7896e − 04	6.6263e − 04
300	0.9787	0.9733	6.5032e − 05	1.7771e − 04
1200	0.9893	0.9900	3.0731e − 06	1.8360e − 05
	E2		Opt. Gap	
B	GreedyCut (Sum of Squared Error)	GreedyCut (MAE)	GreedyCut (Sum of Squared Error)	GreedyCut (MAE)
90	0.0689	0.0905	0.0033	0.0060
150	0.0435	0.0839	0.0011	0.0089
300	0.0233	0.0457	0.0018	0.0017
1200	0.0048	0.0126	4.5576e − 04	4.1921e − 04

1.4. Matrix Convergence over number of samples

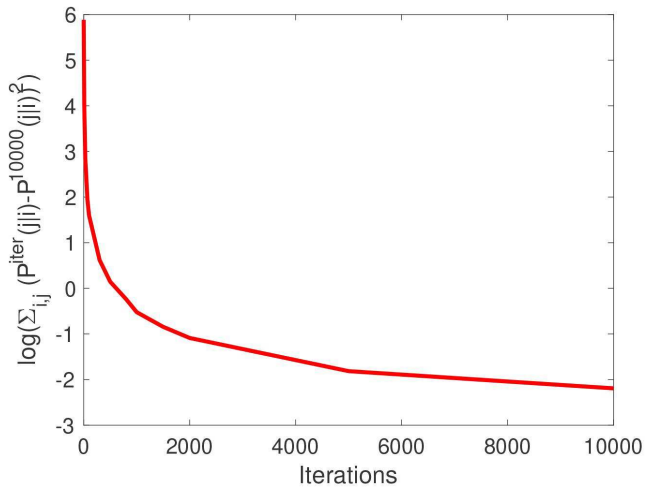

Figure A1. Matrix convergence over time.

1.5. Distribution of discretization regions generated

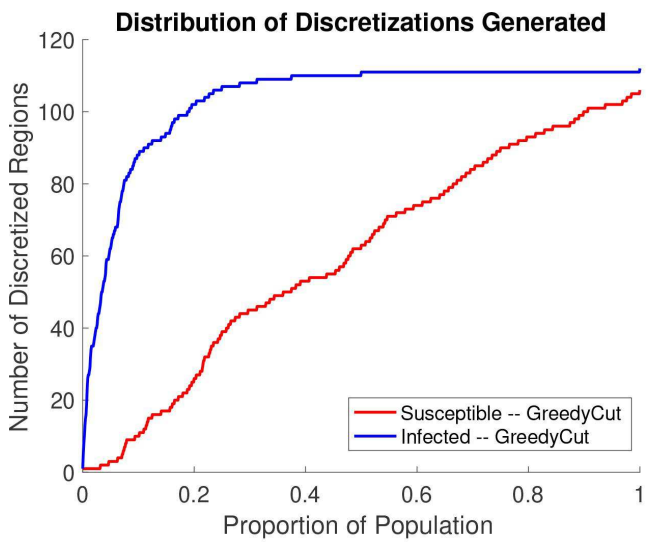

Figure A2. The GreedyCut algorithm allocated most of its cuts to states that are more likely to be visited based on sample trajectories. Unlike the InverseProportional method, which focuses solely on frequently visited states, GreedyCut also made cuts in less frequently visited but still possible regions, potentially offering a more reliable and robust discretization.

1.6. MDP outcomes using different numbers of state time pairs

Table A3. Comparison on MDP solutions.

	ACC		MSE	
B	GreedyCut (3000 pairs)	GreedyCut (6000 pairs)	GreedyCut (3000 pairs)	GreedyCut (6000 pairs)
90	0.9657	0.9597	4.3239e − 04	3.1057e − 04
150	0.9850	0.9711	1.7896e − 04	1.3284e − 04
300	0.9787	0.9772	6.5032e − 05	5.4063e − 05
1200	0.9893	0.9927	3.0731e − 06	2.3932e − 06
	E2		Opt. Gap	
B	GreedyCut (3000 pairs)	GreedyCut (6000 pairs)	GreedyCut (3000 pairs)	GreedyCut (6000 pairs)
90	0.0689	0.0546	0.0033	0.0036
150	0.0435	0.0350	0.0011	0.0019
300	0.0233	0.0212	0.0018	0.0019
1200	0.0048	0.0041	4.5576e − 04	2.8458e − 04

1.7. Sensitivity analysis: Uncertainty in transmission rate

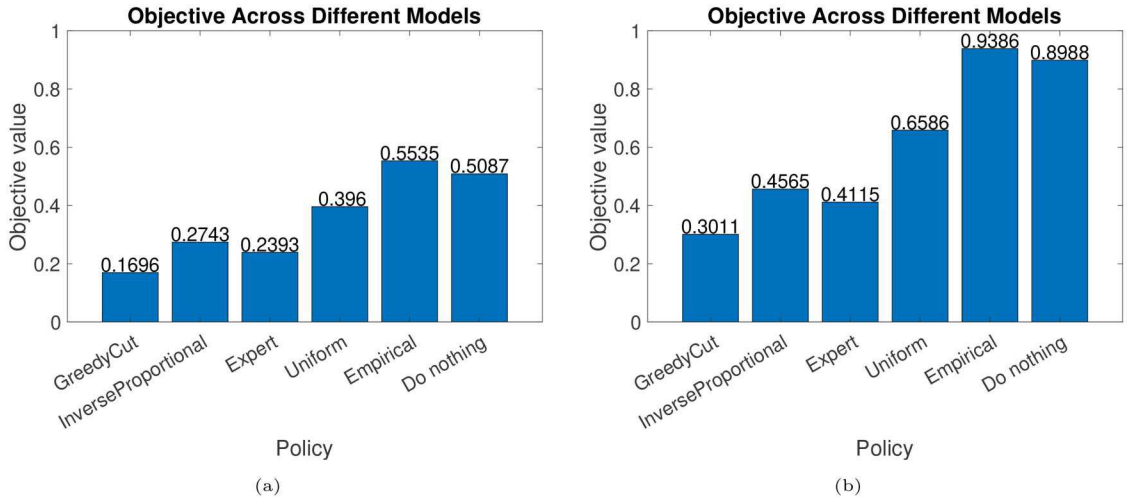

Figure A3. The objective of MDP solutions. (1) The actual transmission rate is 30% less than the calibrated values used in the discretization generation. (2) The actual transmission rate is 30% more than the calibrated values used in the discretization generation.