

[COMP90005] Advanced Computing Subject research project report

James Sun, Zexi Liu, Yuqing Xiao

July 2, 2020

1 Introduction

This project ~~background~~ ^{considers} is under the setting of a pandemic outbreak. During this outbreak, it is assumed there are a set of parameters, θ , that can be controlled by public policy. The vision of this project is to give the policymakers at any time, sets of optimal parameters θ^* , so that the public health risks can be reduced as small as possible.

The question is hence raised to what extent should we mitigate the pandemics? Intuitively, we would like to give θ^* such that it will achieve the virus-free outcome. However, under the assumptions, nudging θ towards such θ^* by implementing policy has increasing cost, meaning that such θ^* will potentially impose infeasible government expenses.

Due to this cost constraint, we introduce the hospital capacity as the mitigation threshold for choosing such θ^* . In contrast to the virus-free outcome, this not only makes sure the pandemics are well contained and managed under the healthcare capacity, but ultimately, as the motivation of this project, achieving the lowest government expenses.

2 Problem Formulation

We've established that containing the infected population slightly below the hospital capacity will lead to the lowest government expense. In other words, we would like to minimize a probabilistic measurement such that this measurement quantifies the infected population going over the hospital's capacity.

For a stochastic process $X_t(\theta)$, this probability

what is $\mu(\theta)$

$$P(X_t(\theta) > H) = \frac{E_{\mu(\theta)}(T_H)}{E_{\mu(\theta)}(T_{\text{totalTime}})} \quad (1)$$

is interpreted to the "long term" accumulating fractions of time and in those times the infected population is over the capacity, denoted by T_H . We would like this limiting probability in (1) to target at a small enough value, β .

2.1 Model for underlying process

In order to use target tracking to find the optimal parameter θ^* which keeps the limiting probability of exceeding hospital capacity H equal to value β , we must first model the underlying process $\{\xi_t(\theta)\}$ which we will use to simulate the number of infected people over time. We do so by using a birth-death process, where a state i represents i number of people infected in a population N . This is modelled as a continuous-time Markov chain $\{X_t(\theta)\}$ with transition intensities from a state i defined by:

$$\begin{aligned} q_{i,i+1} &= \frac{\lambda \cdot p \cdot 2i(N-1)}{N(N-1)} \\ q_{i,i-1} &= i\alpha \end{aligned}$$

Is
 $\xi_t(\theta)$ = number of infected ?

For the underlying process, we assume the following:

- The total population N is fixed for the duration of the process.
- An individual can either be infected or healthy and there are no asymptomatic cases.
- When $i = 0$, no new infections can occur.

Figure 1 shows the transition graph for the underlying process:

Needs reference (Ross)

Needs introduction of your parameters and some explanation (or citation) to where this comes from.

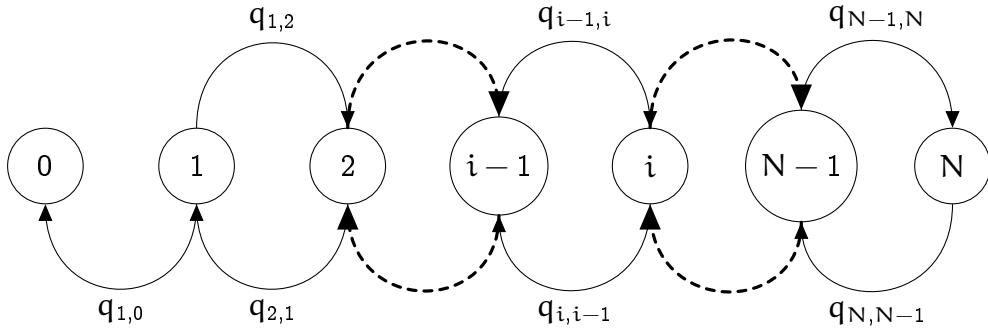


Figure 1: Transition graph of birth-death process model

very nice!

2.2 Model parameters

From the underlying process, the set of model parameters that are of interest are $\theta = [\lambda, p, \alpha]^T$, and are defined as follows:

(should be defined before $q_{i,i+1}$, $q_{i,i-1}$)

- λ : The contact rate between individuals
- p : The probability of becoming infected when an individual comes into contact with an infected person
- α : The recovery rate

We are interested in these parameters as they will be tied directly to the response by policymakers to an ongoing or expected outbreak of disease. Some examples of factors which affect these parameters are given below:

- λ : social distancing, lockdowns, quarantines
- p : hygiene, vaccinations, masks
- α : increasing hospital staffing, available medical equipment

2.2.1 Monotonic Argument

The relation between the probability of exceeding hospital capacity and λ, p, α is assumed to be strictly monotonic. For example, intuitively, increasing the contact rate between individual λ will increase the chances of infection hence increase the probability of the infected population going over the hospital capacity. The same arguments apply for the rest two parameters.

In order to reduce the probability of exceeding hospital capacity, policymakers will need to implement policies that reduce λ and p and increase α .

However, we can see from the factors which affect these parameters, there will be varying accumulated economic and social costs associated with the implementation of any policy.

This is another motivating factor behind finding different sets of θ^* , as the policymaker can choose between different bundles of policy according to the different sets of θ^* . The flexibility

*✓ needs to be defined
in form of
 $\{S_t(\theta)\}$.*

can also reduce the cost. each set of θ^* will always ensure the probability of exceeding hospital capacity does not exceed a certain value, but most importantly, this gives the policymaker options to choose the cheapest policy bundle.

Optimal value is not unique; needs better justification.

2.3 Target Tracking

the bundle of parameters θ has effects on the limiting probability, $L(\theta) := P(X(\theta) > H)$, so we seek the value θ^* such that $L(\theta) = \beta$, but function $L(\cdot)$ is unknown.

This general multi-variate problem can be formulated as an optimization problem. Let \downarrow what is $X(\theta)$?

$$J(\theta) = \frac{1}{2}(L(\theta) - \beta)^T(L(\theta) - \beta) \quad (2)$$

For $L : \mathbb{R}^d \rightarrow \mathbb{R}$ and $\beta \in \mathbb{R}$. The problem is to find

$$\min_{\theta} J(\theta) \quad (3)$$

Please define $L(\cdot)$ clearly: how is this related to (1) \swarrow

In this tracking problem, β is the target and it is assumed that there exist $\theta^* \in \mathbb{R}^d$ such that $L(\theta^*) = \beta$. The gradient-based algorithm has the deterministic formulation [1]:

$$\theta_{n+1} = \theta_n - \epsilon_n \nabla J(\theta)^T = \theta_n - \epsilon_n ((L(\theta_n) - \beta)^T \nabla L(\theta_n))^T \quad (4)$$

In chapter 3 [1], if the function $L(\theta)$ is monotone non-decreasing (or non-increasing) in each of its components, then the gradient information, $\nabla L(\theta)$ in above algorithm can be replaced by the matrix $e = \text{sign}(\nabla L(\theta))$ that is independent of θ and has fixed value. hence one can use instead the recursion [1]:

$$\theta_{n+1} = \theta_n - \epsilon_n ((\hat{L}(\theta_n) - \beta)^T e)^T \quad (5)$$

2.3.1 Well Posedness

[1] under strict monotonicity argument 2.21, the chosen direction, $(\hat{L}(\theta_n) - \beta)^T e$ is always the negative descent direction. This means that the field is coercive for the well-posed optimization problem as (3).

3 Methodology and Convergence Analysis

e has different signs!

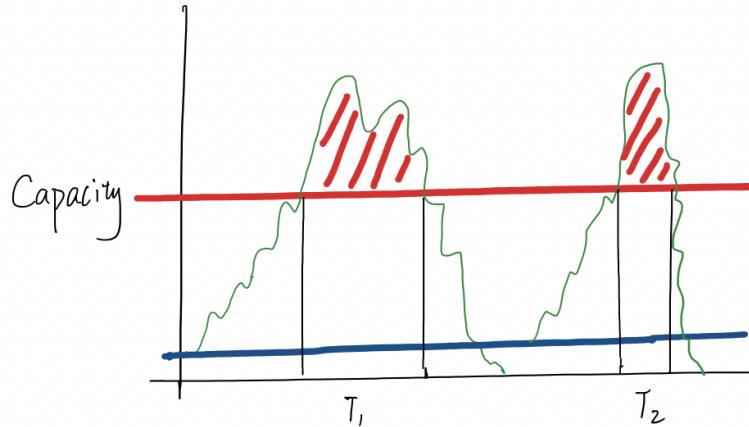
$$G(\theta) = (L(\theta_n) - \beta)^T e = \frac{E_{\mu(\theta)}(T_H(\theta))}{E_{\mu(\theta)}(T(\theta))} - \beta$$

missing Δ
your B&D does not have a stat measure $\mu(\theta)$ (6)

3.1 convergence verification - theorem 5.3 in Chapter 5 [1]

We will use those following equations to prove that we can use theorem 5.3[1] and We plan to build a function with those three parameters to value the infected population, the threshold capacity we have here is H . We will count the binary event as the following Figure 2 showed.

? I think you do not consider the binary model



you forgot β !

Figure 2: Curve of infected population

For a stochastic process, the probability

$$P(X(\theta) > C)$$

is interpreted to the "long term" fraction of time that the process is overcapacity. Based on a statistical approach like this

$$\frac{E(T_H)}{E(T_{\text{total}})}$$

the process is a continuous time MC

$$\hat{T}_H(n) = \text{total time above } H \quad (8)$$

until iteration n of the process. Then this formula shows the elapsed time ρ between jumps.

what is this?

$$\hat{T}_H(n+i) = \hat{T}_H(n) + \rho \cdot 1_{X_n(\theta) < H}$$

Needs better explanation

Lastly, we construct the estimator

$$Y_n = \frac{\hat{T}_H(n)}{K}$$

$-\beta$ I think you mean

$$f_n \sim \exp(\nu_{X_n})$$

K here is a constant batch size in days. For each K days, we construct such Y_n to update the parameters. This will be further explained in section 5.

It is noted that we estimate T_{total} by using a constant K . However there are cases when the infected reaches zero before finish K -days simulation and once we reach zero, we start another new K -days simulation. Therefore T_{total} is almost certain to be overestimated.

a1:

The transition probability for the fixed- θ is defined in equation (20). It is continuous with respect to θ

are you building the feedback for the stochastic approximation? clarify.

a2:

From figure 1, we can obviously observe the underlying process is not an ergodic markov chain because there exists an absorbing state at 0. Hence the model in Chapter 5 [1] can not be directly applied.

As such we adjust our simulation process and impose the ergodic process by programming the underlying process jumping right back to the initial infected patient number x_0 with probability 1 when $X(\theta)$ hits zero. Because the chain is recurrent and finite, the unique stationary measure μ_θ exists and the set $\{\mu_{\theta(\cdot)}, \theta \in \Theta\}$ of stationary measures of the fixed- θ processes is tight for each compact set Θ .

a3:

$$E[Y_n | \mathcal{F}_{n-1}] = g(\xi_{n_1}, \theta_n) + \beta_n \quad (11)$$

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^N g(\xi_n, \theta_n) = G(\theta) \quad (12)$$

$$E(T(\theta)) \approx \frac{NK}{3} \quad (13)$$

$$E(T_H(\theta)) = \frac{1}{3} \left(\sum_{1st\ cycle} T_H^{(k)} + \sum_{2nd\ cycle} T_H^{(k)} + \sum_{3rd\ cycle} T_H^{(k)} \right) \quad (14)$$

$$\frac{1}{N} \sum_{n=1}^N \frac{T_H^{(k)}(n)}{K} = \frac{\frac{1}{N} \sum_{n=1}^N T_H^{(k)}(n)}{NK} \quad (15)$$

this should have been explained in section 2.

Like what we have in equations (13) and (14), to calculate the probability as the equation (7), we will have an existing our $g(\theta)$ as the equation(15) shows. With that $g(\theta)$ like what we have exactly in equations (11) and (12), we will have a $G(\theta)$ like the right part of equation (15) to satisfies equation (12).

a4: The feedback Y_n is defined as a probability, hence it will always be bounded less than 1, meaning that it has bounded variance, which is sufficient to imply uniformly integrable. *what's β_n ?*

a5:

it should subtract β

From the $G(\theta)$ that we have from a3 and with the epsilon for Y_n we have in a4. Because we are truncating, we have all variables Y_n are uniformly bounded. According to (11), we will have lots of uniformly bounded pairs of variables in $g(\theta)$ which looks like (θ, \cdot) . Hence, we will have this set $(\xi_n^\epsilon, \theta_n^\epsilon)$ tight.

Not very clear: $\theta \in \mathbb{R}^3$, you should have $Y_n \in \mathbb{R}^3$ as well.

Explain better.

4 Other Discussions

In future extension work, We are considering use gradient estimation of chapter 8 [1] to make our parameters updating more reasonable. As what we designed in the last section, the objective function is

$$J(\theta) = \frac{1}{2} \left(\frac{E(T_H)}{E(T)} - \beta \right)^T \left(\frac{E(T_H)}{E(T)} - \beta \right) \quad (16)$$

To find the best policy with the least cost, we need to minimize $J(\theta)$ function. We apply the stochastic version of the gradient search method in Chapter 1 [1] and update our parameters

$$\theta_{n+1} = \theta_n + \epsilon Y_n \quad (17)$$

Similar to the equation (8.9) of chapter 8 [1], we consider the probability of the population of infected people is smaller than our hospital capacity is as follows, the β means a positive value which we assume for capacity.

? no "n" ? what is $X_n(\theta)$?

$$P(X(\theta) > C) = E[1_{X_n(\theta) \leq H}] \quad (18)$$

Based on that, in some future work, every time we update our parameters, we can apply an IPA estimator. We will give a small enough perturbation and check if our system is stable. If we have lots of broad enough perturbation, based on

$$\theta_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\theta)^2}{2\sigma^2}} \quad (19)$$

Then, we might be able to get more reasonable parameters.[2]

not correct. You need to refer to your own model. I suppose you still use the ergodic model by repeating cycles ; if so then you need the evaluation of the gradient of your original function

$$D_\theta \xrightarrow{\frac{E[T_H]}{E[\text{total}]}}$$

Does not make any sense. RHS is "almost" a density for a normal dist, LHS is a control variable,

why do you introduce a perturbation? why normal? what is the purpose of this? Not clear at all.

5 Experimental results

In order to find the optimal parameters θ^* , we need to obtain simulations of $\{X_t(\theta)\}$. This is achieved by first simulating the waiting time between transitions T_i using the following expressions:

you had used f

$$v_i = q_{i,i+1} + q_{i,i-1}$$

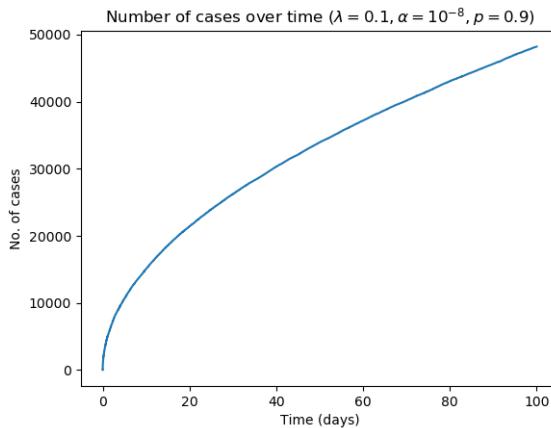
$$T_i \sim \text{Exp}(v_i)$$

} should be in section 2.
inverse of scale

After simulating T_i from the exponential distribution (note that v_i is the scale parameter), we then increment $X_t(\theta)$ using the following as a parameter in the Bernoulli distribution:

$$\Pr(X_{t+T_i} - X_t = 1 | X_t) = \frac{q_{i,i+1}}{v_i} \quad (20)$$

An example of this process over 100 days with fixed parameters is shown below:



(notation is not consistent throughout your paper)

It looks almost deterministic
(Bad example to illustrate a B&D.)

Figure 3: Simulation of birth-death process with fixed parameters

Interpreting the parameters for this simulation as not having any policies in place to reduce infections, we can see that the process appears to be deterministic, with very few recoveries and that the hospital capacity is reached in about 20 days.

not really (use long term)

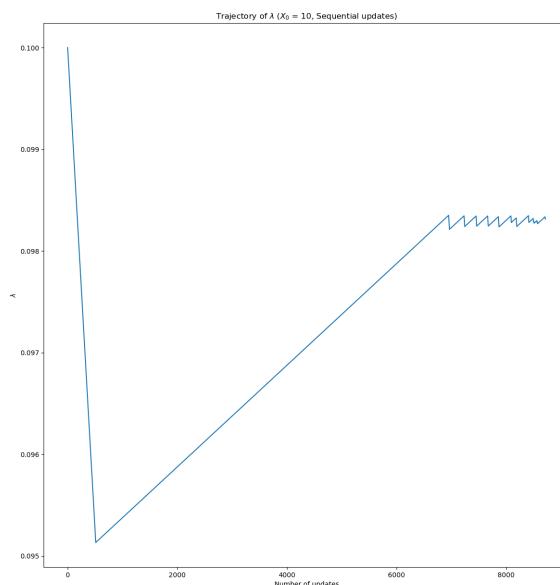
From section 3, we use $\frac{\mathbb{E}[T_{H(n)}]}{K}$ as an estimator for $\mu(X_t(\theta) > H)$, where n is the batch number. For our experiments, we choose $K = 14$, so we allow a large enough batch to be simulated for an estimator value, whilst also imitating the behaviour of actual policymakers, who will adjust their policies regularly every few weeks to contain the spread of the disease. We will also set $N = 4000000$ and $H = 20000$ to model the number of infections in a population roughly similar to the size of Melbourne, AU and set the target value $\beta = 0.05$.

n is the "batch" number

The experiments will all be performed with $p = 0.9$, representing a population which is more susceptible to infection when contact with an infected person. Two methods will be used for updating the parameters: sequential, updating λ at the end of one batch then α at the end of another batch, and simultaneous, updating both variables at the end of each batch. We

will compare the results for these two methods, and also compare the results of different initial conditions with $X_0 = 10$ and 1000 . Constant step sizes were used to update the parameters, with different step sizes for each. The step sizes used were $\epsilon_\lambda = 0.00001$ and $\epsilon_\alpha = 2 * 10^{-9}$, chosen so that the updates in parameter would not be too large. The initial values of the parameters were set to $\lambda = 0.1$ and $\alpha = 10^{-8}$. The code used to implement these simulations can be found in section 7.

Shown below in Figures 4-9 are the trajectories of the parameters and number of cases of one simulation each using sequential and simultaneous updates, with $X_0 = 10$:



very good.
This is much better
section than the
preceding ones

Figure 4: Trajectory of λ , Sequential

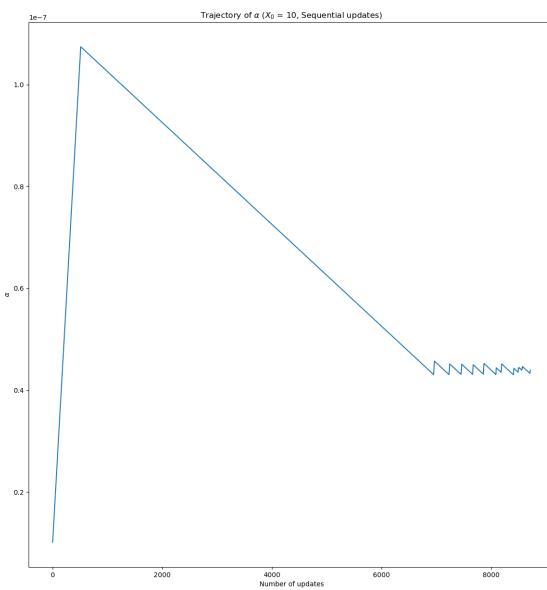


Figure 5: Trajectory of α , Sequential

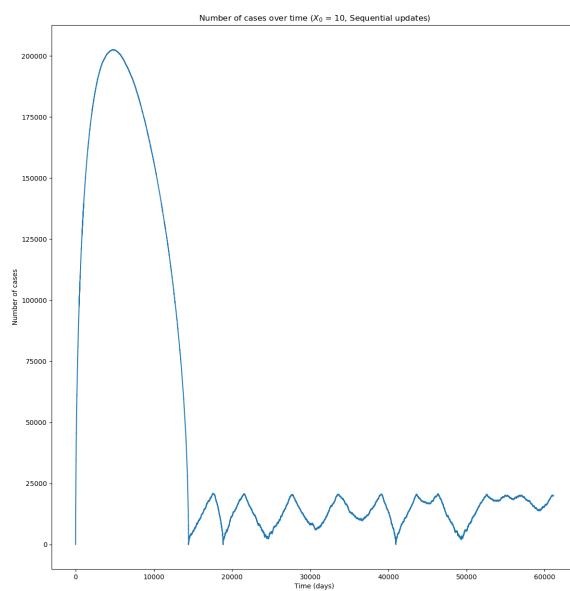


Figure 6: Number of cases over time, Sequential

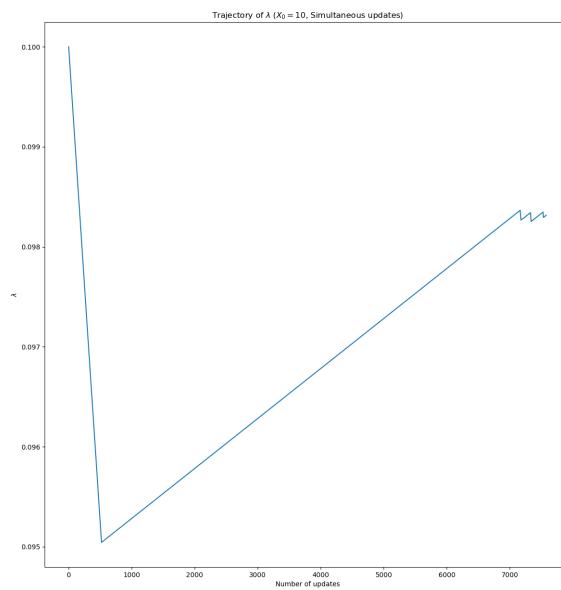


Figure 7: Trajectory of λ , Simultaneous

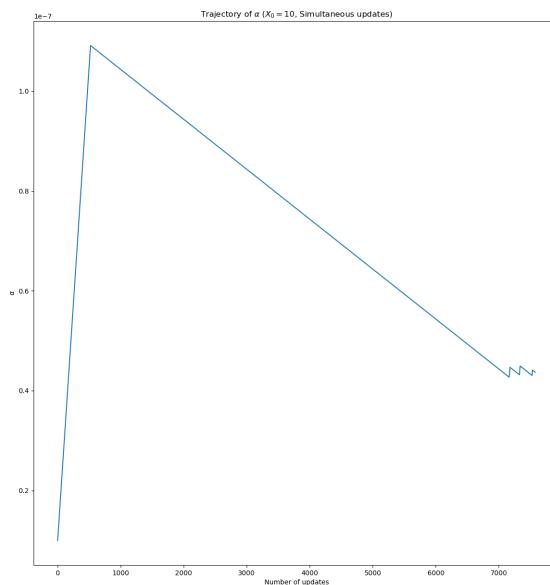
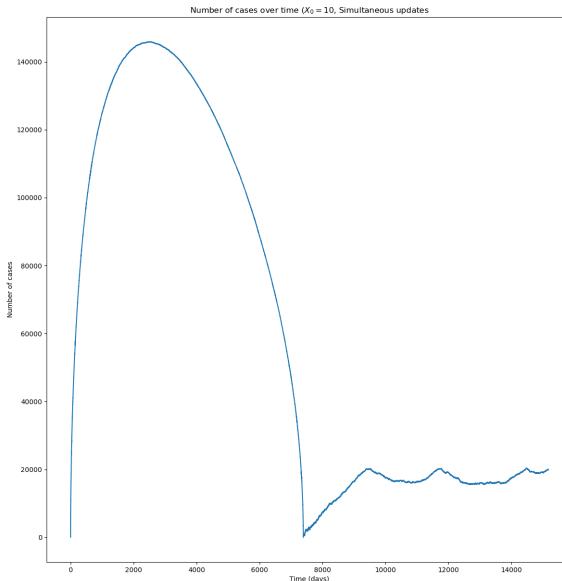


Figure 8: Trajectory of α , Simultaneous



I can't read
the numbers in plots
very well.

Figure 9: Number of cases over time, Simultaneous

From Figures 4-9, we can see that compared to the simultaneous updates, the sequential update method requires more iterations before convergence, has a higher number of cases in the first birth-death process before $X_t = 0$ and the process runs for a longer period of time (60000 vs 14000 days). However, we can also see that the values for each parameter obtained from both methods are roughly the same.

For each initial condition and parameter update method, a total of 5 simulations were run. Using the results from these simulations, we are able to construct confidence intervals (CIs) for each parameter using the following expression (at significance level γ , n is the number of samples, from Chapter 6):

$$I_\gamma(\hat{\theta}_n) = \hat{\theta}_n \pm t_{n-1, 1-\frac{\gamma}{2}} \sqrt{\frac{S_n^2}{n}}$$

very good!

(21)

where $\hat{\theta}_n$ is the sample average of θ and S_n^2 is the sample variance of θ . Using the raw data given in Appendix A, we get the following CIs for the parameters ($\gamma = 0.05$):

(Include references to support your model)

$n=5$

This is strange. Didn't we establish that the solution θ^* is not unique?

Method	X_0	Parameter	Lower bound	Upper bound
Sequential	10	λ	0.09828	0.09836
		α	4.2756e-8	4.4196e-8
	1000	λ	0.09823	0.09833
		α	4.3688e-8	4.5200e-8
Simultaneous	10	λ	0.09824	0.09838
		α	4.2470e-8	4.5233e-8
	1000	λ	0.09822	0.09840
		α	4.1996e-8	4.5539e-8

include
actual
estimate
of param.

Table 1: CIs for parameters

From Table 1, it can be seen that for the simultaneous updates, the CIs are slightly wider than the ones for the parameters using the sequential updates. However, even for a small sample size, it appears that the CI for both methods and all parameters are very small. In addition, it should be noted that for $X_0 = 1000$, the value for α for the two methods has a larger difference than the other parameters using different initial conditions and methods, likely caused by the higher transition rate given by the initial condition, as policies will need to be more aggressive in order to contain an outbreak. The differences may also seem small between parameter values, but they could still represent a relatively large difference in costs. *Is this statistically significant?*

From Figures 6 and 9, it is shown that beyond the initial first wave of an outbreak, we are able to successfully find the optimal parameters θ^* which minimises the cost of keeping the stationary probability of exceeding capacity equal to a certain value. However, more work is required to keep the time scale to more realistic values.

I don't understand how you got only one limit value. It's supposed to be a curve. (α^, λ^*)*

6 Conclusion

This project has explored how we can model a disease outbreak, how policies can be adjusted to keep the stationary probability of exceeding hospital capacity equal to a small value and to provide a simulation model to assist policymakers in implementing policy bundles to achieve this goal. The aim of keeping the probability equal to a certain value is to minimize the economic and social costs incurred as a result of policies implemented to control a disease outbreak within a population.

By using a birth-death process and associating the parameters in the process with certain policy implementations, we were able to build an estimator for the stationary probability and confirm convergence using Theorem 5.3 [1]. The estimator was then used in conjunction with target tracking (detailed in Chapter 3 [1]) to find the optimal parameters which kept the stationary probability at a certain level. Once a sample of optimal parameters was obtained, CIs were built for these parameters using Chapter 6 [1].

From the experimental results, it was found that for two methods of updating the parameters and with different initial conditions, the optimal values of λ and α were similar for all the experi-

ments, with the constructed CIs being relatively narrow, even for a small sample. Although the differences were small, it is possible that in reality, these small differences could represent large differences in the costs incurred by implementing certain policies.

In the future, this work could be expanded upon by implementing the gradient descent methods detailed in section 4 and Chapter 8 [1]. In addition, as the model used for the underlying process is relatively simple, we could attempt to construct a more realistic model which allows for deaths and population changes. Another aspect to explore would be to find how the process is affected by setting different initial values of λ and α and also exploring how changes in p affects the target tracking process.

7 Code

```
1 import numpy as np
2 import math
3 import matplotlib.pyplot as plt
4 # simulate the infected patients number for a specific period T_total
5 # (written in python)
6 # this function simulate not strictly numbers of X_n in T_total time, given
7 # fixed Lambda, P, Alpha
8 # Random Horizon: means that the whole trajectory stop once X_n hits 0
9 # @ param N: is the number of total population can be infected
10 # @ param t_0: a number that is the time when starting the T-time simulation
11 # @ param x_0: initial infected patients (greater than 0)
12 # @ param T_total: is total time for this simulation
13 # @ param H: is the hospital capacity
14 def simulation_infected(la, p, al, N, t_0, x_0, T_total, H):
15     # initialize time and infected patient
16     # bind variables to the initial conditions
17     x_n = x_0
18     # make sure the start is not 0
19     # assert(x_n!=0)
20     # assign initial time to a know value, noted it can start from
21     # not just at 0 time
22     t_n = t_0
23     # introduce T_H := "the total time above the hospital capacity"
24     T_H = 0
25     # initiate X_n process trajectory and corresponding T_n process
26     X_trajectory = []
27     T_trajectory = []
28     # loop
29     # break condition
30     # @ condition1: when the time exceeds the maximum time
31     # @ condition2: when the infected patients go to zero
32     # @ condition3: when the whole populations are infected! :(
33     while (x_n != 0 and
34            x_n < N
35            ):
36         i = x_n
37         # q_i - i+1
38         q_forward_i = la*p*2*x_n*(N-i) / (N*(N-1))
39         # q_i - i-1
40         q_backward_i = al*i
41         # waiting time rate v_i = (q_i - i+1) + (q_i - i-1)
42         v_i = q_forward_i + q_backward_i
43         t_i = np.random.exponential(v_i)
44         t_n = t_n+t_i
45         # calculate T_H before jumping
```

```

44     if (x_n >= H):
45         # makes sure T_H don't exceed T_total
46         T_H = min(T_total ,T_H+t_i)
47
48     if ((t_n-t_0) > T_total):
49         # take the unjumped x_n
50         # take the truncated T (multiple of 14)
51         X_trajectory.append(x_n)
52         T_trajectory.append(T_total+t_0)
53     return X_trajectory, T_trajectory, T_H, T_total
54
55 elif ((t_n-t_0) < T_total):
56     # jumping probability to STATE i+1 is (q_i - i+1)/v_i
57     jump = np.random.binomial(n=1,p=(q_forward_i/v_i))
58
59     # change x_n
60     if (jump ==1):
61         x_n += 1
62     elif (jump == 0):
63         x_n -= 1
64
65     # add the jumped X_n at t_i time
66     X_trajectory.append(x_n)
67     # increase time
68     T_trajectory.append(t_n)
69
70
71     # if hits then zero return
72     if (x_n == 0):
73         return X_trajectory, T_trajectory, T_H, T_total
74
75 # function takes that updates lambda with negative sign
76 def update_lambda(_lambda, T_H, T, Beta ,stepsize):
77     Y_n = T_H/T
78     _lambda -= float(stepsize) * (Y_n-Beta)
79     return _lambda
80
81 # function that updates p with negative sign
82 def update_p(_p, T_H, T, Beta, stepsize):
83     Y_n = T_H/T
84     _p -= float(stepsize) * (Y_n-Beta)
85     return _p
86
87 # function that updates alpha with positive sign
88 def update_alpha(_alpha, T_H, T, Beta, stepsize):
89     Y_n = T_H/T
90     _alpha += float(stepsize) * (Y_n-Beta)

```

```

91     return _alpha
92
93 # @ x_0 initial infected patient
94 # @ _lambda the initial lambda value
95 # @ _alpha the initial alpha value
96 # @ p the initial p value
97 # @ C is the hospital capacity
98 # @ Beta is the target tracking probability
99 # @ K is the batch size of each estimate
100 # @ The batch is defined here as the total K days of simulation starting from
101 #   ↵ a time point
102 # @ It is not defined as the number of replications of each simulation with T
103 #   ↵ days
104 x_0 = 10
105 _lambda = 0.1
106 _alpha = 1e-8
107 _p = 0.9
108 K = 14
109 H = 2e4
110 Beta = 0.05
111 N = 4e6
112 stepsize = [0.00001, 2e-10]
113
114 trajectory_lambda = []
115 trajectory_alpha = []
116 trajectory_X_n = []
117 trajectory_T_n = []
118 iteration = 0
119
120 x_n = x_0
121 t_n = 0
122
123 # used to decide which parameter to update for sequential updates
124 # remove this variable and lines 135, 138, 139 and 142 for simultaneous
125 #   ↵ updating of parameters
126 update = 0
127
128 # placeholder value for stationary probability
129 mu = 1
130
131 # iterates past the instability in parameters, and then iterates until target
132 #   ↵ is found, set iteration < 6000 if updating parameters simultaneously
133 while iteration < 12000 or abs(mu - Beta)/Beta > 0.1:
134     # simulate K days of X_n obtain T_H/T, starting from t_0 = 0 and x_0 =
135     #   ↵ x_n
136     X_n, T_n, T_H, T = simulation_infected(_lambda, _p, _alpha, N, t_n, x_n, K, H)
137
138     mu = T_H/T

```

```

133
134     # use K days simulation to update lambda and alpha
135     if update == 1:
136         _alpha = max(0, update_alpha(_alpha, T_H, T, Beta, stepsize\cite{book}))
137         trajectory_alpha.append(_alpha)
138         update = 0
139     else:
140         _lambda = max(0, update_lambda(_lambda, T_H, T, Beta, stepsize[0]))
141         trajectory_lambda.append(_lambda)
142         update = 1
143
144     # update the simulation trajectory
145     trajectory_X_n.extend(X_n)
146     trajectory_T_n.extend(T_n)
147
148     # update x_n and t_n using the last (latest) element of objective
149     # → simulation X_n and T_n
150     x_n = X_n[-1]
151     t_n = T_n[-1]
152
153     # if x_n is zero, jump up to x_0, restart simulation
154     if (x_n==0):
155         x_n = x_0
156
iteration+=1

```

Appendix A

λ	α
0.098346	4.30759e-8
0.098319	4.31294e-8
0.098321	4.37330e-8
0.098352	4.30583e-8
0.098275	4.43842e-8

Table A1: Sequential updates, $X_0 = 10$

λ	α
0.098277	4.45428e-8
0.098342	4.33913e-8
0.098258	4.48121e-8
0.098229	4.49086e-8
0.098283	4.45646e-8

Table A2: Sequential updates, $X_0 = 1000$

λ	α
0.098362	4.27654e-8
0.098237	4.52662e-8
0.098345	4.30922e-8
0.098260	4.48082e-8
0.098334	4.33233e-8

Table A3: Simultaneous updates, $X_0 = 10$

λ	α
0.098213	4.57387e-8
0.098258	4.48384e-8
0.098356	4.28764e-8
0.098363	4.27374e-8
0.098368	4.26476e-8

Table A4: Simultaneous updates, $X_0 = 1000$

References

- [1] F.J. Vazquez-Abad B. Heidergott. Stochastic gradient techniques for optimisation and learning. 2020.
- [2] Felisa J Vázquez-Abad and Bernd Heidergott. Gradient estimation for a problem in public transportation: A comparison of spa, sf and mvd. In *invited paper, submitted to Conference on Decision and Control*, 2003.