Optimal Policies for a Pandemic: A Stochastic Game Approach and a Deep Learning Algorithm

Yao Xuan[1], Robert Balkin[1], Jiequn Han[2], Ruimeng Hu[1,3], Hector D. Ceniceros[1]

[1] *Department of Mathematics, University of California, Santa Barbara*

[2] *Department of Mathematics, Princeton University*

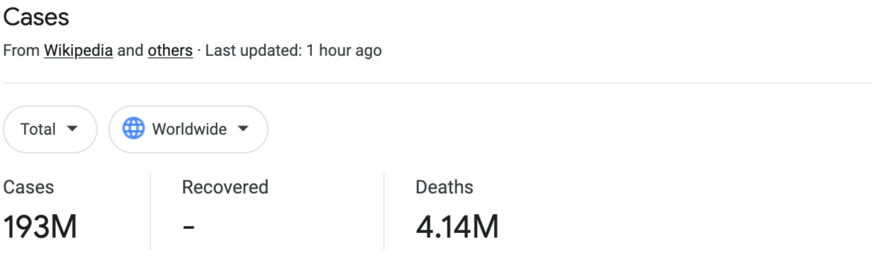
[3] *Department of Statistics and Applied Probability, University of California, Santa Barbara*

July 31, 2021

Content

- Introduction
- A multi-region SEIR model
- Enhanced deep fictitious play algorithm
- Experimental results

Pandemic: affecting a substantial number of people



- Covid-19 causes millions of deaths.
- Covid-19 significantly reduces economic growth.

Optimal Policies for a Pandemic

lockdown

vaccination

Figure: Two tools to fight with a pandemic.

Content

- Introduction
- A multi-region SEIR model
- Enhanced deep fictitious play algorithm
- Experimental results

A multi-region SEIR model

Susceptible, **E**xposed, **I**nfectious, and **R**emoved in a pandemic,

$$
\begin{aligned}
\mathrm{d}S_t^n &= -\sum_{k=1}^{N} \beta^{nk} S_t^n I_t^k (1-\theta \ell_t^n)(1-\theta \ell_t^k)\,\mathrm{d}t - v(h_t^n) S_t^n\,\mathrm{d}t - \sigma_{s_n} S_t^n\,\mathrm{d}W_t^{s_n},\\
\mathrm{d}E_t^n &= \sum_{k=1}^{N} \beta^{nk} S_t^n I_t^k (1-\theta \ell_t^n)(1-\theta \ell_t^k)\,\mathrm{d}t - \gamma E_t^n\,\mathrm{d}t + \sigma_{s_n} S_t^n\,\mathrm{d}W_t^{s_n} - \sigma_{e_n} E_t^n\,\mathrm{d}W_t^{e_n},\\
\mathrm{d}I_t^n &= (\gamma E_t^n - \lambda(h_t^n) I_t^n)\,\mathrm{d}t + \sigma_{e_n} E_t^n\,\mathrm{d}W_t^{e_n},\\
\mathrm{d}R_t^n &= \lambda(h_t^n) I_t^n\,\mathrm{d}t + v(h_t^n) S_t^n\,\mathrm{d}t, \quad n \in \mathcal{N} := \{1, 2, \dots, N\},
\end{aligned}
$$

Each planner n seeks to minimize its region's cost within a period $[0, T]$:

$$
J^n(\boldsymbol{\ell}, \boldsymbol{h}) := \mathbb{E}\left[\int_0^T e^{-rt} P^n \left[(S_t^n + E_t^n + I_t^n)\ell_t^n w + a(\kappa I_t^n \chi + p I_t^n c)\right] + e^{-rt}\eta (h_t^n)^2\,\mathrm{d}t\right]. \tag{1}
$$

Nash equilibrium

Definition

A Nash equilibrium is a tuple $(\boldsymbol{\ell}^*, \boldsymbol{h}^*) = (\ell^{1,*}, h^{1,*}, \ldots, \ell^{N,*}, h^{N,*}) \in \mathbb{A}^N$ *such that*

$$\forall n \in \mathcal{N}, \text{ and } (\ell^n, h^n) \in \mathbb{A}, \quad J^n(\boldsymbol{\ell}^*, \boldsymbol{h}^*) \leq J^n((\boldsymbol{\ell}^{-n,*}, \ell^n), (\boldsymbol{h}^{-n,*}, h^n)), \tag{2}$$

where $\boldsymbol{\ell}^{-n,*}$, $\boldsymbol{h}^{-n,*}$ *represent strategies of players other than the n-th one:*

$$\begin{aligned} \boldsymbol{\ell}^{-n,*} &:= [\ell^{1,*}, \ldots, \ell^{n-1,*}, \ell^{n+1,*}, \ldots, \ell^{N,*}], \\ \boldsymbol{h}^{-n,*} &:= [h^{1,*}, \ldots, h^{n-1,*}, h^{n+1,*}, \ldots, h^{N,*}], \end{aligned} \tag{3}$$

$\mathbb{A}$ *denotes the set of admissible strategies for each player and* $\mathbb{A}^N$ *is the produce of N copies of* $\mathbb{A}$.

Derivation of HJB equations

We derive below the Hamilton-Jacobi-Bellman (HJB) equations characterizing the Markovian Nash equilibrium.

$$\boldsymbol{X}_t \equiv [\boldsymbol{S}_t, \boldsymbol{E}_t, \boldsymbol{I}_t]^{\mathrm{T}} \equiv [S_t^1, \cdots, S_t^N, E_t^1, \cdots, E_t^N, I_t^1, \cdots, I_t^N]^{\mathrm{T}} \in \mathbb{R}^{3N}$$

The dynamics of $\boldsymbol{X}_t$ reads:

$$\mathrm{d}\boldsymbol{X}_t = b(t, \boldsymbol{X}_t, \boldsymbol{\ell}(t, \boldsymbol{X}_t), \boldsymbol{h}(t, \boldsymbol{X}_t))\,\mathrm{d}t + \Sigma(\boldsymbol{X}_t)\,\mathrm{d}\boldsymbol{W}_t, \tag{4}$$

Each player n aims to minimize the expected running cost

$$\mathbb{E}\left[\int_0^T f^n(t, \boldsymbol{X}_t, \ell^n(t, \boldsymbol{X}_t), h^n(t, \boldsymbol{X}_t))\,\mathrm{d}t\right]. \tag{5}$$

Define the value function of player n by

$$V^n(t, \boldsymbol{x}) = \inf_{(\ell^n, h^n)\in\mathbb{A}} \mathbb{E}\left[\int_t^T f^n(s, \boldsymbol{X}_s, \ell^n(s, \boldsymbol{X}_s), h^n(s, \boldsymbol{X}_s))\,\mathrm{d}s \middle| \boldsymbol{X}_t = \boldsymbol{x}\right]. \tag{6}$$

Derivation of HJB equations

By dynamic programming, it solves the following HJB system

$$\begin{cases} \partial_t V^n + \inf_{(\ell^n, h^n) \in [0,1]^2} H^n(t, \boldsymbol{x}, (\ell, \boldsymbol{h})(t, \boldsymbol{x}), \nabla_{\boldsymbol{x}} V^n) + \frac{1}{2} \mathrm{Tr}(\Sigma(\boldsymbol{x})^{\mathrm{T}} \mathrm{Hess}_{\boldsymbol{x}} V^n \Sigma(\boldsymbol{x})) = 0, \\ V^n(T, \boldsymbol{x}) = 0, \quad n \in \mathcal{N}, \end{cases} \tag{7}$$

where H^n is the usual Hamiltonian defined by

$$H^n(t, \boldsymbol{x}, \ell, \boldsymbol{h}, \boldsymbol{p}) = b(t, \boldsymbol{x}, \ell, \boldsymbol{h}) \cdot \boldsymbol{p} + f^n(t, \boldsymbol{x}, \ell^n, h^n), \tag{8}$$

Curse of dimensionality: N-coupled $3N + 1$ dimensional nonlinear equations

- Deep fictitious play (Han-Hu '20): Deep learning + fictitious play
- Enhanced deep fictitious play: break bottlenecks of time complexity and memory complexity.

Content

- Introduction
- A multi-region SEIR model
- Enhanced deep fictitious play algorithm
- Experimental results

Enhanced deep fictitious play algorithm

To solve such a high-dimensional stochastic game($\alpha = (l, h)$):

- Initialize $V^{n,0}$ and $\alpha^{n,0}$, $n \in \mathcal{N}$ by $2N$ neural networks.
- At the beginning of stage $m+1$, value functions $\tilde{\boldsymbol{V}}^m$ and best responses $\tilde{\alpha}^m$ at stage m is observable by all players.
- During stage $m+1$, fictitious play $\xrightarrow{\text{decoupling}}$ N optimization problems $\rightarrow$ solved simultaneously $\rightarrow$ $\tilde{\boldsymbol{V}}^{m+1}, \tilde{\alpha}^{m+1}$
  - usually not analytically tractable, solve numerically
  - use deep BSDE method (cf. Han-Jentzen-E, CMS('17), PNAS('18))
- Repeat $\tilde{\alpha}^{m+1}$ converges $\rightarrow$ a Nash equilibrium.

Remark: On top of *deep fictitious play*, N additional neural networks $\tilde{\alpha}$ are introduced to approximate policies, which is cheaper to evaluate.

$$DFP : \tilde{\alpha}^{m+1}(\tilde{\boldsymbol{V}}^1, \ldots, \tilde{\boldsymbol{V}}^m) \rightarrow EDFP : \text{neural networks } \tilde{\alpha}^{m+1}$$

From high dimensions to low dimensions

HJB equation decoupled by fictitious play to *N* separate equations

$$\partial_t V^n + \frac{1}{2}\mathrm{Tr}(\Sigma(\boldsymbol{x})^{\mathrm{T}} \mathrm{Hess}_{\boldsymbol{x}} V^n \Sigma(\boldsymbol{x})) + \mu^n(t, \boldsymbol{x}; \boldsymbol{\ell}^{-n}, \boldsymbol{h}^{-n}) \cdot \nabla_{\boldsymbol{x}} V^n + g^n(t, \boldsymbol{x}, \Sigma(\boldsymbol{x})^{\mathrm{T}} \nabla_{\boldsymbol{x}} V^n; \boldsymbol{\ell}^{-n}, \boldsymbol{h}^{-n}) = 0, \quad (9)$$

with some functions μ^n and g^n.

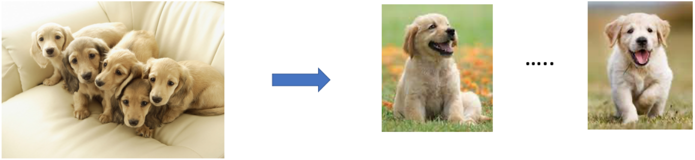


Figure: Decoupling *N* coupled equations to *N* separate equations to be solved in parallel.

Enhanced deep fictitious play

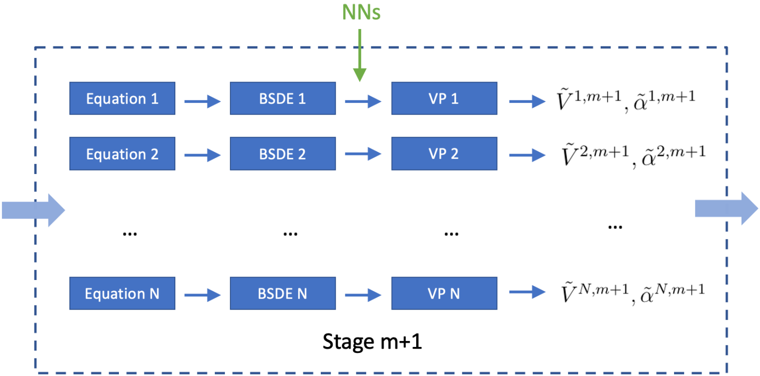


Figure: Illustration of one stage of *enhanced deep fictitious play.* BSDE: Backward stochastic differential equations. VP: variational problem.

BSDE

The solution is then approximated by solving the equivalent BSDE $(\boldsymbol{X}_t^n, Y_t^n, Z_t^n) \in \mathbb{R}^{3N} \times \mathbb{R} \times \mathbb{R}^{2N}$:

$$\begin{cases} \boldsymbol{X}_t^n = \boldsymbol{x}_0 + \int_0^t \mu^n(s, \boldsymbol{X}_s^n; (\ell^{-n}, \boldsymbol{h}^{-n})(s, \boldsymbol{X}_s^n)) \, \mathrm{d}s + \int_0^t \Sigma(\boldsymbol{X}_s^n) \, \mathrm{d}\boldsymbol{W}_s, \\ Y_t^n = \int_t^T g^n(s, \boldsymbol{X}_s^n, Z_s^n; (\ell^{-n}, \boldsymbol{h}^{-n})(s, \boldsymbol{X}_s^n)) \, \mathrm{d}s - \int_t^T (Z_s^n)^\top \, \mathrm{d}\boldsymbol{W}_s, \end{cases} \tag{10}$$

in the sense of

$$Y_t^n = V^n(t, \boldsymbol{X}_t^n) \quad \text{and} \quad Z_t^n = \Sigma(\boldsymbol{X}_t^n)^\top \nabla_{\boldsymbol{x}} V^n(t, \boldsymbol{X}_t^n). \tag{11}$$

Variational problem

The BSDE is solved by a variational problem,

$$\begin{aligned} &\inf_{Y_0^n, \tilde{\alpha}^n, \{Z_t^n\}_{0\le t\le T}} \mathbb{E}\left(\left|Y_T^n\right|^2 + \tau \int_0^T \left\|\alpha^n(s, \boldsymbol{X}_s^n) - \tilde{\alpha}^n(s, \boldsymbol{X}_s^n)\right\|_2^2 \,\mathrm{d}s\right) \\ \text{s.t. } &\boldsymbol{X}_t^n = \boldsymbol{x}_0 + \int_0^t \mu^n(s, \boldsymbol{X}_s^n; \tilde{\alpha}^{-n}(s, \boldsymbol{X}_s^n))\,\mathrm{d}s + \int_0^t \Sigma(\boldsymbol{X}_s^n)\,\mathrm{d}\boldsymbol{W}_s, \\ &Y_t^n = Y_0^n - \int_0^t g^n(s, \boldsymbol{X}_s^n, Z_s^n; \tilde{\alpha}^{-n}(s, \boldsymbol{X}_s^n))\,\mathrm{d}s + \int_0^t (Z_s^n)^{\mathsf{T}}\,\mathrm{d}\boldsymbol{W}_s, \\ &\alpha^n(s, \boldsymbol{X}_s^n) = \arg\min_{\beta^n} H^n(s, \boldsymbol{X}_s^n, (\beta^n, \tilde{\alpha}^{-n})(s, \boldsymbol{X}_s^n), Z_s^n), \end{aligned} \tag{12}$$

Repeat updating $\tilde{\alpha}^{n,\pi}$ until convergence.

Numerical Discretization to BSDE

$$\inf_{\psi_0 \in \mathcal{N}_0^{n'}, \{\phi_k \in \mathcal{N}_k^{n}, \xi_k \in \mathcal{N}_k^{n''}\}_{k=0}^{N_T-1}} \mathbb{E}\{|Y_T^{n,\pi}|^2 + \tau \sum_k \left\|\alpha_k^{n,\pi} - \tilde{\alpha}_k^{n,\pi}(\boldsymbol{X}_k^{n,\pi})\right\|_2^2 \Delta t_k\}$$

$$\begin{aligned}
\text{s.t. } & \boldsymbol{X}_0^{n,\pi} = \boldsymbol{X}_0, \quad Y_0^{n,\pi} = \psi_0\left(\boldsymbol{X}_0^{n,\pi}\right), \quad Z_k^{n,\pi} = \phi_k\left(\boldsymbol{X}_k^{n,\pi}\right), \quad \tilde{\alpha}_k^{n,\pi}(\boldsymbol{X}_k^{n,\pi}) = \xi_k(\boldsymbol{X}_k^{n,\pi}), \\
& \alpha_k^{n,\pi} = \arg\min_{\beta^n} H^n(t_k, \boldsymbol{X}_k^{n,\pi}, (\beta^n, \tilde{\boldsymbol{\alpha}}_k^{-n,\pi})(\boldsymbol{X}_k^{n,\pi}), Z_k^{n,\pi}), \quad k = 0, \ldots, N_T - 1 \\
& \boldsymbol{X}_{k+1}^{n,\pi} = \boldsymbol{X}_k^{n,\pi} + \mu^n\left(t_k, \boldsymbol{X}_k^{n,\pi}; \tilde{\boldsymbol{\alpha}}_k^{-n,\pi}(\boldsymbol{X}_k^{n,\pi})\right)\Delta t_k + \Sigma\left(t_k, \boldsymbol{X}_k^{n,\pi}\right)\Delta \boldsymbol{W}_k, \\
& Y_{k+1}^{n,\pi} = Y_k^{n,\pi} - g^n\left(t_k, \boldsymbol{X}_k^{n,\pi}, Z_k^{n,\pi}; \tilde{\boldsymbol{\alpha}}_k^{-n,\pi}(\boldsymbol{X}_k^{n,\pi})\right)\Delta t_k + \left(Z_k^{n,\pi}\right)^{\mathrm{T}}\Delta \boldsymbol{W}_k,
\end{aligned}$$

Repeat updating $\tilde{\alpha}^{n,\pi}$ until convergence.

Improvement of time and memory complexity

	DFP	EDFP
Memory complexity	$O(m)$	$O(1)$
Time complexity	$O(m^2)$	$O(m)$

Table: Memory and time cost of *Deep Fictitious Play* (DFP) and *Enhanced Deep Fictitious Play* (EDFP) in solving each equation.

Content

- Introduction
- A multi-region SEIR model
- Enhanced deep fictitious play algorithm
- Experimental results

Numerical Results

- Case studies on optimal policies of COVID-19.
- Stochastic game among 3 states, NY, NJ and PA.
- Simulation from 03/15/2020 to 09/15/2020, vaccination was not available ($v = 0$).
- Simulations on how lockdown policies influence the pandemic in different settings.

Dependence of policies on a

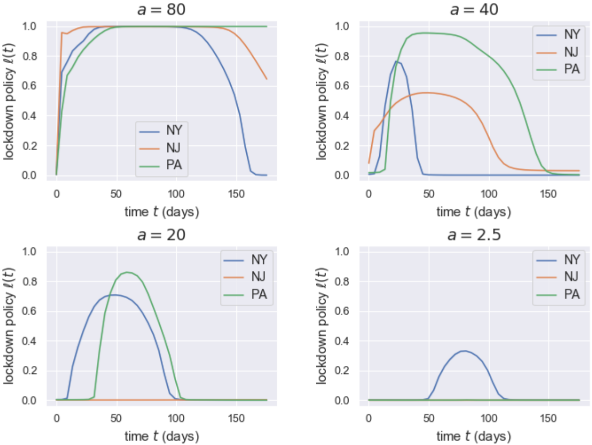


Figure: Optimal policies $\ell(t)$ with different choice of a (planners' view on the death of human beings) for three states: New York (blue), New Jersey (orange) and Pennsylvania (green), lockdown efficiency $\theta = 0.9$.

Dependence of policies on θ

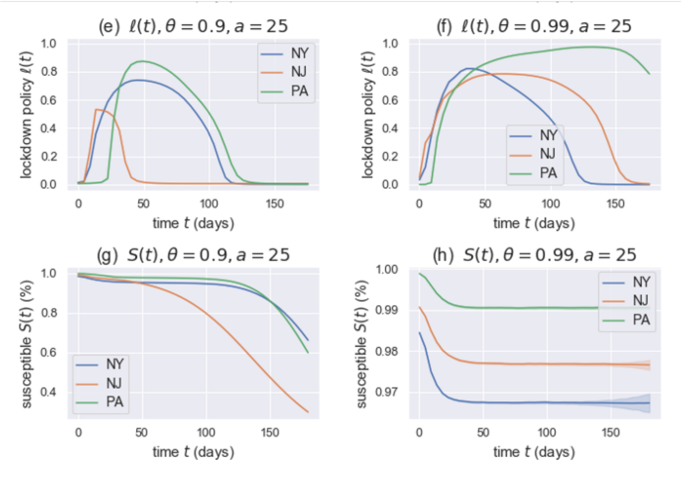


Figure: Comparison of optimal policies for three states (NY = blue, NJ = orange, PA = green) and their susceptibles under different θ (lockdown efficiency, residents' willingness to comply with the policy).

Summary

- Build a multi-region SEIR model for a pandemic to find optimal policies.
- Propose *enhanced deep fictitious play* algorithm to solve the high dimensional problem.
- Case studies on COVID-19 show the importance of θ (planners' view on the death of human beings) and a (residents' willingness to comply with the policy).

Thanks!