Comparison : DQN vs. Adjoint Method on Optimal control

Boyeon Kim

Department of Mathematics, School of Mathematics and Computing
Mathematics
Yonsei University

January 6, 2023

SLIAR Optimal control

- GAOL : SLIAR Model optimal control
- Method 1 : Adjoint Method
- Method 2 : DQN

SLIAR Model

- SLIAR Model Equations and structure.

$$\begin{cases} S' &= -\beta(1-\sigma)S\Lambda - \nu S \\ L' &= \beta(1-\sigma)S\Lambda - \kappa L \\ I' &= p\kappa L - \alpha I - \tau I \\ A' &= (1-p)\kappa L - \eta A \end{cases} \qquad \text{with } \Lambda = \epsilon L + (1-q)I + \delta A$$

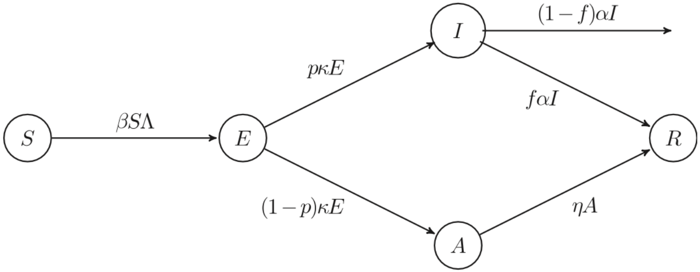


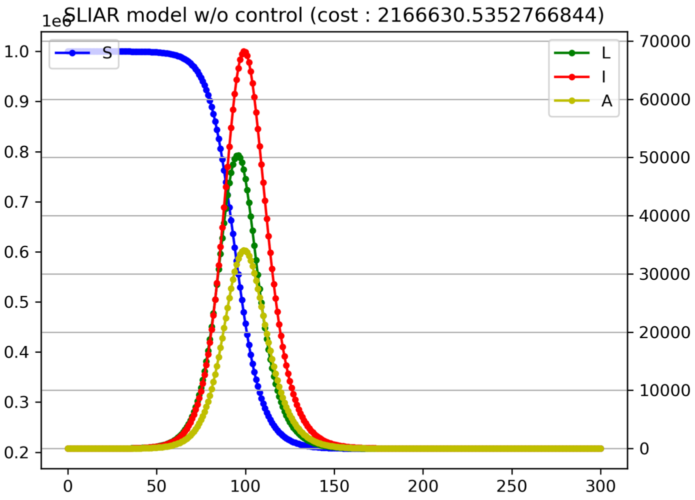
Fig. 1. SEIAR epidemic model.

SLIAR model parameters

```
Start: 0
End: 300

S0: 1000000
L0: 0
I0: 1
A0: 0

Reproduction number
: 1.9847
```

```
beta: 7.26582E-07
sigma: 0
kappa: 0.526
alpha: 0.244
tau: 0
p: 0.667
eta: 0.244
epsilon: 0
q: 0.5
delta: 1

P: 1
Q: 1
R: 0
W: 0
```

SLIAR w/o constrol



SLIAR optimal control

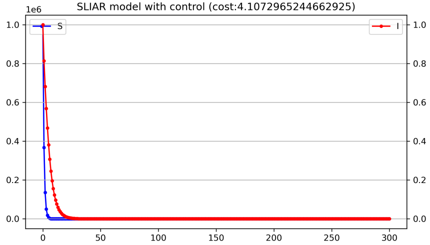
$$\min_{u \in \mathcal{U}_{ad}} \int_0^T PI(t) + Q\nu^2(t) + R\tau^2(t) + W\sigma^2(t)dt$$

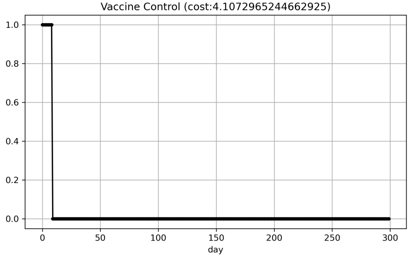
subject to

$$\begin{cases} S' &= -\beta(1-\sigma)S\Lambda - \nu S \\ L' &= \beta(1-\sigma)S\Lambda - \kappa L \\ I' &= p\kappa L - \alpha I - \tau I \\ A' &= (1-p)\kappa L - \eta A \end{cases} \qquad with \quad \Lambda = \epsilon L + (1-q)I + \delta A$$

SLIAR optimal control

- $\min_{u \in \mathcal{U}_{ad}} \int_0^T PI(t) + Q\nu^2(t) + R\tau^2(t) + W\sigma^2(t) dt$
- Method : DQN
- P = 1, Q = 1, iteration : 10,000





SLIAR optimal control

- Method : Adjoint Method
- Hamilonian

$$
\begin{aligned}
H &= f + \lambda \cdot g \\
&= PI + Q\nu^2 + R\tau^2 + W\sigma^2 + \begin{bmatrix} \lambda_S \\ \lambda_L \\ \lambda_I \\ \lambda_A \end{bmatrix}' \cdot \begin{bmatrix} -\beta(1-\sigma)S\Lambda - \nu S \\ \beta(1-\sigma)S\Lambda - \kappa L \\ p\kappa L - \alpha I - \tau I \\ (1-p)\kappa L - \eta A \end{bmatrix} \\
\lambda' &= -\frac{\partial H}{\partial x}
\end{aligned}
$$

SLIAR optimal control

- Method : Adjoint Method
- Adjoint Equations

$$\begin{cases} \lambda_S' &= \nu\lambda_S + \beta(1-\sigma)\Lambda(\lambda_S - \lambda_L) \\ \lambda_L' &= \beta(1-\sigma)\epsilon S(\lambda_S - \lambda_L) - \kappa p(\lambda_I - \lambda_A) + \kappa(\lambda_L - \lambda_A) \\ \lambda_I' &= -P + \beta(1-q)(1-\sigma)S(\lambda_S - \lambda_L) + \lambda_I(\alpha + \tau) \\ \lambda_A' &= \beta(1-\sigma)\delta S(\lambda_S - \lambda_L) + \eta\lambda_A \end{cases}$$

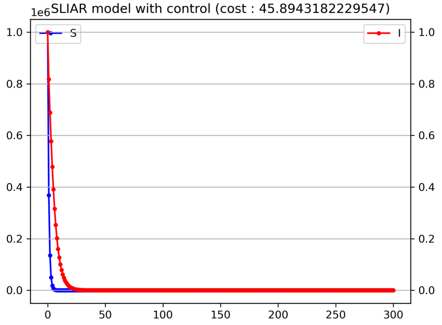
SLIAR optimal control

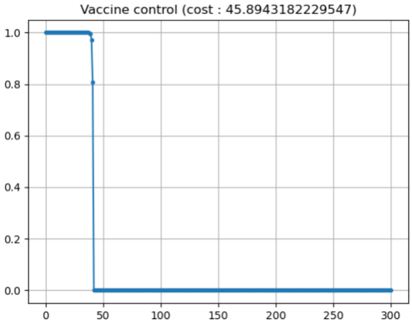
- Method : Adjoint Method
- Foward-Backward Sweep

```python
alpha = 1E-5
old_cost = 1E8
for it in range(MaxIter + 1):
    # State (Forward)
    nu_intp = lambda tc: np.interp(tc, t, nu0)
    sol = odeint(sliar_by_sympy, y0, t, args=(beta, sigma, kappa, alpha, tau, p, eta, epsilon, q, delta, P, Q, nu_intp))

    # Cost
    S, L, I, A = np.hsplit(sol, 4)
    S_mid = (S[1:] + S[:-1]) / 2.
    L_mid = (L[1:] + L[:-1]) / 2.
    I_mid = (I[1:] + I[:-1]) / 2.
    A_mid = (A[1:] + A[:-1]) / 2.
    nu_mid = (nu0[1:] + nu0[:-1]) / 2.
    cost1 = dt * np.sum(I_mid.flatten())
    cost2 = dt * np.sum(nu_mid.flatten() ** 2)
    cost = cost1 + cost2

    # Adjoint (Backward)
    nu_intp = lambda tc: np.interp(tf - tc, t, nu0)
    x_intp = lambda tc: np.array([np.interp(tf - tc, t, sol[:, 0]), np.interp(tf - tc, t, sol[:, 1]), np.interp(tf - tc, t, sol[:, 2]), np.interp(tf - tc, t, sol[:, 3])])
    y_T = np.array([0,0,0,0])
    l_sol = odeint(adjoint_sliar_by_sympy, y_T, t, args=(x_intp, beta, sigma, kappa, alpha, tau, p, eta, epsilon, q, delta, P, Q, nu_intp))


    # Simple Gradient
    # FOR optimality condition
    Hnu = 2 * P * nu0 - l_sol[:,0] * sol[:, 0]
    nu1 = np.clip(nu0 - alpha * Hnu , 0, 1)

    if old_cost < cost:
        alpha = alpha / 10 # simple adaptive learning rate

    # Convergence
    if np.abs(old_cost - cost) / alpha  <= 1E-7:
        break
```

SLIAR optimal control

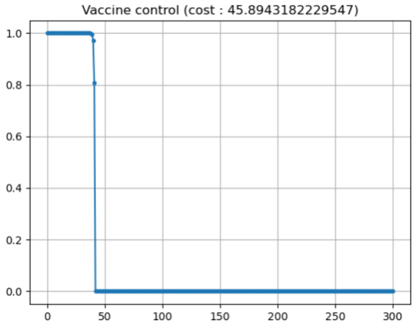
- $\min_{u\in\mathcal{U}_{ad}} \int_0^T PI(t) + Q\nu^2(t) + R\tau^2(t) + W\sigma^2(t)dt$
- Method : Adjoint Method
- P = 1, Q = 1

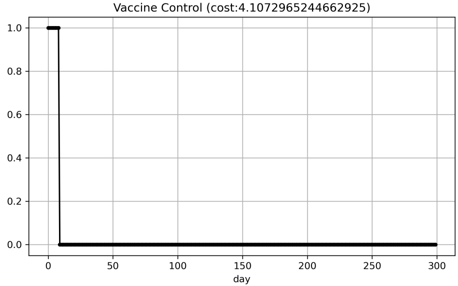




Adjoint vs. DQN

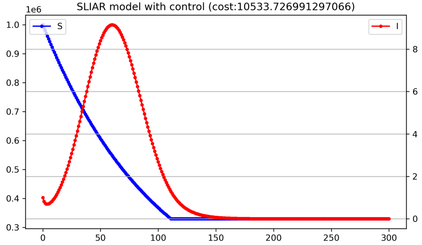
- $\min_{u \in \mathcal{U}_{ad}} \int_0^T PI(t) + Q\nu^2(t) + R\tau^2(t) + W\sigma^2(t)dt$
- Adjoint Method (left) vs. DQN (right)

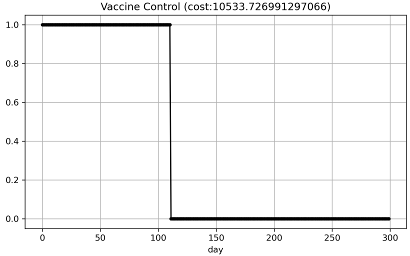




Another result of DQN

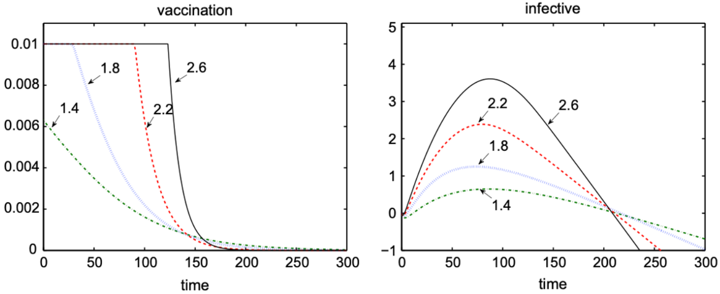
- $\min_{u \in \mathcal{U}_{ad}} \int_0^T PI(t) + Q\nu^2(t) + R\tau^2(t) + W\sigma^2(t)dt$
- Method : DQN
- P = 1, Q = 1e6, $\nu_max = 0.01$, iteration : 10,000





Result of paper

- $\min_{u \in \mathcal{U}_{ad}} \int_0^T PI(t) + Q\nu^2(t) + R\tau^2(t) + W\sigma^2(t)dt$
- Method : Adjoint Method
- P = 1, Q = 1e6, $\nu_{max} = 0.01$



Next todo list

- Fix the adjoint method
- Comparison : Adjoint method vs. DQN
- Apply $\nu_{max} = 0.01$ to the adjoint method