**Pseudo code for optimization**

**Finite Horizon Q-learning**

* + **Step 1**

1. Run the PATH model till 2016 and export the "World-Results-dryrun2.csv".
2. In the beginning of the optimization code import "World-Results-dryrun2.csv", so that there is no need to run the dry run in each simulation.
3. Define l (size of state space), u (size of action space), S (size of stage)
4. Define kmax (max number of iterations) and smax (max number of stage)
5. Initialize Q(i,a,s) = 0 for all i ϵ Ω, a ϵ A, s ϵ S
6. Initialize Q(j,T+1,b) = 0 for all j ϵ Ω, a ϵ A(i, S)
7. Start simulation at the starting state, which is assumed to be known with certainty (2015:S = 1)
8. Find current state from the "World-Results-dryrun2.csv”, “i”
   * **Step 2**
9. Current state “i”, Current stage “s”
10. Select action with probability
11. Simulate action “a”
12. Find the state system transition to: next state “j”
13. Next stage: “s+1”
14. Find the immediate reward: r(i,s,a,j,s+1)

r(i,s,a,j,s+1) = QALY – costs

QALY = sum QALYs whole population in a year

Costs = sum costs whole population in a year

In PATH: costs = util-cost + regimen-cost-quarter + oi-cost-quarter + care-service-cost + test-cost

* + **Step 3**

1. Update Q(i,s,a)

Q(i,s,a) = (1 – α) Q(i,s,a) + α[r(i,s,a,j,s+1) + λ max Q(j,s+1,b)]

1. k = k+1, s = s+1, if s = T+1 >>>> s = 1
2. if k<kmax >>> i = j go to 2

O.W >>> go to 4

* + **Step 4**

1. d(l,s) ϵ argmax Q(l,s,b) b ϵ A(i, S)

for each l ϵ Ω

s ϵ S