Optimisation of Multiple Control Strategies on a Dengue Fever Model.

Sandra Montes (@slmontes)

2025-03-10

Sandra Montes (@slmontes), 2025-02-20

## Introduction

This example explores the optimal control on a dengue fever model using multiple combinations of NPIs strategies. The model and strategies tested match those presentented by [Asamoah et al. (2021)](https://www.sciencedirect.com/science/article/pii/S2211379721009487?via%3Dihub). The model takes into account 2 populations: human and female mosquito or vector population. The human population is divided into 5 compartments: susceptible S\_h, infected (symptomatic) I\_h, carrier (asymptomatic) I\_hA, partially immune P, and recovered R\_h. While the mosquito population is formed by 2 compartments: susceptible S\_v and infected I\_v.

The model is described by the following differential equations:

## Libraries

using JuMP  
using Ipopt  
using OrdinaryDiffEq  
using Plots  
using Measures  
using MAT

## Functions

Model

function dengue\_ode!(du,u,p,t)  
 (S\_h, I\_h, I\_hA, P, R\_h, S\_v, I\_v) = u  
 (β1, β2, β3, ρ, ψ, γ\_h, ω, μ\_h, μ\_v, φ, r\_0, u1, u2, u3, u4) = p  
 N\_h = S\_h + I\_h + I\_hA + P + R\_h  
 N\_v = S\_v + I\_v  
 λ\_h = ((1 - u1) \* b \* β1 / N\_h) \* I\_v  
 λ\_h1 = ((1 - u1) \* b \* β2 / N\_h) \* I\_v  
 λ\_v = (b \* β3 / N\_h) \* (I\_h + I\_hA)  
 @inbounds begin  
 du[1] = μ\_h \* N\_h - λ\_h \* S\_h - S\_h \* u2 - μ\_h \* S\_h # dS\_h/dt  
 du[2] = ψ \* λ\_h \* S\_h + ω \* λ\_h1 \* P - (μ\_h + u3 + γ\_h) \* I\_h # dI\_h/dt  
 du[3] = (1 - ψ) \* λ\_h \* S\_h + (1 - ω) \* λ\_h1 \* P - (μ\_h + γ\_h) \* I\_hA # dI\_hA/dt  
 du[4] = u2 \* S\_h + ρ \* u3 \* I\_h + φ \* γ\_h \* (I\_h + I\_hA) - λ\_h1 \* P - μ\_h \* P # dP/dt  
 du[5] = (1 - ρ) \* u3 \* I\_h + (1 - φ) \* γ\_h \* (I\_h + I\_hA) - μ\_h \* R\_h # dR\_h/dt  
 du[6] = μ\_v \* N\_v \* (1 - u4) - λ\_v \* S\_v - μ\_v \* S\_v - r\_0 \* u4 \* S\_v # dS\_v/dt  
 du[7] = λ\_v \* S\_v - μ\_v \* I\_v - r\_0 \* u4 \* I\_v # dI\_v/dt  
 end  
 nothing  
end;

To create plots

function create\_combined\_plot(ts, u\_opts, S\_h\_opts, I\_h\_opts, I\_hA\_opts, P\_opts, R\_h\_opts, S\_v\_opts, I\_v\_opts, scenario\_label; P\_axis=3000, P\_legend=:topright, save=false)  
 color\_palette = [:red, :blue, :green, :magenta, :purple, :cyan, :orange, :yellow];  
 linestyle\_palette = [:solid, :dash, :dot, :dashdot, :dashdotdot, :dash, :dot, :dashdot];  
  
 combined\_plot = plot(layout = (3, 2), dpi=300, size=(1200,1500),  
 left\_margin=10mm, right\_margin=10mm, top\_margin=10mm, bottom\_margin=10mm)  
  
 # First subplot:   
 for (i, I\_h\_opt) in enumerate(I\_h\_opts)  
 plot!(combined\_plot[1, 1], ts, I\_h\_opt,   
 label="Scenario $(scenario\_label[i])",   
 linewidth=3, color=color\_palette[i], thickness\_scaling=1,  
 xlim=(0, ts[end]),  
 ylim=(0, 600),  
 xtickfontsize=12, ytickfontsize=12,  
 xguidefontsize=14, yguidefontsize=14,  
 legendfontsize=12, legend=:topright)  
 end  
 xlabel!(combined\_plot[1, 1], "Time (days)")  
 ylabel!(combined\_plot[1, 1], "Symptomatic humans, I\_h")  
 title!(combined\_plot[1, 1], "Symptomatic humans, I\_h")  
   
   
 # Second subplot:   
 for (i, I\_hA\_opt) in enumerate(I\_hA\_opts)  
 plot!(combined\_plot[1, 2], ts, I\_hA\_opt,   
 label="Scenario $(scenario\_label[i])",   
 linewidth=3, color=color\_palette[i], thickness\_scaling=1,  
 xlim=(0, ts[end]+1),  
 ylim=(0, 800),  
 xtickfontsize=12, ytickfontsize=12,  
 xguidefontsize=14, yguidefontsize=14,  
 legendfontsize=12, legend=:topright)  
 end  
 xlabel!(combined\_plot[1, 2], "Time (days)")  
 ylabel!(combined\_plot[1, 2], "Asymptomatic humans, I\_hA")  
 title!(combined\_plot[1, 2], "Asymptomatic humans, I\_hA")  
   
 # Third subplot:   
 for (i, S\_v\_opt) in enumerate(S\_v\_opts)  
 plot!(combined\_plot[2, 1], ts, S\_v\_opt,   
 label="Scenario $(scenario\_label[i])",   
 linewidth=3, color=color\_palette[i], thickness\_scaling=1,  
 xlim=(0, ts[end]),  
 ylim=(0, 6500),  
 xtickfontsize=12, ytickfontsize=12,  
 xguidefontsize=14, yguidefontsize=14,  
 legendfontsize=12,  
 legend=:right)  
 end  
 xlabel!(combined\_plot[2, 1], "Time (days)")  
 ylabel!(combined\_plot[2, 1], "Susceptible mosquitoes, S\_v")  
 title!(combined\_plot[2, 1], "Susceptible mosquitoes, S\_v")  
  
 # Fourth subplot:   
 for (i, I\_v\_opt) in enumerate(I\_v\_opts)  
 plot!(combined\_plot[2, 2], ts, I\_v\_opt,   
 label="Scenario $(scenario\_label[i])",   
 linewidth=3, color=color\_palette[i], thickness\_scaling=1,  
 xlim=(0, ts[end]),  
 ylim=(0, 3500),  
 xtickfontsize=12, ytickfontsize=12,  
 xguidefontsize=14, yguidefontsize=14,  
 legendfontsize=12, legend=:topright)  
 end  
 xlabel!(combined\_plot[2, 2], "Time (days)")  
 ylabel!(combined\_plot[2, 2], "Infected mosquitoes, I\_v")  
 title!(combined\_plot[2, 2], "Infected mosquitoes, I\_v")  
  
 # Fifth subplot:   
 for (i, P\_opt) in enumerate(P\_opts)  
 plot!(combined\_plot[3, 1], ts, P\_opt,   
 label="Scenario $(scenario\_label[i])",   
 linewidth=3, color=color\_palette[i], thickness\_scaling=1,  
 xlim=(0, ts[end]),  
 ylim=(0, P\_axis),  
 xtickfontsize=12, ytickfontsize=12,  
 xguidefontsize=14, yguidefontsize=14,  
 legendfontsize=12, legend=(P\_legend))  
 end  
 xlabel!(combined\_plot[3, 1], "Time (days)")  
 ylabel!(combined\_plot[3, 1], "Partially immune, P")  
 title!(combined\_plot[3, 1], "Partially immune, P")  
  
 if !isempty(u\_opts)  
 # u\_opts = [u1\_opt\_1, u2\_opt\_1, u3\_opt\_1, u4\_opt\_1]  
 for (i, u\_opt) in enumerate(u\_opts)  
 plot!(combined\_plot[3, 2], ts, u\_opt, label="u$i", linewidth=3, color=color\_palette[i], linestyle=linestyle\_palette[i], thickness\_scaling=1,  
 xlim=(0, ts[end]+1),  
 ylim=(0, 1.1),  
 xtickfontsize=12, ytickfontsize=12,  
 xguidefontsize=14, yguidefontsize=14,  
 legendfontsize=12, legend\_columns=4,  
 legend=:topright)  
 end  
 xlabel!(combined\_plot[3, 2], "Time (days)")  
 title!(combined\_plot[3, 2], "Control Profiles Scenario $(scenario\_label[2])")  
 end  
   
 display(combined\_plot)  
  
 # Save the plot if `save=true`  
 if save  
 filename = "OptControl\_Dengue\_$((scenario\_label[2])).png"  
 savefig(combined\_plot, filename)  
 println("Plot saved as $filename")  
 end  
end;

To easily run the optimisation of different strategies

function run\_optimization(u1\_max, u2\_max, u3\_max, u4\_max)  
  
 # Model definition with Ipopt optimizer  
 model = Model(Ipopt.Optimizer)  
 set\_optimizer\_attribute(model, "max\_iter", 10000)  
 set\_optimizer\_attribute(model, "print\_level", 5)   
 set\_optimizer\_attribute(model, "tol", 1e-6)  
  
 u1\_init = 0  
 u2\_init = 0  
 u3\_init = 0  
 u4\_init = 0  
  
 T = Int(tf/dt)  
  
 @variable(model, S\_h[1:(T+1)] >= 0) # Susceptible humans  
 @variable(model, I\_h[1:(T+1)] >= 0) # Infected symptomatic  
 @variable(model, I\_hA[1:(T+1)] >= 0) # Carriers asymptomatic  
 @variable(model, P[1:(T+1)] >= 0) # Partially immune  
 @variable(model, R\_h[1:(T+1)] >= 0) # Recovered humans  
 @variable(model, S\_v[1:(T+1)] >= 0) # Susceptible mosquitoes  
 @variable(model, I\_v[1:(T+1)] >= 0) # Infected mosquitoes   
   
 @variable(model, 0 <= u1[1:(T+1)] <= u1\_max) # Treated bednet   
 @variable(model, 0 <= u2[1:(T+1)] <= u2\_max) # Vaccination  
 @variable(model, 0 <= u3[1:(T+1)] <= u3\_max) # Treatment (prophylactics)  
 @variable(model, 0 <= u4[1:(T+1)] <= u4\_max) # Insecticides  
   
 # Initial conditions  
 @constraints(model, begin  
 S\_h[1] == S\_h0  
 I\_h[1] == I\_h0  
 I\_hA[1] == I\_hA0  
 P[1] == P0  
 R\_h[1] == R\_h0  
 S\_v[1] == S\_v0  
 I\_v[1] == I\_v0  
 u1[1] == u1\_init  
 u2[1] == u2\_init  
 u3[1] == u3\_init  
 u4[1] == u4\_init  
 [t=(T+1)], u1[t] == u1[t-1]  
 [t=(T+1)], u2[t] == u2[t-1]  
 [t=(T+1)], u3[t] == u3[t-1]  
 [t=(T+1)], u4[t] == u4[t-1]  
 end);  
  
 # Population sizes and infection rates  
 @expressions(model, begin  
 N\_h[t=1:T], (S\_h[t] + I\_h[t] + I\_hA[t] + P[t] + R\_h[t])  
 N\_v[t=1:T], (S\_v[t] + I\_v[t])  
 lambda\_h[t=1:T], ((1 - u1[t]) \* b \* beta\_1 / N\_h[t]) \* I\_v[t]  
 lambda\_h1[t=1:T], ((1 - u1[t]) \* b \* beta\_2 / N\_h[t]) \* I\_v[t]  
 lambda\_v[t=1:T], (b \* beta\_3 / N\_h[t]) \* (I\_h[t] + I\_hA[t])  
 end)  
  
 # ODEs  
 @constraints(model, begin  
 [t=1:T], S\_h[t+1] == S\_h[t] + (mu\_h \* N\_h[t] - lambda\_h[t] \* S\_h[t] - S\_h[t] \* u2[t] - mu\_h \* S\_h[t]) \* dt  
 [t=1:T], I\_h[t+1] == I\_h[t] + (psi \* lambda\_h[t] \* S\_h[t] + omega \* lambda\_h1[t] \* P[t] - (mu\_h + u3[t] + gamma\_h) \* I\_h[t]) \* dt  
 [t=1:T], I\_hA[t+1] == I\_hA[t] + ((1 - psi) \* lambda\_h[t] \* S\_h[t] + (1 - omega) \* lambda\_h1[t] \* P[t] - (mu\_h + gamma\_h) \* I\_hA[t]) \* dt  
 [t=1:T], P[t+1] == P[t] + (u2[t] \* S\_h[t] + rho \* u3[t] \* I\_h[t] + phi \* gamma\_h \* (I\_h[t] + I\_hA[t]) - lambda\_h1[t] \* P[t] - mu\_h \* P[t]) \* dt  
 [t=1:T], R\_h[t+1] == R\_h[t] + ((1 - rho) \* u3[t] \* I\_h[t] + (1 - phi) \* gamma\_h \* (I\_h[t] + I\_hA[t]) - mu\_h \* R\_h[t]) \* dt  
 [t=1:T], S\_v[t+1] == S\_v[t] + (mu\_v \* N\_v[t] \* (1 - u4[t]) - lambda\_v[t] \* S\_v[t] - mu\_v \* S\_v[t] - r\_0 \* u4[t] \* S\_v[t]) \* dt  
 [t=1:T], I\_v[t+1] == I\_v[t] + (lambda\_v[t] \* S\_v[t] - mu\_v \* I\_v[t] - r\_0 \* u4[t] \* I\_v[t]) \* dt  
 end);  
  
  
 # Objective function  
 @objective(model, Min, sum(  
 C\_1 \* I\_h[t] + C\_2 \* I\_hA[t] + C\_3 \* (S\_v[t] + I\_v[t]) + (D\_1 \* u1[t]^2)/2 + (D\_2 \* u2[t]^2)/2 + (D\_3 \* u3[t]^2)/2 + (D\_4 \* u4[t]^2)/2  
 for t in 1:T  
 ));  
  
 if silent  
 set\_silent(model)  
 end  
  
 # Run optimization  
 optimize!(model)  
 println("Termination status: $(termination\_status(model))")  
  
 return value.(u1), value.(u2), value.(u3), value.(u4), value.(S\_h), value.(I\_h), value.(I\_hA), value.(P), value.(R\_h), value.(S\_v), value.(I\_v)  
end;

## Running the model without interventions

Parameters

beta\_1 = 0.75; #Transmission prob from I\_v to S\_h  
beta\_2 = 0.375; #Transmission prob from I\_h to S\_v  
beta\_3 = 0.75; #Transmission prb from I\_v to P  
b = 0.5; #Avg bitting rate per mosquito per person  
rho = 0.01; #Proportion of treated individuals with partial immunity  
psi = 0.4; #Proportion of incidence rate from S\_h to I\_h  
gamma\_h = 0.3288330; #Disease related death rate of humans  
omega = 0.54; #Proportion of incidence rate from P to I\_h  
mu\_h = 0.0045; #Natural mortality rate and recruitment rate of humans  
mu\_v = 0.0323; #Natural mortality rate and recruitment rate of vector  
phi = 0.48; #Proportion of natural Recovery  
r\_0 = 0.005; #Enhance death rate  
  
u\_1 = 0.0; #Treated bednet  
u\_2 = 0.0; #Vaccination  
u\_3 = 0.0; #Treatment (prophylactics)  
u\_4 = 0.0; #Insecticides

Time and initial conditions

t0 = 0.0;  
tf = 300.0;  
dt = 0.5;  
ts = collect(0:dt:tf);  
  
S\_h0 = 10000;  
I\_h0 = 100;  
I\_hA0 = 500;  
P0 = 100;  
R\_h0 = 1000;  
S\_v0 = 6000;  
I\_v0 = 600;  
  
δ = 0.001;  
  
alg = Tsit5();  
  
u0 = [S\_h0, I\_h0, I\_hA0, P0, R\_h0, S\_v0, I\_v0];  
params = [beta\_1, beta\_2, beta\_3, rho, psi, gamma\_h, omega, mu\_h, mu\_v, phi, r\_0, u\_1, u\_2, u\_3, u\_4];

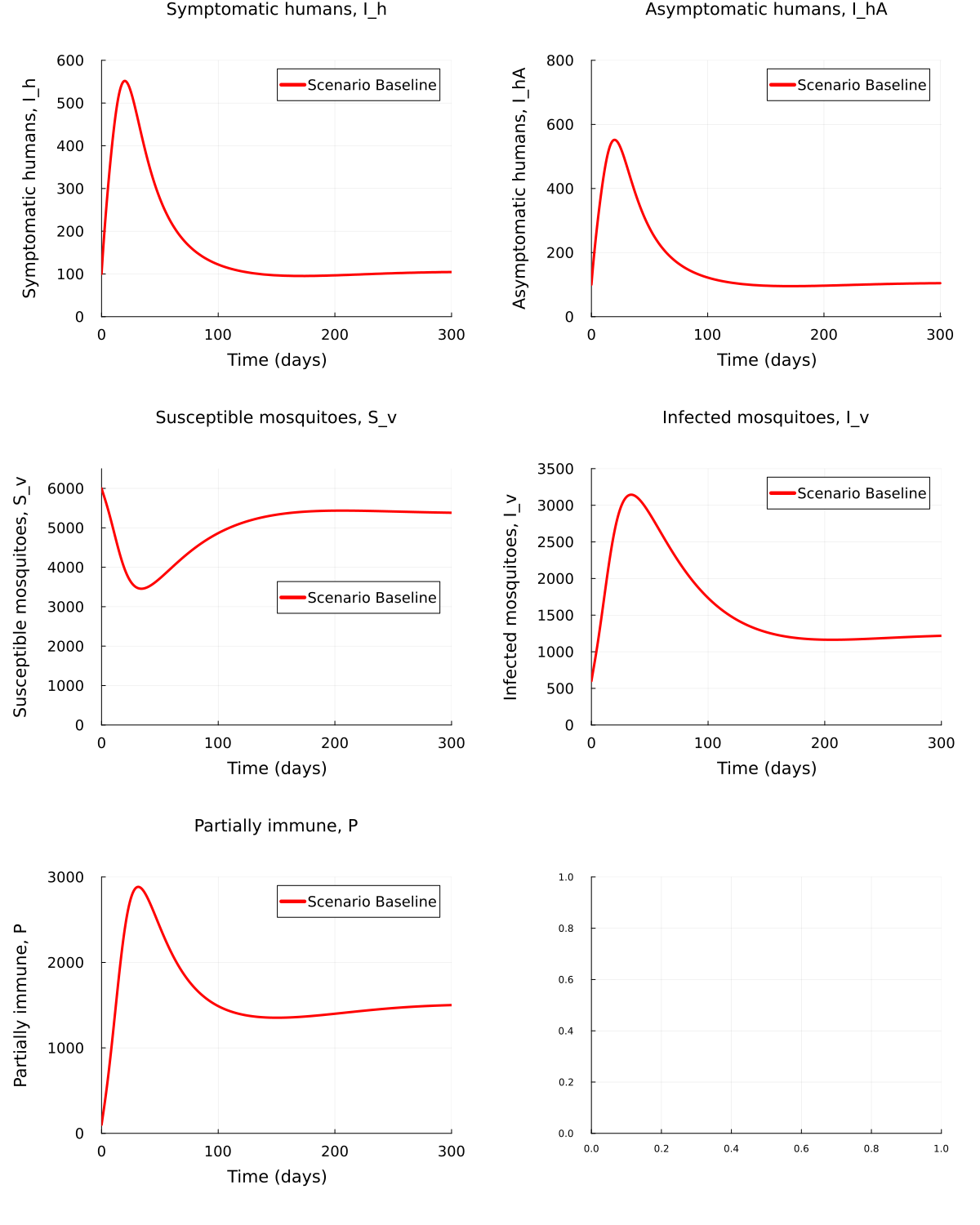
prob\_base = ODEProblem(dengue\_ode!, u0, (t0, tf), params)  
sol\_base = solve(prob\_base, alg, saveat=ts);

Extract data:

t\_base = sol\_base.t;  
S\_h\_baseline = sol\_base[1, :];  
I\_h\_baseline = sol\_base[2, :];  
I\_hA\_baseline = sol\_base[3, :];  
P\_baseline = sol\_base[4, :];  
R\_h\_baseline = sol\_base[5, :];  
S\_v\_baseline = sol\_base[6, :];  
I\_v\_baseline = sol\_base[7, :];

Plot baseline

Scenario\_test = create\_combined\_plot(t\_base,   
[],   
[S\_h\_baseline],   
[I\_h\_baseline],  
[I\_h\_baseline],   
[P\_baseline],   
[R\_h\_baseline],   
[S\_v\_baseline],   
[I\_v\_baseline],  
["Baseline"], save=false)



# Strategies

# Weights for the objective function  
C\_1 = 5;   
C\_2 = 5;   
C\_3 = 5;   
D\_1= 16.62;   
D\_2 = 2.5;   
D\_3 = 5;   
D\_4 = 16.62;  
  
silent = true;

### Strategy A

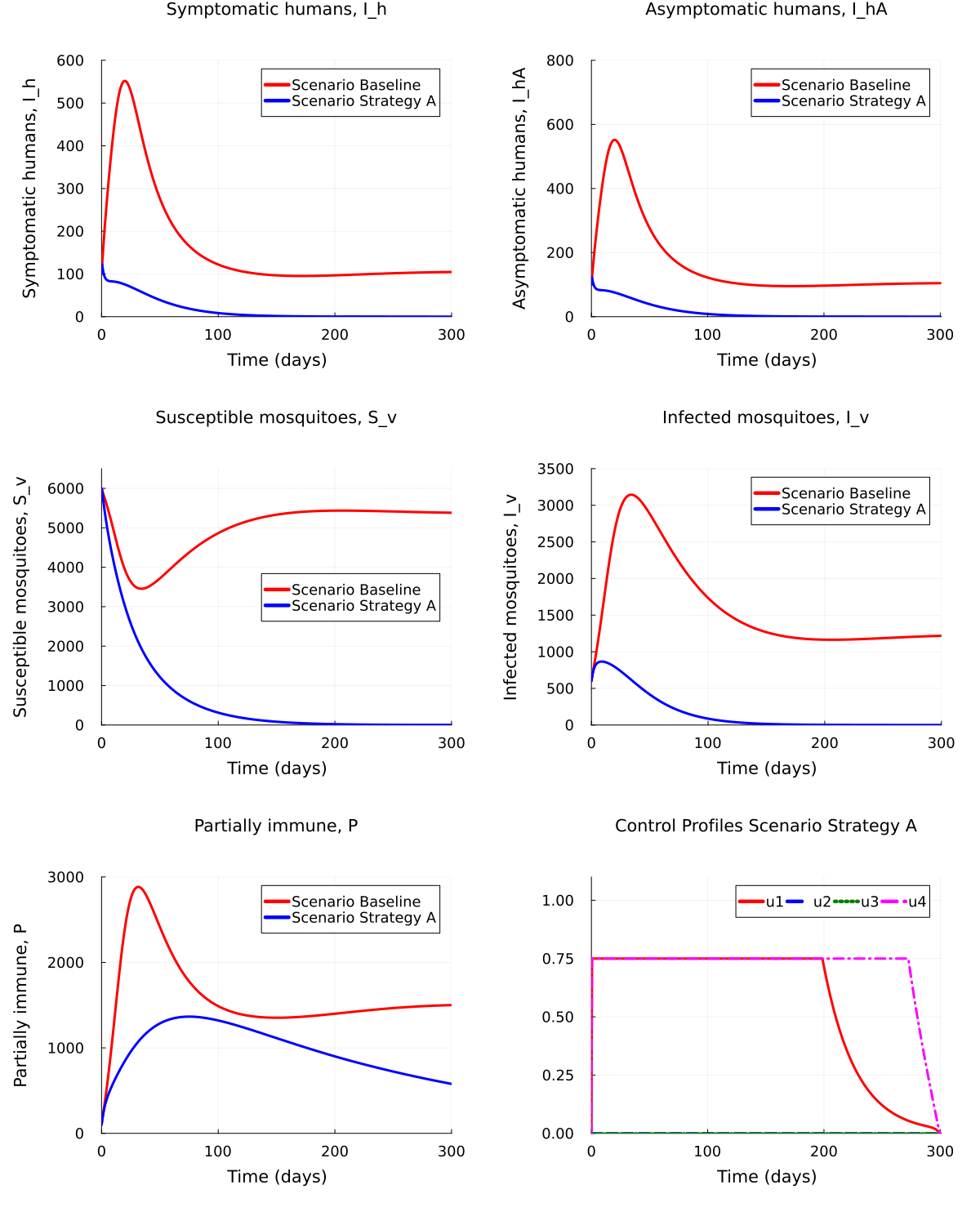
This strategy combines the use of treated bednets and insectisides

outputA\_values = open("OptDengue\_StratA\_log.txt", "w") do file  
 redirect\_stdout(file) do  
 return run\_optimization(0.75, 0, 0, 0.75)  
 end  
end  
  
# Now extract values from output\_values  
u1\_opt\_A, u2\_opt\_A, u3\_opt\_A, u4\_opt\_A, S\_h\_opt\_A, I\_h\_opt\_A, I\_hA\_opt\_A, P\_opt\_A, R\_h\_opt\_A, S\_v\_opt\_A, I\_v\_opt\_A = outputA\_values;

┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185

Plot optimal results obtained from Strategy A vs Baseline

Scenario\_A = create\_combined\_plot(t\_base,   
 [u1\_opt\_A, u2\_opt\_A, u3\_opt\_A, u4\_opt\_A],   
 [S\_h\_baseline, S\_h\_opt\_A],   
 [I\_h\_baseline, I\_h\_opt\_A],  
 [I\_h\_baseline, I\_h\_opt\_A],   
 [P\_baseline, P\_opt\_A],   
 [R\_h\_baseline, R\_h\_opt\_A],   
 [S\_v\_baseline, S\_v\_opt\_A],   
 [I\_v\_baseline, I\_v\_opt\_A],  
 ["Baseline", "Strategy A"], save=false)



### Strategy B

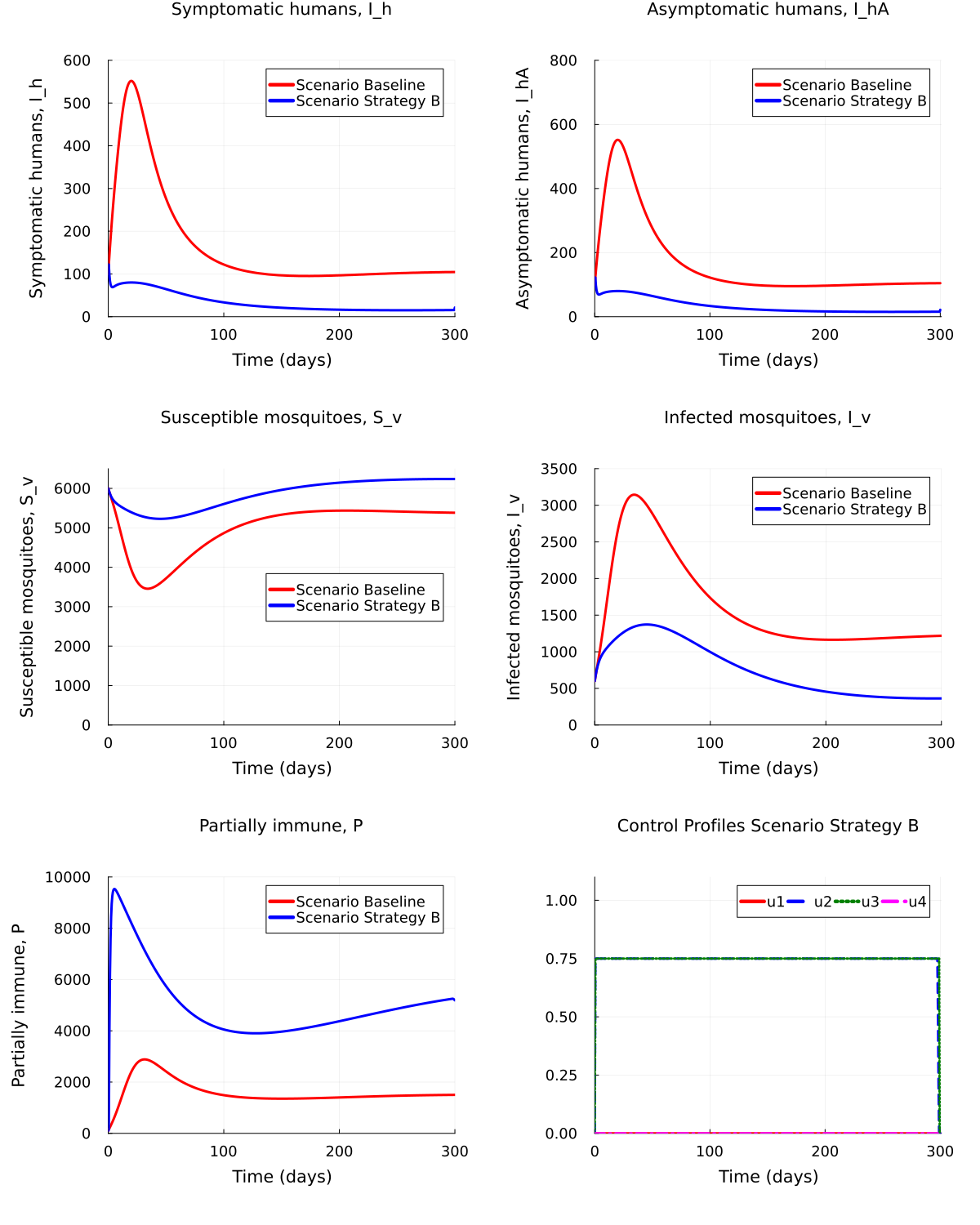
This strategy combines the use of treatment (prophylactics) and vaccination

outputB\_values = open("OptDengue\_StratB\_log.txt", "w") do file  
 redirect\_stdout(file) do  
 return run\_optimization(0, 0.75, 0.75, 0)  
 end  
end  
  
# Now extract values from output\_values  
u1\_opt\_B, u2\_opt\_B, u3\_opt\_B, u4\_opt\_B, S\_h\_opt\_B, I\_h\_opt\_B, I\_hA\_opt\_B, P\_opt\_B, R\_h\_opt\_B, S\_v\_opt\_B, I\_v\_opt\_B = outputB\_values;

┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185

Plot optimal results obtained from Strategy B vs Baseline

Scenario\_B = create\_combined\_plot(t\_base,   
 [u1\_opt\_B, u2\_opt\_B, u3\_opt\_B, u4\_opt\_B],   
 [S\_h\_baseline, S\_h\_opt\_B],   
 [I\_h\_baseline, I\_h\_opt\_B],  
 [I\_h\_baseline, I\_h\_opt\_B],   
 [P\_baseline, P\_opt\_B],   
 [R\_h\_baseline, R\_h\_opt\_B],   
 [S\_v\_baseline, S\_v\_opt\_B],   
 [I\_v\_baseline, I\_v\_opt\_B],  
 ["Baseline", "Strategy B"], P\_axis=10000, save=false)



### Strategy C

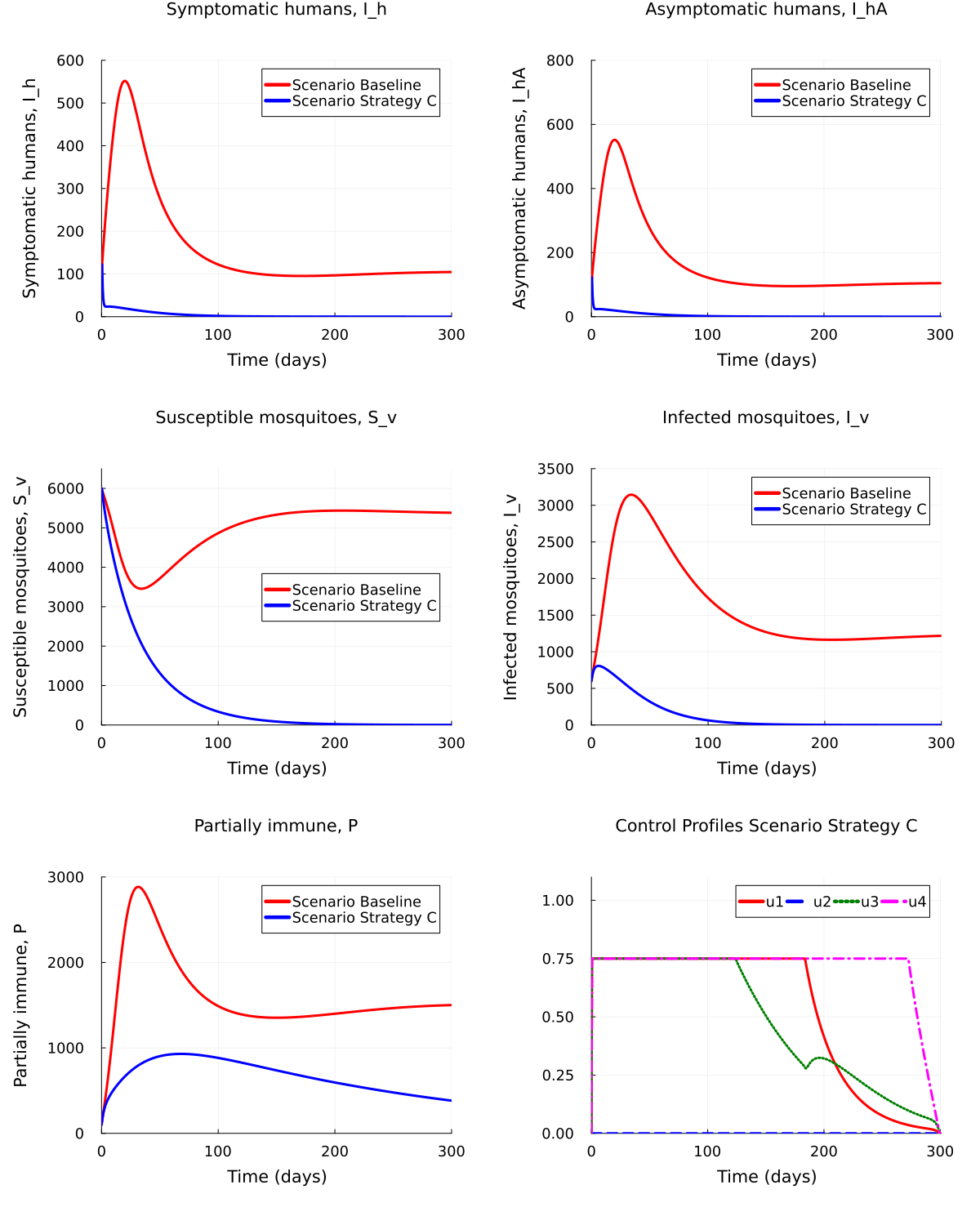
This strategy combines the use of treated bednets, treatment (prophylactics) and insectisides

outputC\_values = open("OptDengue\_StratC\_log.txt", "w") do file  
 redirect\_stdout(file) do  
 return run\_optimization(0.75, 0, 0.75, 0.75)  
 end  
end  
  
# Now extract values from output\_values  
u1\_opt\_C, u2\_opt\_C, u3\_opt\_C, u4\_opt\_C, S\_h\_opt\_C, I\_h\_opt\_C, I\_hA\_opt\_C, P\_opt\_C, R\_h\_opt\_C, S\_v\_opt\_C, I\_v\_opt\_C = outputC\_values;

┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185

Plot optimal results obtained from Strategy C vs Baseline

Scenario\_C = create\_combined\_plot(t\_base,   
 [u1\_opt\_C, u2\_opt\_C, u3\_opt\_C, u4\_opt\_C],   
 [S\_h\_baseline, S\_h\_opt\_C],   
 [I\_h\_baseline, I\_h\_opt\_C],  
 [I\_h\_baseline, I\_h\_opt\_C],   
 [P\_baseline, P\_opt\_C],   
 [R\_h\_baseline, R\_h\_opt\_C],   
 [S\_v\_baseline, S\_v\_opt\_C],   
 [I\_v\_baseline, I\_v\_opt\_C],  
 ["Baseline", "Strategy C"], save=false)



### Strategy D

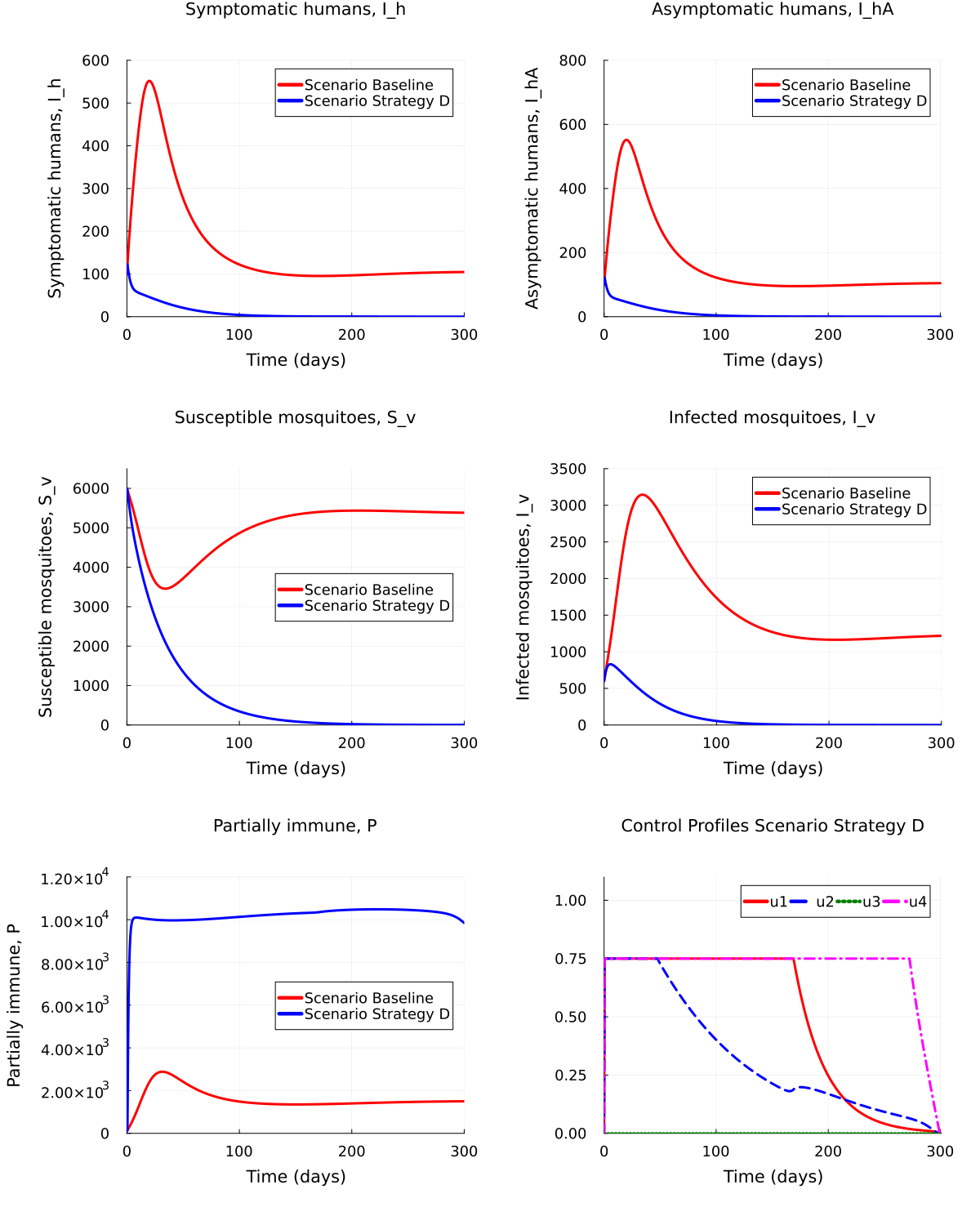
This strategy combines the use of treated bednets, insectisides and vaccination

outputD\_values = open("OptDengue\_StratD\_log.txt", "w") do file  
 redirect\_stdout(file) do  
 return run\_optimization(0.75, 0.75, 0, 0.75)  
 end  
end  
  
# Now extract values from output\_values  
u1\_opt\_D, u2\_opt\_D, u3\_opt\_D, u4\_opt\_D, S\_h\_opt\_D, I\_h\_opt\_D, I\_hA\_opt\_D, P\_opt\_D, R\_h\_opt\_D, S\_v\_opt\_D, I\_v\_opt\_D = outputD\_values;

┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185

Plot optimal results obtained from Strategy D vs Baseline

Scenario\_D = create\_combined\_plot(t\_base,   
 [u1\_opt\_D, u2\_opt\_D, u3\_opt\_D, u4\_opt\_D],   
 [S\_h\_baseline, S\_h\_opt\_D],   
 [I\_h\_baseline, I\_h\_opt\_D],  
 [I\_h\_baseline, I\_h\_opt\_D],   
 [P\_baseline, P\_opt\_D],   
 [R\_h\_baseline, R\_h\_opt\_D],   
 [S\_v\_baseline, S\_v\_opt\_D],   
 [I\_v\_baseline, I\_v\_opt\_D],  
 ["Baseline", "Strategy D"], P\_axis=12000, P\_legend=:right, save=false)



### Strategy E

This strategy combines the use of the four control interventions treated bednets, insectisides, treatment and vaccination

outputE\_values = open("OptDengue\_StratE\_log.txt", "w") do file  
 redirect\_stdout(file) do  
 return run\_optimization(0.75, 0.75, 0.75, 0.75)  
 end  
end  
  
# Now extract values from output\_values  
u1\_opt\_E, u2\_opt\_E, u3\_opt\_E, u4\_opt\_E, S\_h\_opt\_E, I\_h\_opt\_E, I\_hA\_opt\_E, P\_opt\_E, R\_h\_opt\_E, S\_v\_opt\_E, I\_v\_opt\_E = outputE\_values;

┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185  
┌ Warning: Axis contains one element: 601. If intended, you can safely ignore this warning. To explicitly pass the axis with one element, pass `[601]` instead of `601`.  
└ @ JuMP.Containers ~/.julia/packages/JuMP/xlp0s/src/Containers/DenseAxisArray.jl:185

Plot optimal results obtained from Strategy E vs Baseline

Scenario\_E = create\_combined\_plot(t\_base,   
 [u1\_opt\_E, u2\_opt\_E, u3\_opt\_E, u4\_opt\_E],   
 [S\_h\_baseline, S\_h\_opt\_E],   
 [I\_h\_baseline, I\_h\_opt\_E],  
 [I\_h\_baseline, I\_h\_opt\_E],   
 [P\_baseline, P\_opt\_E],   
 [R\_h\_baseline, R\_h\_opt\_E],   
 [S\_v\_baseline, S\_v\_opt\_E],   
 [I\_v\_baseline, I\_v\_opt\_E],  
 ["Baseline", "Strategy E"], P\_axis=12000, P\_legend=:right, save=false)

