Lab 7: Parking Garage Case Study

Jiayue Yin(jy152)

Thu., Mar. 28

using Revise  
using ParkingGarage  
using Distributions

and also regular packages

using Plots  
Plots.default(; margin=5Plots.mm)

## Formal problem framing

let  
 sow = ParkingGarageSOW()  
 years = 1:(sow.n\_years)  
 demand = [  
 ParkingGarage.calculate\_demand(year, sow.demand\_growth\_rate) for year in years  
 ]  
 plot(  
 years,  
 demand;  
 ylabel="Demand [cars/day]",  
 xlabel="Year",  
 legend=false,  
 title="Demand Growth Rate: $(sow.demand\_growth\_rate) Cars/Year",  
 size=(800, 400),  
 marker=:circle,  
 )  
end

## Static Policy

let  
 sow = ParkingGarageSOW(; demand\_growth\_rate=80.0, n\_years=20, discount\_rate=0.12)  
 n\_levels = 2:12  
 policies = [StaticPolicy(i) for i in n\_levels]  
 profits = [simulate(sow, policy) for policy in policies]  
 plot(  
 n\_levels,  
 profits;  
 ylabel="NPV Profits [Million USD]",  
 xlabel="Number of levels",  
 legend=false,  
 title="$(sow.n\_years) Year Horizon, $(sow.discount\_rate) Discount, $(sow.demand\_growth\_rate) Demand Growth",  
 size=(800, 400),  
 marker=:circle,  
 xticks=n\_levels,  
 )  
 hline!([0])  
end

## Uncertainty

I take the same n\_years = 20 and the same discount rate, but use the Normal fuction to create different demanad growth rate and use its mean value.

rates = Normal(80,30)  
demand\_growth\_rates = [rand(rates) for \_ in 1:1000]  
let  
 sows = [ParkingGarageSOW(; demand\_growth\_rate=rate, n\_years=20, discount\_rate=0.12) for rate in demand\_growth\_rates]  
 n\_levels = 2:12  
 profits = [  
 mean([simulate(sow, StaticPolicy(i)) for sow in sows]) for i in n\_levels  
 ]  
 plot(  
 n\_levels,  
 profits;  
 ylabel="Average NPV Profits [Million USD]",  
 xlabel="Number of levels",  
 legend=false,  
 title="20 Year Horizon, 0.12 Discount, Variable Demand Growth",  
 size=(800, 400),  
 marker=:circle,  
 xticks=n\_levels,  
 )  
 hline!([0])  
end

By compareing those two figures, we can find the uncertainty has a great impact.

## Adapative Comparisons

# single SOW and mutiple SOWs

sow = ParkingGarageSOW(; demand\_growth\_rate=80.0, n\_years=20, discount\_rate=0.12)  
n\_levels = 2:12  
policies = [StaticPolicy(i) for i in n\_levels]  
profits = [simulate(sow, policy) for policy in policies]  
adaptive\_policies = [AdaptivePolicy(i) for i in n\_levels]  
adaptive\_profits = [simulate(sow, policy) for policy in adaptive\_policies]  
  
plot(  
 n\_levels,  
 adaptive\_profits;  
 ylabel="NPV Profits [Million USD]",  
 xlabel="Number of levels",  
 marker=:circle,  
 xticks=n\_levels,  
 label="Adaptive Policy single SOW"  
)  
hline!([0], label="0 NPV")  
  
rates = Normal(80,30)  
demand\_growth\_rates = [rand(rates) for \_ in 1:1000]  
  
sows = [ParkingGarageSOW(; demand\_growth\_rate=rate, n\_years=20, discount\_rate=0.12) for rate in demand\_growth\_rates]  
n\_levels = 2:12  
adaptive\_results = []  
  
for policy in adaptive\_policies  
 # take the mean of all SOWs for the current level (policy) and add it to the results  
 result\_allsow = [simulate(sow, policy) for sow in sows]  
 push!(adaptive\_results, mean(result\_allsow))  
end  
  
plot!(  
 n\_levels,  
 adaptive\_results;  
 ylabel="NPV Profits [Million USD]",  
 xlabel="Number of levels",  
 marker=:circle,  
 xticks=n\_levels,  
 label="Adaptive Policy SOWs"  
)

In those figures, we can see that mulitple SOWs is slightly higher than the single SOW. And Introducing uncertainty and adopting adaptive policy can more truly reflect the real world situation. , and potentially improve profits.