

Assignment 2

A).

```
using Evolutionary
using CSV
using DataFrames

#Set Filename and Read CSV into DataFrame
filename="knapPI_16_50_1000.csv"
data=CSV.read(filename,DataFrame,skipto=6,header=false)

#Also read in constraint value
con=open(filename) do file
    fileText= readlines(file)
    con=fileText[3][3:end]
    return parse{Int,con}
end

#Create variables, knapsack options (n), weights (w), and values (v)
n=size(data)[1]
w=data[:,3]
v=data[:,2]

#Create objective function, which defaults to 0 if it violates the constraint,
#and the sum is set to negative as this EA uses minimization
f(x)=((sum(w.*x)<=con) ? -sum(v.*x) : 0)

#Create bitvector to represent the selected items; provide at least 1 zero, so the ob
#starting out; otherwise it just prematurely converges to 0.
x0=BitVector(vcat(zeros(n-1),1))

#Defined uniform bit flipping function, provided a mutation rate; can't currently ext
#the function definitions below, as this is not in the original package
function uflip(recombinant::T) where {T <: BitVector}
    p = 0.2
    s = length(recombinant)
    check = rand(s).<p
    for i in 1:length(check)
```

```

        if check[i]==1
            recombinant[i]=!recombinant[i]
        end
    end
    return recombinant
end

#Define different combinations of genetic algorithms with different types of mutation
#uniform and singlepoint.
gafu = GA(populationSize=1000,selection=roulette,mutation=flip,crossover=uniform,mutat
gafs = GA(populationSize=1000,selection=roulette,mutation=flip,crossover=singlepoint,m
gaus = GA(populationSize=1000,selection=roulette,mutation=uflip,crossover=singlepoint,m
gauu = GA(populationSize=1000,selection=roulette,mutation=uflip,crossover=uniform,muta

#Run the optimization, suppress output.
rfu = Evolutionary.optimize(f, x0, gafu, Evolutionary.Options(iterations=10000));
rfs = Evolutionary.optimize(f, x0, gafs, Evolutionary.Options(iterations=10000));
rus = Evolutionary.optimize(f, x0, gaus, Evolutionary.Options(iterations=10000));
ruu = Evolutionary.optimize(f, x0, gauu, Evolutionary.Options(iterations=10000));

#Print minima, total iterations, and function calls
println("GA:RLT:INV:SP (-objfun) => M: $(minimum(rfu)), I: $(Evolutionary.iterations(r
println("GA:RLT:INV:SP (-objfun) => M: $(minimum(rfs)), I: $(Evolutionary.iterations(r
println("GA:RLT:INV:SP (-objfun) => M: $(minimum(rus)), I: $(Evolutionary.iterations(r
println("GA:RLT:INV:SP (-objfun) => M: $(minimum(ruu)), I: $(Evolutionary.iterations(r

GA:RLT:INV:SP (-objfun) => M: -1994, I: 21 F: 22003
GA:RLT:INV:SP (-objfun) => M: -2135, I: 32 F: 33006
GA:RLT:INV:SP (-objfun) => M: -2410, I: 21 F: 22003
GA:RLT:INV:SP (-objfun) => M: -1061, I: 12 F: 13001

```

B).

C).

D).