

# Homework 5

## Population Based-Heuristics

Daniel Carpenter

April 2022

### Contents

Global Variables	2
Question 1: Simulated Annealing	3
Question 2: Genetic Algorithm	4
Summary Output of each Model	5

*Please see the final page for the summary output.*

## Global Variables

Input variables like the *random seed*, *values* and *weights* data for knapsack, and the *maximum allowable weight*

Please assume these are referenced by following code chunks

```
# Import python libraries
import copy
import math
from random import Random
import numpy as np

# to setup a random number generator, we will specify a "seed" value
seed = 5113
myPRNG = Random(seed)

# to setup a random number generator, we will specify a "seed" value
# need this for the random number generation -- do not change
seed = 51132021
myPRNG = Random(seed)

# number of elements in a solution
n = 150

# create an "instance" for the knapsack problem
value = []
for i in range(0, n):
    # value.append(round(myPRNG.expovariate(1/500)+1,1))
    value.append(round(myPRNG.triangular(150, 2000, 500), 1))

weights = []
for i in range(0, n):
    weights.append(round(myPRNG.triangular(8, 300, 95), 1))

# define max weight for the knapsack
maxWeight = 2500
```

## Question 1: Simulated Annealing

## Question 2: Genetic Algorithm

## Summary Output of each Model

```
# Get the list of questions to send to R
# output = [q2, q3, q4_1, q4_2, q5_1, q5_2]

# library(reticulate) # Package to convert Python to R / R to Python
#
# # >> Convert Python list to R list object <<
# pyList <- py$output
#
# # Create empty data frame to append results to
# df <- data.frame()
#
# for (i in 1:length(pyList)) {
#   newRow <- as.data.frame(pyList[i][1])
#   colnames(newRow) <- c('Iterations', '# Items Selected', 'Weight', 'Objective')
#
#   # Append the new row
#   df <- rbind(df, newRow)
# }
#
# rownames(df) <- c('Local Search with Best Improvement',
#                   'Local Search with First Improvement',
#                   'Local Search with Random Restarts (k=10)',
#                   'Local Search with Random Restarts (k=50)',
#                   'Local Search with Random Walk, using First Acceptance (p=0.75)',
#                   'Local Search with Random Walk, using First Acceptance (p=0.01)'
#                   )

# knitr::kable(df)
```