HOMEWORK 2 OF MODERN OPTIMIZATION METHODS

R09946006 | 何青儒 | HO, Ching-Ru | 09/29/2020

1. Optimization of a Function using Simulated Annealing.

- (a) Find the minimum of the function using simulated annealing. Assume suitable parameters and show detailed calculations for 2 iterations.
  - Objective function is $f(X) = 6x_1^2 - 6x_1x_2 + 2x_2^2 - x_1 - 2x_2$.
  - (0) Initialized ($i = 0$)
    - Randomly selected four points in the design space.
    - Let $X^{(i)} = [x_1^{(i)}, x_2^{(i)}]$, $X^{(1)} = [2, 0]$, $X^{(2)} = [5, 10]$, $X^{(3)} = [8, 5]$, $X^{(4)} = [1, 1]$, where i in here denotes sampling.
    - Get objective function $f(X^{(1)}) = 22$, $f(X^{(2)}) = 25$, $f(X^{(3)}) = 176$, $f(X^{(4)}) = -1$.
    - We assume the initial temperature to be $T_0 = \bar{f} = 55.5$, and temperature reduction factor is chosen as $c = 0.5$.
  - (1) 1st Iteration ($i = 1$)
    - Select the initial design points as $X_1 = [4, 5]$, and get $f(X_1) = 12$.
    - Select two uniformly distributed RV $u_1 = 0.31$ and $u_2 = 0.57$ for the vicinity of 4 and 5 respectively. The range of ± 6 imply the available ranges of $x_1 \in [-2, 10]$ and $x_2 \in [-1, 11]$.
    - The uniformly distributed RV r_1 and r_2, in the range of x_1 and x_2, and corresponding to u_1 and u_2, can be focus as:
      - $r_1 = -2 + 0.31 \cdot (10 - (-2)) = 1.72$
      - $r_2 = -1 + 0.57 \cdot (11 - (-1)) = 5.84$
    - The objective function value of $X_2 = [1.72, 5.84]$ and $f(X_2) \simeq 12.2928$
    - Get $f(X_2) > f(X_1)$. Since $\Delta f = f(X_2) - f(X_1) = 12.2928 - 12 = 0.2928 \geq 0$, we use Metropolis criterion to decide whether to accept or reject the current point.
    - Randomly choose a RV in the range of $(0, 1)$ as $r = 0.83$. Assume Boltzmann constant (k) is 1 for simplicity, and we have $f(X_2) \simeq 12.2928$ and $\Delta f = 0.2928$ for X_2, so we get $P(X_2)$.
    - $P(X_2) = e^{-\frac{\Delta f}{kT}} = e^{\frac{-0.2928}{1 \cdot 12.2928}} \simeq 0.97 > r = 0.83$, the possibility is greater than threshold r. We accept $X_2 = [1.72, 5.84]$ as the next design point.
  - (2) 2nd Iteration ($i = 2$)
    - We have $X_2 = [1.72, 5.84]$ from step (1).
    - Select two uniformly distributed RV $u_1 = 0.52$ and $u_2 = 0.13$ for the vicinity of 1.72 and 5.84 respectively. The range of ± 6 imply the available ranges of $x_1 \in [-4.28, 7.72]$ and $x_2 \in [-0.16, 11.84]$.
    - The uniformly distributed RV r_1 and r_2, in the range of x_1 and x_2, and corresponding to u_1 and u_2, can be focus as:
      - $r_1 = -4.28 + 0.52 \cdot (7.72 - (-4.28)) = 1.96$
      - $r_2 = -0.16 + 0.13 \cdot (11.84 - (-0.16)) = 1.40$

    - The objective function value of $X_3 = [1.96, 1.40]$ and $f(X_3) \simeq 5.7456$
    - Get $f(X_3) < f(X_2), \Delta f = f(X_3) - f(X_2) < 0$, solution X_3 is better than X_2 trivially, so we accept the current point.
    - Update the iteration number as $i = 3$, however, $i > n$, it reaches the termination criterion. Update the number of cycle as $p = 2$.
    - Reduce the temperature to a new value of $T = cT_0 = c\bar{f} = 0.5 \times 55.5 = 27.75$.
    - Reset the current iteration number as $i = 1$. back to (2).
- (b) Run a random walk to roughly scan the landscape of $f(X)$.
    - Assume run 5 times, $X(i) = [x_1, x_2]$
    - Minimum candidate solutions in Random Walk is $X = [1, 0]$, and the objective function value is $f(X) = 5$.

```
>>X( 0 )= [-3, 1] , and f(X)= 75
>>X( 1 )= [1, 0] , and f(X)= 5
>>X( 2 )= [0, -3] , and f(X)= 24
>>X( 3 )= [-1, 4] , and f(X)= 55
>>X( 4 )= [0, 3] , and f(X)= 12
```

- (c) See `hw2ass1.py` files.
- (d) Set parameters:
    - initial solution from Random Walk results: $X = [1, 0]$
    - initial temperature from Random Walk results: $f(\bar{X}(i)) = \frac{1}{5}(75 + 5 + 24 + 55 + 12) = 34.2$
    - temperature reduction factor$(= c) = 0.9$
    - Boltzmann constant$(= k) = 1$
    - Find the best solution $X = [0.59506437, 1]$, and the best objective function $f(X) = -2.0408409632983524$.

```
Cycle: 0 with Temperature: 34.2
Cycle: 1 with Temperature: 30.780000000000005
Cycle: 2 with Temperature: 27.702000000000005
Cycle: 3 with Temperature: 24.931800000000006
Cycle: 4 with Temperature: 22.438620000000007
Cycle: 5 with Temperature: 20.194758000000007
Cycle: 6 with Temperature: 18.17528220000001
Cycle: 7 with Temperature: 16.35775398000001
Cycle: 8 with Temperature: 14.721978582000009
Cycle: 9 with Temperature: 13.249780723800008
Cycle: 10 with Temperature: 11.924802651420007
Cycle: 11 with Temperature: 10.732322386278007
Cycle: 12 with Temperature: 9.659090147650206
Cycle: 13 with Temperature: 8.693181132885185
Cycle: 14 with Temperature: 7.823863019596667
Cycle: 15 with Temperature: 7.041476717637
Cycle: 16 with Temperature: 6.3373290458733
Cycle: 17 with Temperature: 5.703596141285971
Cycle: 18 with Temperature: 5.133236527157374
Cycle: 19 with Temperature: 4.619912874441637
Cycle: 20 with Temperature: 4.157921586997473
Cycle: 21 with Temperature: 3.7421294282977255
Cycle: 22 with Temperature: 3.367916485467953
Cycle: 23 with Temperature: 3.031124836921158
Cycle: 24 with Temperature: 2.728012353229042
Cycle: 25 with Temperature: 2.455211117906138
```

```
Cycle: 26 with Temperature: 2.2096900061155242
Cycle: 27 with Temperature: 1.988721005503972
Cycle: 28 with Temperature: 1.7898489049535748
Cycle: 29 with Temperature: 1.6108640144582174
Cycle: 30 with Temperature: 1.4497776130123956
Cycle: 31 with Temperature: 1.304799851711156
Cycle: 32 with Temperature: 1.1743198665400405
Cycle: 33 with Temperature: 1.0568878798860364
Cycle: 34 with Temperature: 0.9511990918974328
Cycle: 35 with Temperature: 0.8560791827076896
Cycle: 36 with Temperature: 0.7704712644369206
Cycle: 37 with Temperature: 0.6934241379932286
Cycle: 38 with Temperature: 0.6240817241939057
Cycle: 39 with Temperature: 0.5616735517745152
Cycle: 40 with Temperature: 0.5055061965970636
Cycle: 41 with Temperature: 0.4549555769373573
Cycle: 42 with Temperature: 0.4094600192436216
Cycle: 43 with Temperature: 0.3685140173192594
Cycle: 44 with Temperature: 0.3316626155873335
Cycle: 45 with Temperature: 0.29849635402860014
Cycle: 46 with Temperature: 0.26864671862574013
Cycle: 47 with Temperature: 0.24178204676316611
Cycle: 48 with Temperature: 0.21760384208684952
Cycle: 49 with Temperature: 0.19584345787816457
Best Solution X: [0.59506437 1.        ]
Best Objective Function: -2.0408409632983524
```

2. Optimization of Travel Routes for South Korea Cities

- (a) Create the distance of location table.

```
# create the distance(km) of location table.
distance=[
    [0, 27, 335, 244, 141, 257, 33, 316, 186, 115, 304, 439, 102, 95, 275],
    [27, 0, 330, 237, 144, 268, 31, 307, 195, 113, 301, 453, 75, 111, 290],
    [335, 330, 0, 95, 199, 193, 304, 54, 189, 221, 35, 291, 330, 271, 233],
    [244, 237, 95, 0, 117, 171, 212, 75, 130, 130, 72, 324, 236, 191, 215],
    [141, 144, 199, 117, 0, 137, 114, 192, 61, 36, 167, 323, 175, 74, 171],
    [257, 268, 193, 171, 137, 0, 238, 222, 77, 173, 161, 186, 311, 162, 44],
    [33, 31, 304, 212, 114, 238, 0, 284, 164, 84, 274, 423, 91, 83, 260],
    [316, 307, 54, 75, 192, 222, 284, 0, 198, 205, 67, 341, 296, 266, 265],
    [186, 195, 189, 130, 61, 77, 164, 198, 0, 96, 154, 263, 234, 97, 111],
    [115, 113, 221, 130, 36, 173, 84, 205, 96, 0, 190, 359, 139, 74, 205],
    [304, 301, 35, 72, 167, 161, 274, 67, 154, 190, 0, 275, 306, 237, 202],
    [439, 453, 291, 324, 323, 186, 423, 341, 263, 359, 275, 0, 498, 344, 165],
    [102, 75, 330, 236, 175, 311, 91, 296, 234, 139, 306, 498, 0, 170, 340],
    [95, 111, 271, 191, 74, 162, 83, 266, 97, 74, 237, 344, 170, 0, 180],
    [275, 290, 233, 215, 171, 44, 260, 265, 111, 205, 202, 165, 340, 180, 0],
]
# totally 15 cities, where the array indices correspond to the locations in the following
order:
# (0)Incheon - (1)Seoul - (2)Busan - (3)Daegu - (4)Daejeon - (5)Gwangju - (6)Suwon-si -
(7)Ulsan - (8)Jeonju - (9)Cheongju-si - (10)Changwon - (11)Jeju-si - (12)Chuncheon -
(13)Hongsung - (14)Muan
```

- (b) It will need to take $15!$ times to try all possible in exhaustive enumeration algorithm.
- (c) Use random walk after 100 iterations, get the optimal path 2235(km) and the order is $[3, 2, 10, 11, 6, 0, 9, 12, 13, 4, 14, 5, 1, 8, 7]$.

```
>> Random Walk Best Distance = 2235
>> Random Walk Best Order =  [3, 2, 10, 11, 6, 0, 9, 12, 13, 4, 14, 5, 1, 8, 7]
```

- (d) See `hw2ass2.py` files.
- (e) Run `$ python hw2ass2.py`
    - Thus, use Tabu Search Algorithm on this problem, we get:
        - iteration times: 20 times
            - Tabu Algorithm Best Distance: 1069 (km)
            - Tabu Algorithm Best Order: $[7, 2, 10, 3, 4, 9, 6, 12, 1, 0, 13, 8, 5, 14, 11]$
            - Final Tabu List:
              $[[10, 9], [9, 6], [9, 4], [10, 5], [10, 9], [10, 6], [5, 4], [3, 2], [3, 0], [2, 0]]$

```
>> (...omitted)
>> 20  times:
>> TABU LIST:  [[10, 9], [9, 6], [9, 4], [10, 5], [10, 9], [10, 6], [5, 4], [3, 2], [3, 0],
[2, 0]]
>> 當前最短解:  [7, 2, 10, 3, 4, 9, 6, 12, 1, 0, 13, 8, 5, 14, 11]
>> 當前最短: 1069
>> 檢查鄰居數:  105
>> 所有鄰居最佳交換: [2, 0]
>> 所有鄰居最佳解: []
>> 所有鄰居之中最佳解沒有更好，略過加入tabu的動作
>> Tabu Algo Best Distance: 1069
>> Tabu Algo Best Order: [7, 2, 10, 3, 4, 9, 6, 12, 1, 0, 13, 8, 5, 14, 11]
```

- (f)