HOMEWORK 2 FOR MODERN OPTIMIZATION METHODS FALL SEMESTER 2021

1. **Optimization of a Function via Particle Swarm Optimization.** *(30 points)* The goal of this problem is to find the maximum of the function $f = -x^5 + 5x^3 + 20x - 5$ in the range $-4 \leq x \leq 4$ using the PSO method. Here are some settings: Use 4 particles with the initial positions $x_1 = -2$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 3$. Assume $c_1 = c_2 = w = 1$ for simplicity.

   (a) Show detailed calculations for 2 iterations, which is similar to that in the lecture note.

$f(x) = -x^5 + 5x^3 + 20x - 5 \quad (-4 \le x \le 4)$

$x_1 = -2,\ x_2 = 0,\ x_3 = 1,\ x_4 = 3.$

$c_1 = c_2 = w = 1$

Number of particles $= N = 4$

(Step1) $x_{1,0} = -2,\ f(x_{1,0}) = -53$

($i=1$) $x_{2,0} = 0,\ f(x_{2,0}) = -5$

$x_{3,0} = 1,\ f(x_{3,0}) = 19$

$x_{4,0} = 3,\ f(x_{4,0}) = -53$

Set initial velocities of each particle to zero: $V_{1,0} = V_{2,0} = V_{3,0} = V_{4,0} = 0$.

(Step2)

($i=1$) Find global best position $\Rightarrow P_G = 1\ (f_G = 19)$

Using the random numbers, in the range $(0,1)$ as

$r_1 = 0.3294,\ r_2 = 0.9542$

$\Rightarrow V_{1,1} = 0 + 0.3294(-2+2) + 0.9542(1+2) = 2.8626$

$V_{2,1} = 0 + 0.3294(0-0) + 0.9542(1-0) = 0.9542$

$V_{3,1} = 0 + 0.3294(1-1) + 0.9542(1-1) = 0$

$V_{4,1} = 0 + 0.3294(3-3) + 0.9542(1-3) = -1.9084.$

(Step3) $\Rightarrow$ ($i=1$)

$x_{1,1} = -2 + 2.8626 = 0.8626 \rightarrow f(x_{1,1}) = 14.9836$

$x_{2,1} = 0 + 0.9542 = 0.9542 \rightarrow f(x_{2,1}) = 17.6369$

$x_{3,1} = 1 + 0 = 1 \rightarrow f(x_{3,1}) = 19$

$x_{4,1} = 3 + (-1.9084) = 1.0916 \rightarrow f(x_{4,1}) = 21.7858.$

Since the values of x_j did not converge, we raise the i to 2.

(Step2) $\Rightarrow$ Find the global best position: $P_G = 1.0916,\ (f_G = 21.7858)$

($i=2$) $r_1 = 0.1482,\ r_2 = 0.4867$

$\Rightarrow V_{1,2} = 2.8626 + 0.1482(0.8626 - 0.8626) + 0.4867(1.0916 - 0.8626) = 2.9741$

$V_{2,2} = 0.9542 + 0.1482(0.9542 - 0.9542) + 0.4867(1.0916 - 0.9542) = 1.0211$

$V_{3,2} = 1 + 0.1482(1-1) + 0.4867(1.0916 - 1) = 1.0446$

$V_{4,2} = -1.9084 + 0.1482(1.0916 - 1.0916) + 0.4867(1.0916 - 1.0916) = -1.9084$

(Step3) $\Rightarrow$ ($i=2$)

$x_{1,2} = 0.8626 + 2.9741 = 3.8367 \rightarrow f(x_{1,2}) = -477.2399$

$x_{2,2} = 0.9542 + 1.0211 = 1.9753 \rightarrow f(x_{2,2}) = 42.9100$

$x_{3,2} = 1 + 1.0446 = 2.0446 \rightarrow f(x_{3,2}) = 42.8914$

$x_{4,2} = 1.0916 + (-1.9084) = -0.8168 \rightarrow f(x_{4,2}) = -23.6971$

Since the values of x_j didn't converge, we raise the iteration to $i=3$

Repeat the same procedure until the convergence of the process is achieved.

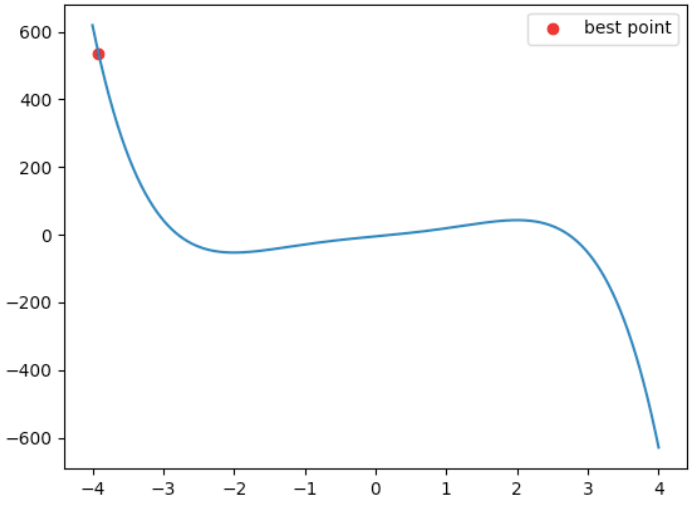
(b) Write a program of PSO for function optimization problem. Please write your own code using any programming lanaguages. If you directly use a package or library from a program language, you will lose the points here.

(c) Find the optimal solution using your program. The termination criterion has to be either until convergence or reaching the maximum number of iteration (set to be 20).

See code 1.py.

→ Result:

```
================== RESTART: /Users/liuqingxuan/Desktop/1.py ==================
x =  -3.9152423294654506
f(x) =  536.6204666549296
```



藍色線條部分為函數於-4~4 區間內的變化，紅色點為 code 找到的最佳解。

2. **Knapsack Problem via GA** *(70 points)*. Suppose you are going to fight for survival in a war. The goal is to maximize your survival points. The following table is a list of weapons you can choose:

Item	Type	Weight	Survival Points
Shadow Daggers	Knife	3.3	7
Huntsman Knife	Knife	3.4	8
Gut Knife	Knife	6.0	13
228 Compact Handgun	Pistols	26.1	29
Night Hawk	Pistols	37.6	48
Desert Eagle Magnum	Pistols	62.5	99
Ingram MAC-10 SMG	Primary	100.2	177
Leone YG1265 Auto Shotgun	Primary	141.1	213
M4A1 Carbine	Primary	119.2	202
AK-47 Rifle	Primary	122.4	210
Krieg 550 Sniper Rifles	Primary	247.6	380
M249 Machine Gun	Primary	352.0	485
Gas Mask	Equipment	24.2	9
Night-Vision Goggle	Equipment	32.1	12
Tactical Shield	Equipment	42.5	15

You can carry at most 529 weight units in your inventory bag. In addition, here are additional requirements and bonus survival points:

(a) You must carry at least one knife, one pistol and one equipment in your inventory bag.

(b) If you carry shadow daggers and desert eagle magnum at the same time, add an additional 5 survival points.

(c) If you carry 228 compact handgun and either AK-47 rifle or M4A1 carbine at the same time, add an additional 15 survival points.

(d) If you carry either Leone YG1265 Auto Shotgun or Krieg 500 Sniper Rifles as primary weapon, plus Desert Eagle Magnum and tactical shield, add an additional 25 survival points.

(e) If you carry all three equipments in your inventory, add an additional 70 survival points.

(a) Write down the statement of the optimization problem from the given information above. State clearly the objective function and the constraints.

See code 2.py. (def function)

將 15 種武器按照順序形成一個 list，並以 list 的方式操作：

（ ans 代表 survival point 的數值）

(a) if x[0]==1 and x[5] == 1: ans += 5

(b) if x[3]==1 and (x[8]==1 or x[9]==1): ans += 15

(c) if (x[7]==1 or x[10]==1) and (x[5]==1 and x[14]==1): ans += 25

(d) if (x[12]==1 and x[13]==1 and x[14]==1): ans += 70

(b) Calculate the maximum number of possible combinations of inventory bags.

ANS: 6455 種。 See code 2-2.py.

```
================= RESTART: /Users/liuqingxuan/Desktop/2-2.py =================
總共可能解數量: 6455
>>>
```

(c) Write a program of Genetic Algorithm. We consider the roulette-wheel selection, uniform crossover (crossover probability = 0.1) and multi bit flip mutation (consecutive based on item types) as three operators.

(d) Use your GA program to optimize your survival points. The population size is set at 10 and the maximum number of iteration is set at 20 steps. No variation convergence criterion is used for termination.

See code 2.py.

(e) Use the hill climbing algorithm and random walk algorithms with the mutation operator defined by your own on this problem. The maximum number of iteration is set at 200 steps.

See code 2.py.

(f) Draw the progress diagrams (best objective function value vs number of function evaluation) of GA, Hill Climbing and Random Walk. Comment on how their performances are associated with the choice of their operators. (NOTE: think carefully how to compare GA and other two methods on the same plot with number of FE as x-axis).

- **Hill climbing:**

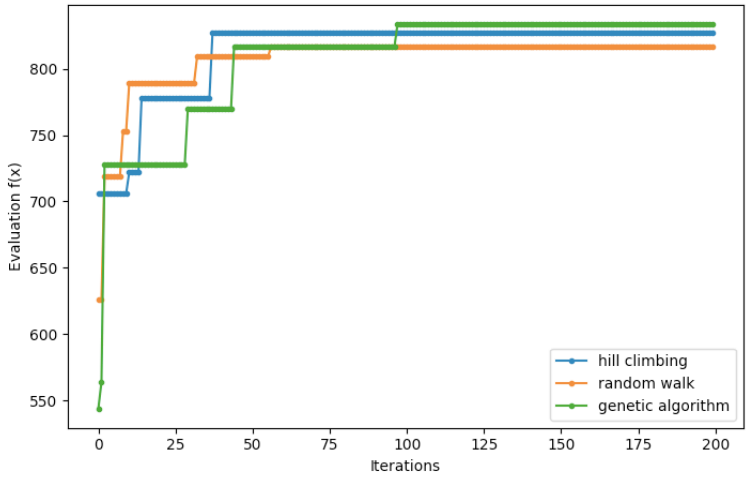
```
================== RESTART: /Users/liuqingxuan/Desktop/2.py ==================
最終最佳點: [1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0]
最終最佳解: 827
```

- **Random walk:**

```
最終最佳點: [1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0]
最終最佳解: 817
```

- **Genetic Algorithm:**

```
>0, new best f([1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1]) = 564.000
>0, new best f([1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1]) = 728.000
>2, new best f([1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1]) = 770.000
>4, new best f([1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0]) = 817.000
>9, new best f([1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]) = 834.000
Done!
f([1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]) = 834.000000
```



→ Comparison :

- GA 演算法的優點：
  (1) 可群體 (多變數) 搜尋，易於並行化處理；(2) 不受連續、可微等條件的約束，適用範圍很廣。
- GA 演算法的缺點：
  很容易陷入區域最佳解的情況中，全域搜尋能力並不強。
- Hill climbing 跟 random walk 主要使用於單變數的優化，且 Hill climbing 更容易陷入區域最佳解當中，很難更加優化。

→ 整體來說 GA 的表現是最佳的，在同樣的 iteration 之內，找到某一個解之後，比起其他方法更有機會再優化答案，不像 random walk 靠運氣找更佳的解，也不像 hill climbing 會很早被困於最佳解中很難再優化。

3. **Linear Programming** *(Bonus 10 points)* Can Question 1 and Question 2 be solved by linear programming? If yes, solve it using a linear program (you can use package or library of your own programming language in this part). If no, solve it using simulated annealing. Compare their optimal solution with the solutions you obtained from PSO (Question 1) and GA (Question 2).

See code 3.py.

- **Q1:**

```
================== RESTART: /Users/liuqingxuan/Desktop/3.py ==================
Q1 Answer:
x =  2.0000005641859877
f(x) =  42.999999999984084
```

- **Q2:（1 代表攜帶該武器，0 代表不攜帶該武器，Objective 為最佳解）**

最佳解：835。

```
Q2 Answer:
Shadow Daggers = 1.000000
Huntsman Knife = 1.000000
Gut Knife = 0.000000
228 Compact Handgun = 1.000000
Night Hawk = 0.000000
Desert Eagle Magnum = 0.000000
Ingram MAC-10 SMG = 1.000000
Leone YG1265 Auto Shotgun = 0.000000
M4A1 Carbine = 0.000000
AK-47 Rifle = 1.000000
Krieg 550 Sniper Rifles = 1.000000
M249 Machine Gun = 0.000000
Gas Mask = 1.000000
Night-Vision Goggle = 0.000000
Tactical Shield = 0.000000
Objective = -835.000000
```