Metaheuristics Homework-1

Student : M11015Q03 童暘修

Homework Github URL : https://github.com/tongysmember/Metaheuristic_hw

Notes on Homework:

1. Clean Code
2. Pseudocode
3. Search Feasibility
4. Convergence Curve

Question 2:

$$f(x_1, x_2) = |x_1^2 + x_2^2 + x_1 x_2| + |sin(x_1)| + |cos(x_2)| \quad (2)$$

Bound at $[-500 \leq x_i \leq 500]$. Global minimum solution at $x^* = (0, 0)$, where $f(x^*) = 1$.

1. Clean Code

Reference material for coding cleaner, separate every functional code into classes. And naming become reasonable. Class files as described below.

- **ALGO Folder** : Algorithm Files
  - **SA.py** : Simulated annealing algorithm
  - **GA.py** : Genetic algorithm
- **FUNC Folder** : Functional and Question.2 Files
  - **GRAPH.py** : Show 2D and 3D graph, made user understand convergence and iterate position easily.
  - **Objective.py** : Assigned question, it's fitness in algorithm method.
- **HW1.py** : Main Program in homework1, just run this file by python3, it will show compute process information and results.

2. Pseudocode

2.1 Simulate Annealing Algorighm

```
Algorithm. Simulated Annealing
-------------------------------------------------
initinial value:
  x1, x2, fitness

while TemperatureNow > TemperatureMin:
  for iterationTimes:
    generate new x1, x2
    fitnessNew => fitness(x1, x2)
    if fitnessNew < fitness:
      Update(x1, x2)
    TemperatureNow = Temperature_decrease()

until TemperatureNow <= TemperatureMin
return Best x1, x2
```

2.1 Genetic Algorithm

```
Algorithm. Genetic Algorithm
-------------------------------------------------
initinial value:
  DNA_Size, Population_Size, CrossoverRate, MutationRate, IterationTimes

for 1,2,3,... IterationTimes:
  x1, x2 => GenerateDNA()
  Population, Fitness => Selection()
    if Fitness <= Evalate:
      return Best x1, x2 Value
  Child => Crossover()
  Child_Next => Mutation()
```

3. Search Feasibility

3.1 Assign random solution

In algorithm files, initial points and generate next-points which iterate time increase used random API method. Generated variable code as show below.

Simulated Annealing Algorithm:

```
30        x1 = np.random.uniform(low=self.lower_bound,high=self.upper_bound)
31        x2 = np.random.uniform(low=self.lower_bound,high=self.upper_bound)
```

Genetic Algorithm:

```
99        pop = np.random.randint(2, size=(self.POP_SIZE, self.DNA_SIZE*2))
```

3.2 Fit the to the end-border

Iteration result are close end-border nearly. Position of point varies very slightly. Compare with questions minimum value are very closely.

Simulated Annealing Algorithm:

```
Times: 990  Temprate_now: 0.10090817356205853 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 991  Temprate_now: 0.10080645161290322 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 992  Temprate_now: 0.10070493454179255 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 993  Temprate_now: 0.1006036217303823 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 994  Temprate_now: 0.10050251256281408 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 995  Temprate_now: 0.10040160642570281 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 996  Temprate_now: 0.10030090270812438 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 997  Temprate_now: 0.10020040080160321 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 998  Temprate_now: 0.1001001001001001 x1:-0.009956356476807926, x2:-0.11293759141657622
Times: 999  Temprate_now: 0.1 x1:-0.009956356476807926, x2:-0.11293759141657622
```

Genetic Algorithm:

```
min_fitness: 1.0396797186379436
(x, y): (-0.004386063669244322, 0.2692095006528916)
min_fitness: 1.0396797186379436
(x, y): (-0.004386063669244322, 0.2692095006528916)
min_fitness: 1.004609574190532
(x, y): (-0.004386063669244322, 0.025068875425517945)
```

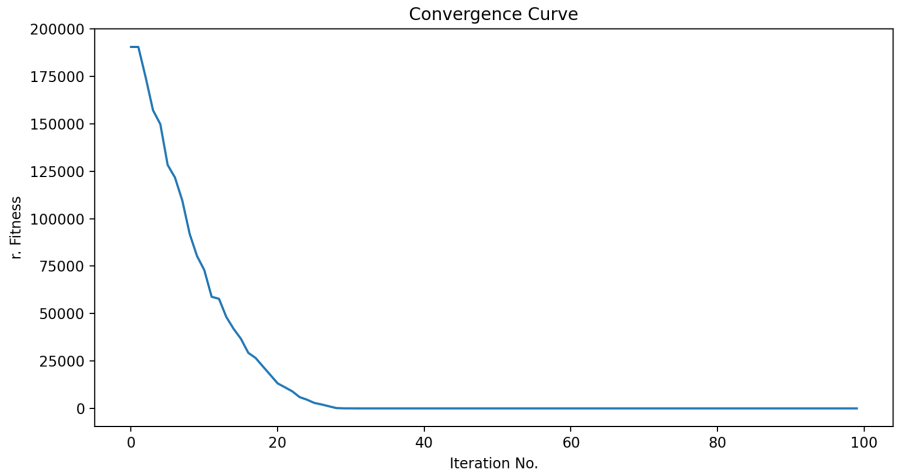
3.3 Overlap the move

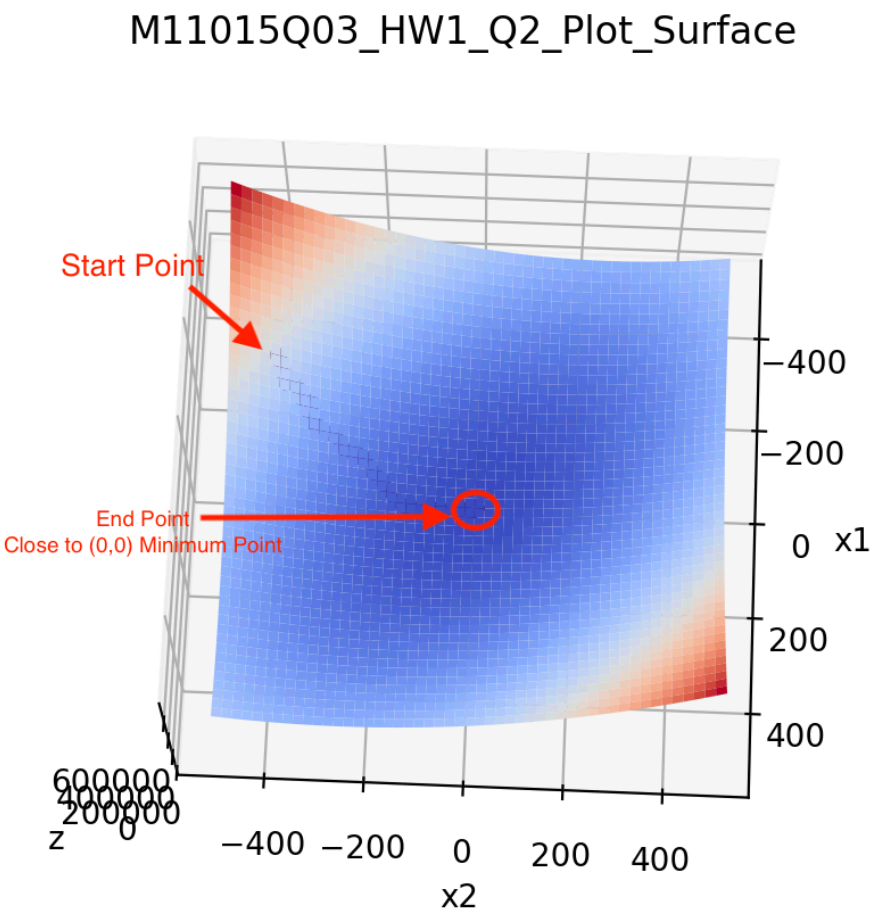
The overlapping movement will be verified by the fitness function in the algorithm. Compared with Best_Fitness, it is determined whether it needs to move to the next area. It is found through the 3D chart that it can move towards the minimum position.

4. Convergence Curve

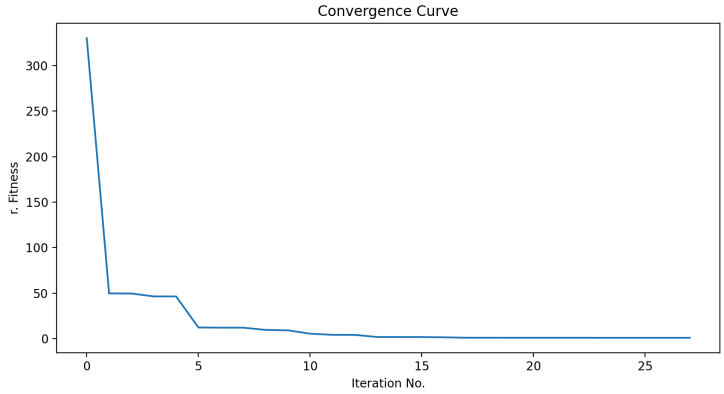
The following chart shows the convergence curve. Through the calculation results of the two algorithms, it can be seen that the fitness value continues to decrease and converge to the minimum value. The 3D graph shows that the algorithm is towards the lowest point.

4.1 Simulate Annealing Algorighm





4.1 Genetic Algorithm



5. Experimental experience

In report, I used 2 algorithms to solve minimum value. Each simulated annealing algorithm and genetic algorithm. Simulated annealing process more simple and easily implement, setting variable are few than genetic algorithm. Genetic algorithm is evolute by multiple sub process. And it needs setting more algorithm numerical value(crossover_rate, mutation_rate, population_size, indivition_size). But crossover and mutation mechanism can prevent solution stay in local optimal. It's suitable for more complex situation and need more computation.

Question 2 is a smooth surface plot. This situation isn't easy to encounter local optimal. Thus, between 2 algorithms' result are very small value. But genetic algorithm need more computation for process running.