AI 프로그래밍

#HOMEWORK 11

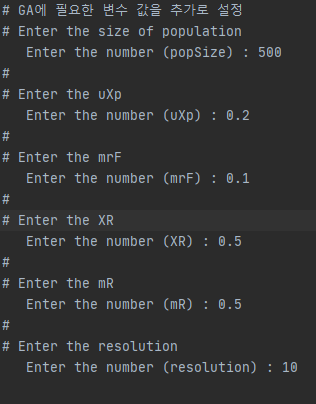


학번: 201724461

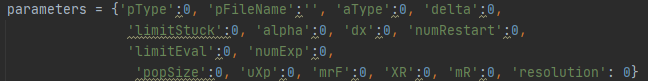
이름: 류지환

Adding GA

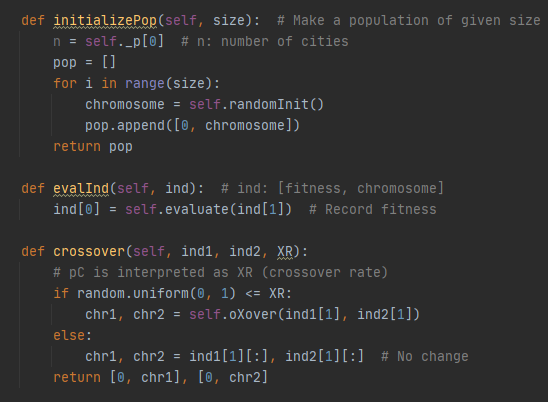
1. Exp.txt
   1. GA에서 요구하는 변수를 추가 후 수치를 조정.

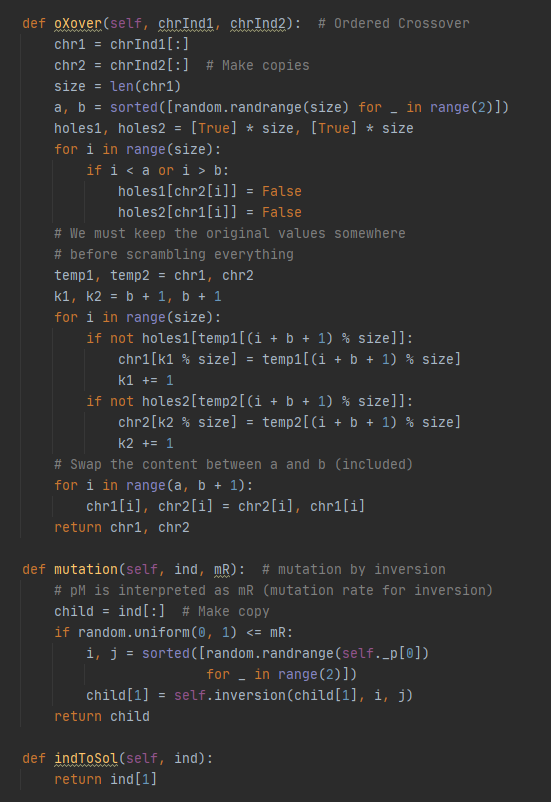


1. main.py
   1. Parameters.key에 앞서 exp.txt에 추가한 변수들을 추가.

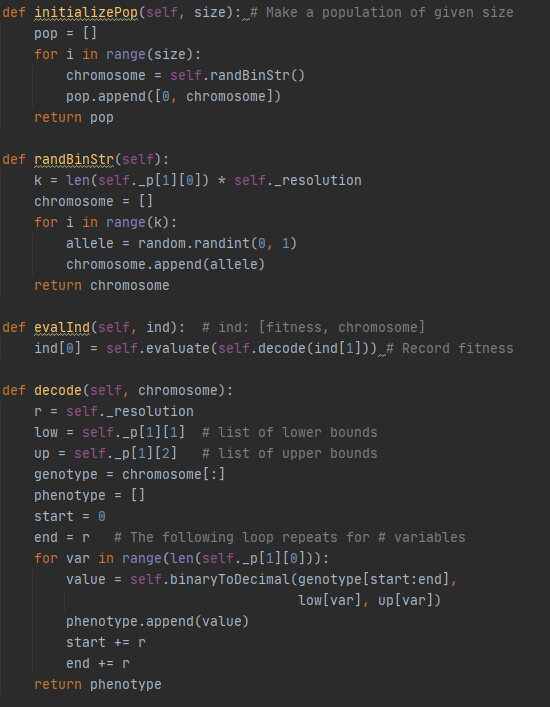


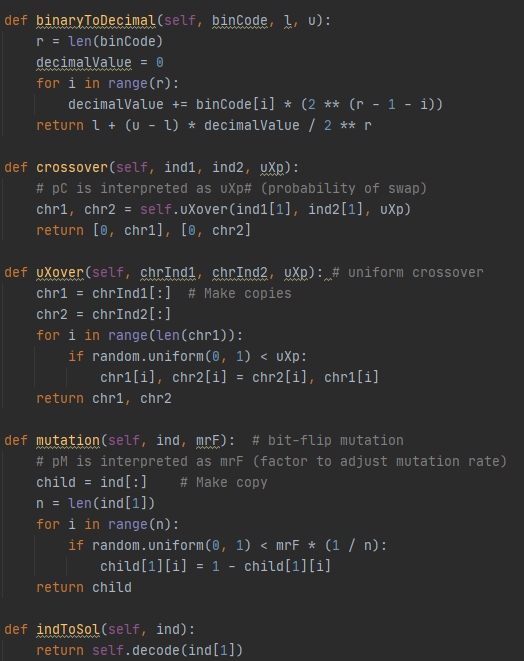
1. Problem.py
   1. 사전에 제공된 코드를 tsp와 numeric 내부 함수로 각각 추가.
      1. Tsp:



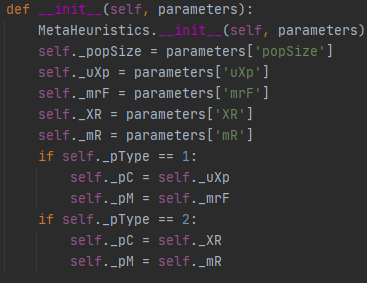


* + 1. numeric:

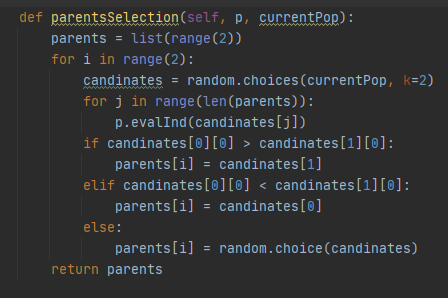




1. Optimizer.py
   1. GA
      1. 제공된 코드 중 def setVariable(self, parameters)의 기능을 기존 본인 코드에 맞추어 def \_\_init\_\_(self, p)에서 동작하도록 수정.



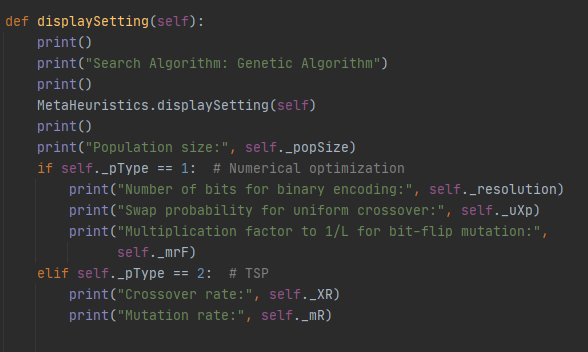
* + 1. child의 기반이 될 parents를 설정하는 함수 parentsSelection(self, p, currentPop)을 추가.



* + 1. run(self, p) 함수 구현.

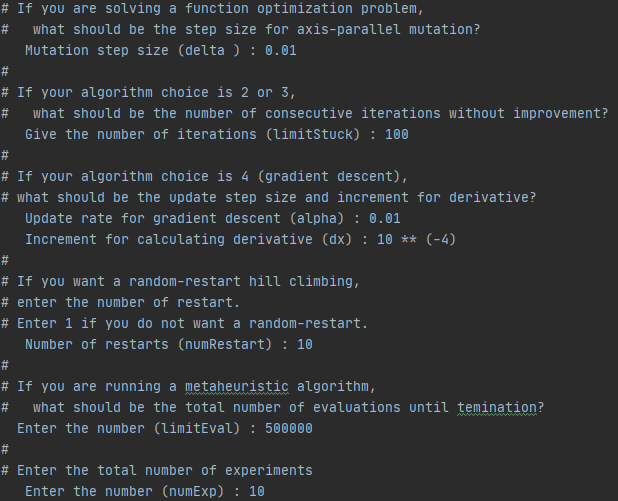


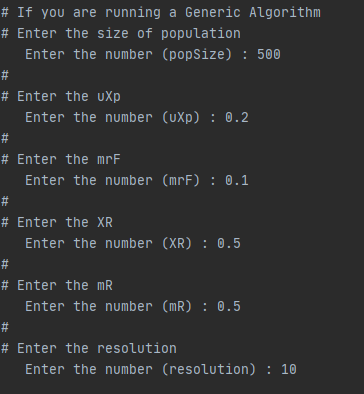
* + 1. 기존에 제공된 displaySetting(self) 함수를 추가.



Experiments

1. 설정





1. 결과

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **results** | **Convex** | **Griewank** | **Ackley** | **tsp30** | **tsp50** | **tsp100** |
| **Steepest-Ascent** | Average Objective  Value | 0.000 | 0.154 | 17.766 | 517.930 | 767.572 | 1,202.496 |
| Best Objective  Value  Found | 0.000 | 0.032 | 17.198 | 468.001 | 705.125 | 1,090.253 |
| Average  Number of  Evaluations | 80,138 | 85,190 | 1,414 | 733 | 1,964 | 8,687 |
| **First-Choice** | Average Objective  Value | 0.000 | 0.159 | 17.461 | 460.556 | 706.276 | 1,339.535 |
| Best Objective  Value  Found | 0.000 | 0.067 | 16.135 | 433.692 | 647.940 | 1,284.452 |
| Average  Number of  Evaluations | 29,079 | 3,116 | 515 | 629 | 1,266 | 1,996 |
| **Stochastic** | Average Objective  Value | 0.000 | 0.168 | 17.971 | 429.670 | 638.589 | 1,132.446 |
| Best Objective  Value  Found | 0.000 | 0.052 | 16.443 | 419.829 | 605.566 | 1,054.901 |
| Average  Number of  Evaluations | 217,954 | 31,467 | 5,300 | 17,944 | 53,026 | 185,871 |
| **Gradient Descent** | Average Objective  Value | 0.000 | 0.142 | 17.233 | - | - | - |
| Best Objective  Value  Found | 0.000 | 0.027 | 11.545 | - | - | - |
| Average  Number of  Evaluations | 6,996 | 44,311 | 557 | - | - | - |
| **Simulated Annealing** | Average Objective  Value | 0.000 | 0.414 | 18.712 | 410.091 | 578.614 | 836.504 |
| Best Objective  Value  Found | 0.000 | 0.214 | 16.172 | 407.979 | 565.014 | 804.123 |
| Average  Number of  Evaluations | 500,000 | 500,000 | 500,000 | 500,000 | 500,000 | 500,000 |
| Average  Iteration of Finding  The best  Solution | 122,497 | 124,274 | 59,720 | 470,358 | 382,609 | 337,339 |
| **Genetic Algorithm** | Average Objective  Value | 1.949 | 0.024 | 0.195 | 531.512 | 805.038 | 1,632.364 |
| Best Objective  Value  Found | 0.046 | 0.000 | 0.000 | 473.148 | 678.097 | 1,363.647 |
| Average  Number of  Evaluations | 500,000 | 500,000 | 500,000 | 500,000 | 500,000 | 500,000 |
| Average  Iteration of Finding  The best  Solution | 72,065 | 108,697 | 59,872 | 135,960 | 243,344 | 456,409 |

1. 평가 지표
   1. 속도
      1. Average Number of Evaluations으로 평가.
      2. 값이 클수록 느리고 작을수록 빠름.
      3. Metaheuristic Algorithms의 경우 Average Iteration of Finding the best solution으로 평가.
   2. 안정성
      1. Average Objective Value - Best Objective Value Found로 평가.
      2. 차가 클수록 낮고 작을수록 높음.
   3. 질
      1. Best Objective Value Found로 평가.
      2. 타 알고리즘들과 비교해 값이 클수록 낮고 작을수록 높음.
2. 평가
   1. Hill Climbing Algorithms
      * 1. Metaheuristic Algorithms에 비해 상대적으로 알고리즘들의 안정성이 높음.
      1. Steepest-Ascent
         1. **Griewank에서 속도가 느리나 질이 높음.**
         2. **TSP에서 방문 도시 수가 커질수록 질이 높아짐.**
      2. First-Choice
         1. 전체 알고리즘 중 속도가 가장 빠름.
         2. TSP에서 방문 도시 수가 커질수록 질이 낮아짐.
      3. Stochastic
         1. Hill Climbing Algorithms 중 속도가 가장 느림.
         2. 모든 유형의 problems에서 질이 높음.
      4. Gradient Descent
         1. 전체 알고리즘 중 numeric problems에서 질이 가장 높음.
         2. Convex와 Ackley에서 속도가 가장 빠름.
         3. 다만 Ackley의 경우 안정성이 낮음.
   2. Metaheuristic Algorithms
      * 1. Hill Climbing Algorithms에 비해 상대적으로 알고리즘들의 속도가 느림.
      1. Simulated Annealing
         1. 전체 알고리즘 중 Griewank에서의 질이 가장 낮음.
         2. 이외 problems에서의 질이 높음.
      2. Genetic Algorithm
         1. 전체 알고리즘 중 전반적인 problems에서의 안정성이 가장 낮음.
         2. 전체 알고리즘 중 전반적인 problems에서의 질이 낮음.
         3. Griewank와 Ackley에서 알고리즘의 질이 높음.
         4. 특히 Ackley에서의 질이 타 알고리즘들에 비해 매우 높음.