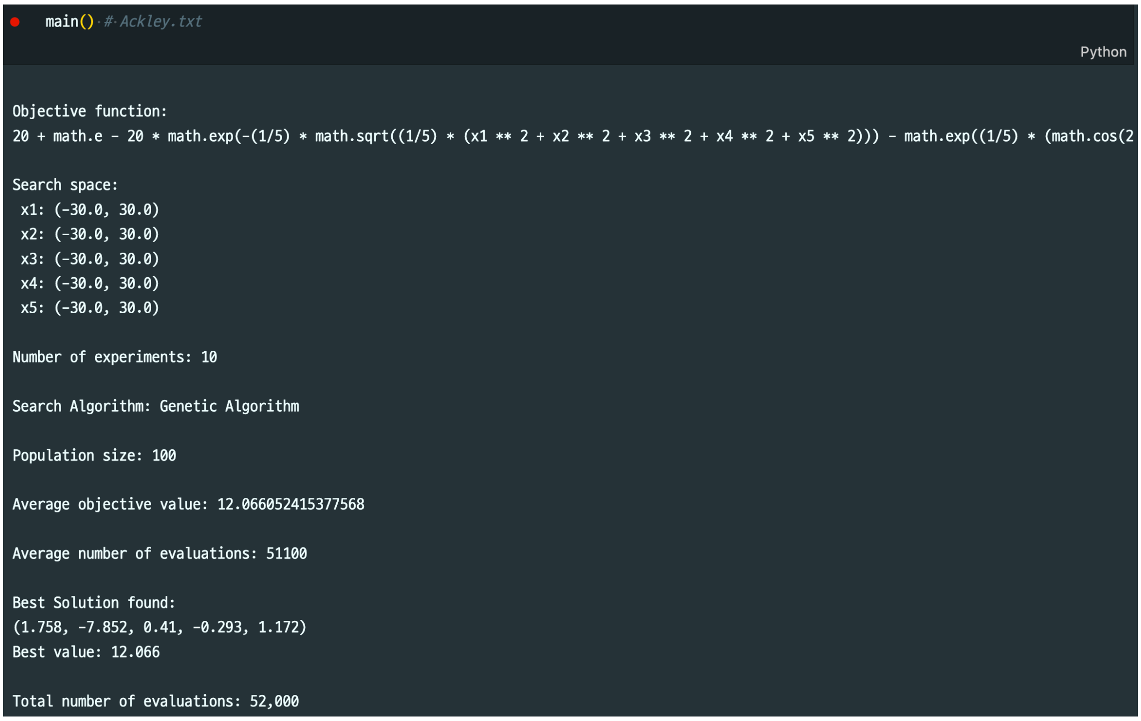
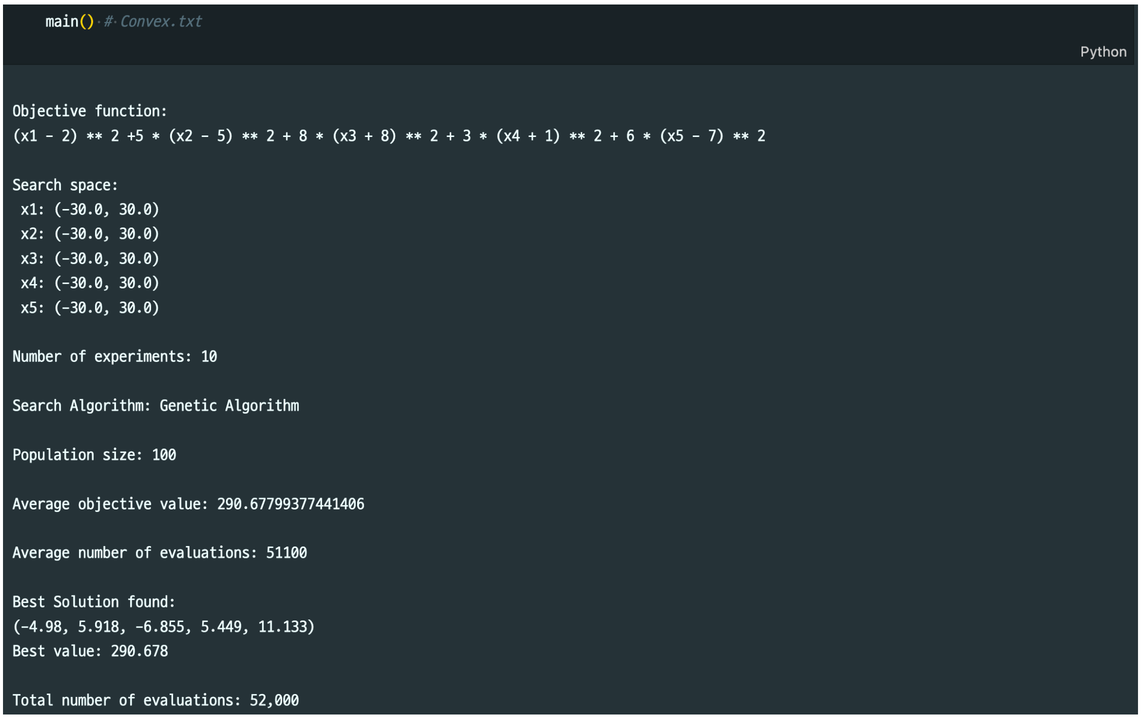
## 1. Genetic Algorithm(n), Ackley.txt



```
main() # Ackley.txt
```

```
Objective function:
20 + math.e - 20 * math.exp(-(1/5) * math.sqrt((1/5) * (x1 ** 2 + x2 ** 2 + x3 ** 2 + x4 ** 2 + x5 ** 2))) - math.exp((1/5) * (math.cos(2

Search space:
 x1: (-30.0, 30.0)
 x2: (-30.0, 30.0)
 x3: (-30.0, 30.0)
 x4: (-30.0, 30.0)
 x5: (-30.0, 30.0)

Number of experiments: 10

Search Algorithm: Genetic Algorithm

Population size: 100

Average objective value: 12.066052415377568

Average number of evaluations: 51100

Best Solution found:
(1.758, -7.852, 0.41, -0.293, 1.172)
Best value: 12.066

Total number of evaluations: 52,000
```

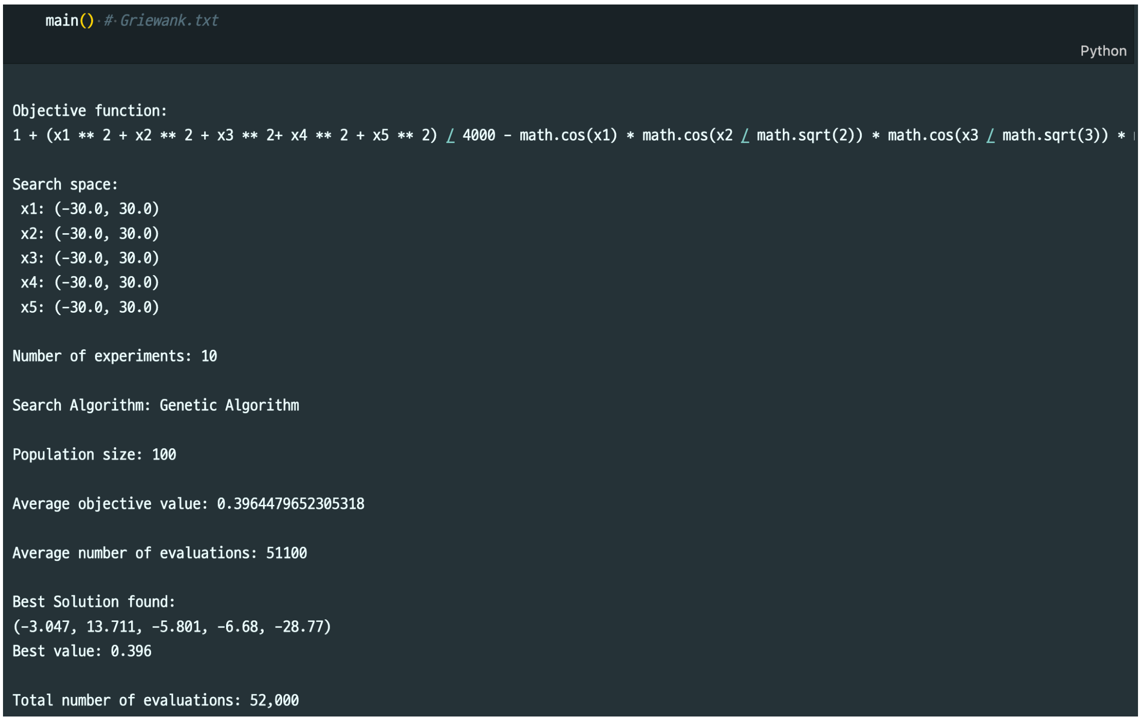
## 2. Genetic Algorithm(n), Convex.txt



```
main() # Convex.txt
```

```
Objective function:
(x1 - 2) ** 2 +5 * (x2 - 5) ** 2 + 8 * (x3 + 8) ** 2 + 3 * (x4 + 1) ** 2 + 6 * (x5 - 7) ** 2

Search space:
 x1: (-30.0, 30.0)
 x2: (-30.0, 30.0)
 x3: (-30.0, 30.0)
 x4: (-30.0, 30.0)
 x5: (-30.0, 30.0)

Number of experiments: 10

Search Algorithm: Genetic Algorithm

Population size: 100

Average objective value: 290.67799377441406

Average number of evaluations: 51100

Best Solution found:
(-4.98, 5.918, -6.855, 5.449, 11.133)
Best value: 290.678

Total number of evaluations: 52,000
```

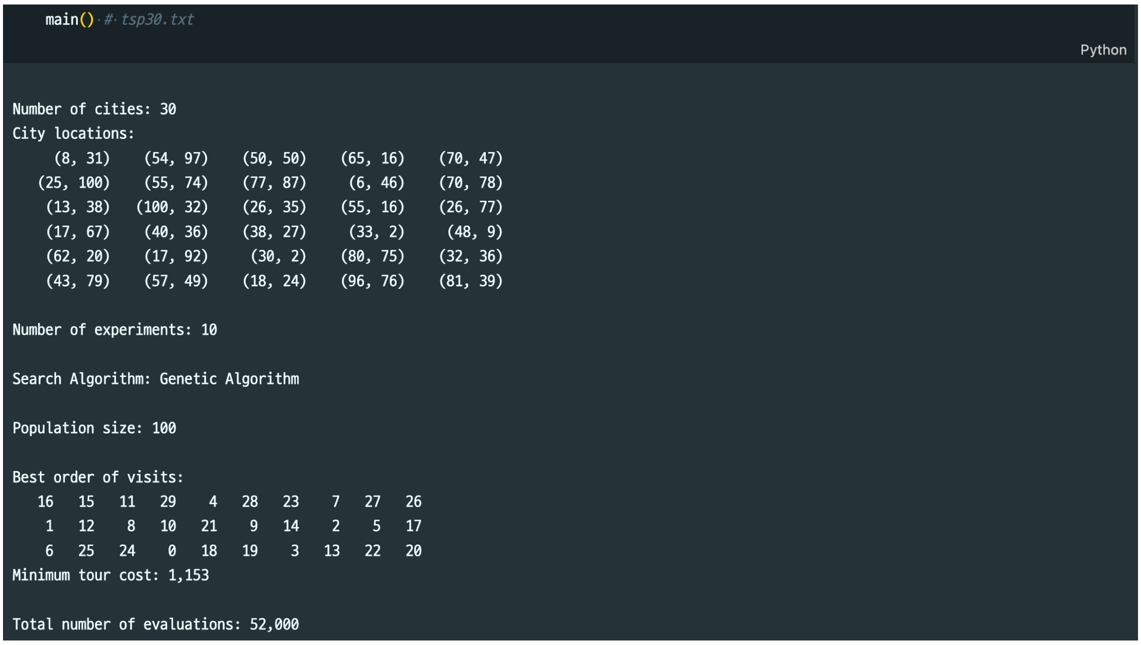
## 3. Genetic Algorithm(n), Griewank.txt



```
    main() # Griewank.txt
```

```
Objective function:
1 + (x1 ** 2 + x2 ** 2 + x3 ** 2+ x4 ** 2 + x5 ** 2) / 4000 - math.cos(x1) * math.cos(x2 / math.sqrt(2)) * math.cos(x3 / math.sqrt(3)) *

Search space:
 x1: (-30.0, 30.0)
 x2: (-30.0, 30.0)
 x3: (-30.0, 30.0)
 x4: (-30.0, 30.0)
 x5: (-30.0, 30.0)

Number of experiments: 10

Search Algorithm: Genetic Algorithm

Population size: 100

Average objective value: 0.3964479652305318

Average number of evaluations: 51100

Best Solution found:
(-3.047, 13.711, -5.801, -6.68, -28.77)
Best value: 0.396

Total number of evaluations: 52,000
```

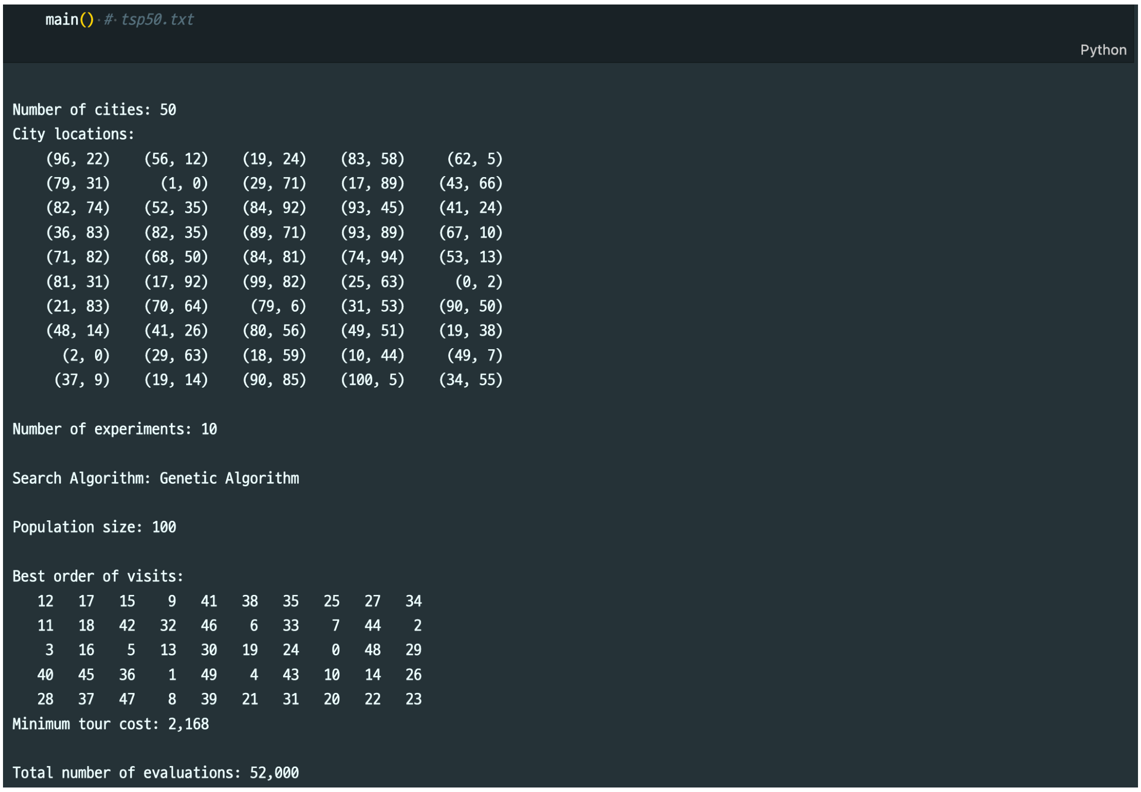
## 4. Genetic Algorithm(tsp), tsp30.txt



```
    main() # tsp30.txt
```

```
Number of cities: 30
City locations:
     (8, 31)    (54, 97)    (50, 50)    (65, 16)    (70, 47)
   (25, 100)    (55, 74)    (77, 87)     (6, 46)    (70, 78)
    (13, 38)   (100, 32)    (26, 35)    (55, 16)    (26, 77)
    (17, 67)    (40, 36)    (38, 27)     (33, 2)     (48, 9)
    (62, 20)    (17, 92)     (30, 2)    (80, 75)    (32, 36)
    (43, 79)    (57, 49)    (18, 24)    (96, 76)    (81, 39)

Number of experiments: 10

Search Algorithm: Genetic Algorithm

Population size: 100

Best order of visits:
   16   15   11   29    4   28   23    7   27   26
    1   12    8   10   21    9   14    2    5   17
    6   25   24    0   18   19    3   13   22   20
Minimum tour cost: 1,153

Total number of evaluations: 52,000
```

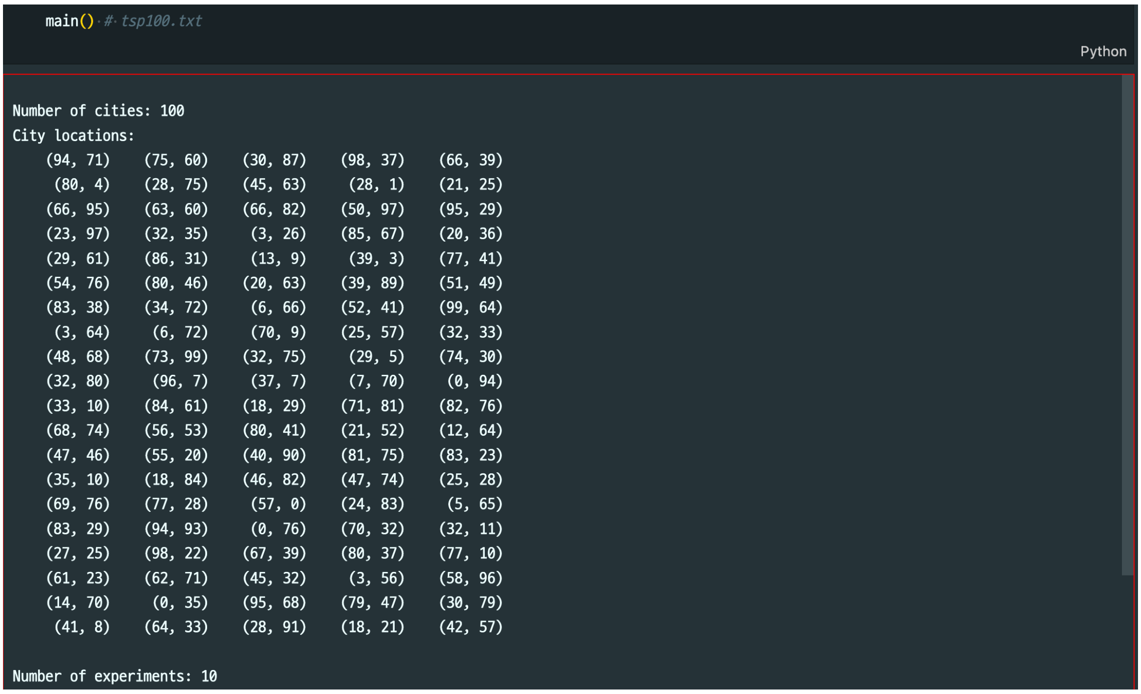
## 5. Genetic Algorithm(tsp), tsp50.txt



```python
    main() # tsp50.txt
```

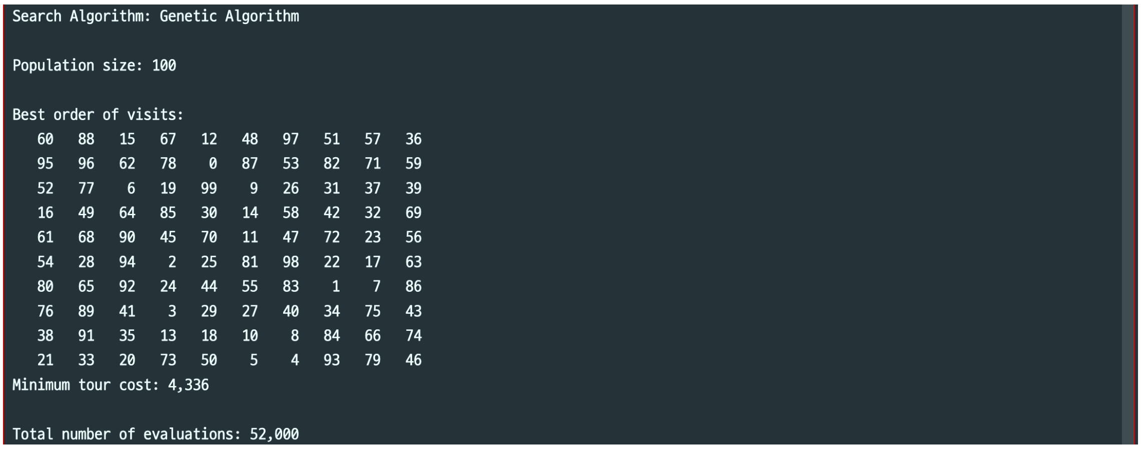
```
Number of cities: 50
City locations:
    (96, 22)    (56, 12)    (19, 24)    (83, 58)     (62, 5)
    (79, 31)      (1, 0)    (29, 71)    (17, 89)    (43, 66)
    (82, 74)    (52, 35)    (84, 92)    (93, 45)    (41, 24)
    (36, 83)    (82, 35)    (89, 71)    (93, 89)    (67, 10)
    (71, 82)    (68, 50)    (84, 81)    (74, 94)    (53, 13)
    (81, 31)    (17, 92)    (99, 82)    (25, 63)      (0, 2)
    (21, 83)    (70, 64)     (79, 6)    (31, 53)    (90, 50)
    (48, 14)    (41, 26)    (80, 56)    (49, 51)    (19, 38)
      (2, 0)    (29, 63)    (18, 59)    (10, 44)     (49, 7)
     (37, 9)    (19, 14)    (90, 85)    (100, 5)    (34, 55)

Number of experiments: 10

Search Algorithm: Genetic Algorithm

Population size: 100

Best order of visits:
   12   17   15    9   41   38   35   25   27   34
   11   18   42   32   46    6   33    7   44    2
    3   16    5   13   30   19   24    0   48   29
   40   45   36    1   49    4   43   10   14   26
   28   37   47    8   39   21   31   20   22   23
Minimum tour cost: 2,168

Total number of evaluations: 52,000
```

## 6. Genetic Algorithm(tsp), tsp100.txt



```python
main() # tsp100.txt
```

```
Number of cities: 100
City locations:
    (94, 71)    (75, 60)    (30, 87)    (98, 37)    (66, 39)
     (80, 4)    (28, 75)    (45, 63)     (28, 1)    (21, 25)
    (66, 95)    (63, 60)    (66, 82)    (50, 97)    (95, 29)
    (23, 97)    (32, 35)     (3, 26)    (85, 67)    (20, 36)
    (29, 61)    (86, 31)     (13, 9)     (39, 3)    (77, 41)
    (54, 76)    (80, 46)    (20, 63)    (39, 89)    (51, 49)
    (83, 38)    (34, 72)     (6, 66)    (52, 41)    (99, 64)
     (3, 64)     (6, 72)     (70, 9)    (25, 57)    (32, 33)
    (48, 68)    (73, 99)    (32, 75)     (29, 5)    (74, 30)
    (32, 80)     (96, 7)     (37, 7)     (7, 70)     (0, 94)
    (33, 10)    (84, 61)    (18, 29)    (71, 81)    (82, 76)
    (68, 74)    (56, 53)    (80, 41)    (21, 52)    (12, 64)
    (47, 46)    (55, 20)    (40, 90)    (81, 75)    (83, 23)
    (35, 10)    (18, 84)    (46, 82)    (47, 74)    (25, 28)
    (69, 76)    (77, 28)     (57, 0)    (24, 83)     (5, 65)
    (83, 29)    (94, 93)     (0, 76)    (70, 32)    (32, 11)
    (27, 25)    (98, 22)    (67, 39)    (80, 37)    (77, 10)
    (61, 23)    (62, 71)    (45, 32)     (3, 56)    (58, 96)
    (14, 70)     (0, 35)    (95, 68)    (79, 47)    (30, 79)
     (41, 8)    (64, 33)    (28, 91)    (18, 21)    (42, 57)

Number of experiments: 10
```



```
Search Algorithm: Genetic Algorithm

Population size: 100

Best order of visits:
   60   88   15   67   12   48   97   51   57   36
   95   96   62   78    0   87   53   82   71   59
   52   77    6   19   99    9   26   31   37   39
   16   49   64   85   30   14   58   42   32   69
   61   68   90   45   70   11   47   72   23   56
   54   28   94    2   25   81   98   22   17   63
   80   65   92   24   44   55   83    1    7   86
   76   89   41    3   29   27   40   34   75   43
   38   91   35   13   18   10    8   84   66   74
   21   33   20   73   50    5    4   93   79   46
Minimum tour cost: 4,336

Total number of evaluations: 52,000
```