

به نام خدا

گزارش پروژه درس هوش مصنوعی

امیرمحمد پیرحسین لو

9531014

مقدمه

در این پروژه ، تعدادی از الگوریتم های جستجو پیاده سازی شدند و مورد ارزیابی قرار گرفتند. ارزیابی الگوریتم ها به وسیله 3 مسئله که در ادامه خواهند آمد ، انجام گرفت.

نکاتی در مورد ساختار کد های پروژه:

- الگوریتم های جستجو در پکیج SearchAlgorithms و الگوریتم های بهینه سازی در پکیج OptimizationAlgorithms قرار دارند.
 - مسئله 1 در پکیج problem1 و مسئله 2 در پکیج problem2 و مسئله 3 در پکیج problem3 قرار دارد.
 - در پکیج مسائل ، کلاس Problem در problem.py و کد نقطه شروع اجرای الگوریتم ها در فایل های start_XXX.py قرار دارد که XXX مخفف نام الگوریتم است ، مثلاً start_dls.py شامل کد برای اجرای الگوریتم limited_depth_search با ورودی کلاس Problem است.
 - مسیر طی شده در خروجی الگوریتم های جستجو از آخر به اول نمایش داده شده است.
1. مسئله اول : مسیریابی شهر ها

a. Breadth first search:

visited = 9 , expanded = 7 , max_memory = 9 , path_cost = 3

Path:

8 , 19 , 18 , 1 ,

b. Depth limited search:

Depth = 2

DLS Cutoff.

visited = 9 , expanded = 4

Depth = 3

visited = 10 , expanded = 7 , max_memory = 0 , path_cost = 3

Path:

8 , 19 , 18 , 1 ,

Depth = 4

visited = 14 , expanded = 9 , max_memory = 0 , path_cost = 3

Path:

8 , 19 , 18 , 1 ,

- Depth = 10
 visited = 8 , expanded = 7 , max_memory = 0 , path_cost = 7
 Path:
 8 , 7 , 6 , 5 , 4 , 3 , 2 , 1 ,
- c. Depth first search
 visited = 8 , expanded = 7 , max_memory = 0 , path_cost = 7
 Path:
 8 , 7 , 6 , 5 , 4 , 3 , 2 , 1 ,
- d. Iterative deepening depth first search
 Depth = 2
 DLS Cutoff.
 visited = 14 , expanded = 5
- Depth = 3
 visited = 10 , expanded = 7 , max_memory = 0 , path_cost = 3
 Path:
 8 , 19 , 18 , 1 ,
- Depth = 5
 visited = 10 , expanded = 7 , max_memory = 0 , path_cost = 3
 Path:
 8 , 19 , 18 , 1 ,
- e. Uniform cost search
 visited = 13 , expanded = 11 , max_memory = 12 , path_cost = 3
 Path:
 8 , 19 , 18 , 1 ,
- f. Greedy best first search
 visited = 8 , expanded = 3 , max_memory = 7 , path_cost = 3
 Path:
 8 , 19 , 18 , 1 ,
- g. A*
 visited = 8 , expanded = 3 , max_memory = 7 , path_cost = 3
 Path:
 8 , 19 , 18 , 1 ,

2. رنگ آمیزی گراف

نحوه ذخیره سازی گراف : لیست مجاورت

a. رنگ آمیزی گراف نمونه

- i. Hill climbing
 visited = 67 , expanded = 3 , local minimum = 1
 Path:
 [1, 2, 3, 3, 3, 1, 2, 2, 3, 3, 1] cost:4
 [1, 2, 3, 3, 1, 1, 2, 2, 3, 3, 1] cost:2

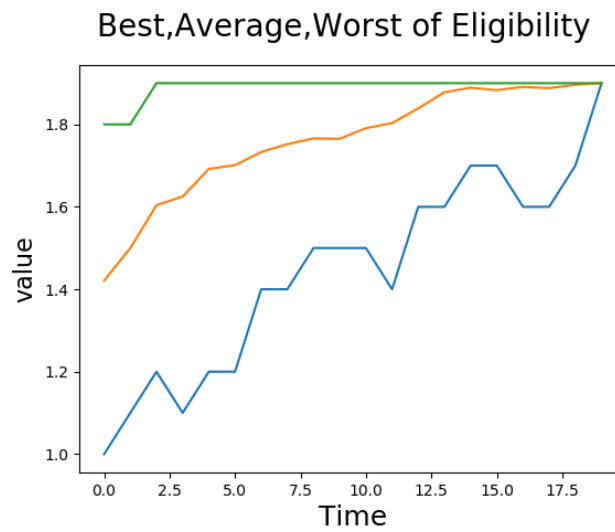
- [1, 2, 3, 3, 1, 1, 2, 2, 3, 2, 1] cost:1
- ii. Stochastic hill climbing
visited = 67 , expanded = 3 , local minimum = 1
[1, 2, 3, 3, 3, 1, 2, 2, 3, 3, 1] cost:4
[1, 2, 2, 3, 3, 1, 2, 2, 3, 3, 1] cost:3
[1, 2, 2, 3, 3, 1, 2, 2, 3, 2, 1] cost:1
- iii. First choice hill climbing
visited = 58 , expanded = 4 , local minimum = 1
[1, 2, 3, 3, 3, 1, 2, 2, 3, 3, 1] cost:4
[1, 2, 2, 3, 3, 1, 2, 2, 3, 3, 1] cost:3
[1, 2, 2, 3, 1, 1, 2, 2, 3, 3, 1] cost:2
[1, 2, 2, 3, 1, 1, 2, 2, 3, 2, 1] cost:1
- iv. Random restart hill climbing
Parameters: steps = 10000
visited = 886348 , expanded = 39834 , local minimum = 1
[3, 1, 1, 2, 1, 2, 3, 1, 3, 1, 2] cost:8
[3, 1, 1, 2, 2, 2, 3, 1, 3, 1, 2] cost:4
[3, 1, 1, 1, 2, 2, 3, 1, 3, 1, 2] cost:3
[3, 1, 1, 1, 2, 2, 3, 1, 1, 1, 2] cost:2
[3, 1, 1, 1, 2, 2, 3, 1, 1, 3, 2] cost:1
- v. Simulated annealing

به دلیل طولانی بودن دنباله جواب ، تنها بخش کوچکی از
ابتدا و انتهای آن ذکر می شود.

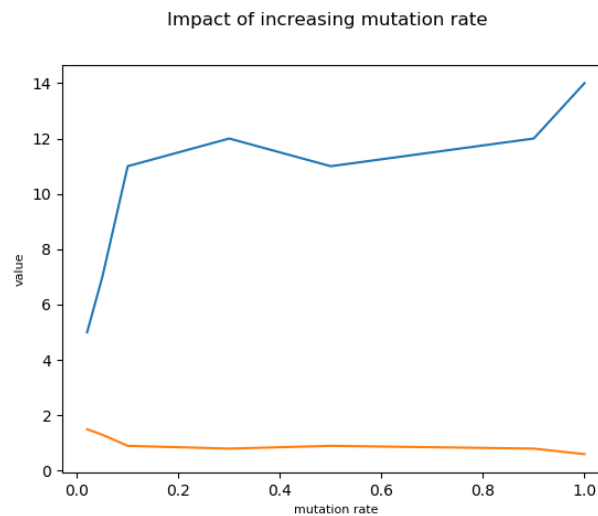
Scheduler: 1.0 / (1.1 ** t)
visited = 7448 , expanded = 7447 , local minimum = 1
[1, 2, 3, 3, 3, 1, 2, 2, 3, 3, 1] cost:4
[1, 2, 3, 3, 3, 2, 2, 2, 3, 3, 1] cost:5
[1, 2, 3, 3, 3, 2, 2, 3, 3, 3, 1] cost:5
[1, 2, 3, 3, 1, 2, 2, 3, 3, 3, 1] cost:2
[1, 2, 3, 3, 1, 2, 2, 3, 1, 3, 1] cost:2
[1, 2, 3, 3, 1, 2, 2, 3, 1, 3, 2] cost:1
[3, 2, 3, 3, 1, 2, 2, 3, 1, 3, 2] cost:1
[1, 2, 3, 3, 1, 2, 2, 3, 1, 3, 2] cost:1
[1, 2, 3, 3, 1, 2, 2, 3, 3, 3, 2] cost:1
.
.
.
[1, 2, 2, 1, 3, 1, 2, 2, 3, 1, 3] cost:1
[1, 2, 2, 1, 3, 3, 2, 2, 3, 1, 3] cost:1
[1, 2, 2, 1, 3, 1, 2, 2, 3, 1, 3] cost:1
[1, 2, 2, 1, 3, 1, 2, 2, 3, 2, 3] cost:1

vi. Genetic

```
population_size = 100  
number_of_generations = 20  
mutation_rate = 0.02  
tornument_size = 4
```



```
population_size = 100  
number_of_generations = 20  
mutation_rate = [0.02, 0.05, 0.1, 0.3, 0.5, 0.9, 1]  
tornument_size = 4
```

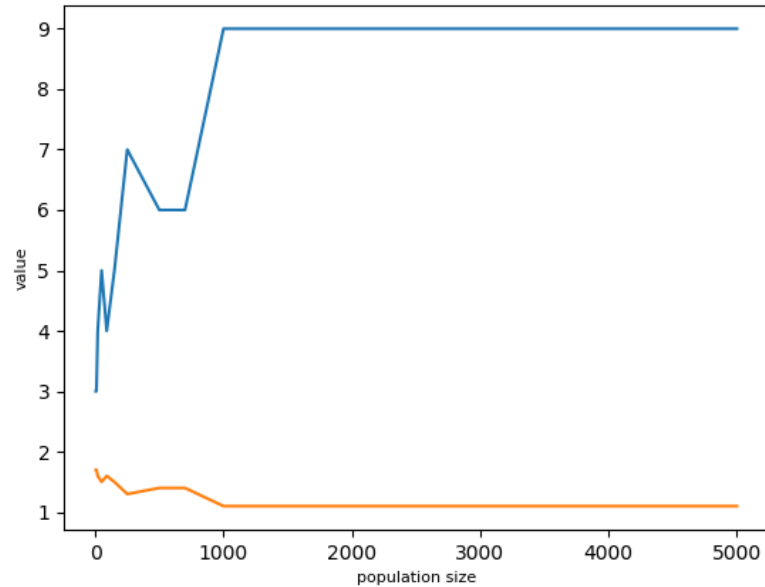


```

population_sizes = [5, 10, 20, 50, 90, 150, 250, 500, 700, 1000,
5000]
number_of_generations = 20
mutation_rate = 0.02
tornument_size = 4

```

Impact of increasing population on convergence of genetic algorithm

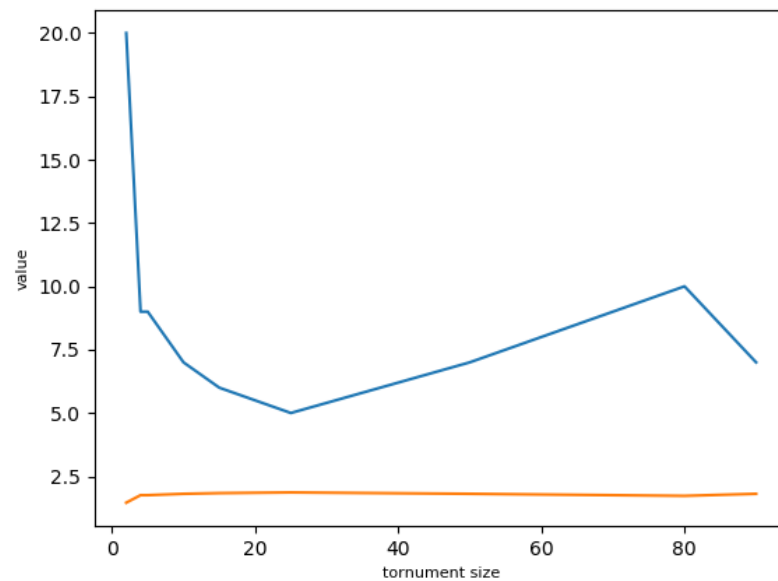


```

population_size = 100
number_of_generations = 20
mutation_rate = 0.02
tornument_size = [2, 4, 5, 10, 15, 25, 50, 60, 70, 80, 90]

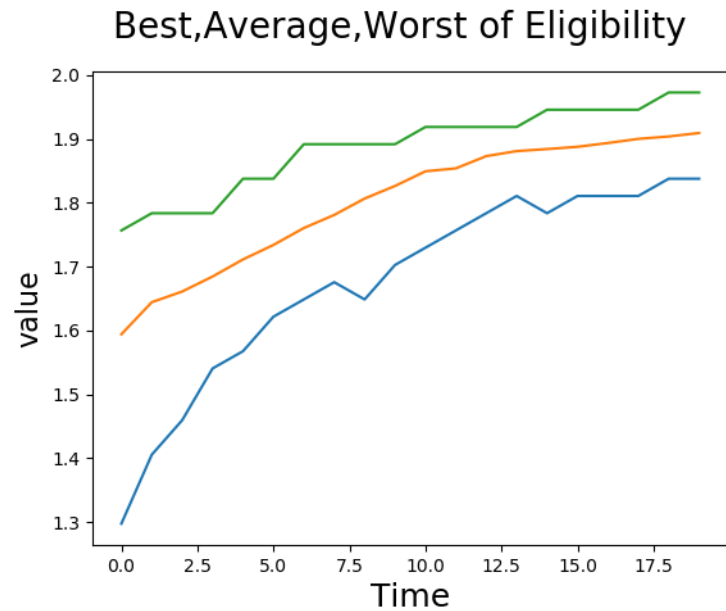
```

Impact of increasing tornument size

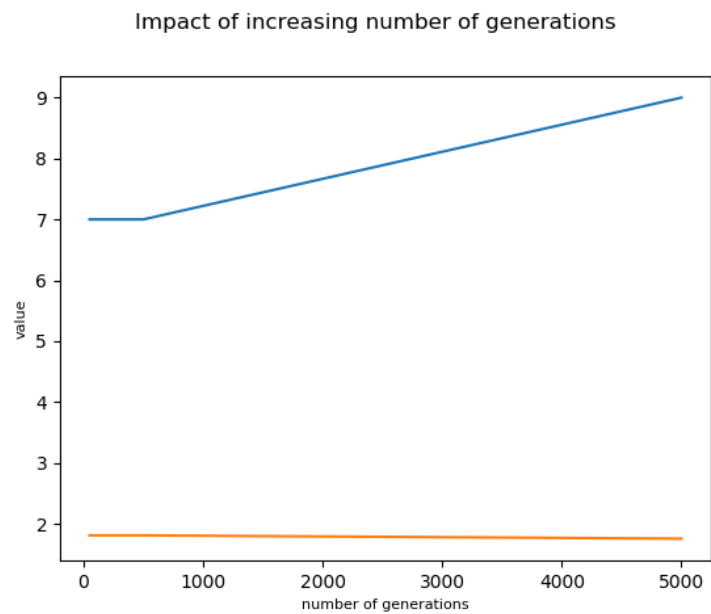


b. رنگ آمیزی گراف شهرهای ایران

```
i. population_size = 100  
number_of_generations = 20  
mutation_rate = 0.02  
tornument_size = 4
```

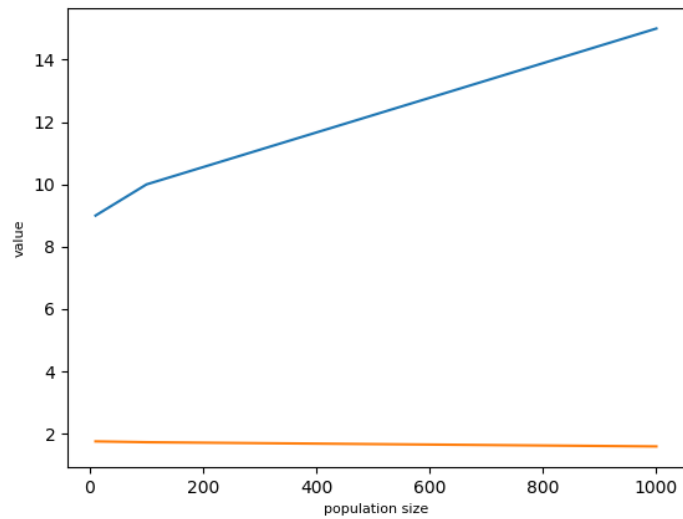


```
ii. population_size = 100  
number_of_generations = [50, 500, 5000]  
mutation_rate = 0.02  
tornument_size = 4
```



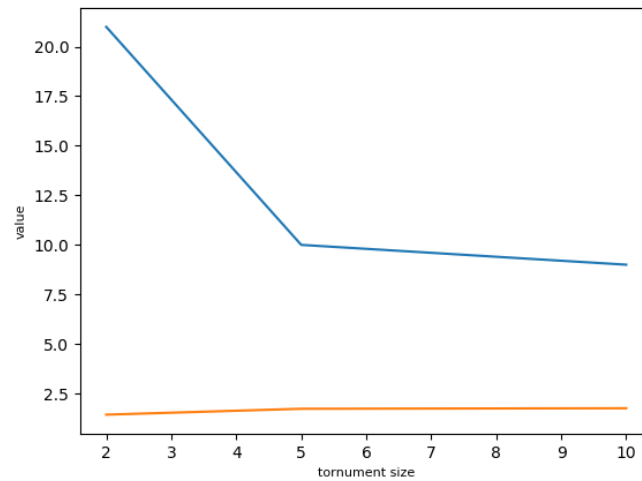
```
iii. population_size = [10, 100, 1000]
    number_of_generations = 20
    mutation_rate = 0.02
    tournament_size = 4
```

Impact of increasing population size



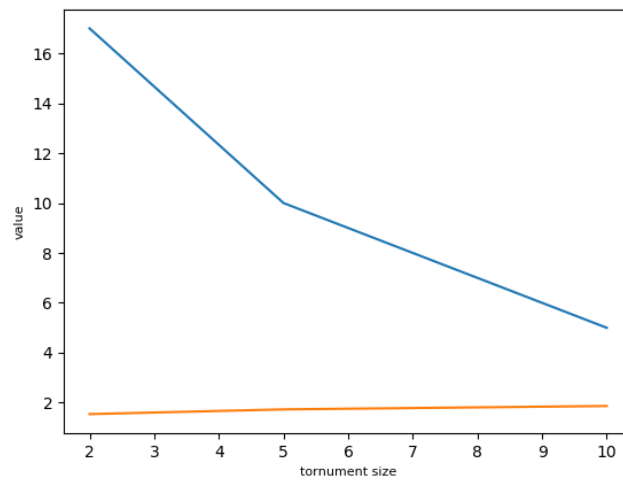
```
iv. population_size = 10
    number_of_generations = 20
    mutation_rate = 0.02
    tournament_size = [2, 5, 10]
```

Impact of increasing tournament size



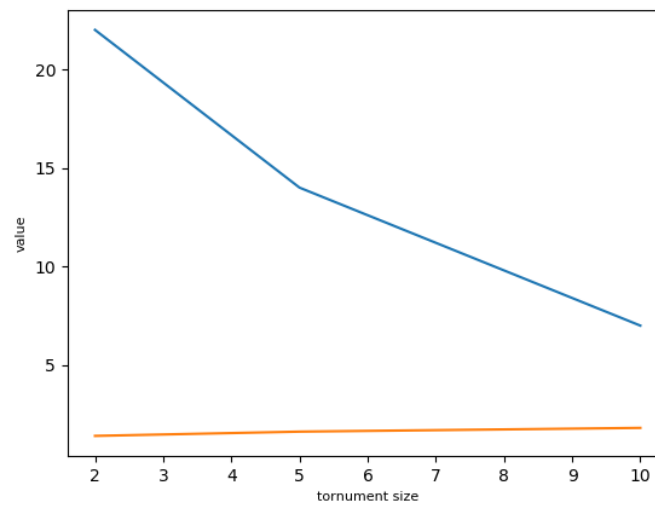
```
v. population_size = 100
    number_of_generations = 20
    mutation_rate = 0.02
    tournament_size = [2, 5, 10]
```

Impact of increasing tournament size



```
vi. population_size = 1000
    number_of_generations = 20
    mutation_rate = 0.02
    tournament_size = [2, 5, 10]
```

Impact of increasing tournament size



```
vii. population_size = 100
    number_of_generations = 20
    mutation_rate = [0.01, 0.02, 0.05, 0.1]
    tournament_size = 4
```


Impact of increasing mutation rate

