به نام خدا

پروژه اول درس رایانش تکاملی

(کوله پشتی)

[redacted]

آبان ۹۹

مسئله‌ای به نام توریست، وجود داره که شخصی قراره آخر هفته به سفر بره و در واقع ۲۲ آیتم داره و ظرفیت کوله‌ای که هم به همراه داره برابره ۴۰۰ هستش. من از مفروضات و داده این مسئله استفاده می‌کنم و آیتم‌های من به شرح زیر هستش :

Item	Weight	Value
map	9	150
compass	13	35
water	153	200
sandwich	50	160
glucose	15	60
tin	68	45
banana	27	60
apple	39	40
cheese	23	30
beer	52	10
suntan cream	11	70
camera	32	30
T-shirt	24	15
trousers	48	10
umbrella	73	40
waterproof trousers	42	70
waterproof overclothes	43	75
note-case	22	80
sunglasses	7	20
towel	18	12
socks	4	50
book	30	10

قصد داریم این مسئله رو با الگوریتم ژنتیک حل کنیم. در مورد نمایش کروموزوم‌ها، ما برای هر فرد، یک آرایه به طول ۲۲ داریم که اگر مقدار هر آیتم برابر ۱ باشه، یعنی اون وسیله در کوله گذاشته می‌شه و اگه ۰ باشه یعنی در کوله نمی‌زاریم.

به مقدار کافی توضیحات رو در کد گذاشتم و کاملا کارکرد کد رو توضیح می‌ده. اما توضیحات مختصری رو می‌دم.

یک کلاس به نام knapsack01problem در فایل knapsack.py وجود داره که حاوی متدهای زیر هستش :

__init_data()

دیتای مسئله رو مقدار دهی اولیه می‌کنه. البته با لیستی از تاپل‌ها. هر تاپل حاوی نام آیتم، وزن و ارزش هر آیتم هستش.

getValue(zeroOneList)

ارزش آیتم‌های موجود در لیست رو محاسبه می‌کنه. در حالی که آیتم‌هایی رو که باعث میشه وزن کوله از ماکس بیشتر بشه رو نادیده می‌گیره

printItems(zeroOneList)

آیتم های انتخاب شده در لیست رو چاپ می‌کنه در حالی که آیتم‌هایی رو که باعث شدن وزن کوله از ماکس بیشتر شه رو نادیده می‌گیره

متد main کلاس، یک نمونه از کلاس knapsack01problem رو می‌سازه. سپس یک راه‌حل رندم می‌سازه و اطلاعات رو چاپ می‌کنه. در واقع اگر این کلاس رو ران کنیم، یک نمونه خروجی مانند زیر مشاهده می‌کنیم.

```
Random Solution =
[1 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0]
- Adding map: weight = 9, value = 150, accumulated weight = 9, accumulated
value = 150
- Adding compass: weight = 13, value = 35, accumulated weight = 22,
accumulated value = 185
- Adding water: weight = 153, value = 200, accumulated weight = 175,
accumulated value = 385
- Adding sandwich: weight = 50, value = 160, accumulated weight = 225,
accumulated value = 545
- Adding glucose: weight = 15, value = 60, accumulated weight = 240,
accumulated value = 605
- Adding beer: weight = 52, value = 10, accumulated weight = 292,
accumulated value = 615
- Adding suntan cream: weight = 11, value = 70, accumulated weight = 303,
accumulated value = 685
- Adding camera: weight = 32, value = 30, accumulated weight = 335,
accumulated value = 715
- Adding trousers: weight = 48, value = 10, accumulated weight = 383,
accumulated value = 725
- Total weight = 383, Total value = 725
```

اگر به تصویر توجه کنین،آخرین ا ! که در راه‌حل رندم وجود داره در واقع آیتم note case رو نشون می‌ده. ولی چون وزنی برابر با ۲۲ داره، اگه اضافه بشه به کوله، باعث میشه که وزن کوله از ٤٠٠ بیشتر شه، در نتیجه این آیتم به راه‌حل اضافه نمی‌شه.

حالا می‌خوایم جواب بهینه رو با استفاده از الگوریتم ژنتیک پیدا کنیم.

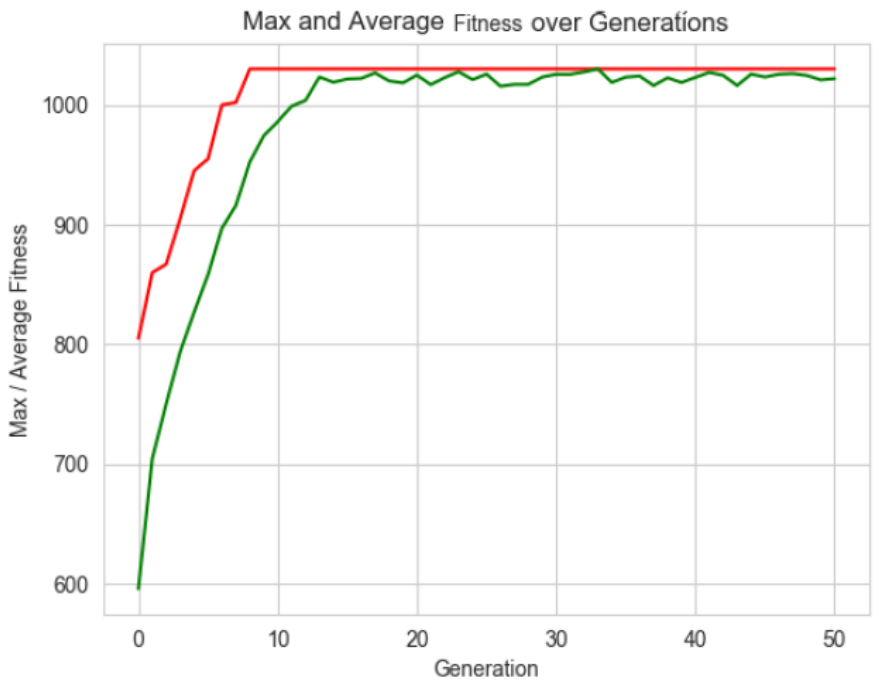
من یک فایل دیگه به نام solve-knapsack.py نوشتم. در این کد، برای ۵۰ نسل با سایز جمعیت برابر با ۵۰، ران گرفتم. چون این مسئله، یک رشته باینری با طول کم بود. در واقع تنظیم کردن پارامترهای این الگوریتم راحت‌تر بود. و تصویر زیر بهترین جوابی بود که بعد از چند تنظیم بدست آوردم .

```
-- Best Ever Individual = [1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 1, 1, 1, 0, 1, 1]
-- Best Ever Fitness = 1030.0
-- Knapsack Items =
- Adding map: weight = 9, value = 150, accumulated weight = 9,
accumulated value = 150
- Adding compass: weight = 13, value = 35, accumulated weight =
22, accumulated value = 185
- Adding water: weight = 153, value = 200, accumulated weight =
175, accumulated value = 385
- Adding sandwich: weight = 50, value = 160, accumulated weight
= 225, accumulated value = 545
- Adding glucose: weight = 15, value = 60, accumulated weight =
240, accumulated value = 605
- Adding banana: weight = 27, value = 60, accumulated weight =
267, accumulated value = 665
- Adding suntan cream: weight = 11, value = 70, accumulated
weight = 278, accumulated value = 735
- Adding waterproof trousers: weight = 42, value = 70,
accumulated weight = 320, accumulated value = 805
- Adding waterproof overclothes: weight = 43, value = 75,
accumulated weight = 363, accumulated value = 880
- Adding note-case: weight = 22, value = 80, accumulated weight
= 385, accumulated value = 960
- Adding sunglasses: weight = 7, value = 20, accumulated weight
= 392, accumulated value = 980
- Adding socks: weight = 4, value = 50, accumulated weight =
396, accumulated value = 1030
- Total weight = 396, Total value = 1030
```

مقدار کلی برابره ۱۰۳۰ هستش که به عنوان راه حل بهینه برای این مسئله شناخته شده. همچنین در این تصویر هم می‌بینیم که آخرین ۱ در کروموزوم بهترین جواب، که به آیتم book اشاره داره، قربانی شده تا وزن کوله از ماکس بیشتر نشه.

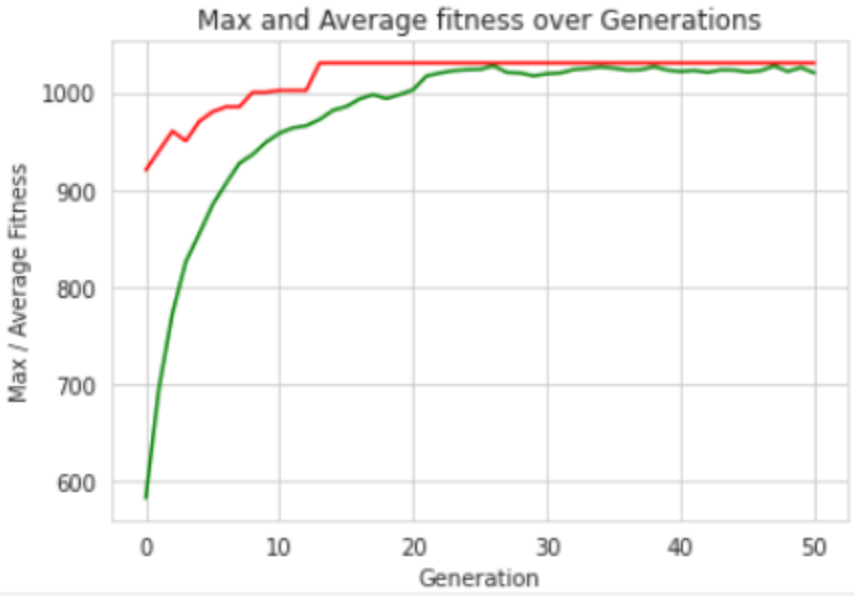
گرافی که فیتنس ماکس و میانگین رو در طول نسل‌ها نشون می‌ده در اینجا آورده شده. همون‌طور که ملاحظه می‌کنید، بهترین جواب در کمتر از ۱۰ نسل پیدا شده.

1.




```
# Genetic Algorithm constants:
POPULATION_SIZE = 50
P_CROSSOVER = 0.9  # probability for crossover
P_MUTATION = 0.1   # probability for mutating an individual
MAX_GENERATIONS = 50
HALL_OF_FAME_SIZE = 1
```

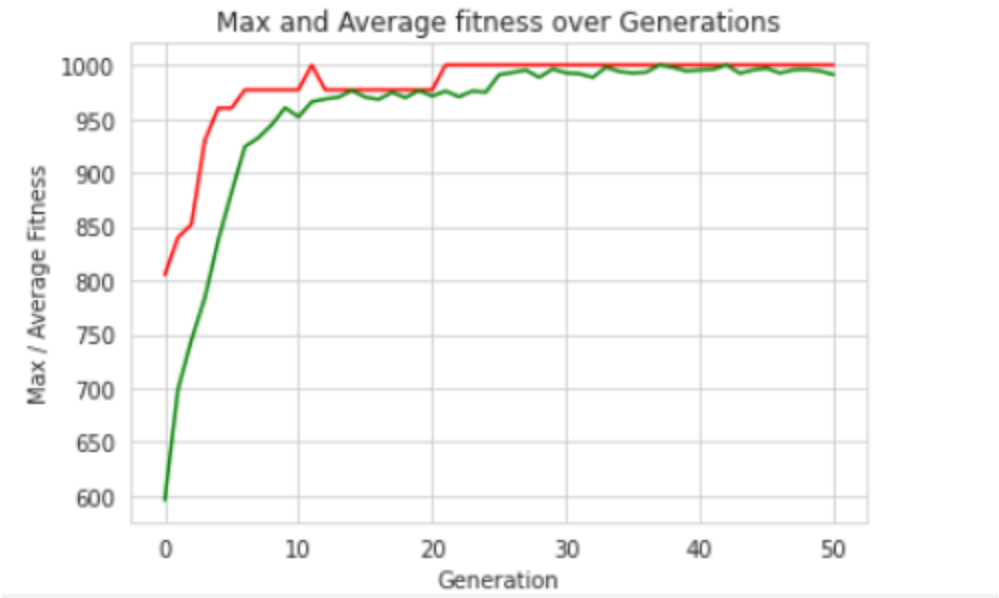
2.




```
# Genetic Algorithm constants:
POPULATION_SIZE = 200
P_CROSSOVER = 0.9  # probability for crossover
P_MUTATION = 0.1   # probability for mutating an individual
MAX_GENERATIONS = 50
HALL_OF_FAME_SIZE = 1
```

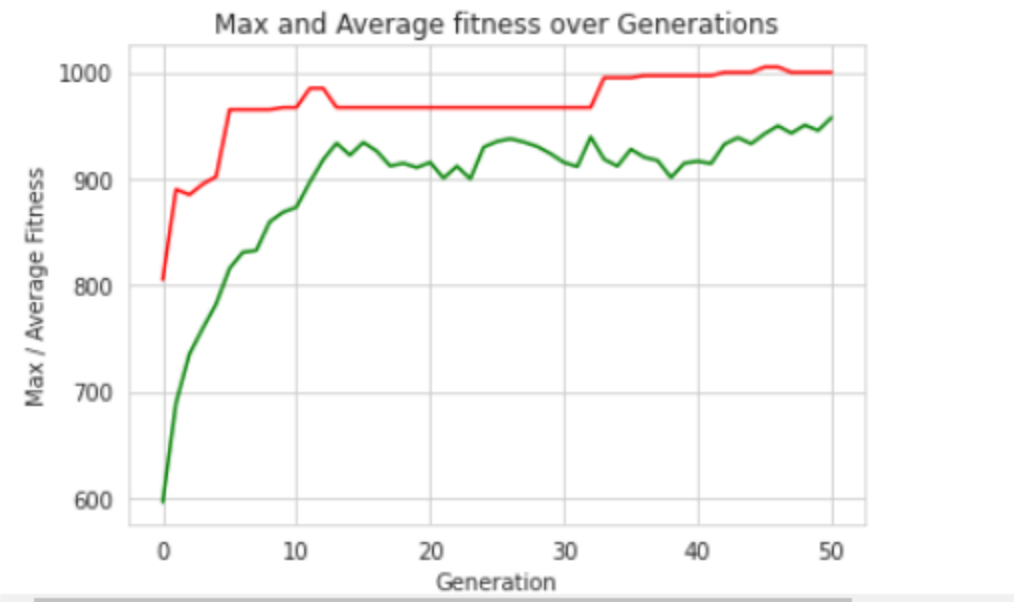
3.

```
# Genetic Algorithm constants:
POPULATION_SIZE = 50
P_CROSSOVER = 0.4  # probability for crossover
P_MUTATION = 0.1   # probability for mutating an individual
MAX_GENERATIONS = 50
HALL_OF_FAME_SIZE = 1
```



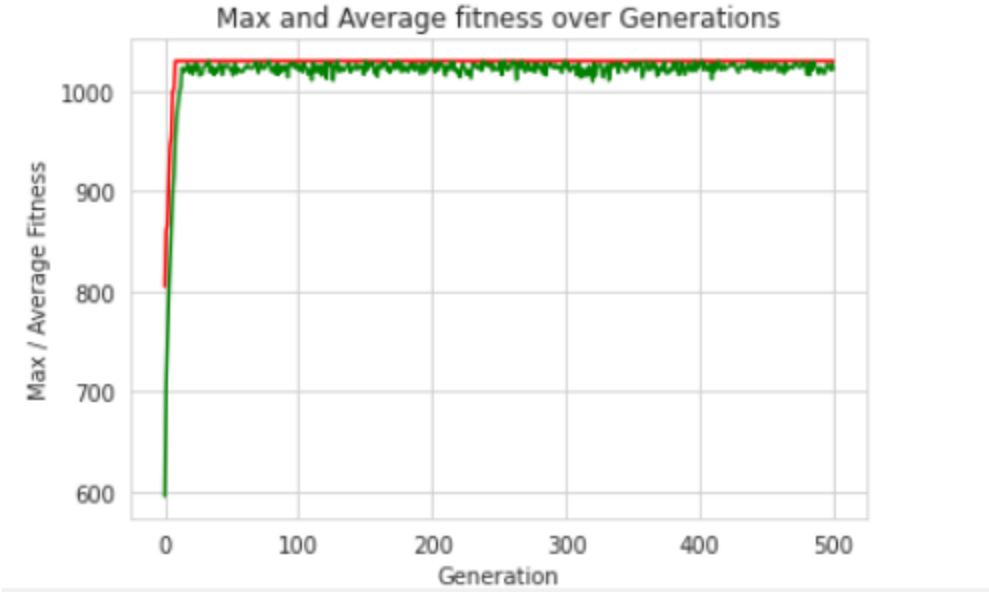
4.

```
# Genetic Algorithm constants:
POPULATION_SIZE = 50
P_CROSSOVER = 0.9  # probability for crossover
P_MUTATION = 0.7   # probability for mutating an individual
MAX_GENERATIONS = 50
HALL_OF_FAME_SIZE = 1
```



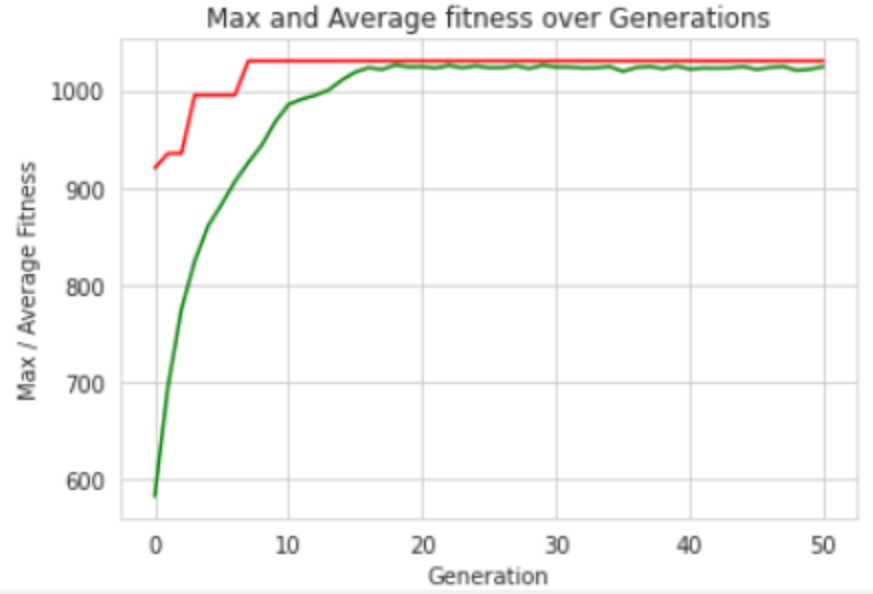
5.

```
# Genetic Algorithm constants:
POPULATION_SIZE = 50
P_CROSSOVER = 0.9  # probability for crossover
P_MUTATION = 0.1   # probability for mutating an individual
MAX_GENERATIONS = 500
HALL_OF_FAME_SIZE = 1
```



6.

```
# Genetic Algorithm constants:
POPULATION_SIZE = 300
P_CROSSOVER = 0.3  # probability for crossover
P_MUTATION = 0.1   # probability for mutating an individual
MAX_GENERATIONS = 50
HALL_OF_FAME_SIZE = 1
```



نکته : نیازی به ران کردن کد نیست، چون قبلا ران شده و خروجی‌ها رو در نوت‌بوک می‌تونین مشاهده کنین.

//با تشکر

//محمد علی‌آبادی

aliabadi4mohammad@gmail.com//