Shiraz University

# گزارش تمرین اول

(پردازش تکاملی)


تهیه شده توسط:

**سارا لیمویی**

استاد:

**دکتر حمزه**


پاییز ۱۴۰۰


بسم الله الرحمن الرحیم

# فهرست مطالب

## توضیحات

در این تمرین می خواهیم مساله ی ۸ وزیر را به کمک genetic algorithm پیاده سازی کنیم. برای این منظور، کلاس GeneticAlgorithm8Queens را داریم که دو تابع ()__init__ و ()loop دارد. در تابع ()__init__، متغیر ها و operator هایی که در صورت سوال تعریف شده اند و به عنوان ورودی به تابع ()__init__ دادیه ایم را انتساب می دهیم. در loop، حلقه ی اصلی الگوریتم genetic پیاده سازی شده است.

هر کدام از operator های پیاده سازی شده در این تمرین در ادامه توضیح داده می شوند:

**Initialization:**

در این مرحله به اندازه ی population-size یک مجموعه از کروموزوم ها به صورت رندوم ولید می کنیم که جمعیت اولیه ما را تشکیل می دهند.

**Parent selection:**

در این مرحله به صورت رندوم ۵ parent را انتخاب کرده و بهترین والد ها، بر اساس fitness-function، را انتخاب می کنیم.

**Crossover:**

در این مرحله با دو والدی که انتهاب کردیم، به روش cut & crossfill یا همان one-point، دو فرزند جدید تولید می کنیم. Cross-over probability را ۱۰۰ درصد در نظر گرفته ایم چون در هر مرحله در این تمرین، باید حتما دو فرزند ایجاد شوند.

**Mutation:**

به احتمال mutation probability، بر روی هر یک از فرزندان ایجاد شده، عملیات mutation انجام می دهیم. برای این کار، یک عدد رندوم بین ۰ تا ۱۰۰ ایجاد کرده و اگر از ۸۰ کوچکتر بود، عملیات را انجام می دهیم.

### Survival selection:

در این مرحله دو تا از بدترین individual های درون جمعیت را با دو فرزند جدید جایگرین می کنیم.

### Termination:

در ادامه چک می کنیم که آیا این دو فرزند جدید جواب می باشند یا خیر. اگر هنوز به جواب نرسیده بودیم، تمام مراحل ذکر شده را با جمعیت جدید تکرار میکنیم. این کار را تا ۱۰۰۰۰ باز تکرار می کنیم.

### حلقه ی اصلی:

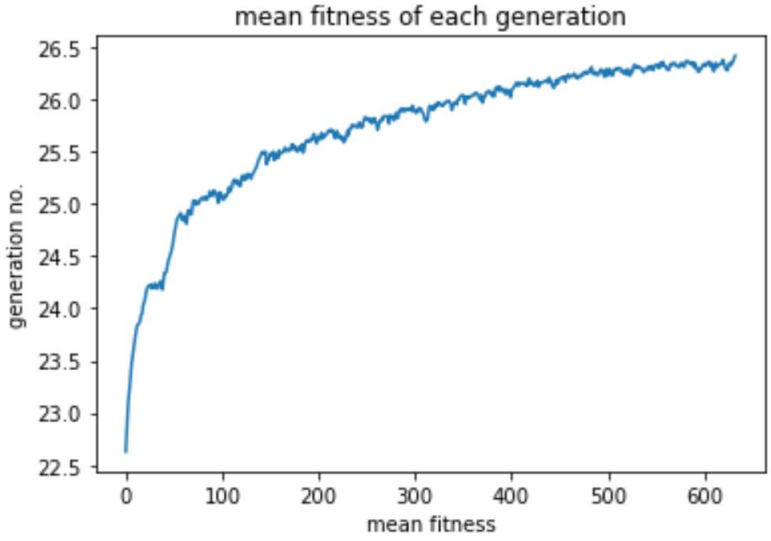
در تابع loop حلقه ی اصلی و اجرای توابع ذکر شده در بالا را می نویسیم که در ادامه شبه کد آن آورده شده است:

```
initialize population
found-ans : check if answer is in initial population
if found-ans:
   break
   return answer
else:
   for i=1 to 10'000
      parent1, parent2 : parent selection
      child1 , child2 : cross-over(parent1, parent2, crossover-probability)

      mutate(child1, mutation-probability)
      mutate(child2, mutation-probability)

      found-ans : check if any new born child is answer or not
      if found-ans:
         break
         return answer
      else
         new-population : survival selection(old-population, child1, child2)
```

**نتایج:**

در این قسمت به اجرای کد و بررسی نتایج بدست آمده می پردازیم. در ابتدا الگوریتم را با پارامتر های population-size=100 و crossover-prob=100 و mutation-prob=80 اجرا می کنیم که در ادامه نتیجه ی آن آورده شده است.

```
Done! Finished 633 fitness evaluation.
chromosoem:  [0, 2, 4, 7, 1, 4, 1, 5]    fitness:  28.0
  ||  0  ||  1  ||  2  ||  3  ||  4  ||  5  ||  6  ||  7  ||
-----------------------------------------------------------
0 ||  Q  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||
1 ||  -  ||  -  ||  -  ||  -  ||  Q  ||  -  ||  Q  ||  -  ||
2 ||  -  ||  Q  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||
3 ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||
4 ||  -  ||  -  ||  Q  ||  -  ||  -  ||  Q  ||  -  ||  -  ||
5 ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  Q  ||
6 ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||  -  ||
7 ||  -  ||  -  ||  -  ||  Q  ||  -  ||  -  ||  -  ||  -  ||
-----------------------------------------------------------
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
```

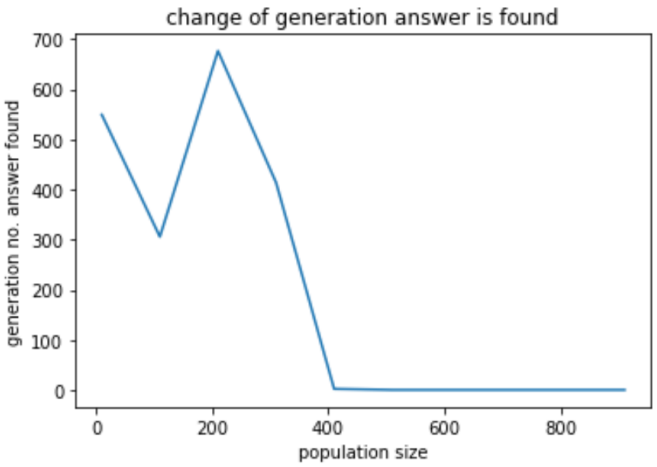


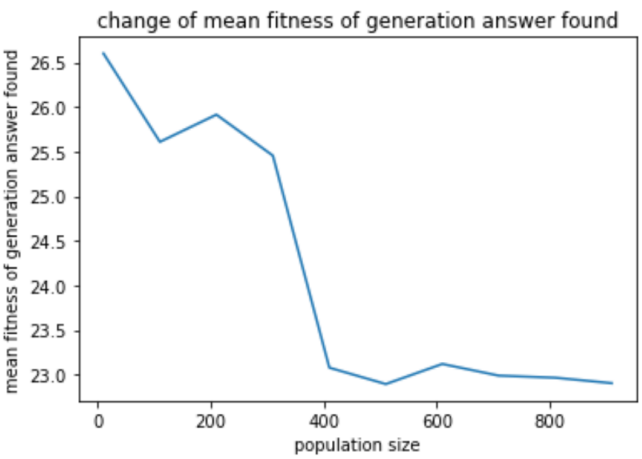
اکنون می خواهیم تاثیر تغییر هر یک از پارامتر های mutation-prob و population-size را بر روی نتیجه ی بدست آورده جواب بررسی کنیم. ابتدا mutation-prob را همان ۸۰٪ ثابت در نظر می گیریم تا تاثیر تغییرات population-size را بر نتیجه برررسی کنیم.

```
Done! Finished 558 fitness evaluation with population-size 10.
chromosoem:  [4, 1, 3, 6, 2, 7, 5, 0]    fitness:  28.0

Done! Finished 390 fitness evaluation with population-size 50.
chromosoem:  [6, 1, 5, 2, 0, 7, 3, 1]    fitness:  28.0

Done! Finished 212 fitness evaluation with population-size 100.
chromosoem:  [5, 3, 6, 0, 7, 4, 1, 5]    fitness:  28.0

Done! Finished 1 fitness evaluation with population-size 500.
chromosoem:  (0, 6, 3, 5, 7, 1, 4, 2)    fitness:  28.0

Done! Finished 1 fitness evaluation with population-size 1000.
chromosoem:  (2, 0, 6, 4, 7, 1, 3, 5)    fitness:  28.0
```

همان طور که در نتایج بدست آمده در بالا مشاهده می کنید، با افزایش تعداد population-size، احتمال اینکه در همان generation اول حواب به صورت رندوم تولید شده باشد بیشتر است.

در ادامه نمودارهای مربوط به تغییرات population-size و تاثیر آن بر generation ای که در آن به نتیجه میرسیم و همچنین میانگین fitness تمام individual های مربوط به generation آخر هر بار تست را مشاهده می کنید.



اکنون population-size را ۱۰۰ ثابت در نظر میگیریم و تاثیر تغییرات mutation-prob را از ۰٪ تا ۱۰۰٪ بررسی میکنم.