موضوع :

**حل مسئله n-وزیر با الگوریتم ژنتیک**

استاد :

خانم دکتر اسدی

تهیه و تنظیم :

سید عمید اسدالهی مجد (941221102)

دانشگاه مازندران

1396

**روند کلی پیاده سازی :**

در نرم افزار برای اجرای الگوریتم ژنتیک به ترتب مراحل زیر انجام شد:

1. Initialize
2. fitness\_measure
3. selection
4. cross over
5. mutation
6. evaluation

برنامه از ۴ ماژول اصلی زیر تشکیل میشود:

۱. **main**:

ماژول اصلی برنامه که تمام قسمت های نرم افزار را به هم متصل کرده و نقطه شروع اجرای نرم افزار با اجرای این ماژول میباشد.

۲. **DNA**:

این ماژول ساختار کروموزم ها را میسازد و وظیفه تولید کروموزم ها را بر عهده دارد.

۳.**population** :

این ماژول با ایجاد جمعیت اولیه شروع شده و شامل تمامی قسمت های اصلی الگوریتم ژنتیک است.

۴.**GUI** :

در این ماژول طراحی رابط کاربری نرم افزار انجام گرفته است .

**کتابخانه های جانبی استفاده شده :**

numpy : محاسبات عددی و آرایه ها

matplotlib : رسم اشکال گرافیکی

PyQt : رابط گرافیکی برنامه

**توابع و کلاس های ماژول main :**

* 1. کلاس GA\_thread که الگوریتم ژنتیک را به صورت موازی با رابط کاربری مدریت میکند و اجرای آن بر عهده این کلاس است.
  2. کلاس Main که رابط گرافیکی را راه اندازی کرده و با کلاس GA\_thread تعامل دارد و پس از دریافت ۴ ورودی از کاربر در ابتدا شروع به بررسی صحت ورودی ها و سپس اجرای الگوریتم میپردازد.

تابع plot\_stuf هم برای رسم و ذخیره شکل نهایی استفاده شده است.

**توابع و کلاس های ماژول population :**

کلاس main که شامل توابع زیر است :

۲-۱. تابع init که شروع کننده کلاس است و با دریافت ۴ ورودی

(dna\_n, pop\_len, mutation\_rate, max\_iter)شروع به کار میکند.

۲-۲. تابع initialize جمعیت اولیه را میسازد.

۲-۳. تابع fitness برای اندازه گیری امتیاز هر کروموزم ورودی تابع میباشد.(از توان دوم امتیاز برای افزایش تاثیرگزاری هر شاخص استفاده شده است.)

۲-۴. تابع fitness\_measure آرایه ای شامل امتیاز هر کروموزم جمعیت کل را میسازد.

۲-۵. تابع cross\_over که برای پیاده سازی عمل cross over الگوریتم ژنتیک است و به روش signle point این کار را انجام میدهد.

۲-۶. تابع mutation که برای پیاده سازی عمل mutation الگوریتم ژنتیک است و با دریافت هر کروموزم به عنوان ورودی به محاسبه یک عدد رندم به ازای هر ژن پرداخته و در صورت کوچکتر بودن این عدد از mut\_rate عمل جایگزینی ژن با یک عدد متناسب مسئله را برای ژن های کروموزم انجام میدهد.

۲-۷. تابع selection در ابتدا اصطلاحا یک استخر از امتیاز های هر کروموزم جمیت با احتمال بالاتر برای امتیازات بالاتر ساخته و سپس دو کروموزم والد به صورت رندم انتخاب و سپس عمل cross over و در ادامه mutation را روی فرزندان جدید انجام داده و نسل جدید ایجاد شده را با جمعیت سابق تعویض میکند.

۲-۸. تابع evaluate که برای بررسی هر نسل ایجاد شده از جهت به وجود آمدن جواب مسئله یا رسیدن به حد نساب نسل ها میباشد.

۲-۹. تابع get\_answer که جواب پیدا شده را برمیگرداند.