

به نام خدا



عنوان: پروژه اول درس هوش محاسباتی

استاد: حسین کارشناس

اعضای گروه:

علی پورقیصری

طاها داوری

محمد امین مولوی زاده

## مقدمه:

در این پروژه، با استفاده از الگوریتم ژنتیک سعی شده تا برپایه یک شبکه ارتباطی بی‌سیم، اطلاعات برپایی دکل‌های مخابراتی برای ارائه خدمات هرچه بهتر به کاربران مشخص و ارائه شود. همچنین نتایج به دست آمده به صورت نمودار در ادامه این گزارش آمده است.

## روش انجام آزمایش:

برای انجام این پروژه، ابتدا یک لیست از کلاس شهر می‌سازیم. در این فرایند فایل‌های مورد نیاز برای اجرای پروژه که شامل دو فایل *blocks\_population.txt* و *problem\_config.txt* می‌باشد که محتویات فایل اول شامل جمعیت هر محله و محتویات فایل دوم شامل هزینه اولیه ساخت هر دکل، هزینه نگهداری هر دکل، سطح رضایت کاربر بر اساس میزان پهنای باند و امتیاز رضایت‌مندی کاربر است. این دو فایل را در کنار کدهای پروژه قرار می‌دهیم و آنها را با استفاده از توابع نوشته شده می‌خوانیم و این اطلاعات را در فیلدهای مربوطه کلاس ذخیره می‌کنیم.

سپس برای هر شهر تعداد تصادفی از دکل‌ها در مکان‌های تصادفی در نظر می‌گیریم. در ادامه یک لیست از برازندگی هر شهر ساخته و برازندگی هر کدام را با توجه به مقادیر تصادفی در نظر گرفته شده حساب می‌کنیم.

حال به تعداد نسل مورد نظر عملیات انجام می‌دهیم به این صورت که ابتدا بسته به ضریب *crossover*، تعداد مشخصی از بهترین والدین انتخاب می‌کنیم و بر روی آنها عملیات *crossover* انجام می‌دهیم و تعدادی فرزند به دست می‌آوریم؛ سپس با توجه به ضریب *mutation*، تعدادی از این فرزندان انتخاب می‌کنیم و بر روی هر کدام عملیات *mutation* انجام می‌دهیم. حال این فرزندان را به لیست والدین که همان کلاسی از لیست شهر است اضافه می‌کنیم و برازندگی هر کدام را حساب کرده و بهترین اعضا را با توجه به تعداد مشخص شده انتخاب می‌کنیم.

این عملیات را به طور مکرر و بسته به میزان خواسته شده انجام می‌دهیم تا بهترین اعضا به دست آیند به طوری که دیگر در نمودار شاهد تغییرات زیادی نباشیم.

## نتایج به دست آمده:

نتیجه بدست آمده از این کد، چهار فایل است به صورتی که در هر فایل مشخصات مورد نیاز پروژه شامل تعداد دکل، مختصات قرارگیری هر دکل در شهر و پهنای باند متعلق به آن، و چگونگی تخصیص هر محله به دکل آمده است.

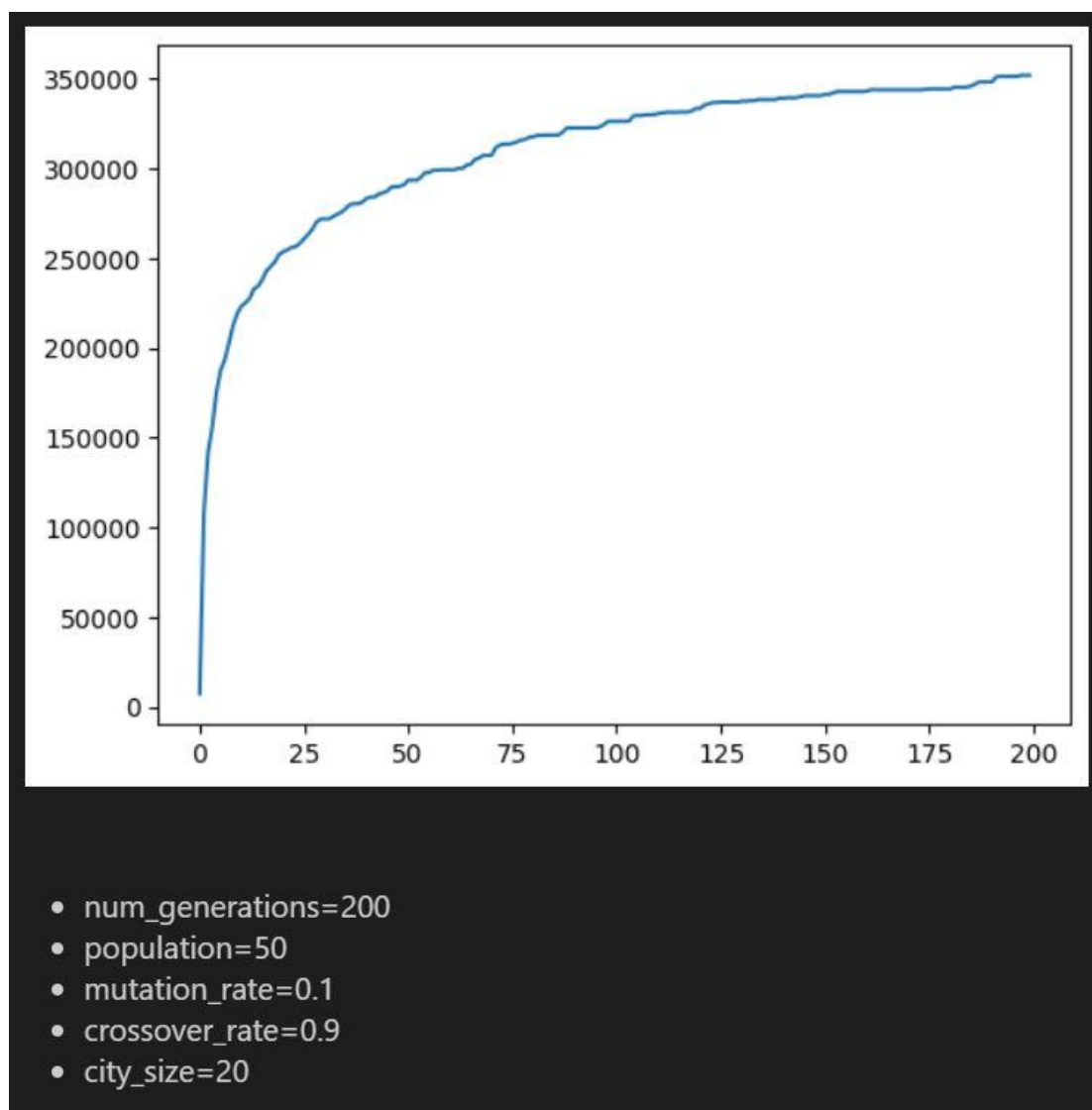
این نتایج به طور حدودی در بهینه‌ترین حالت خود می‌باشند.

همچنین مقدار برازندگی کل با هربار اجرای فرایند به دست آمده که به صورت نمودار نشان داده شده است.

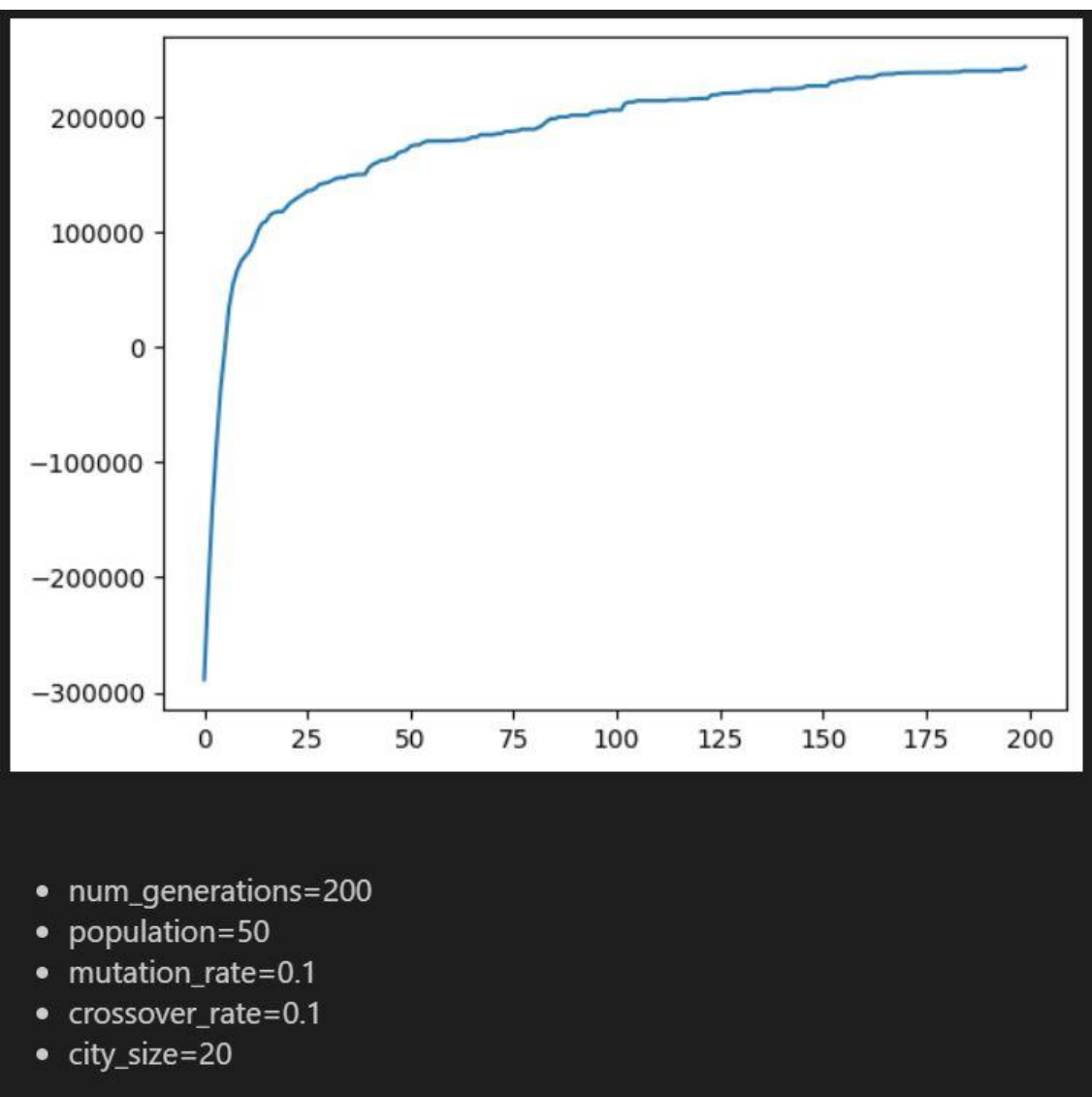
مقایسه این نمودارها در قسمت بعدی آمده است.

## تحلیل نتایج:

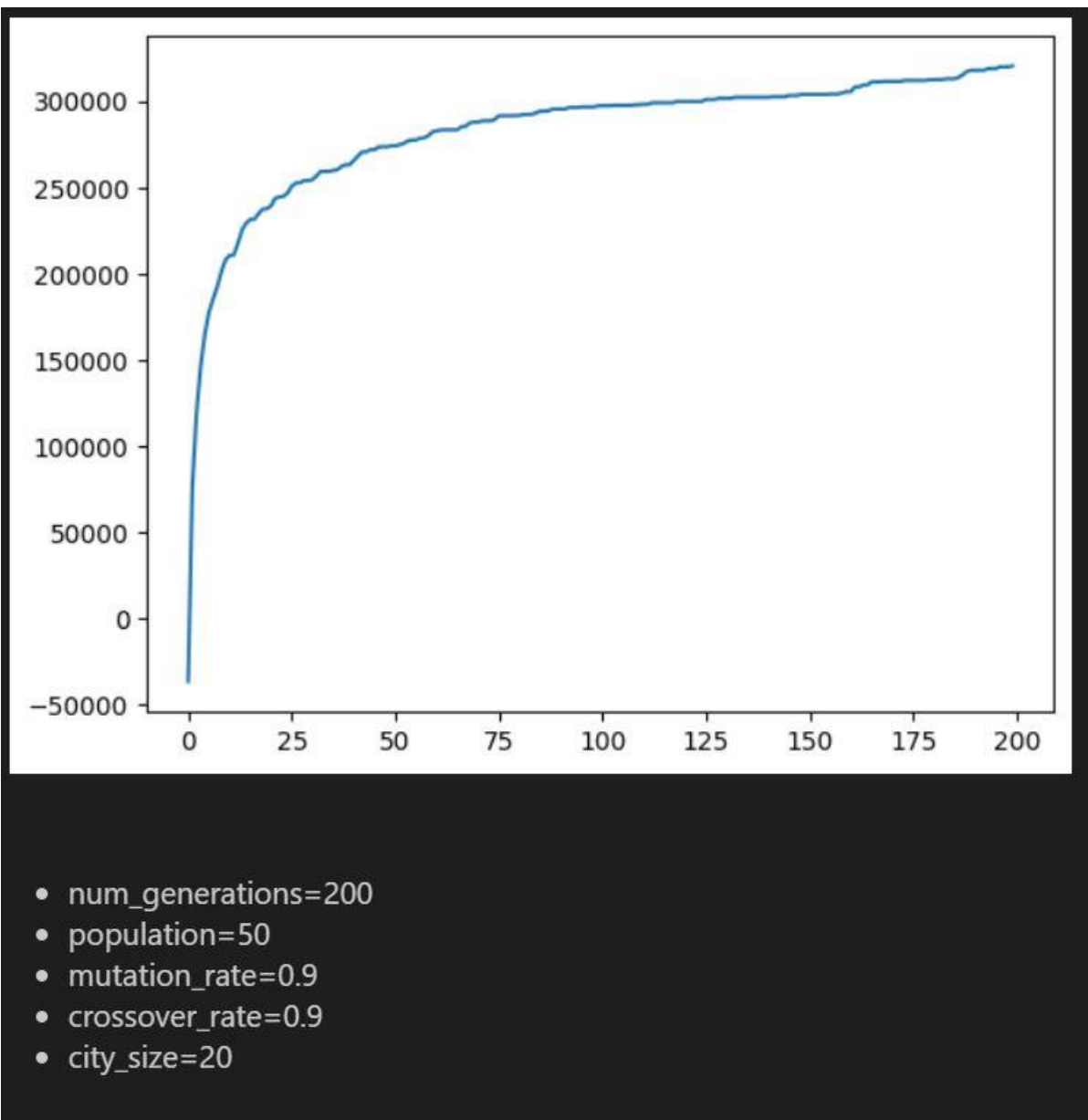
نمودار های به دست آمده از تغییرات برازندگی میانگین در همه نسل ها با توجه به  $mutation\ rate$  و  $crossover\ rate$  به صورت زیر آمده است که مشاهده می شود که  $mutation\ rate = 0.1$  و  $crossover\ rate = 0.9$  بهترین انتخاب است.



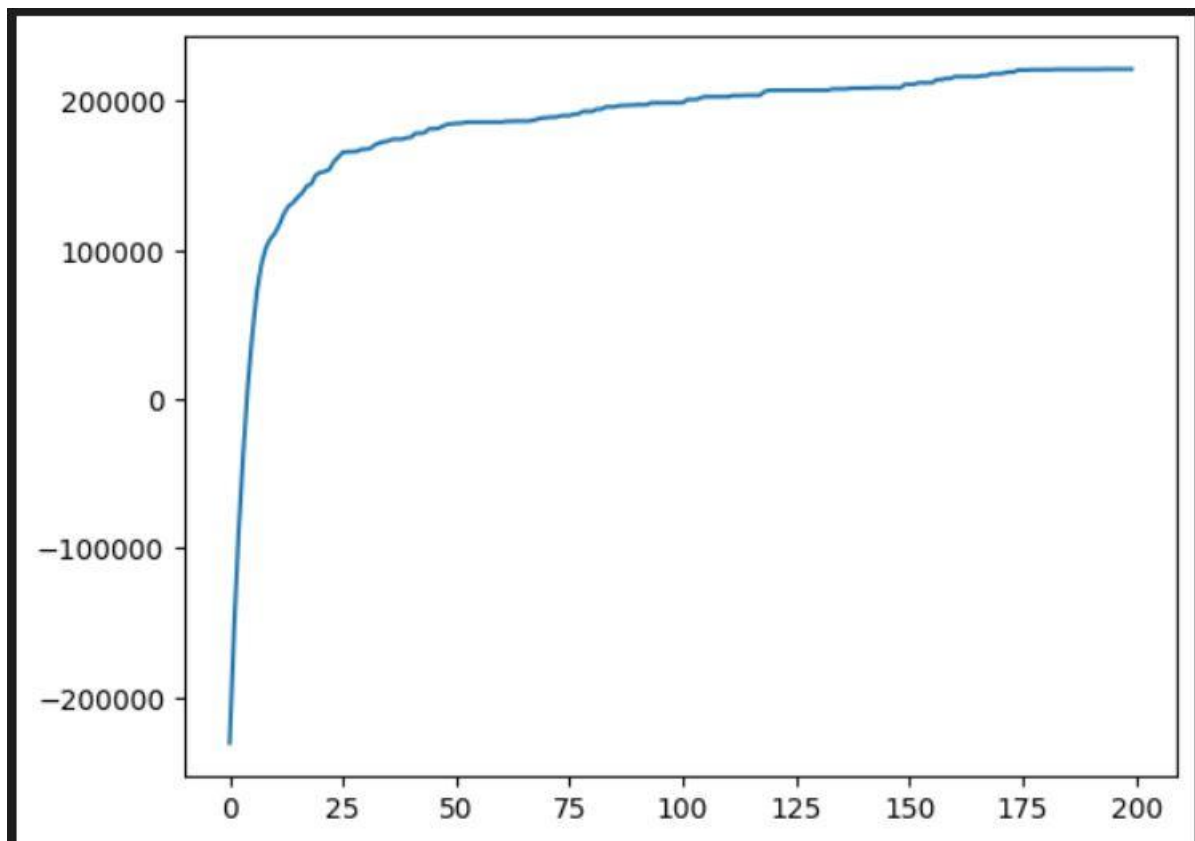
شکل ۱: نمودار تغییرات برازندگی میانگین در نسل های مختلف



شکل ۲: نمودار تغییرات برازندگی میانگین در نسل های مختلف



شکل ۳: نمودار تغییرات برازندگی میانگین در نسل های مختلف



- num\_generations=200
- population=50
- mutation\_rate=0.9
- crossover\_rate=0.1
- city\_size=20

شکل ۴: نمودار تغییرات برازندگی میانگین در نسل های مختلف

شیب همگرایی توابع تقریباً شبیه به هم هستند و می‌بینیم هرچه تعداد نسل‌ها زیادتر می‌شود شیب همگرایی کمتر و کمتر شده تا حدی که می‌توان از تفاوت شیب چشم‌پوشی کرده و الگوریتم را متوقف کرد. مدت زمان اجرای هر یک از الگوریتم‌ها با توجه به *crossover rate* و *mutation rate* متفاوت است. زمان اجرا به ترتیب برابر سه دقیقه و سی و پنج ثانیه، هجده ثانیه، سه دقیقه و هفده ثانیه می‌باشد.

#### پیشنهاد:

در صورت نداشتن محدودیت زمانی، با افزایش *crossover rate* عملکرد الگوریتم بهبود می‌یابد اما مدت زمان اجرا به تبع آن افزایش می‌یابد.

#### منابع:

<https://blog.faradars.org/genetic-algorithm/>

<https://treatta.com/genetic-algorithms-introduction/>

[Machine Intelligence - Lecture 18 \(Evolutionary Algorithms\) - YouTube](#)