دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گروه فناوری اطلاعات

درس کارگاه متلب

عنوان پروژه:

پیاده‌سازی و تحلیل الگوریتم ژنتیک برای مسئله فروشنده دوره گرد

نسخه اول

تهیه‌کننده:

میلاد محمدی

نام استاد درس:

دکتر حمیدرضا حر

نیم‌سال اول تحصیلی ۱۴۰۰-۱۴۰۱

فهرست مطالب:

چکیده:

هدف اصلی این برنامه پیاده‌سازی الگوریتم ژنتیک برای حل مسئله فروشنده دوره گرد است. در مسئله فروشنده دوره گرد، مجموعه‌ای از شهرها (رئوس گراف) و جاده‌ها (یال‌های گراف) با وزن و هزینه متفاوت وجود دارند و هدف پیاده کردن کوتاه‌ترین مسیری است که از هر شهر یکبار و فقط یکبار گذر کند و به شهر اولیه بازگردد.

برای حل این مسئله الگوریتم‌های مختلفی از جمله برنامه‌نویسی پویا (Dynamic Programming) وجود دارد. اما در این برنامه ما از الگوریتم تکاملی ژنتیک برای حل این مسئله استفاده کرده‌ایم.

این برنامه چند بخش اصلی شامل دریافت گراف اولیه، دریافت پارمترهای الگوریتم ژنتیک، انتخاب جمعیت اولیه، محاسبه برازندگی جمعیت و مرتب‌کردن جمعیت براساس برازندگی، انتخاب والدها، تولید و انتخاب فرزندان برای ساخت جمیعیت جدید و سپس تکرار این عملیات است.

در این سند به طور خلاصه به کارکرد کلی این الگوریتم و شیوه پیاده‌سازی آن پرداخته شده است. و هدف از نوشتن هر بخش آورده شده است. همچنین پیشنهاداتی در جهت بهبود هر قسمت برای تکمیل و نوشتن ورژن‌های بعدی آورده شده است.

پس از پایان معرفی کد، بخش تحلیل الگوریتم ژنتیک آورده شده که به بررسی کارکرد الگوریتم با تغییر پارامترها و محاسبه شاخص‌های آماری اختصاص یافته.

در پایان نیز یادگیری‌های از درس آز متلب و برنامه‌نویسی در محیط متلب به همراه پیشنهاداتی آورده شده است. و منابع استفاده شده جهت برنامه‌نویسی این برنامه نوشته شده.

معرفی بخش‌های مختلف پروژه:

در این بخش به معرفی فایل‌های مختلف کد پروژه و توابع پیاده‌سازی شده و استفاده شده هرکدام خواهیم پرداخت. هدف از این بخش تشریح کارکرد کلی هر بخش است و از بیان جزئیات پیاده‌سازی صرف نظر شده است. به جز موارد به خصوصی که عدم توضیح آن می‌تواند باعث به وجود آمدن ابهام یا سوال شود.

جدول توضیح خلاصه توابع پیاده‌سازی شده:

در جدول زیر توابع پیاده‌سازی شده به شکل فهرست‌وار به همراه یک توضیح مختصر آورده شده است.

به دلیل اینکه سعی شده در پیاده‌سازی این برنامه از Design Pattern مهم Single Responsibility استفاده شود، تعداد توابع پیاده‌سازی شده به نسبت تعداد زیادی دارند و این جدول می‌تواند راهنمای خوبی برای فهم کد نوشته شده باشد.

نام تابع	توضیح
GetCitiesLocations	دریافت مختصات شهرها (رئوس گراف)
CalcCitiesDistances	محاسبه فاصله شهرها (یال‌های گراف)
GetParameters	دریافت پارمترهای الگوریتم ژنتیک
GeneratePopulation	تولید جمعیت اولیه
GetStatus	دریافت وضعیت Crossover و Mutation در هر دور
CalcFitness	محاسبه Fitness یک جمعیت
CalcSingleFit	محاسبه Fitness یک مسیر
GetBestRoadFit	دریافت بهترین مسیر هر جمعیت و بهترین مسیر کنونی
visualizeGeneration	مصور سازی بهترین مسیر هر دور
Top20per	دریافت ۲۰ درصد برتر جمعیت کنونی (نخبگان)
Selection	انتخاب والدها
CreateCrossovers	تولید و انتخاب فرزندها
CrossOver2Par	تولید فرزندان از دو والد
CrossOver3Par	تولید فرزندان از سه والد
Combine	ساخت جمعیت از تلفیق نخبگان و فرزندان انتخاب شده
Mutation	جهش بر روی جمعیت

بخش مشخص‌کردن نقاط و فاصله‌ها (طرح مسئله):

در این بخش ابتدا از طریق تابع اول، نقاط گراف را دریافت می‌کند. در حالت درست باید نقاط را از طریق فایل دریافت کند اما اکنون به صورت دستی در ۴ نمونه با تعداد و حالت‌های مختلف نوشته شده است.

در تابع دوم فرض می‌شود بین هر دو شهر یک مسیر وجود دارد و فاصله اقلیدسی آنها را حساب می‌کند و در یک ماتریس n در n ذخیره می‌کند. اما در حالت درست باید وجود یا وجود نداشتن مسیر بین دو راس گراف و همچنین وزن آن مسیر از طریق فایل دریافت شود. (در غیر این صورت می‌توان گفت همیشه کوتاه‌ترین مسیر، مسیر محیط گراف خواهد بود.)

مشخص‌کردن پارامترهای الگوریتم ژنتیک:

از طریق یک تابع ساده پارمترهای الگوریتم ژنتیک دریافت می‌شود. این پارمترها شامل:

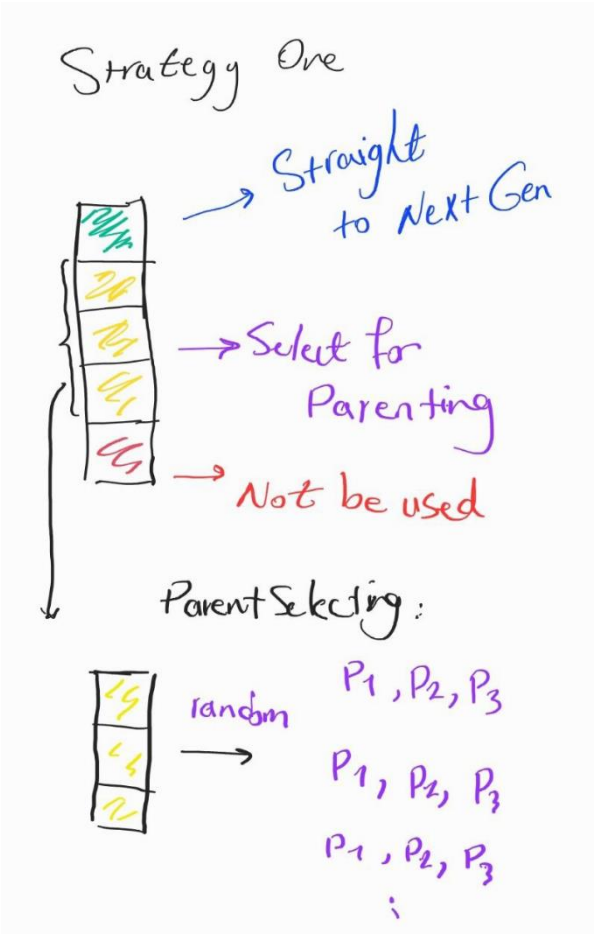
- تعداد جمعیت اولیه و هر دور
- تعداد تولید نسل‌ها
- نرخ CrossOver
- نرخ رخداد Mutation
- نرخ تعداد Mutation

همچنین پارمترهایی مثل تعداد نخبگان جمعیت براساس پارمترهای بدست آمده، محاسبه می‌شوند.

الگوریتم ژنتیک بخش صفرم - جمعیت اولیه:

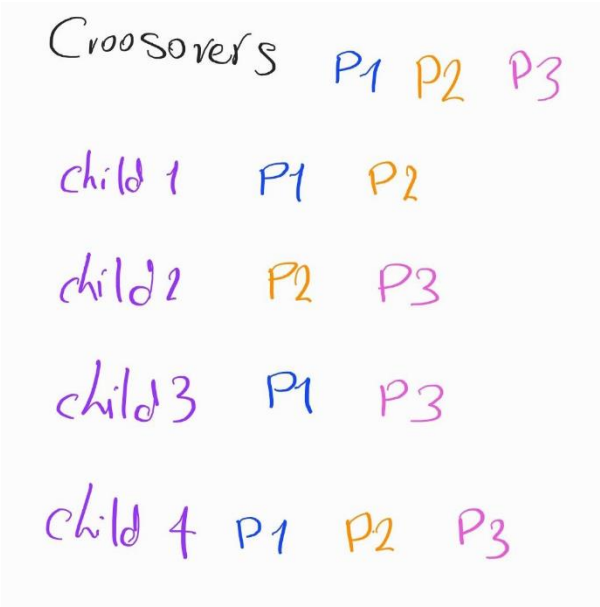
الگوریتم ژنتیک بخش اول – محاسبه برازندگی فعلی:

الگوریتم ژنتیک بخش دوم – انتخاب والدها:

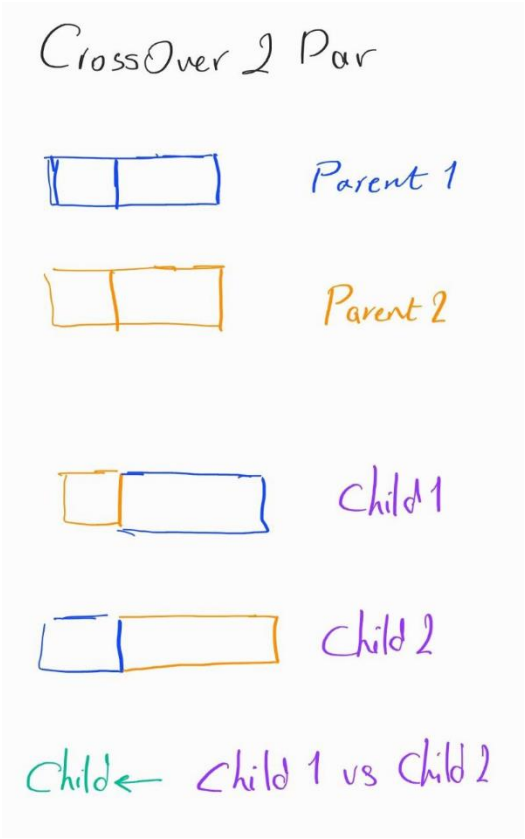


پس دریافت پارمترهای اولیه و قبل از وارد شدن به حلقه اصلی الگوریتم ژنتیک یک متغییر repeat_num و یک حلقه تعریف شده است. این بخش از برنامه برای این است که باتوجه به ماهیت تصادفی بودن الگوریتم ژنتیک بتوان با استفاده از پارمترهای یکسان، چندین بار الگوریتم را اجرا و در هر دفعه در یک ماتریس بهترین fitness بدست آمده و مدت زمان اجرا الگوریتم ذخیره شود و سپس در انتها میانگین و انحراف معیار بدست آید.

الگوریتم ژنتیک بخش سوم – انتخاب تولید / انتخاب فرزندان:

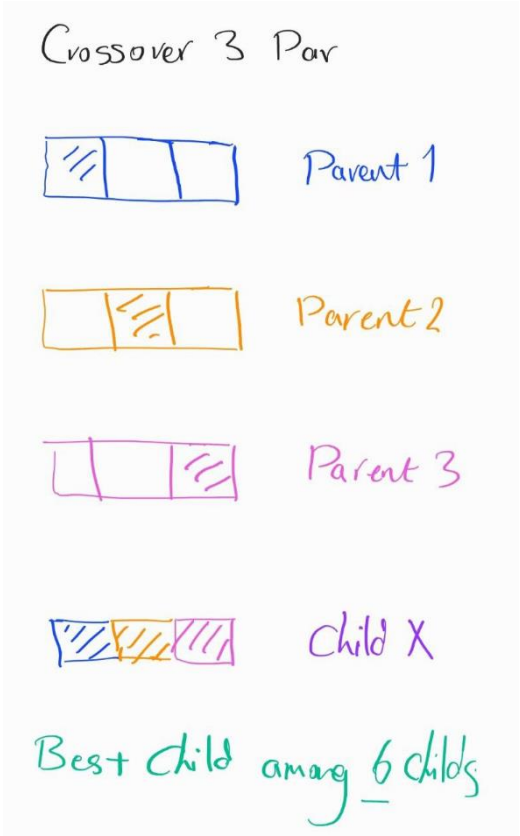


ابتدا توسط تابع Selection مجموعه‌ای تصادفی از سه‌والد انتخاب می‌شود. سپس از هر سه والد ما در انتها چهار فرزند خواهیم داشت. که فرزند اول فرزند برتر تولید شده از والد اول دوم، فرزند دوم فرزند برتر تولید شده از والد دوم و سوم، فرزند سوم فرزند برتر تولید شده از والد اول و سوم و فرزند آخر فرزند تولید شده از هر سه والد است.



برای تولید فرزند از دو والد، به شیوه توضیح داده شده در درس اکتفا کردیم. البته با یک پیاده‌سازی طولانی‌تر می‌توانستیم از تابع any و با کمک if پیاده‌سازی را انجام دهیم.

از هر دو والد دو فرزند تولید می‌شود، که سپس برازندگی فرزندان محاسبه و فرزند با برازندگی بهتر به عنوان خروجی انتخاب می‌شود.



برای تولید آخرین فرزند ما از CrossOver سه والد و تمام جایگشت آنها که باعث تولید ۶ فرزند می‌شود استفاده می‌کنیم.

برای Sliceing هم نیاز است تا دو نقطه انتخاب شود. یک نقطه قبل از میانه و یک نقطه بعد از آن.

و سپس پیاده‌سازی آن انجام شود، تمام فرزندان در یک ماتریس نگهداری شوند و سپس بهترین فرزند در بین آنها از لحاظ برازندگی (پس از محاسبه برازندگی انتخاب شود.)

[پیاده‌سازی این بخش تکمیل نشده است.]

الگوریتم ژنتیک بخش چهارم – جهش:

جهش به سادگی با sawp کردن دو نقطه از یک مسیر تصادفی انجام می‌شود.

تحلیل الگوریتم:

پس دریافت پارمترهای اولیه و قبل از وارد شدن به حلقه اصلی الگوریتم ژنتیک یک متغییر repeat_num و یک حلقه تعریف شده است. این بخش از برنامه برای این است که باتوجه به ماهیت تصادفی بودن الگوریتم ژنتیک بتوان با استفاده از پارمترهای یکسان، چندین بار الگوریتم را اجرا و در هر دفعه در یک ماتریس بهترین fitness بدست آمده و مدت زمان اجرا الگوریتم ذخیره شود و سپس در انتها میانگین و انحراف معیار بدست آید.

بهترین برازندگی بدست آمده در هر اجرای الگوریتم ژنتیک و زمان اجرا برنامه در یک ماتریس دوبعدی ذخیره می‌شود.

یافته‌ها:

در این قسمت به طور خلاصه و کلی به یافته‌های درس و انجام این پروژه پرداخته شده است.

چالش‌ها الگوریتم ژنتیک:

مزایای و معایب برنامه‌نویسی در متلب:

پیشنهاداتی برای درس:

مراجع:

1. ویدیوهای تدریس شده توسط استاد درس برای حل مسئله مربع جادویی
2. https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35
3. https://www.geeksforgeeks.org/traveling-salesman-problem-using-genetic-algorithm/
4. https://github.com/zahidkizmaz/TSP-Genetic-Algorithms