

دانشگاه آزاد اسلامی

واحد تهران شمال

دانشکده برق و کامپیوتر

پروژه پایانی کارشناسی کامپیوتر گرایش نرم افزار

عنوان

**زمانبندی کلاس ها دانشکده با استفاده از الگوریتم ژنتیک**

استاد راهنما:

دکتر سروش مبشری

تهیه و تنظیم :

سید علی مجتبوی – امیر حسین کفاشی – امیر رضا فصیحی راد

خرداد 1401



سپاسگزاری

پروردگارا مرا یاری کن تا دانش اندکم نه نردبانی باشد برای فزونی و تکبر و غرور، نه حلقهایی برای اسارت و نه دستمایهایی برای تجارت، بلکه گامی باشد برای تجلیل از تو و تعالی ساختن زندگی خود و دیگران.  
قبل از هر چیز، خداوند بزرگ را به خاطر لطفی که همواره شامل حال من نموده شاکرم. سپس، از زحمات استاد  
محترم راهنما، جناب آقای دکتر سروش مبشری که نه تنها به عنوان استاد بلکه همچون همکاری در تمام مراحل  
انجام این تحقیق از رهنمودها و کمک های بیدریغ ایشان بهرهمند شده ایم، به ویژه به خاطر ساعتهای طولانی که به بحث و تبادل نظر در مورد موضوع تحقیق بنده اختصاص داده اند که همواره برای ما­ الهامبخش ایده و دیدگاهی تازه نسبت به موضوع بوده است، تشکر و قدردانی میکنیم.

#### **فهرست مطالب**

عنوان صفحه

[چکیده **Error! Bookmark not defined.**](#_Toc109291410)

[مقدمه 9](#_Toc109291411)

[پیکره 11](#_Toc109291412)

[فصل اول: چارچوب مفهومی نرم افزار 12](#_Toc109291413)

[فصل دوم: اجزای تشکیل دهنده سیستم: 14](#_Toc109291414)

[الگوریتم 14](#_Toc109291415)

[رابط کاربری 15](#_Toc109291416)

[دیتابیس 15](#_Toc109291417)

[مدیریت سناریو 16](#_Toc109291418)

[فصل سوم: ساختار پروژه 17](#_Toc109291419)

[دایرکتوری Components 17](#_Toc109291420)

[دایرکتوری Containers 17](#_Toc109291421)

[دایرکتوری py\_ui 17](#_Toc109291422)

[فایل gas.db 18](#_Toc109291423)

[فایل requirements.txt 18](#_Toc109291424)

[فایل settings.json 18](#_Toc109291425)

[فایل timeslots.json 18](#_Toc109291426)

[فصل چهارم : ساختار ساخت زمانبندی 19](#_Toc109291427)

[کتابخانه های مورد نیاز الگوریتم 19](#_Toc109291428)

[کلاس کروموزم در الگوریتم ژنتیک 20](#_Toc109291429)

[کلاس GeneticAlgorithm 26](#_Toc109291430)

[متد selectRoom 31](#_Toc109291431)

[متد selectInstructor 31](#_Toc109291432)

[متد selectTimeDetails 32](#_Toc109291433)

[تابع ارزیابی 33](#_Toc109291434)

[تنظیم جمعیت 36](#_Toc109291435)

[تنظیم نرخ جهش 37](#_Toc109291436)

[انتخاب جمعیت هدف 38](#_Toc109291437)

[جفت گیری و تولید نسل جدید 40](#_Toc109291438)

[جهش 41](#_Toc109291439)

[دامنه و محدودیت 41](#_Toc109291440)

[خروجی نرم افزار 42](#_Toc109291441)

[تنظیمات الگوریتم ژنتیک 42](#_Toc109291442)

[فصل پنج : رابط گرافیکی 44](#_Toc109291443)

[ساختار کلی: 44](#_Toc109291444)

[پنجره اصلی: 44](#_Toc109291445)

[نمایش داده ها: 44](#_Toc109291446)

[ستون عملیات 45](#_Toc109291447)

[ستون دردسترس 45](#_Toc109291448)

[جدول زمانی 45](#_Toc109291449)

[اساتید 46](#_Toc109291450)

[کلاس ها 47](#_Toc109291451)

[درس ها 48](#_Toc109291452)

[گروه ها 50](#_Toc109291453)

[مدریت سناریو 52](#_Toc109291454)

[ایجاد 52](#_Toc109291455)

[ساختار کد 53](#_Toc109291456)

[ابزار های مورد استفاده 53](#_Toc109291457)

[طراحی GUI 53](#_Toc109291458)

[اجزای GUI 54](#_Toc109291459)

[Py\_ui 54](#_Toc109291460)

[Containers 54](#_Toc109291461)

[Instructor 55](#_Toc109291462)

[Subject 62](#_Toc109291463)

[Room 66](#_Toc109291464)

[Section 66](#_Toc109291465)

# مقدمه

امروزه یکی از سخت ترین مشکلاتی است که دانشگاه ها و کالج ها با آن مواجه هستند، مسئله زمان بندی کلاس های دانشگاه (UCSP) است، که یک مسئله بهینه سازی بسیار محدود ترکیبی، در دنیای واقعی است. حل UCSP به معنای ایجاد یک برنامه بهینه با اختصاص دوره ها به اتاق های خاص، مدرسان، دانش آموزان و بازه های زمانی خاص با در نظر گرفتن محدودیت های داده شده است. UCSP یک مسئله زمان‌بندی کلاسیک است و-NP کامل در نظر گرفته می‌شود. مسئله های زمان بندی باید دو نوع محدودیت را برآورده کند، یعنی محدودیت‌های سخت و نرم که در آن محدودیت‌های سخت شرایطی هستند که برای یک جدول زمانی کاری حتما باید رعایت شوند، در حالی که محدودیت‌های نرم شرایطی هستند که ممکن است نقض شوند اما بر کیفیت راه‌حل تأثیر می‌گذارند.

برنامه ریزی کلاس ها در یک موسسه یکی از کارهایی است که می تواند تحت دسته بندی "مدیریت عملیات" طبقه بندی   
می شود. اهداف مدیریت عملیات برای به حداکثر رساندن کارایی در یک زمنیه مشخص استفاده خواهد شد. برنامه ریزی کلاس ها اغلب توسط مدیر گروه (نیروی انسانی) انجام می شود. تا کارایی لازم را داشته باشد. ولی این به معنی بی نقص بودن برنامه نیست. استفاده از کامپیوتر برای یافتن بهترین راه حل برای این مسئله ها با استفاده از روش های مرسوم بسیار ناکارآمد است. بنابراین کامپیوتر های برای مدتی برای حل این مسئله کنار گذاشته شده بودند با این حال، افزایش قدرت محاسبه کامپیوتر ها و استفاده از الگوریتم های جدید، راهی را برای حل مسئله های زمانبندی باز کرده است.

مدیریت عملیات اغلب به عنوان ستون فقرات و شاکله بسیاری شرکت ها از در نظر گرفته می شود. به دست آوردن کارآمدترین عملکرد به معنای سود بیشتر است. در یک سناریوی مؤسسه آموزشی، به دست آوردن کارآمدترین برنامه زمانبندی، نه تنها به کاهش هزینه ها کمک می کند، بلکه عمدتاً به دانشجو کمک می کند. در بیشتر مواقع برنامه مناسب و کارآمد به دانشجو ها و مربیان کمک می کند تا کنترل بهتری بر زمان خود داشته باشند. ایجاد برنامه زمانی توسط انسان ها بهتر از روش های تکنیک های محاسباتی معمول انجام می شود زیرا انسان ها دارای مغز قدرتمندی برای ارزیابی محدودیت ها و ترکیب ها هستند.

با این حال، مسئله های زمان‌بندی به عنوانnon-deterministic polynomial time (NP) شناخته می‌شود. که بهترین راه حل برای آنها هنوز پیدا نشده است و برای یافتن بهترین آن، باید هر ترکیب ممکن را اجرا کرد. انسان ها نمی توانند هر ترکیب ممکن را محاسبه کنند و بنابراین با پر کردن و اطمینان از رعایت محدودیت ها مشکل را حل می کنند. این راه حل ها به اسم "به اندازه کافی خوب[[1]](#footnote-1)" شناخته شده است. این فرآیند مستعد خطا، ناکارآمدی و نقض محدودیت‌ها است، به‌ویژه زمانی که در یک سناریوی بسیار فشرده کار می‌کنید. تکرار روش انسانی برای حل مسئله (الگوریتم Greedy) در رایانه به دلیل هزینه محاسباتی آن بسیار ناکارآمد است.

کامپیوترها به ندرت برای حل این مشکل استفاده می شوند، زیرا غیر قابل اعتماد و ناکارآمد هستند. با این حال، این به زمانی برمی گردد که کامپیوترها کند بودند و برای حل مشکلات محاسباتی در نظر گرفته نمی شدند. با انفجار کلان داده ها [[2]](#footnote-2)و علوم داده [[3]](#footnote-3)و افزایش تصاعدی قدرت محاسباتی، استفاده از رایانه ها برای حل مسائل مجموعه های جدیدی که قبلاً فقط توسط انسان قابل حل بود، آسان شد. زمان بندی یکی از مسائلی است که هم اکنون توسط رایانه ها با راه حل های قابل قبول یا حتی بهتر قابل حل است. هوش مصنوعی امروزه به دلیل رشد صنعت محاسبات رایج تر شده است. چندین راه حل مانند یادگیری ماشینی، یادگیری عمیق و الگوریتم های ژنتیک تحت هوش مصنوعی قرار می گیرند. همه این فرآیندها مزایا و معایب خاص خود را دارند. الگوریتم ژنتیک برای مسئله زمانبندی مناسب است زیرا در طول زمان بر اساس قوانین و معیارها راه حلی جدیدی ایجاد می کند.

در دانشگاه آزاد اسلامی ، مدیر گروه مسئول ایجاد زمان‌بندی است. زمان بندی به صورت نیمه دستی با کمک کامپیوتر انجام می شود. این فرآیند هنوز نمی تواند راه حلی ارائه دهد که مطمئن باشد از خطاها، نتایج بد ظاهری و نقض محدودیت ها عاری باشد. با شرایط فعلی استفاده از کامپیوتر برای کمک به حل مشکل زمان بندی می تواند یکی از گزینه ها باشد. هدف این پروژه ایجاد یک راه حل نرم افزاری کاندید با استفاده از **الگوریتم های ژنتیک** برای تولید زمان بندی است.

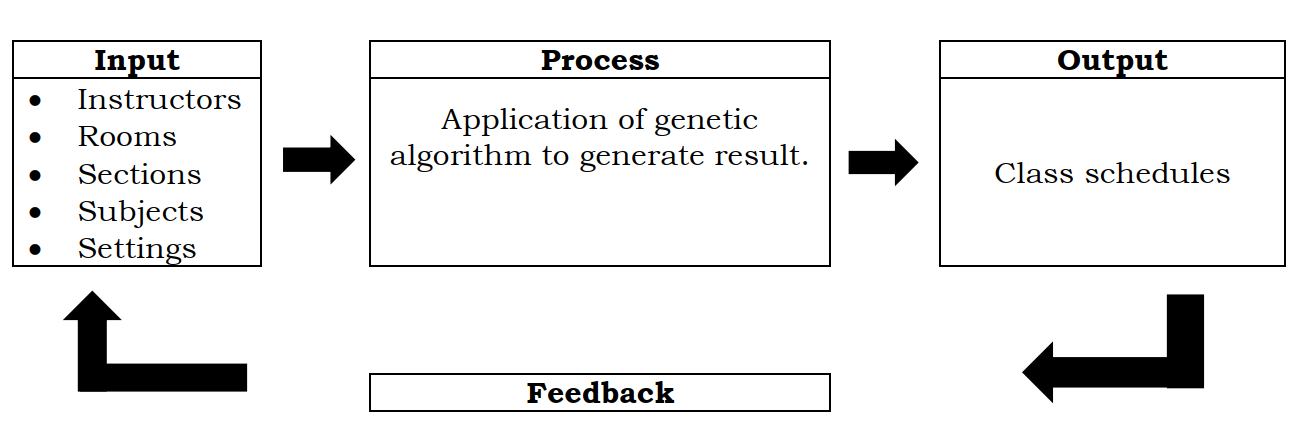
# پیکره

برنامه ریزی دانشگاه آزاد اسلامی به صورت نیمه دستی با کمک کامپیوتر انجام می شود که می تواند خطاها، نقض محدودیت ها و ناکارآمدی ایجاد کند مسئله را می توان به تقسیم کرد؛

* زمان‌بندی هنوز به صورت نیمه دستی با اپراتور انجام می‌شود که این فرآیند به بسیار شبیه الگوریتم حریصانه است که می‌تواند نتایجی با خطا ایجاد کند.

# فصل اول: چارچوب مفهومی نرم افزار

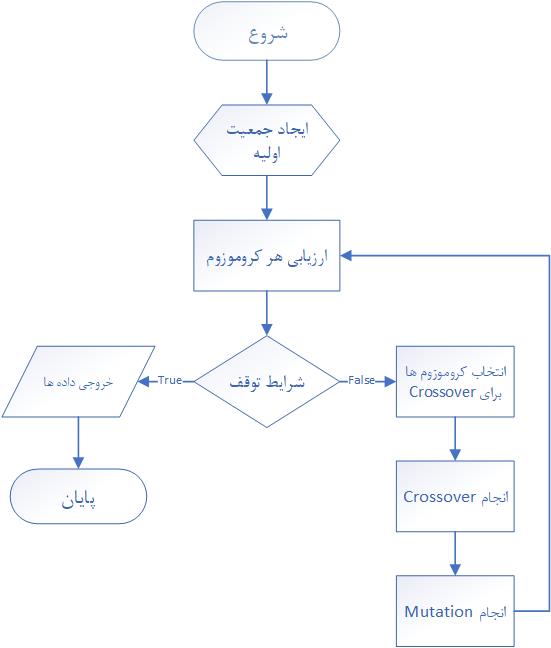
این برنامه را می توان به راحتی به عنوان یک نرم افزار محاسباتی زمان بندی توصیف کرد که در آن مجموعه داده های اولیه را وارد می کنید و یک نتیجه ساختاریافته به دست می آورید. نمودار جریان داده نرم افزار به اینصورت خواهد بود:



الگوریتم ژنتیک یک روش محاسباتی است که انتخاب طبیعی را انجام می دهد که فرآیندی در تکامل بیولوژیکی است. این فرآیند تکراری است و با استفاده از نمودار جریان به بهترین شکل مفهوم سازی می شود. در الگوریتم ژنتیک، به یک محلول به جای انفرادی، کروموزوم گفته می شود. شکل زیر فرآیند اصلی الگوریتم ژنتیک را نشان می دهد اما در عمل به سیستم، مرحله دیگری پس از ارزیابی اضافه می شود که تنظیم محیط[[4]](#footnote-4) است.

تولید جمعیت باید به طور تصادفی با ترکیبی از رویکرد حریصانه انجام شود که در آن یک نقطه تصادفی انتخاب می‌شود و به عنوان نقطه شروع عمل می‌کند و سپس شروع به پر کردن جدول با هر موجودی مناسب می‌کند. در طول تولید جمعیت، همه محدودیت‌های سخت باید برآورده شوند. محدودیت‌های متوسط ​​چندین بار برای پیگیری انجام می‌شوند در حالی که محدودیت‌های نرم کاملاً نادیده گرفته می‌شوند.

ارزیابی[[5]](#footnote-5)، فرآیند محاسبه تناسب[[6]](#footnote-6) یک کروموزوم است. فیتنس معیار کیفیت کروموزوم نسبت به راه حل مورد نظر است. همچنین فرآیند با رسیدن به مقدار مشخصی فیتنس پایان می یابد. محاسبه فیتنس با ارزیابی هر کروموزوم به محدودیت‌های متوسط ​​و نرم انتخاب شده به‌ویژه محل قرارگیری همه ی واحد ها[[7]](#footnote-7)، استراحت ناهار انجام می‌شود.



# فصل دوم: اجزای تشکیل دهنده سیستم:

سیستم از چهاربخش اصلی تشکیل شده است

1. الگوریتم
2. رابط کاربری
3. دیتابیس
4. مدیریت سناریو

## الگوریتم

برای حل این مسئله راه حل متفاوتی وجود دارد یکی از این راه حل ها استفاده از الگوریتم های هوش جمعی مانند الگوریتم ژنتیک است.

در این پروژه به این دلیل از این الگوریتم استفاده شده است که الگوریتم ژنتیک قدرت و دوام بیشتری نسبت به سایر روش های مبتنی بر هوش مصنوعی دارد. بر خلاف سیستم های هوش مصنوعی قدیمی تر، الگوریتم ژنتیک با تغییر اندک مقادیر ورودی و یا با وجود مقادیر قابل توجهی از نویز در سیستم به راحتی قطع نمی شود. علاوه بر این، در جستجوی یک فضای حالت بزرگ، فضای حالت Multimodal و یا یک رویه چند بعدی، استفاده از الگوریتم ژنتیک مزیت های بسیار بیشتری نسبت به روش های جستجوی متداول در سایر تکنیک های بهینه سازی مانند برنامه ریزی خطی، جستجوی تصادفی و یا روش های جستجوی اول عمق، اول سطح و یا praxis دارد.

الگوریتم های ژنتیک برای حل مسئله، اصل بقای اصلح را در میان اعضای یک جمعیت در طی نسل های متوالی شبیه سازی می کنند. هر نسل شامل جمعیتی از رشته کاراکتر ها است که مشابه کروموزوم هایی که در DNA ما دیده می شوند می باشند. هر فرد، نمایانگر یک نقطه در فضای جستجو و یک راه حل احتمالی خواهد بود. سپس اعضای هر نسل، در فرآیندی مشابه فرآیند تکامل موجودات زنده وارد می شوند.

الگوریتم های ژنتیک با استفاده از مبانی خاص، مشابه ساختار های ژنتیکی و رفتار کروموزوم ها در میان جمعیتی از افراد، عمل می کنند. این مبانی عبارتند از:

* اعضای یک جمعیت برای منابع و جفت گیری با یکدیگر رقابت می کنند.
* اعضایی که در هر رقابت موفق تر هستند فرزندان بیشتری را نسبت به اعضایی که عملکرد مناسبی نداشته اند ایجاد می کنند.
* ژن هایی از اعضای با عملکرد خوب در درون جمعیت منتشر می شود بنابراین، والدین خوب گاهی اوقات صاحب فرزندی می شوند که از هر کدام از والدین بهتر است.
* بنابراین، هر نسل متوالی برای زندگی در محیط اطراف خود مناسب تر خواهد بود.

## رابط کاربری

برای ایجاد رابط کاربری در این پروژه از کتابخانه PtQt5 استفاده شده است. PyQt یک binding (استفاده از کتابخانه های سی پلاس پلاس را در زبان پایتون ممکن می سازد) از فریم ورک Qt (بخوانید کیوت) می باشد که برای استفاده در زبان پایتون ایجاد شده است. خود Qt در حقیقت مجموعه ای از کتابخانه ها و ابزارهای توسعه ی مستقل از پلتفرم در زبان سی پلاس پلاس است. PyQt شامل انتزاع از مفاهیم رابط گرافیکی و کتابخانه های مخصوص شبکه، پردازش موازی، عبارت های قاعده دار، پایگاه داده SQL و... می باشد.

این کتابخوانه قدرتمند می‌تواند تا حد بسیار زیادی در ایجاد پروژه‌های گرافیکی کمک می کند. این کتابخوانه قدرتمند در حقیقت مانند یک چاقوی همه کاره است. تقریبا می‌تواند هر برنامه دلخواهی را با استفاده از کتابخانه pyqt5 تولید کرد. در ابتدا بسیاری زبان برنامه‌نویسی پایتون را چندان جدی نمی‌گرفتند اما با روی کار آمدن کتابخانه‌های قدرتمند این زبان پیشرفت‌های جالبی ایجاد شد. PyQt که توسط Riverbank Computing ایجاد شده است، یک نرم افزار رایگان (دارای مجوز GPL ) است و از سال 1999 در حال توسعه است. PyQt5 در سال 2016 منتشر شد و آخرین بار در اکتبر 2021 به روز شد. هدف از انتخاب این ابزار برای ایجاد رابط کاربری، سبکی و درعین حال قدرتمندی این کتابخانه است که ما را قادر می سازد تا برای سیستم عامل های مختلف بتوانیم رابط کاربری بسازیم. یا به بیان دقیق تر برنامه cross-platform ایجاد کنیم

## دیتابیس

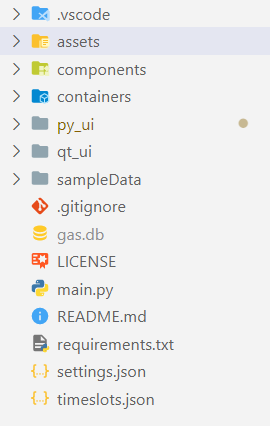
برای ذخیره سازی و ساماندهی داده های اولیه و همچنین نتایج ایجاد شده نرم افزار باید از یک دیتابیس استفاده کنیم. در این پروژه به دلایل زیر از دیتابیس SQLite استفاده شده است:

* تمرکز اصلی SQLite بر ارائه‌ی بانک داده‌ی قدرتمند سازگار با SQL، بدون وابستگی به موارد اضافه است. همان‌طور که از نام این پایگاه داده نیز برمی‌آید، می‌توان آن را یک راهکار سبک محسوب کرد که تقریبا روی هر بستری که از C و ذخیره‌سازی فایل پشتیبانی می‌کند، اجرا می‌شود. امکان ارتباط بین بانک داده‌ی SQLite با زبان‌های برنامه‌نویسی سطح‌بالا در دسترس توسعه‌دهندگان قرار دارد.
* از آنجا که بانک داده‌ی SQLite، در واقع یک فایل ساده است، بنابراین قابلیت حمل بسیار بالایی دارد و می‌توان به‌راحتی نسخه‌ی پشتیبان آن را ایجاد کرد. عدم نیاز به بخش سرور باعث شده است راه‌اندازی SQLite و استفاده از آن بسیار ساده باشد. شما می‌توانید حتی بدون راه‌اندازی محیط توسعه‌ی کامل، از مزایای SQLite در پروژه‌های خود بهره ببرید. علاوه‌بر این، استفاده از بانک داده به طی کردن فرآیندهای طولانی، راه‌اندازی مجدد یا بررسی مسائل امنیتی نیاز ندارد.
* استفاده از SQLite در برنامه‌های کاربردی باعث افزایش انعطاف‌پذیری و کاهش زمان توسعه خواهد شد. بهره گرفتن از فایل‌های متنی ساده به‌عنوان بانک داده‌ی SQLite برای پیکربندی، ذخیره‌سازی و دسترسی به داده‌ها در دستگاه‌های مختلف یکسان است و همین مورد به حفظ عملکرد ثابت این فناوری کمک می‌کند.
* بانک‌های داده معمولا از سیستم‌هایی برای محافظت در برابر آسیب‌های احتمالی بهره می‌برند که این ویژگی در فایل‌های معمولی دیده نمی‌شود. SQLite یک بانک داده‌ی تراکنشی (transactional) است، بنابراین می‌توانید از رویدادهای موفق جزئی اجتناب کنید. اگر یک عملیات در یک تراکنش با شکست مواجه شود، عملیات موفق نیز بازگردانی می‌شود و پایگاه داده را به حالت اولیه‌ی خود باز می‌گرداند.
* SQLite دربرابر خطاهای ذخیره‌سازی و سناریوهایی که به‌دلیل کمبود منابع رم رخ می‌دهد، آسیب‌پذیر نیست. امکان بازیابی بانک‌های داده درصورت بروز مشکل در سیستم یا قطعی برق وجود دارد که این اقدام به ایمنی داده‌ها کمک می‌کند. پایگاه کد توسط مجموعه‌ی آزمایشی بسیار فراگیری با پوشش ۱۰۰ درصد، داده‌ها را تحت مدیریت خود قرار می‌دهد.

## مدیریت سناریو

مدیریت سناریو در واقع بخش های مختلف پروژه شامل دیتابیس، و رابط کاربری را به یکدیگر متصل می سازد و همچنین thread ها را مدیریت می کند. تا نتیجه دلخواه ایجاد شود.

# فصل سوم: ساختار پروژه

ساختار کلی پروژه را در شکل 1-3 مشاهده می کنید.

شکل 1-3

### دایرکتوری Components

این دایرکتوری شامل ماژول ها و کلاس های اصلی پروژه است که در ادامه هرکدام را به تفصیل توضیح خواهیم داد.تمام ماژول های پروژه به صورت شئ گرا و فانکشنال طراحی شده اند تا روند طراحی و توسعه در آینده راحت تر باشد.

### دایرکتوری Containers

این دایرکتوری شامل فانکشن های مربوط به GUI می باشد. فایل مربوط به هر پنجره جدا می باشد که با نام کلاس اصلی آن مشخص شده است

### دایرکتوری py\_ui

کتابخانه PyQt از روی فایل های این دایرکتوری GUI را می سازد به عبارت بهتر فایل های این دایرکتوری ساختار رابط کاربری را مشخص می کنند.

### فایل gas.db

فایل دیتابیس SQLite اصلی پروژه که داده های اولیه شامل مشخصات اساتید، دروس، کلاس ها (اتاق ها) و نیز نتایج حاصل از الگوریتم ژنتیک در آن ذخیره شده است.

### فایل requirements.txt

پکیج ها و کتابخانه های مورد استفاده در پروژه در این فایل نوشته شده اند که به راحتی با استفاده از دستور pip install -r requirements.txt قابل نصب باشند

### فایل settings.json

تنظیمات کلی برنامه و نیز تنظیمات مربوط به مراحل مختلف الگوریتم ژنتیک بصورت فرمت json در این فایل نوشته شده اند تا از طریق الگوریتم و نیز رابط کاربری قابل دسترسی و تغییر باشند و همچنین با بسته شدن نرم افزار به حالت اولیه بازنگردند.

### فایل timeslots.json

تعداد و نام بازه های زمانی در این فایل به فرم json قرار گرفته اند که تعداد و نام آنها با تغییر این فایل امکا پذیر خواهد بود.

# فصل چهارم : ساختار ساخت زمانبندی

در این فصل با اجزای مختلف الگوریتم ژنتیک و تکنینک استفاده شده در آن آشنا خواهیم شد و کد های این قسمت را بررسی خواهیم کرد. کلاس ها و توابع مربوط به الگوریتم ژنتیک در فایل GeneticAlgortim.py واقع در دایرکتوری components نوشته شده اند.

## کتابخانه های مورد نیاز الگوریتم



**خط اول:** کلاس QtCore برای کار با thread ها در این ماژول استفاده شده است تا الگوریتم بدون ایجاد اختلال در عملکرد رابط کاربری بتواند به عنوان یک Thread پردازشی مجازی به کار خود ادامه دهد.

**خط دوم:** فایل تنظیمات و توابع کمکی که در فایل utilities.py قرار گرفته اند را فراخوانی می کند.

**خط سوم:** تابع Itemgetter را از کتابخانه operator فراخوانی میکند که در این ماژول برای مرتب سازی (sort) های جهت دار لیست داده ها مورد استفاده قرار می گیرد.

**خط چهارم:** تابع Counter را از کتابخانه collections فراخوانی می کند. این تابع مقدار هر کاراکتر در یک رشته یا مقدار هر عضو یک لیست را می شمارد و نتیجه را بصورت یک دیکشنری برمیگرداد که برای محاسبه های آماری جمعیت در این ماژول استفاده شده است

**خط پنجم:** کتابخانه copy را فراخوانی می کند. به بیان ساده این کتابخانه برای ایجاد کپی از یک لیست موجود در حافظه بدون استفاده از اشاره گرها استفاده می شود. و توابع مختلفی دارد.

**خط هفتم:**NumPy که مخفف Numerical Python است، برای محاسبات عددی متنوعی در پایتون به کار می‌رود. محاسبات به کمک آرایه‌ها در NumPy سرعت خوبی دارند و علاوه بر آن، توابع این پکیج در ساخت پکیج‌های محاسباتی دیگر مورد استفاده قرار گرفته است.

## کلاس کروموزم در الگوریتم ژنتیک



ساختار دیتای یک کروموزم در کامنت کد بالا مشاهده می کنید. همانطور که مشاهده می کنید در یک کروموزوم تمام اطلاعات اولیه و همچنین یک شیوه از چیدمان ذخیره خواهد شد. تا در صورت تغییر اطلاعات سایر کروموزم دستخوش تغییر نشود. درست مشابه کروموزم انسان هر سلول (کروموزم) حاوی تمام اطلاعات (ژنهای) آن فرد خواهد بود.



تابع سازنده این کلاس شامل فیتنس و اطالاعات مربوط به فیتنس کروموزم و اطلاعات خام (rawData) بصورت یک ورودی از کلاس دریافت می شوند و تغییراتی بر روی آن صورت خواهد گرفت و پس از تغییرات در متغییر data ذخیره خواهند شد. همچنین تنظیمات که با استفاده از تابع getSettings که از ماژول Settings فراخوانی خواهد شد. داخل هر کروموزوم فراخوانی خواهد شد. تا در موقعیت های مختلف درون هر آبجکت کلاس کروموزم قابل دستیابی باشد.



تابع buildChromosome که در سازنده کلاس Chromosome فراخوانی شده است. یک کروموزم جدید ایجاد   
می کند. در این تابع اطلاعات خام (rawData) که تقریبا بصورت دست نخورده از پایگاه داده نرم افزار واکشی شده بودند دستخوش تغییراتی می شوند تا برای پردازش های بعدی آماده شوند. (Preprocessing)



تابع insertSchedule یک لیست بصورت زیر دریافت می کند. و سپس بررسی می کند که آیا از نظر معیارهای طراح سیستم این زمانبندی قابل جایگذاری در برنامه هفتگی می باشد یا خیر. به عبارت دقیق تر زمانبندی ورودی را بررسی می کند تا با دیگر معیار های زمانبندی تداخل نداشته باشد.

اگر زمانبندی درس وارد شده با یکی از معیار ها ناسازگار باشد تابع یک ارور از 1 تا 4 برمی گرداند که مشخص می کند که این زمانبدی با کدام یک از معیار ها در تداخل بوده تا واکنش مناسب نسبت به آن صورت گیرد.

در غیر اینصورت زمانبندی در دیکشنری اتاق وارد خواهد شد که نشان می دهد که یک کلاس مشخص در یک روز و یک زمان مشخص چه کلاسی دارد. همین رویداد برای دیکشنری استاد (instructor) نیز انجام خواهد شد. و در نهایت False نیز بازگردانده خواهد شد.



تابع validateSchedule نیز که در تابع insertSchedule مورد استفاده قرار گرفته بود.صحت زمانبدی (schedule) مورد نظر را بررسی می کند.

چهار معیار برای هر زمانبندی در نظر گرفته شده است:

1. isRoomTimeslotAvailable : بررسی می کند که اتاق در زمان مورد نظر خالی باشد و با کلاس دیگر تداخل نداشته باشد
2. isInstructorTimeslotAvailable : بررسی می کند که استاد در زمان مورد نظر تایم خالی داشته باشد و کلاس دیگری در آن زمان برای استاد تعریف نشده باشد و یا استاد در آن زمان در دانشگاه حضور داشته باشد.
3. isSectionTimeSlotAvailable : بررسی می کند که گروه درسی در آن زمان اجازه برگزاری کلاس داشته باشد
4. isLaunchTime : بررسی می کند که کلاس در زمان ناهار و نماز برگزار نشود.





### کلاس GeneticAlgorithm



کلاس GeneticAlgorithm، کلاس اصلی اجرای الگوریتم ژنتیک است که الگوریتم را مرحله به مرحله تا رسیدن به جواب ایده آل اجرا می کند. قبل از فرخوانی تابع سازنده، سیگنال های GUI فراخوانی شده اند. اطلاعات از الگوریتم به صورت سیگنال همگام به GUI منتقل می شوند تا اطلاعات پردازشی و مرحله و روند اجرای الگوریتم را به رویت کاربر برسانند.



سازنده این کلاس شامل متغیر هایی است که در ادامه آنها را معرفی خواهیم کرد. متغیر averageFitness که مقدار فیتنس میانگین کروموزوم های جمعیت فعلی را ذخیره می کند. و متغیر pastAverageFitness مقدار فیتنس میانگین جمعیت قبل را ذخیره می کند تا بتوان این دو مقدار را با یکدیگر مقایسه کرد. متغیر running که درحال اجرا بودن الگوریتم را نشان می دهد. لیست chromosomes کروموزوم های جمعیت فعلی را در خود ذخیره می کند.

مقدار tournamentSize برابر است با درصدی از جمعیت فعلی که در الگوریتم tournamentSelection با یکدیگر برای نسل بعدی رقابت می کنند.

مقدار elitePercent برابر است با درصدی از جمعیت که با استفاده از الگوریتم eliteSelection مستقیما به نسل بعد منتقل می شوند.

لیست elites که جمعیت نخبه یا برتر را در خود نگه می دارد.

لیست matingPool که زوج هایی از کروموزوم های که قرار است با یکدیگر crossover داده و تشکیل offspring دهند را در خود نگه داشته و offspring ها در یک لیست offsprings ذخیره خواهند شد.

متغیر mutationRate مقدار فعلی نرخ جهش در جمعیت و متغیر mutationRateStep نرخ تغییر جهش در جمعیت را در خود نگه می دارند.



تابع generateChromosome به اندازه ورودی quantity کروموزوم ایجاد می کند و با فراخوانی تابع generateSubjectPlacementsForSections برای هر گروه یک چیدمان ارائه می دهد (در بخش بعدی با عملکرد این متد بیشتر آشنا خواهیم شد.) در نهایت کروموزوم ایجاد شده در لیست کروموزم های جمعیت ذخیره خواهد شد.



این متد با اجرای متد generateSubjectPlacement برای هر گروه (Subject) سعی در این دارد که برای دروس موجود در هر گروه یک زمابندی ایده آل ارائه دهد. اگر این چیدمان بدون ارور انجام شود False را برمی گرداند ولی اگر این چیدمان همراه با ارور باشد درسی که چیده نشده در لیست unplacement قرار خواهد گرفت.

تابع generateSubjectPlacement تا زمانی که متغیرgenerating برابر True باشد. تلاش می کند تا یک زمانبندی مناسب برای درس (subject) وارد شده پیدا کند.اگر انتخاب زمانبندی ارور داشته باشد سعی در رفع ارور با انتخاب دوباره اتاق، استاد و یا تایم دارد. در غیر اینصورت با استفاده از متد insertSchedule از آبجکت کروموزوم مورد نظر، زمانبندی جدیدی برای درس انتخابی ارائه می دهد.

### متد selectRoom



متد selectRoom یک اتاق بصورت تصادفی از لیست اتاق های مجاز برای درس (subject) مورد نظر، انتخاب می کند.

### متد selectInstructor



متد selectInstructor یک استاد بصورت تصادفی از لیست اساتیدی که درس (subject) مورد نظر را تدریس می کنند، انتخاب می کنند.

### متد selectTimeDetails



متد selectTimeDetails در ابتدا یک روز بصورت کاملا تصادفی انتخاب می کند. سپس یک تایم اسلات از بین تایم اسلات های موجود در روز یکی را بصورت تصادفی انتخاب می کند.

### تابع ارزیابی[[8]](#footnote-8)



محاسبه فیتنس تا حد زیادی بر اساس مقدار موضوع ترسیم شده در مقایسه با موضوعات مورد نیاز است. با این حال، محاسبه همچنان به ماتریس ارزیابی ارائه شده توسط کاربر بستگی دارد. ماتریس ارزیابی مجموعه ای از وزن های محدودیت است که قابلیت شکل دادن به جواب ها را دارد. این فهرستی از اولویت بندی محدودیت ها با استفاده از توزیع صد درصد است.



|  |  |
| --- | --- |
| Function | Name |
| تلاش برای ایجاد برنامه زمانی معتبر برای همه  واحدهای همه بخش ها | Subject Placement |
| تلاش برای به حداقل رساندن تایم های خالی اساتید | idleTime |

برای محاسبه فیتنس کل از فرمول زیر استفاده می‌شود:

x برابر هر کروموزم انتخابی است:

𝒇𝒊𝒕𝒏𝒆𝒔𝒔(𝒙) = (𝑔(𝑥) ∗ 𝑔𝑊) + (ℎ(𝑥) ∗ ℎ𝑊) + (𝑖(𝑥) ∗ 𝑖𝑊) + (𝑗(𝑥) ∗ 𝑗𝑊) + (𝑘(𝑥) 𝑘𝑊)  
+ (𝑙(𝑥) ∗ 𝑙𝑊) + (𝑚(𝑥) ∗ 𝑚𝑊)

**Subject Placement:**

این مرحله ارزیابی فیتنس پایه کروموزوم را بر اساس ایجاد برنامه زمانی معتبر برای همه واحد های همه ی بخش ها محاسبه می کند.

**idleTime:**

این مرحله از محاسبه فیتنس کروموزم براساس تلاش الگوریتم برای به حداقل رساندن تایم های خالی بین دروس یک روز اساتید ارزیابی میشود

### تنظیم جمعیت[[9]](#footnote-9)



**جمعیت اصطلاحی است برای گروهی از راه حل ها**. قبل از تولید راه حل، اپراتور با استفاده از سیستم می تواند تنظیمات اجرا الگوریتم را تغییر دهد. تنظیمات "حداقل جمعیت[[10]](#footnote-10)" و "حداکثر جمعیت[[11]](#footnote-11)" حد تغییر جمعیت را مشخص می کند. محاسبه تراز جمعیت که نشان می دهد که چه مقدار تغییر در جمعیت لازم است با استفاده از یک فرمول زیر انجام می شود. در واقع در این تابع الگوریتم تصمیم می گیرد جمعیت جدید (مهاجر) به جمعیت وارد کند یا خیر

### تنظیم نرخ جهش[[12]](#footnote-12)

نرخ جهش شانسی برای هر کروموزوم برای ایجاد تغییرات تصادفی است. تنظیم نرخ جهش زمانی اتفاق می‌افتد که محرکی که قبل از اجرا تنظیم شده است، برآورده شود. هنگامی که یک نسل بدون تنظیم نرخ جهش کامل می شود، میزان جهش 0.5٪ کاهش می یابد، در غیر این صورت 0.5٪ افزایش می یابد. محرک تنظیم نرخ جهش [[13]](#footnote-13)از فرمول زیر محاسبه می شود:

### انتخاب جمعیت هدف[[14]](#footnote-14)

پس از ارزیابی، کروموزوم ها برای شرکت در تولید مثل انتخاب می شوند. احتمال انتخاب کروموزوم هایی با فیتنس بالاتر بیشتر است. اینجاست که بقای شایسته‌ترین‌ها فرا می‌رسد.



روش‌های مختلفی برای انتخاب کروموزوم‌ها وجود دارد که همگی مزایا و معایب خود را دارند. داشتن مناسب ترین نوع انتخاب [[15]](#footnote-15)به جلوگیری از همگرایی اولیه و ترویج راه حل های متنوع (ایجاد تنوع در جمعیت) کمک می کند. به عنوان یک راه حل شایسته‌گرایانه از الگوریتم ژنتیک، n درصد بالای جمعیت انتخاب می‌شود که در نسل بعدی با همان ژن‌ها (بدون تغییر) پیشروی کنند.

بقیه مراحل انتخاب قرار است با اجرای مسابقات[[16]](#footnote-16) متعدد بر روی جمعیت انجام شود. شرکت کنندگان در مسابقات به صورت تصادفی با تعداد 4 درصد از جمعیت موجود انتخاب می شوند اما سقف آن بیست و پنج (25) است. مسابقات تا زمانی که جفت های کافی انتخاب شوند ادامه خواهد داشت. مثال زیر یک شبیه سازی از این سلکشن را نشان می دهد که در آن 40 درصد از 10 کروموزوم در مسابقات انتخاب می شوند. روش انتخاب مسابقات با شبه کد توضیح داده می شود:



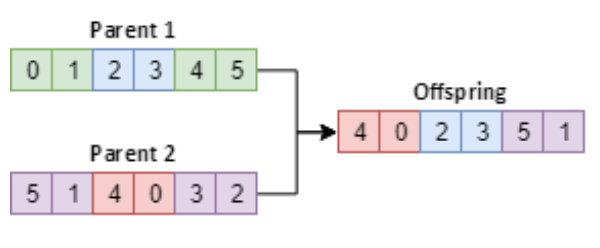
مسابقه قطعی، بهترین فرد را (در صورت p=1) در هر تورنمنت انتخاب می کند. یک انتخاب یک طرفه یعنی زمانی که (k=1) است معادل انتخاب تصادفی است.

اگر 100 کروموزوم در جمعیتی با 10 درصد elite selection وجود داشته باشد، باید 90 بار مسابقات ایجاد کنیم تا 45 کروموزوم را جفت کنیم که 90 فرزند تولید می کند. کروموزوم های فرزندان و شایسته پس از تلاقی در مجموع به 100 می رسد که تعداد جمعیت در تمام طول فرآیند الگوریتم ژنتیک است.

### جفت گیری [[17]](#footnote-17)و تولید نسل جدید[[18]](#footnote-18)

استخر جفت گیری [[19]](#footnote-19)شامل کروموزوم های انتخابی است که قرار است با یکدیگر کراس اور[[20]](#footnote-20) انجام دهند. عملیات کراس اور شامل تولید فرزندان بر اساس ژن های والدین مشابه با همتای بیولوژیکی آن است. روش های متعددی برای انجام کراس اور وجود دارد و مانند انتخاب، استفاده از مناسب ترین روش برای داشتن نتیجه بهتر مهم است.

Order crossover[[21]](#footnote-21) برای مسائل مبتنی بر جایگشت بهتر عمل می کند. این فرآیند بدین صورت است که خوشه ژنی والد اول را به فرزندان می برد و سپس خوشه ژنی والد دوم را به فرزندان می پیچد. برای هر جفت کروموزوم، دو فرزند تولید می شود که نقش والدین برای تولید فرزندان متفاوت معکوس می شود. همه فرزندان جای والدین خود را در جمعیت خواهند گرفت. در شکل زیر نحوه ی اجرا این نوع کراس اور مشخص است:



### جهش

جهش یک عملگر ژنتیکی است که به تنوع در جمعیت کمک می کند. همچنین می تواند از گیر کردن راه حل ها در بهینه محلی [[22]](#footnote-22)جلوگیری کند. جهش فرآیند تغییر ژن/های یک کروموزوم است. معمولاً میزان جهش پایین و ثابت است. با این حال، الگوریتم ژنتیک تطبیقی ​​به این معنی است که نرخ جهش ممکن است بسته به عملکرد جمعیت متفاوت باشد. که در بخش های قبل نحوه ی تغییر و فرمول تغییر نرخ جهش را کامل بررسی کردیم.

### دامنه و محدودیت

هدف این تحقیق ایجاد یک هوش مصنوعی است که می تواند جدول زمانی ایجاد کند. که فقط پردازش داده های ورودی و تولید نتایج معقول انسان را پوشش می دهد. فرآیندی که در ایجاد زمان‌بندی استفاده می‌شود، الگوریتم ژنتیک تطبیقی-شایسته‌گرا [[23]](#footnote-23)است. نوعی الگوریتم تکاملی که شامل حفظ بهترین مجموعه راه‌حل‌ها در نسل‌های بعدی برای حفظ راه‌حل‌های سطح بالا است، اما در عین حال، متغیرهایی مانند تعداد جمعیت، نرخ جهش و فشار انتخاب تغییر می‌کنند تا از هم‌گرایی زودرس جلوگیری شود که منجر به خروجی ضعیف خواهد شد.

محاسبه زمانبندی شامل اعتبارسنجی برای هر محدودیت است. محدودیت ها مجموعه قوانینی هستند که پذیرش خروجی را هدایت می کنند. سه نوع محدودیت برای این مسئله وجود خواهد داشت.

1. **محدودیت های نرم**: مجموعه ای از قوانین که می تواند بدون تأثیر بر اعتبار خروجی شکسته شود.
2. **محدودیت های متوسط:** مجموعه ای از قوانین که می تواند با تأثیر بر اعتبار خروجی شکسته شود. با این حال، این تنها زمانی می تواند شکسته شود که سناریو از نظر منطقی نامعتبر یا غیرممکن باشد.
3. **محدودیت های سخت:** مجموعه ای از قوانین که در صورت شکسته شدن، راه حل نامعتبر ایجاد می کند.

### خروجی نرم افزار

تضمین نمی شود که نتایج برنامه بهترین راه حل ممکن برای سناریو باشد. کیفیت نتیجه به شدت به جهت گیری اجرا بستگی دارد. به دلیل ماهیت تصادفی آن، نتایج ممکن است از ضعیف تا عالی متفاوت باشد. سیستم تضمین نمی‌کند که محدودیت‌های سخت هر راه‌حل برآورده شود، به‌ویژه زمانی که سناریوی ارائه‌شده از نظر منطقی غیرممکن یا به شدت فشرده باشد. این همچنین به این معنی است که وقتی در یک سناریوی سخت، محدودیت‌های نرم یا متوسط ​​در معرض نقض هستند تا محدودیت‌های سخت برآورده شود. محدودیت های سخت باید رعایت شوند و هرگز نقض نخواهند شد.

### تنظیمات الگوریتم ژنتیک

پیکربندی الگوریتم ژنتیک در برنامه برای همه نوع مسائل یکسان نیست که بتواند هر سناریویی را برآورده کند. این به دلیل محدودیت قدرت محاسباتی است. برای مقابله با محدودیت و همچنین پشتیبانی از مقیاس‌پذیری ، تنظیمات اجرا الگوریتم ژنتیک را می‌توان با توجه به قابلیت یا اولویت اپراتور تغییر داد. جدول زیر شرح و توصیفی برای هر ویژگی قابل تغییر الگوریتم را نشان می دهد.

|  |  |  |
| --- | --- | --- |
| تنظیمات | توضیحات | محدودیت |
| حداقل تعداد جمعیت[[24]](#footnote-24) | حداقل و تعداد اولیه کروموزوم ها در یک نسل. پنجاه (50) مقدار پیشنهادی است زیرا برای انواع راه حل ها به اندازه کافی زیاد است. | 50-10000 |
| حداکثر تعداد جمعیت[[25]](#footnote-25) | محدودیت در تعداد کروموزوم های یک نسل. محاسبه میزان تحمل دستگاه محاسباتی به میزان رم آن بستگی دارد. به ازای هر هفتاد (70) موجودیت فعال در یک سناریو (موضوع، مربی، اتاق و بخش)، یک مگابایت (1 مگابایت) حافظه در هر کروموزوم مصرف می‌کند. بنابراین، اگر از پنجاه (50) موجودیت و حداکثر تعداد جمعیت صد (100) استفاده شود، پیک مصرف RAM تقریباً 72 مگابایت به اضافه 25 درصد مقدار برای مدیریت برنامه اصلی خواهد بود. یک کامپیوتر رده پایین باید حداکثر تعداد جمعیت را پایین نگه دارد، می تواند با حداقل پیشنهاد جمعیت پنجاه (50) مطابقت داشته باشد. | 50-10000 |
| حداکثر تعداد نسل[[26]](#footnote-26) | محدودیت تعداد نسلی که الگوریتم می تواند تولید کند. با رسیدن به مقدار تنظیم شده، الگوریتم متوقف می شود. هرچه تعداد نسل بیشتر یا سناریو محدودیت‌های پیچیده‌تری داشته باشد، بیشتر باید اجرا شود، زیرا شانس بیشتری به هوش مصنوعی می‌دهد تا مشکلات جزئی را برطرف کند یا حتی پیشرفتی ایجاد کند که می‌تواند کل مجموعه راه‌حل‌ها را تغییر دهد. | 50-10000 |
| حداکثر کروموزم جدید[[27]](#footnote-27) | حداکثر تلاش هوش مصنوعی برای قرار دادن یک برنامه زمانی معتبر در کروموزوم. یک کامپیوتر قوی می تواند ارزش بیشتری برای این کار قائل شود، زیرا این بیشتر به عنوان یک تاکتیک brute force عمل می کند که می تواند به افزایش کیفیت راه حل کمک کند. | 1500-3000 |
| فعال کننده نرخ جهش[[28]](#footnote-28) | مقدار آستانه برای شروع تغییر نرخ جهش. این مقدار برابر است با تفاوت آخرین میانگین فیتنس را با میانگین فیتنس نسل قبلی آن.  اگر تفاوت کمتر یا برابر با مقدار تنظیم شده باشد، نرخ جهش 5٪ افزایش می یابد و در غیر این صورت تا 5٪ کاهش می یابد. اگر اپراتور نمی خواهد نرخ جهش تغییر کند، مقدار را می تواند روی صفر تنظیم کرد. توصیه می شود برای جلوگیری از تغییرات گسترده در جمعیت، مقدار این تنظیم را روی پیش فرض 0.08 نگه دارید. | 0.00-100.00 |
| حدکثر مقدار فیتنس[[29]](#footnote-29) | ماشه ای برای پایان دادن به الگوریتم. هنگامی که فیتنس یک کروموزوم به مقدار تنظیم شده رسید، الگوریتم متوقف می شود. پیشنهاد می شود برای این مقدار یک مقدار بهینه قرار گیرد که می تواند از 90 تا 98 درصد برای سطح قابل قبول متغیر باشد. | 0-100% |
| جمعیت شایسته[[30]](#footnote-30) | درصدی از اینکه چه مقدار از کروموزوم های جمعیت فعلی با بهترین عملکرد تا نسل بعدی حفظ می شود. مقدار پیشنهادی برای متغیر 5٪ است زیرا تعداد کروموزوم های خوب را پایین نگه می دارد و به جلوگیری از همگرایی زودرس کمک می کند. | 0-100% |
| انحراف  تحمل[[31]](#footnote-31) | غنای مجموعه راه حل جمعیت را اندازه گیری می کند. تناسب هر کروموزوم برای انحراف آماری اندازه گیری و نرمال می شود. اگر یکی از سیگماهای انحراف از مقدار تعیین شده فراتر رود، تعداد جمعیت به میزان درصدی تغییر می کند. در غیر این صورت جمعیت به میزان درصد کمبود آن کاهش می یابد. این به حفظ تنوع بر اساس انحراف تناسب بین کروموزوم ها کمک می کند. مقدار پیشنهادی 55٪ است زیرا از تسلط کامل یک گروه مشترک از کروموزوم ها جلوگیری می کند. | 0-100% |

# فصل پنج : رابط گرافیکی

برای استفاده راحت تر کاربر نیاز به یک رابط گرافیکی احساس می شد.

## ساختار کلی:

### پنجره اصلی:

پنجره اصلی که در زمان باز شدن برنامه مشاهده خواهد شد شامل زبانه های زیر است :

1. اساتید
2. کلاس ها (اتاق ها)
3. درس ها
4. گروه ها
5. مدیریت سناریو

که به صورت پیشفرض زبانه اساتید فعال است.

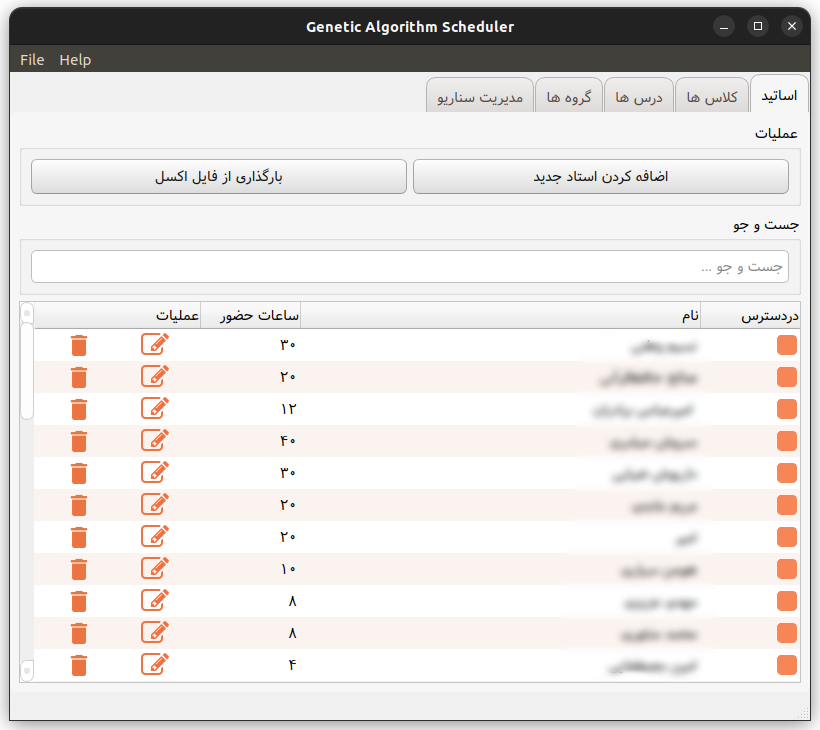
زبانه های اساتید، کلاس ها و درس ها شامل سه بخش هستند :

1. عملیات: در این بخش می توان به ساختمان داده اعضای مربوط به هر کدام از زبانه ها را اضافه کرد. همچنین دکمه ای برای بارگذاری از فایل اکسل (Excel) در نظر گرفته شده است.
2. جست و جو: این بخش برای دسترسی سریع تر به داده مورد نظر برای تغییر یا حذف آن ایجاد شده است.
3. نمایش داده ها: در این بخش داده های مربوط به هر زبانه نمایش داده می شود.

### نمایش داده ها:

در پنجره اصلی در صورت وجود داده مربوط به هر زبانه مشابه (تصویر 1-5) نمایش داده می شود. و شما می توانید برخی از اطلاعات مهم را مشاهده کنید. برای مثال می توانید اسم اساتید را در زبانه اساتید مشاهده کنید. (تصویر 5-1)

امکان مرتب کردن هر ستون بر اساس حروف الفبا برای هر ستون که حاوی اطلاعات است، با کلیک بر روی سرستون ممکن است.



**تصویر 1-5**

### ستون عملیات

با کلیک بروی آیکون ادیت (اولین آیکون از سمت راست) می توان اطلاعات آن ردیف را تغییر داد. همچنین با کلیک بروی آیکون حذف (دومین آیکون) می توان آن را حذف کرد.

### ستون دردسترس

این ستون برای فعال کردن یا غیر فعال کردن کلاس ها، استاد ها و گروه های مورد نظر است. به طوری که با غیر فعال کردن استادی، آن استاد در عملیات زمانبندی در نظر گرفته نمی شود. (تصویر 1-5)

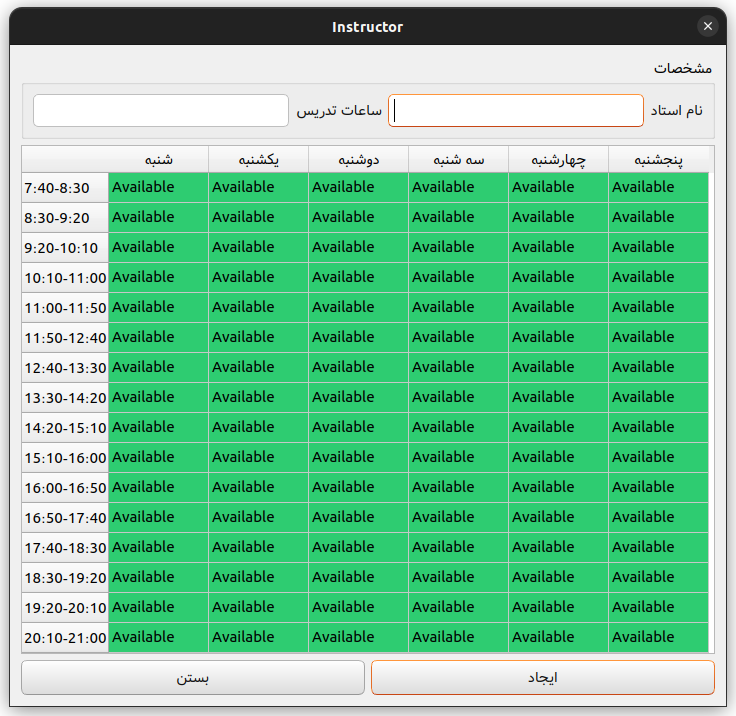
### جدول زمانی

این جدول (تصویر 2-5)، به صورت پیشفرظ تماما دردسترس "Available" است. و شما می توانید با کلیک کردن بروی هر خانه جدول وضعیت آن خانه را تغییر دهید. همچنین با کلیک کردن بر روی سر ستون ها (شنبه، یک شنبه، دوشنبه و ...)، وضعیت آن روز را در تمامی ساعات غیر قابل دسترس "Unavailable" کنید. و با دوبار کلیک کردن به وضعیت دردسترس تغییر دهید. به صورت مشابه برای سرردیف ها نیز قابل استفاده است.

### اساتید

در این قسمت می توان داده های مربوط به اساتید را مشاهده، تغییر، اضافه و یا حذف نمود. (تصویر 1-5).

برای اضافه کردن استاد جدید می توان بر روی دکمه ***اضافه کردن استاد جدید*** کلیک کرد. زیر پنجره باز شده مطابق   
تصویر 2-5 می باشد.

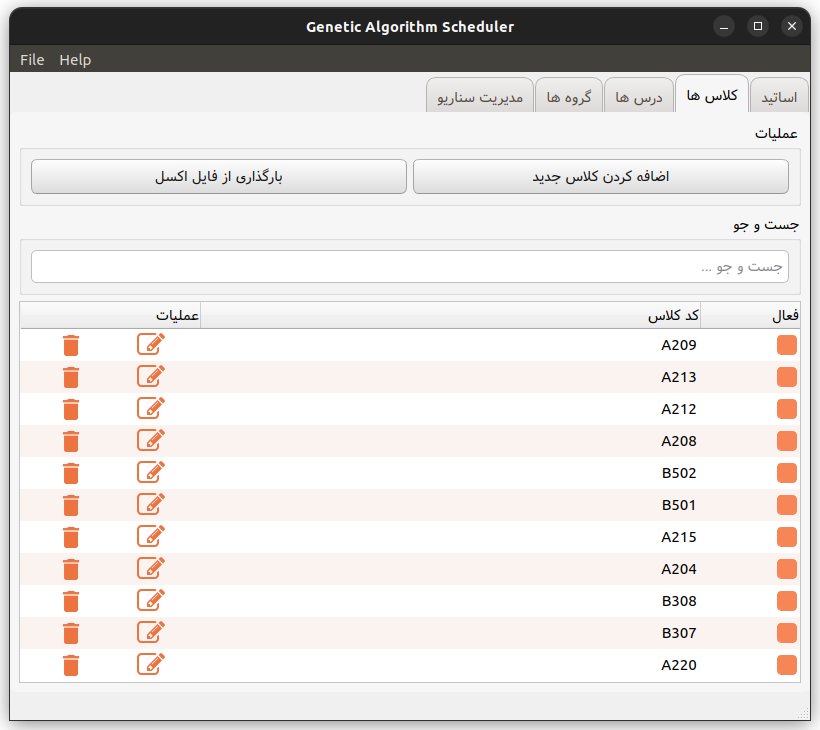


**تصویر 2-5**

1. **نام استاد:** نام و نام خانوادگی استاد مورد نظر.
2. **ساعات تدریس:** مجموع ساعات تدریس استاد در هفته.
3. **جدول زمانی:** در این قسمت روز ها و ساعاتی که استاد امکان تدریس دارد وارد می شد.
4. **دکمه ایجاد:** با کلیک بر روی این دکمه اطلاعات در پایگاه داده ثبت و زیرپنجره بسته می شود.
5. **دکمه بستن:** بدون تغییر در پایگاه داده زیر پنجره را می بندد.

### کلاس ها

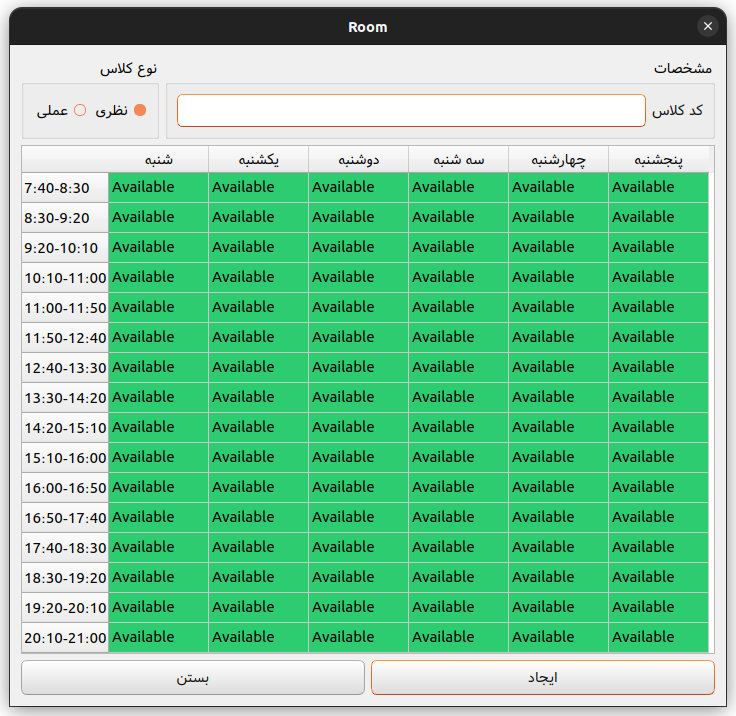
در این قسمت می توان داده های مربوط به کلاس ها را مشاهده، تغییر، اضافه و یا حذف نمود. (تصویر 3-5)



**تصویر 3-5**

برای اضافه کردن کلاس بر روی دکمه ***اضافه کردن کلاس جدید*** کلیک کنید. زیر پنجره باز شده مطابق تصویر 4-5 می باشد.

مطابق تصویر 4-5 کد کلاس را در قسمت مشخصات و نوع کلاس را وارد می شود. و با استفاده از جدول زمانی زمان های در دسترس بودن کلاس را می توان مشخص کرد.



**تصویر 4-5**

### درس ها

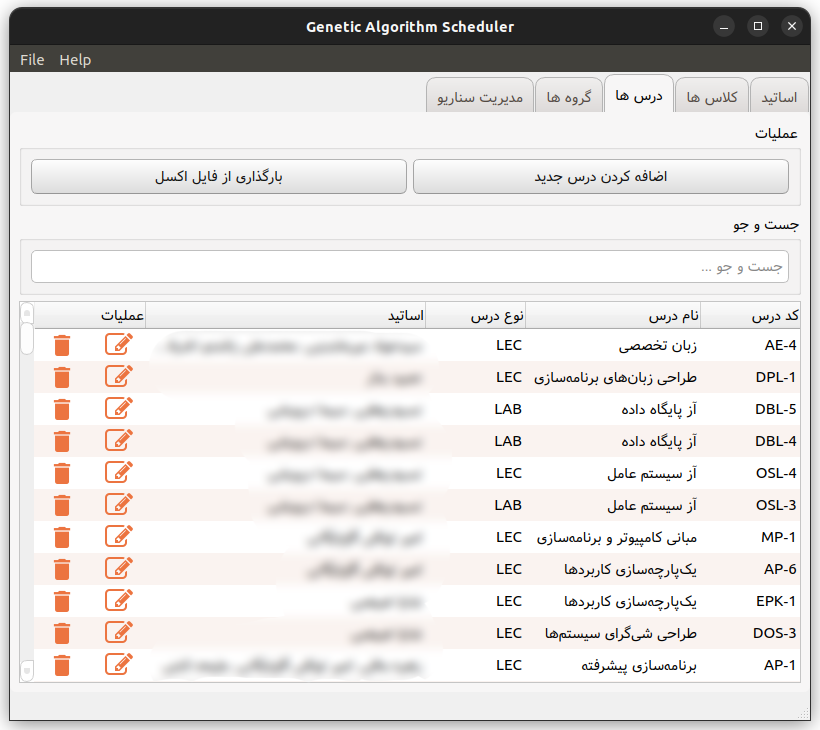
در این قسمت می توان داده های مربوط به درس ها را مشاهده، تغییر، اضافه و یا حذف نمود. تصویر 5-5

برای اضافه کردن درس بر روی دکمه ***اضافه کردن درس جدید*** کلیک کنید. زیر پنجره باز شده مطابق تصویر 6-5 می باشد.

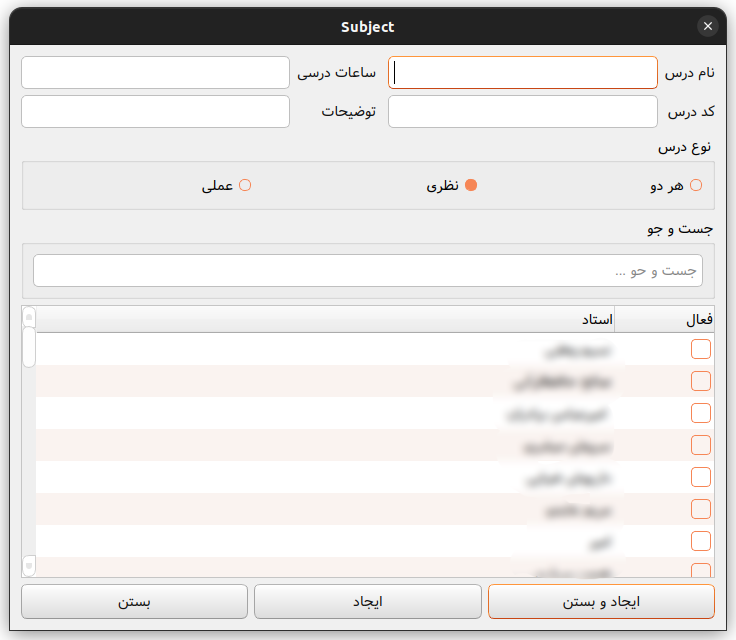
در پنجره جدید نام درس کد درس و مجموع ساعات درسی را وارد کنید (یک درس نظری 3 واحدی، 3 ساعت آموزشی، و یک درس 1 واحدی عملی، 2 ساعت آموزشی را شامل می شود.).

درصورت نیاز می توانید برای هر درس توضیحاتی نیز بنویسید که این توضیحات در الگوریتم استفاده نمی شود و صرفا برای رفع نیاز کاربر نگاه داری می شوند.

در ادامه با فعال کردن اساتید مورد نظر برای آن درس می توانید فرایند ایجاد درس جدید را تکمیل کنید. در صورتی می خواهید از یک درس چند کد ایجاد کنید می توانید از دکمه ***ایجاد*** استفاده کرده و مقادیر را نگاه دارید و تنها کد کلاس تغییر دهید.



**تصویر 5-5**



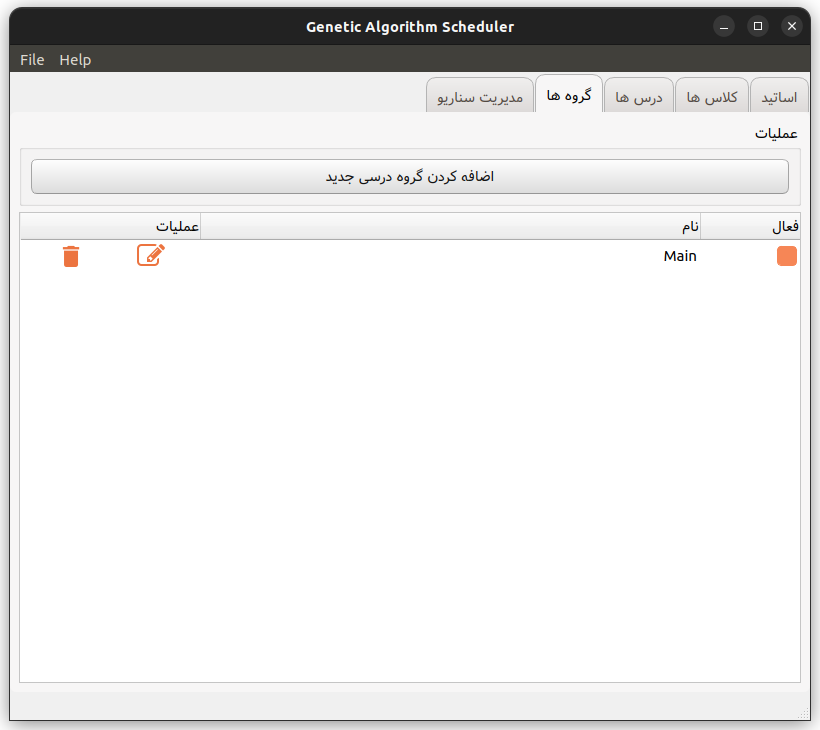
**تصویر 6-5**

### گروه ها

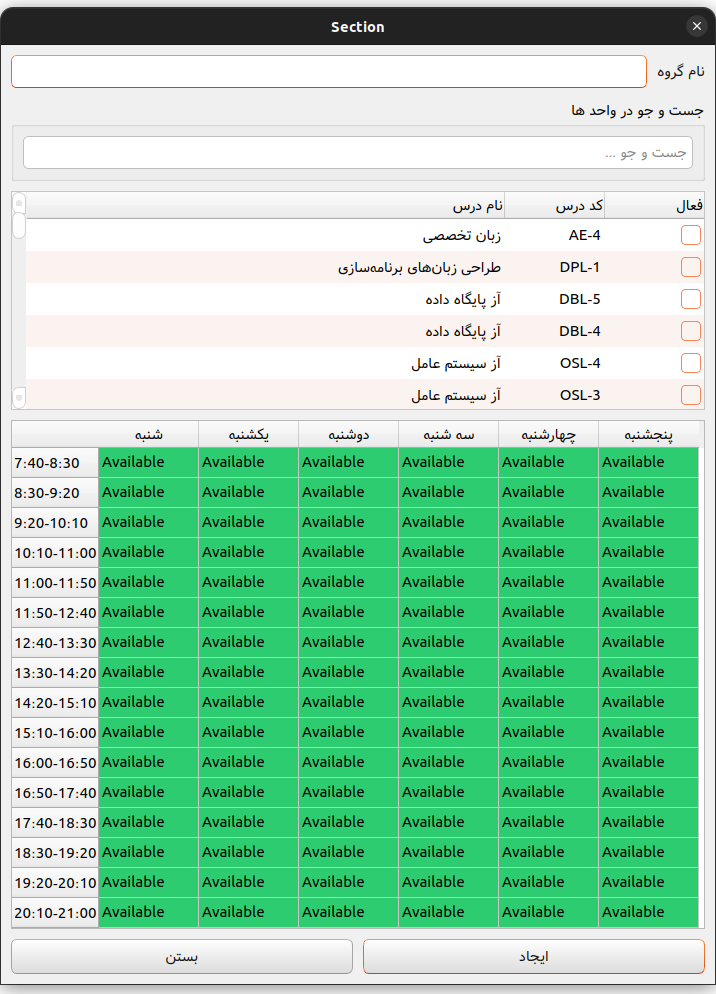
در این قسمت می توان داده های مربوط به گروه ها را مشاهده، تغییر، اضافه و یا حذف نمود. تصویر 7-5

برای اضافه کردن گروه بر روی دکمه ***اضافه کردن گروه درسی جدید*** کلیک کنید. زیر پنجره باز شده مطابق تصویر 8-5   
می باشد.

در پنجره باز شده نام گروه را وارد کرده و درس هایی که می خواهید از فهرست انتخاب کنید، می توانید از قسمت جست و جو کمک بگیرید، در ادامه وارد کرد زمان هایی که می خواهید از آن گروه کلاس ایجاد شود را در جدول زمانبندی وارد کنید.



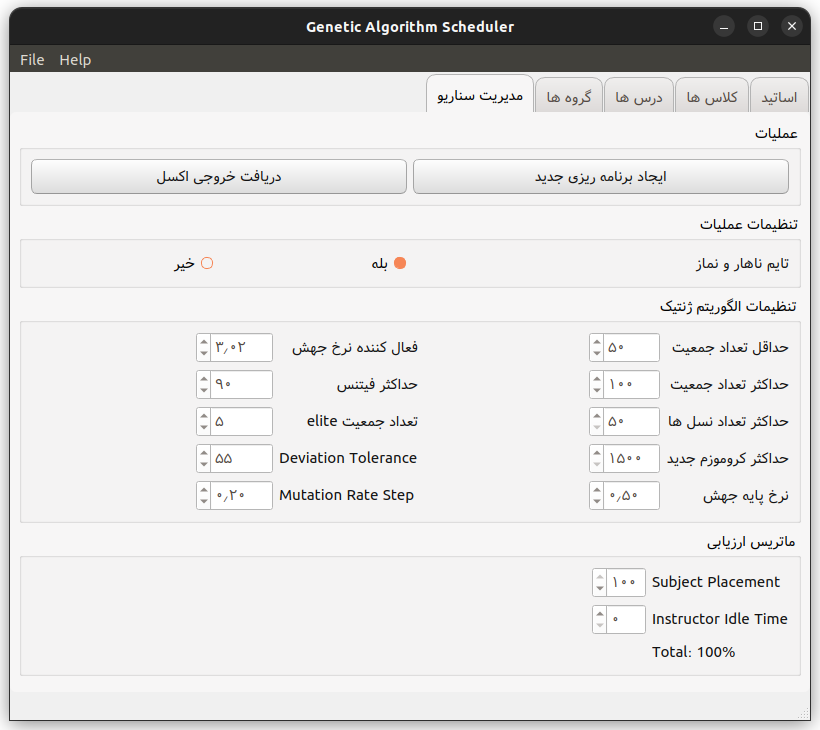
**تصویر 7-5**



**تصویر 8-5**

### مدریت سناریو

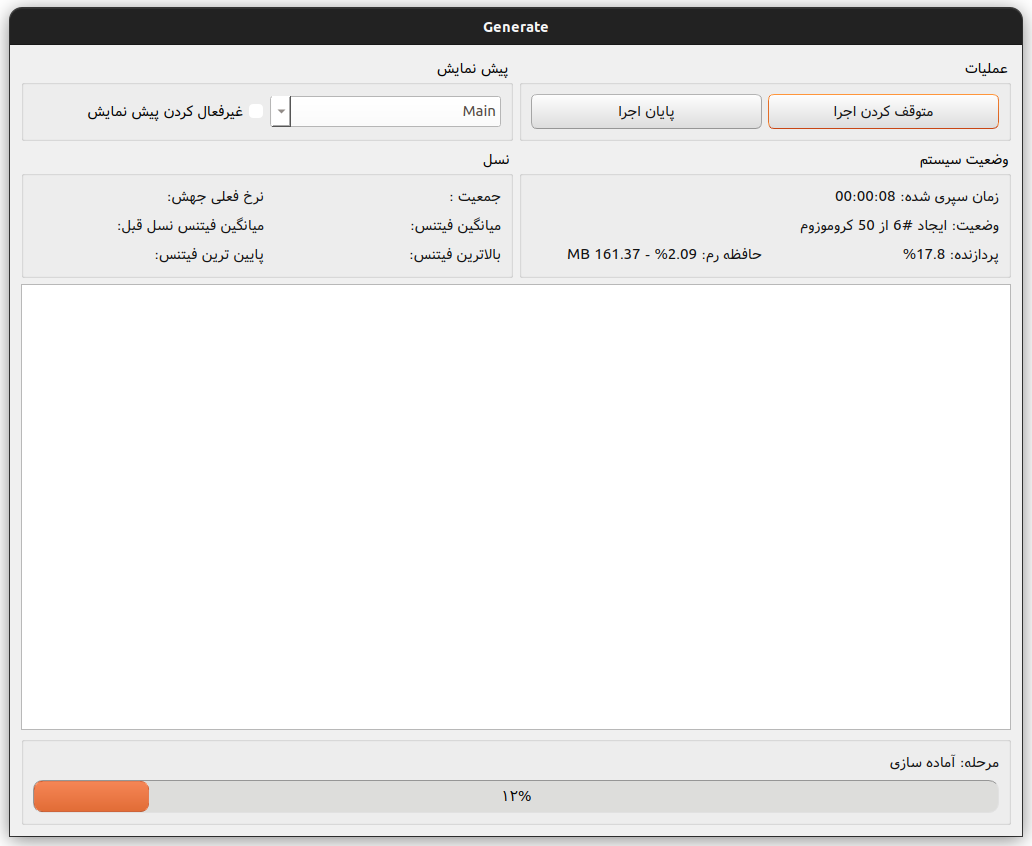
ایجاد برنامه جدید، تغییر تنظیمات و دریافت فایل اکسل (Excel) خروجی در این زبانه امکان پذیر است. که در چهار بخش عملیات، تنظیمات عملیات، تنظیمات الگوریتم ژنتیک و ماتریس ارزیابی دسته بندی شده است. (تصویر 9-5)



**تصویر 9-5**

### ایجاد

این پنجره ای است که هنگام انجام الگوریتم دیده می شود، در این پنجره می توانید اطلاعت وضعیت سیستم مراحل الگوریتم ژنتیک را مشاهده کنید. همچنین بعد از ایجاد نسل اولیه بهترین هر نسل را می توان مشاهده، و یا برای بالاتر بردن سرعت عملیات آن را غیر فعال کرد. (تصویر 10-5)



**تصویر 10-5**

## ساختار کد

### ابزار های مورد استفاده

1. برای طراحی رابط گرافیکی از ابزار Qt Disingner استفاده شد است.
2. برای ساخت و طراحی توابع مربوط به رابط گرافیکی از کتابخانه PyQt استفاده شده است.
3. برای ایجاد icon های داخل برنامه از کتابخانه qtawesome استفاده شده است.

### طراحی GUI

طراحی شامل یک پنجره اصلی:

1. Main

و چند زیر پنجره است :

1. Instructor
2. Subject
3. Room
4. Section
5. Generate

زیر پنجره ها 2 تا 5 برای وارد کردن اطلاعات مربوط به هر کدوم از بخش ها مورد استفاده قرار می گیرد. زیر پنجره 6 در هنگام اجرا الگوریتم نمایش داده می شود، که پیشتر مشاهده کردیم.

### اجزای GUI

#### فایل های UI

فایل های طراحی شده با فرمت .ui در پوشه "qt\_ui" در دسترس هستند.

#### فایل های تبدیل شده UI به py

فایل تبدیل شده به python هر پنجره در پوشه "py\_ui"، و توابع مربوط به هر کدام در پوشه "containers" قرار دارد.

#### فایل های موجود در پوشه "components"

1. TableModel.py : کلاس استاندارد برای ساخت جداول.
2. Timetable.py : کلاس Timetable برای ایجاد جدول زمانی.
3. PreviewScheduleParser.py : کلاس مربوط به جدول پیش نمایش که در پنجره Generate استفاده می شود.

#### فایل های Style

برای ایجاد تغییر در Style پنجره ها از QSS استفاده شده است، که فایل مربوط به آن در پوشه "assets/style" قرار دارد.

### Py\_ui

فایل Main.py کلاس Ui\_MainWindow و سایر فایل ها کلاس Ui\_Dialog دارند. این کلاس ها از کلاس های استاندارد PyQt هستند که برای ساخت اجزای گرافیکی پنجره ها GUI ما مورد استفاده قرار می گیرند. که با فراخوانی تابع setupUi می توان این اجزا را ساخت، البته در این جا ساخت این فایل ها به خود PyQt واگذار شده، در واقع ما با استفاده از Qt Disingner و ساخت فایل ui مورد نظر تحت محیط گرافیکی و تبدیل این فایل با استفاده از دستور

python -m PyQt5.uic.pyuic -x [FILENAME].ui -o [FILENAME].py

آن را به فایل python تبدیل کردیم. البته این فایل ساخته شده صرفا ساختار گرافیکی بوده و عملکردی ندارد.

### Containers

توابع مربوط به GUI در این پوشه قرار دارد که عملکرد مربوط به دکمه ها و ... را شامل می شود.

### Instructor

توابع مربوط به زبانه اساتید و زیرپنجره مربوطه را در ادامه بررسی می کنیم.

در ابتدا باید کتابخانه ها و توابع فریمورک qt را به پروژه وارد کنیم. (خط 1)، سپس پایگاه داده و اجزای مرتبط را به پروژه متصل کرده. (خط 2)، وارد کردن فایل ساختار زیر پنجره اساتید به عنوان Parent، و کتابخانه های مهمjson ، os ،… را به برنامه اضافه می کنیم. (خط 4 و 5)، و در آخر کتابخانه مربوط برای ساخت آیکون ها.



کلاس های تعریف شده در پروژه که خود شامل متد هایی می شوند :

#### کلاس Instructor

گرفتن ورودی و ساخت فیلد در دیتابیس و بررسی خطا ها برای اساتید.



هنگامی که بر روی گزبنه ادیت کلیک می شود تابع زیر اطاعات آن استاد را از پایگاه داده دریافت و در زیرپنجره باز شده وارد می کند.



ثبت اطلاعات وارد شده در پایگاه داده در صورت صحیح بودن.



تابع زیر مربوط به ساخت پیغام های خطا در GUI است که در تابع بالا فراخوانی شده اند.



اضافه کردن اساتید به پلیگاه داده ها، در صورت وجود آن استاد (در صورت ویراش شدن اطلاعات)، اطلاعات آپدیت شده و در غیر این صورت اطلاعات به پایگاه داده وارد می شوند.



#### کلاس SortFilterProxyModel

برای ایجاد جست و جو نیاز به استفاده از ProxyModel است. این کلاس بعنوان واسط میان اطلاعت دریافتی از پایگاه داده و اطلاعت ورودی کلاس Tree عمل کرده و با جست جو در فیلدها، آن هایی را که تشابه دارند را جدا می کند و برای نمایش در اختیار کلاس Tree قرار می دهد.



مرتب سازی ستون در فیلتر شده.



فیلتر کردن فیلد ها بر اساس ریجکس دریافتی.



#### کلاس Tree

این کلاس وظیفه نمایش لیست اساتید در زبانه اساتید را بر عهده دارد. این کلاس با فراخوانی مدل tree از QtGui.QStandardItemModel() اقدام به ساخت ساختار این جدول شامل سرآیندها می نماید. و پس از دریافت اطلاعات از پایگاه داده اطلاعات را نمایش می دهد.



تابع زیر برای تغییر وضعیت ***دردسترس*** بودن اساتید استفاده می شود.



تابع زیر برای نمایش اطلاعات در جدول Tree به کار می رود، و ردیف های مربوط به هر استاد شامل گزینه های ستون عملیات را ایجاد می کند. این کار با ساخت یک کلید و تغییر نوشته آن به وسیله کتابخانه qtawesome انجام می پذیرد.



هنگامی که در فیلد مربوط به جست و جو زبانه اساتید تغییری ایجاد می شد تابع زیر اقدام به دریافت آن متن کرده و به ترتیب تابع مربوط به جست و جو و تابع نمایش اطلاعات را مجددا فراخوانی می کند.



تابع زیر به هنگام کلیک بر روی گزینه ویرایش کلاس Instructor فراخوانی می کند. و پس از اتمام فراید اطلاعات جدول را مجددا نمایش می دهد.



این تابه به هنگام کلیک بر روی گزینه حذف اقدام به نمایش پیغام بروی صفحه کرده و بعد از گرفتن تائیدیه کاربر اقدام به حذف آن داده از پایگاه داده می کند و مجدد جدول را بارگذاری می کند.



### Subject

توابع مربوط به زیر پنجره و زبانه درس ها :

از توضیح کلاس ها و توابع مشابه خوداری شده است.

#### کلاس Subject

کلاس اصلی درس ها که در ادامه به توضیح بخش های آن می پردازیم.

تابع init این کلاس ابتدا به ایجاد صفحه و اتصال دکمه ها و فیلد جست و جو به توابع مربوط به خود.



به صورت مشابه وظیفه تابع زیر دریافت و وارد کردن اطلاعات در زیر پنجره ایجاد شده به هنگام ویرایش درس مورد نظر است.



تابع زیر اقدام ساخت لیست اساتید در زیر پنجره ایجاد کلاس می کند، این کار توسط model Tree صورت می گیرد، همچنین برای ایجاد قابلیت جست و جو از Proxy Model استفاده شده است.



این تابع به هنگام کلیک بر روی دکمه ایجاد و بستن ابتدا اطلاعات را ذخیره کرده و زیرپنجره را می بندد.



این تابع اقدادم به دریافت اطلاعات از زیرپنجره کرده و در صورت صحیح بودن آن، اطلاعات را در پیگاه داده ذخیره می کند. در غیر این صورت پیغام خطا را به کاربر نشان می دهد.



### Room

\*\* کلاس مشابه کلاس های اساتید.

### Section

\*\* کلاس مشابه کلاس های درس ها.

1. good enough [↑](#footnote-ref-1)
2. Big Data [↑](#footnote-ref-2)
3. Data Science [↑](#footnote-ref-3)
4. adapt [↑](#footnote-ref-4)
5. Evaluation [↑](#footnote-ref-5)
6. Fitness [↑](#footnote-ref-6)
7. Subject Placement [↑](#footnote-ref-7)
8. Evaluation [↑](#footnote-ref-8)
9. Population Adaption [↑](#footnote-ref-9)
10. Minimum Population [↑](#footnote-ref-10)
11. Maximum Population [↑](#footnote-ref-11)
12. Mutation Rate Adjustment [↑](#footnote-ref-12)
13. mutation rate adjustment trigger [↑](#footnote-ref-13)
14. Selection [↑](#footnote-ref-14)
15. Selection [↑](#footnote-ref-15)
16. Tournament Selection [↑](#footnote-ref-16)
17. Crossover [↑](#footnote-ref-17)
18. Offspring [↑](#footnote-ref-18)
19. Mating pool [↑](#footnote-ref-19)
20. Crossover [↑](#footnote-ref-20)
21. Order Crossover (ox) [↑](#footnote-ref-21)
22. Local Beam [↑](#footnote-ref-22)
23. adaptive-elitist genetic algorithm [↑](#footnote-ref-23)
24. Minimum Population Count [↑](#footnote-ref-24)
25. Maximum Population [↑](#footnote-ref-25)
26. Maximum Generations [↑](#footnote-ref-26)
27. Maximum Creation Atempt [↑](#footnote-ref-27)
28. Mutation Rate Adjustment Trigger [↑](#footnote-ref-28)
29. Maximum Fitness [↑](#footnote-ref-29)
30. Elite Population [↑](#footnote-ref-30)
31. Deviation Telorance [↑](#footnote-ref-31)