

به نام ایزد منان

پروژه چهارم درس مبانی هوش محاسباتی، «الگوریتم‌های تکاملی»



استاد درس: دکتر عبادزاده

پاییز ۱۴۰۳ - دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

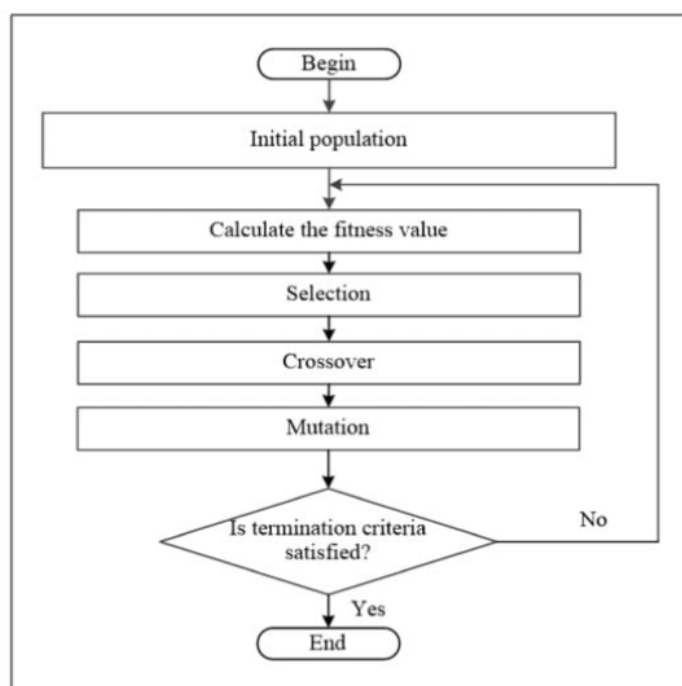


نکاتی در مورد این تمرین نیاز به توجه و دقت دوستان دارد.

- ۱- هرگونه کپی کردن باعث عدم تعلق نمره به تمامی افراد مشارکت کننده در آن می شود.
- ۲- استفاده از چت بات ها در صورت کشف باعث عدم تعلق نمره به پروژه می شود.
- ۳- آخرین مهلت ارسال تمرین، ساعت ۲۳:۵۵ دقیقه روز ۳۰ آذر ۱۴۰۳ می باشد.
- ۴- لطفاً فایل ارسالی خود را به صورت فایل PDF و به صورت «شماره دانشجویی_PRJ4» مانند PRJ4_40031000 نام گذاری کنید (در صورت عدم رعایت نمره کسر می شود).
- ۵- در صورت هرگونه سؤال یا مشکل می توانید با تدریس یاران درس از طریق discussion کانال در ارتباط باشید.

الگوریتم‌های تکاملی

الگوریتم‌های تکاملی الگوریتم‌های بهینه‌سازی الهام‌گرفته از انتخاب طبیعی هستند که در آن، راه‌حل‌های ممکن به عنوان افراد^۱ یک جمعیت در نظر گرفته می‌شوند. الگوریتم با تولید جمعیتی اولیه از راه‌حل‌های تصادفی شروع می‌شود و سپس این جمعیت را از طریق عملیاتی به صورت تکراری تکامل می‌دهد: انتخاب والدین، باز ترکیب و جهش، ارزیابی، انتخاب بازماندگان. در هر تکرار، افراد براساس یک سیاستی انتخاب شده و باز ترکیب شده و جهش داده می‌شوند تا جمعیت جدیدی تشکیل دهند. در نسل‌های متوالی، جمعیت به طور ایده‌آل به سمت راه‌حل‌های بهینه یا نزدیک به بهینه برای مسئله مورد نظر همگرا می‌شود. این دسته از الگوریتم‌ها معمولاً وقتی الگوریتم‌های سنتی به دلیل NP-hard بودن مسئله، بزرگ بودن فضای مسئله کارآمد نیستند و ...، مورد استفاده قرار می‌گیرند. در تصویر زیر شمایی کلی از این دسته از الگوریتم‌ها و مراحل انجام کار آن‌ها را می‌توانید مشاهده کنید.



مسئله فروشنده دوره‌گرد

مسئله فروشنده دوره‌گرد یک مسئله کلاسیک بهینه‌سازی است. در این مسئله، یک "فروشنده" باید به مجموعه‌ای از شهرها سر بزند، به گونه‌ای که هر شهر را فقط یک بار بازدید کرده و مجموع فاصله یا هزینه سفر را به حداقل

¹ Individuals

می‌رساند. نسخه‌های گوناگونی از این مسئله وجود دارد که در هر کدام شرط‌های ویژه‌ای از جمله "بازگشت به شهر اولیه"، "محدودیت زمانی" و ... به مسئله اضافه می‌شود.

تسک اول: حل مسئله فروشنده دوره‌گرد با استفاده از الگوریتم‌های تکاملی

در این پروژه ما قصد داریم مسئله فروشنده دوره‌گرد را با استفاده از یک الگوریتم تکاملی دلخواه حل کنیم. برای این کار تعداد فایل در اختیار شما قرار گرفته است که در ادامه توضیح مختصری در ارتباط با هر یک از مراحل الگوریتم و این فایل‌ها داده می‌شود.

آماده‌سازی مجموعه‌دادگان

مجموعه‌دادگان این پروژه در فایل tsp.txt قرار دارد. همانگونه که مشاهده می‌کنید این مجموعه‌داده شامل ۳۰ سطر است که در هر سطر مختصات یک شهر نوشته شده است (x, y). با فرض اینکه گراف ما گراف کامل است یعنی از هر شهر به شهر دیگر مسیری وجود دارد، تابع گراف همسایگی را در فایل dataset_preparatoin.py پیاده‌سازی کنید.

بازنمایی داده

بازنمایی در الگوریتم‌های تکاملی به معنای نحوه نمایش داده‌های مسئله در قالب افراد است. به بیانی دیگر، بازنمایی تعیین می‌کند که ژنوم هر فرد چگونه به صورت داده‌هایی همانند بیت، رشته‌ها و ... نمایش داده شود تا الگوریتم بتواند از آن برای تولید جمعیت و اعمال عملیات‌های موصوف استفاده کند. در این مسئله پیشنهاد میشود از بازنمایی جایگشت استفاده کنید. به این منظور باید ماژول individual.py را پیاده‌سازی کنید.

ارزیابی شایستگی

تمامی پاسخ‌ها در جمعیت یک معیار شایستگی^۲ دارند که با توجه با همان مسئله تعریف شده و نشان می‌دهد جواب‌ها تا چه اندازه مطلوب هستند. به بیانی دیگر، تابع ارزیابی باید یک معیار برای مقایسه دو پاسخ در آن مسئله ارائه دهد. حال شما باید معیار مناسب در مسئله فروشنده دوره‌گرد را پیدا کرده و سپس آن را در فایل evaluation.py پیاده‌سازی کنید.

^۲ fitness

انتخاب والدین

برای تولید جمعیت نسل بعد و یا انجام عملیات های بازترکیبی نیاز به انتخاب یک سری از افراد از جمعیت فعلی به عنوان والد است. برای این کار نیاز است تابع `select_two_individual_for_crossover` در فایل `selection.py` را پیاده سازی کنید. اینکه انتخاب والدین با جایگذاری و یا بدون جایگذاری باشد دلخواه است.

بازترکیبی

بازترکیبی یکی از عملیات های اصلی در الگوریتم های تکاملی است که در آن با ترکیب ژنوم های دو والد انتخابی، دو فرزند تولید می شود (البته تعداد فرزندان وابسته به الگوریتم است و لزوماً دو نیست!). برای انجام این کار فایل `crossover.py` را پیاده سازی کنید.

جهش

جهش در الگوریتم های تکاملی یکی دیگر از عملیات های اصلی است که در آن تغییر تصادفی در ژنوم صورت می گیرد. این تغییر معمولاً به طور تصادفی و با احتمال کم اتفاق می افتد و هدف آن افزایش تنوع جمعیت است. برای این مورد فایل `mutation.py` را پیاده سازی کنید. دقت کنید که اگر از بازنمایی جایگشت استفاده می کنید، بازترکیبی و جهش تعریف شده نباید به گونه ای باشد که جایگشت را به هم بزند.

انتخاب بازماندگان

در این مرحله باید افرادی را به عنوان جمعیت نسل بعد انتخاب کنید. این کار معمولاً به دو صورت $\mu + \lambda$ و $\mu \lambda$ انجام می شود. روش انتخاب بازماندگان بر عهده شماست و حتی روش های دیگر نیز قابل قبول است.

شرط خاتمه

یکی از مهمترین مراحل در الگوریتم های تکاملی بررسی شرط خاتمه است که با رویکردهای مختلفی بررسی می شود. در این پروژه ما از حد شایستگی استفاده می کنیم. برای پیاده سازی این مورد و همچنین اطلاع از حد شایستگی به فایل `termination_condition.py` مراجعه کنید.

تکرار این عملیات‌ها تا اتمام الگوریتم

در بالا ما ابزار مورد نیاز برای الگوریتم تکاملی را پیاده‌سازی کردیم در این مرحله باید از این ابزارهای برای پیاده‌سازی روند موصوف در تصویر اول استفاده کنیم. برای این کار یک تابع `run_algorithm` در فایل `optimizer.py` وجود دارد که شما باید قسمت‌های خالی این فایل را کامل کنید. همچنین برای تولید جمعیت اولیه باید تابع `primary_population_creator` را پیاده‌سازی کنید.

اجرای الگوریتم

برای اجرای این الگوریتم در روی مجموعه‌داده‌گانی که در ابتدا آن را آماده کردیم، باید فایل `main.ipynb` را پیاده‌سازی کنید. در انتهای این فایل نیز یک قطعه کد برای نمایش خروجی وجود دارد که نیاز به به تغییر و یا پیاده‌سازی ندارد (قسمت `visualization`).

تسک امتیازی اول:

پیاده‌سازی صحیح بازترکیبی جدولی تدریس شده در کلاس نمره امتیازی دارد.

تسک امتیازی دوم:

تکمیل کردن تمیز پیاده سازی ها نمره امتیازی دارد.

نکات پیاده سازی:

- فایل‌های پروژه بر روی کورسز آپلود شده‌اند برای انجام تسک‌ها به تکمیل این فایل‌ها بپردازید. لازم به ذکر است که در صورت نیاز به تغییر سایر بخش‌ها باید به صورت خلاصه دلیل خود را برای تغییر آن قسمت توضیح دهید.
- پیشنهاد می‌شود برای اجرای سریع تر از Google Colab استفاده کرده و به T4GPU متصل شوید.
- انتخاب الگوریتم تکاملی به دلخواه دانشجو است اما باید در قالب قرار گرفته پیاده‌سازی شود.