

سوال:

حل مسئله One-Max با استفاده از الگوریتم ژنتیک

نکته مهم:

این مسئله جز نمره اصلی درس قرار نمی گیرد و صرفاً جهت آشنایی شما با یک نمونه مسئله الگوریتم ژنتیک می باشد.

مقدمه:

الگوریتم های ژنتیک (Genetic Algorithms) یکی از روش های فرا ابتکاری در حوزه ی بهینه سازی و هوش مصنوعی هستند که از فرایند تکامل طبیعی الهام گرفته اند. در این الگوریتم ها، مجموعه ای از جواب های کاندید تحت فرآیند هایی مانند انتخاب، ترکیب (Crossover) و جهش (Mutation) قرار می گیرند تا به سمت یک راه حل بهینه حرکت کنند.

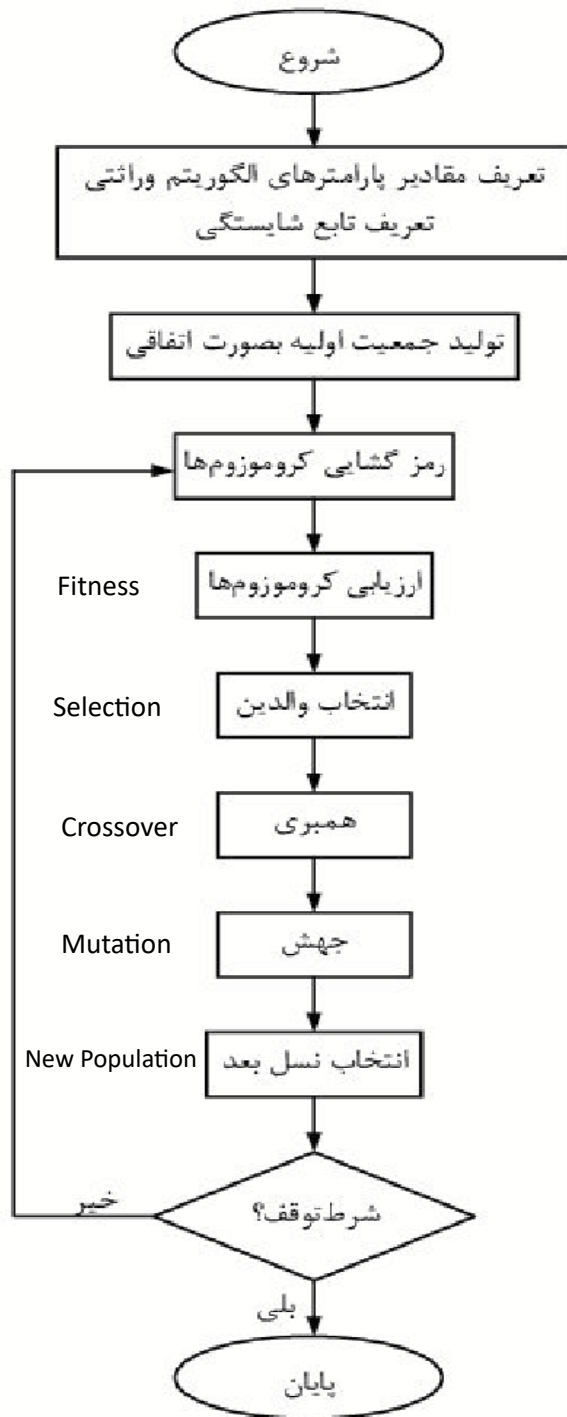
مسئله:

در این تمرین، شما باید یک الگوریتم ژنتیک برای حل مسئله ی One-Max پیاده سازی کنید. مسئله ی One-Max یکی از ساده ترین مسائل در زمینه ی الگوریتم های ژنتیک است که هدف آن یافتن رشته ای از اعداد باینری (۰ و ۱) با بیشترین تعداد ۱ است.

شرح مسئله:

- یک جمعیت اولیه شامل چند رشته باینری (ژنومها) تولید کنید.
 - هر ژنوم شامل تعدادی بیت (0 و 1) است.
 - تابع ارزیابی (Fitness Function) تعداد 1 های موجود در هر رشته را محاسبه کند.
 - از روش های مختلف برای انتخاب والدین استفاده کنید.
 - از ترکیب تک نقطه‌ای (Single-Point Crossover) با احتمال دلخواه (مثلا 70 یا 75) برای تولید فرزندان (Offspring) یا ترکیب دیگری استفاده کنید.
 - از جهش (Mutation) با احتمال کم مثلا 0.1 درصد برای تغییر تصادفی بیت های ژنوم استفاده کنید.
 - این فرایند را برای تعداد نسل دلخواه (200 یا 300) تکرار کنید.
 - در هر نسل، بهترین مقدار Fitness را نمایش دهید.
 - در نهایت، بهترین ژنوم و مقدار Fitness آن را نمایش دهید.
- در مسئله ژنتیک مهم ترین بخش فهمیدن مسئله و بدست آوردن روش و پارامتر های حل مسئله می باشد به عنوان مثال اگر بخواهیم برای این مسئله ژنتیک پارامتر های ورودی رو انتخاب کنیم:
- تعداد افراد جمعیت: 50
 - طول هر ژنوم: 20
 - نرخ جهش: 0.02
 - نرخ ترکیب: 0.65
 - تعداد نسلها: 300

روش حل یک مسئله با الگوریتم ژنتیک در تصویر زیر:



خروجی:

- مقدار Fitness بهترین فرد در هر نسل

- بهترین ژنوم به دست آمده در پایان اجرا

- مقدار Fitness بهترین ژنوم

نکات مهم:

1. کد شما باید یک جمعیت اولیه تصادفی از ژنوم‌ها ایجاد کند.

2. برای هر نسل، باید فرایند انتخاب، ترکیب و جهش را پیاده‌سازی کنید.

3. الگوریتم باید تکامل را برای تعداد نسل دلخواه مثلاً 200 اجرا کند و بهترین فرد را نمایش دهد.

4. کد شما باید دارای توضیحات کافی برای هر تابع باشد تا درک آن برای دیگران آسان باشد.

💡 کاربرد: این مسئله نمونه‌ای ساده از الگوریتم‌های ژنتیک است که می‌تواند در مسائل پیچیده‌تری مانند برنامه‌ریزی زمانی، بهینه‌سازی مسیر، و یادگیری ماشین استفاده شود.

✅ وظیفه شما: مسئله را تحلیل کنید، و کد آن را بنویسید و اجرا کنید و درک کنید که چگونه الگوریتم ژنتیک برای حل این مسئله کار می‌کند. سپس تلاش کنید که پارامترها را تغییر دهید و تأثیر آن را بر نتایج مشاهده کنید.

مسئله:

🎵📺 برنامه ریزی پخش آهنگ بین تبلیغات تلویزیونی

نکته مهم:

برای حل مسائل و مقایسه دو الگوریتم گفته شده ابتدا شما باید state ها رو درست کرده و بعد الگوریتم رو جداگانه ران کرده و نتایج رو مقایسه کنید. (برای این کار می توانید از Jupyter نیز استفاده کنید)

مقدمه:

یک شبکه تلویزیونی در نظر بگیرید که در فواصل بین برنامه ها، تبلیغات مختلفی پخش می کند. بین این تبلیغات، بازه های زمانی خالی وجود دارد که می خواهیم در آن ها آهنگ هایی را پخش کنیم تا کل بازه به بهترین شکل پر شود، بدون اینکه زمانی بدون موسیقی باقی بماند یا آهنگی ناقص قطع شود.

جزئیات مسئله:

- 40 آهنگ مختلف داریم که مدت زمان هر آهنگ تصادفی بین 2 تا 3 دقیقه است.
- 3 تا بازه بین تبلیغاتی وجود دارند که هر کدام بین 10 تا 15 دقیقه هستند.
- هدف ما این است که یک مجموعه مناسب از آهنگ ها انتخاب کنیم تا این بازه ها به طور کامل پر شوند و زمان پرت (بدون موسیقی) نداشته باشیم.
- امکان تکرار آهنگ ها نداریم.
- اگر چندین ترکیب مختلف از آهنگ ها وجود داشته باشد که یک بازه را پر کنند، ترکیبی که از نظر تعداد آهنگ کمترین قطعه را داشته باشد، ارجح است (تا تعداد تغییرات کمتر باشد).

صورت سوال و محدودیت ها:

برای حل این مسئله شما باید دو الگوریتم (ژنتیک و Simulated Annealing) را پیاده سازی کرده و به سوالات آخر صفحه پاسخ دهید و در آخر جواب سوالات و کد رو ارسال کنید.

نکته:

- 1- آهنگی نباید در وسط قطع شود.
- 2- آهنگی نباید از زمان تبلیغات بیشتر شود (اگر در انتها (زمان نزدیک به پایان تبلیغات) چند ثانیه خالی بود مشکلی نیست اما نباید بیشتر یا خیلی کمتر از زمان تبلیغات باشد).
- 3- هدف این است که با کمترین تعداد آهنگ ممکن بازه ها پر شوند، تا کمترین وقفه و تغییر وجود داشته باشد.

راهنمایی:


1. الگوریتم ژنتیک (Genetic Algorithm - GA)

- تولید جمعیت اولیه از ترکیب های تصادفی آهنگ ها
- ارزیابی هر ترکیب بر اساس میزان تطبیق با بازه زمانی
- اعمال انتخاب، تقاطع (Crossover) و جهش (Mutation) برای یافتن ترکیب های بهینه

2. تبرید شبیه سازی شده (Simulated Annealing - SA)

- انتخاب یک ترکیب اولیه تصادفی از آهنگ ها
 - ایجاد تغییرات کوچک و بررسی تأثیر آن ها بر پر شدن بازه زمانی
 - پذیرش تغییرات سودمند و برخی تغییرات غیر سودمند با احتمال کاهش یابنده
- ممکن است که در حل مسئله به عدم وجود ترکیب دقیق از آهنگ ها برای پر کردن یک بازه یا چرخش زیاد بین آهنگ های تازه برخورد کنید که باید حواستون به این مشکل باشد.

خروجی مورد انتظار:

در نهایت، باید یک مجموعه بهینه از آهنگ ها برای هر بازه بین تبلیغاتی پیشنهاد شود که نه بازه خالی باقی بماند و نه آهنگی ناقص قطع شود  .

سوالات:

- ◆ کدام روش به نظرتون (الگوریتم ژنتیک یا تبرید شبیه سازی شده) راه حل بهینه تری ارائه می دهد؟
- ◆ سرعت convergence (همگرایی) در کدام الگوریتم بیشتر است یا سریعتر بهش می رسد؟
- ◆ کدام الگوریتم در یافتن جواب های نزدیک به بهینه در زمان کوتاه تر بهتر عمل می کند؟
- ◆ اگر یک بازه مثلاً 12 دقیقه باشد و هیچ ترکیب دقیقی از آهنگ ها برابر 12 نشود، چه راهکاری پیشنهاد می دهید؟
- ◆ اگر بخواهیم محبوبیت آهنگ ها را هم در نظر بگیریم، چگونه می توانیم مدل را بهبود دهیم؟
- ◆ آیا بهتر است ابتدا آهنگ های طولانی تر را انتخاب کنیم یا ترکیب های تصادفی بسازیم؟
- ◆ اگر به الگوریتم های جستجوی محلیمون یک نوع هوشمندی مثلاً ابتدا الگوریتم آهنگ های کوتاه بعد بلند را انتخاب کند یا بازه تبلیغات رو بگیرد و با توجه به بازه زمانی تبلیغات تقسیم بندی آهنگ را انجام دهد (مثلاً اگر بازه تبلیغات 10 دقیقه باشد میشه 5 تا آهنگ 2 دقیقه ای) اضافه کنیم، چه مزایا و معایبی ایجاد می کند؟