

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



هوش مصنوعی

گزارش پروژه شماره ۲

نام و نام خانوادگی: سجاد علی زاده

شماره دانشجویی: ۸۱۰۱۹۷۵۴۷

مفاهیم ژن و کروموزوم و جمعیت اولیه

ژن: هر یک از گیت‌های منطقی را یک کروموزوم در نظر می‌گیریم. بدیهی است که هر ژن می‌تواند یکی از گیت‌های AND, OR, XOR, NAND, NOR, XNOR باشد.

کروموزوم: به مجموعه‌ای با ترتیب از ژن‌ها یک کروموزوم می‌گوییم. به عنوان مثال اگر ده ورودی داشته باشیم کروموزوم‌هایی با ۹ ژن جمعیت اولیه را می‌سازند. یعنی هر جواب ممکن برای مسئله یک کروموزوم است.

در ابتدا به صورت تصادفی یک جمعیت اولیه از کروموزوم‌ها را تولید می‌کنیم. (یعنی انتخاب گیت‌ها برای هر کروموزوم تصادفی است) اندازه این جمعیت اولیه را برابر $\frac{1}{4000}$ تعداد کل پاسخ‌ها در نظر می‌گیریم. در هر مرحله سعی می‌کنیم این مجموعه پاسخ را بهتر کنیم تا نهایتاً به پاسخ مطلوب دست پیدا کنیم.

اگر اندازه جمعیت اولیه بیش از حد کم باشد باعث می‌شود diversity این مجموعه کم شود و نتوانیم از پاسخ‌های موجود به پاسخ نهایی برسیم و اگر بیش از حد زیاد باشد باعث افزایش زمان اجرای الگوریتم می‌شود.

در هر گام از الگوریتم یک selection بر روی جمعیت اولیه انجام می‌شود. بر اساس fitness (که در ادامه تعریف آن ذکر خواهد شد) یک مجموعه از مجموعه اولیه بدون جایگزینی انتخاب می‌شود تا بر روی آن اعمال crossover و mutation انجام شود. این عملیات به صورت rank base انجام می‌شود تا گوناگونی پاسخ حفظ شود.

معیار سازگاری، crossover، mutation

به ازای هر کروموزوم، تعداد ردیف‌هایی که آن کروموزوم میتواند ارضا کند را به عنوان fitness در نظر می‌گیریم. به عنوان مثال اگر یک کروموزوم به ازای ۵۰۰ ردیف از truth table پاسخ مورد انتظار را بدهد fitness آن برابر با ۵۰۰ خواهد بود.

عملیات crossover به این صورت شکل می‌گیرد که یک عدد تصادفی در بین صفر و طول کروموزوم انتخاب می‌شود. با این عدد هر یک والد‌ها به دو قطعه تبدیل می‌شود. حال دو فرزند ایجاد می‌کنیم که تکه اول فرزند اول تکه اول والد اول و تکه دوم آن تکه دوم والد دوم است و برای فرزند دوم بالعکس. (one point crossover). این عملیات به احتمال pc انجام می‌شود. pc برابر مقدار زیر است:

$$pc = 1 - \frac{\text{میانگین فیتنس کروموزوم‌های جمعیت}}{\text{فیتنس هدف}}$$

هر چه به مقدار هدف نزدیک‌تر باشیم بهتر است کمتر زادولد صورت گیرد زیرا وقتی دو کروموزوم زادولد می‌کنند به دلیل ماهیت fitness تعریف شده ممکن است فرزندان آنها از پاسخ هدف دورتر شوند. در صورتی که پدر و مادر آنها به هدف نزدیک‌تر بودند. به عبارت دیگر وقتی یک کروموزوم به هدف نزدیک است بهتر است آن را نگه داریم تا زادولد کنیم. به همین دلیل هر چه به هدف نزدیک‌تر شویم مقدار احتمال بالا کمتر می‌شود.

برای عملیات mutation به صورت زیر عمل می‌کنیم:

ابتدا چک می‌کنیم که آیا به پاسخ صحیح رسیده‌ایم یا خیر. اگر به پاسخ صحیح نرسیده بودیم به احتمال pm یکی از ژن‌های کروموزوم را تغییر می‌دهیم. انتخاب این ژن به صورت تصادفی خواهد بود. احتمال pm در ابتدا برابر یک تقسیم بر اندازه جمعیت است اما اگر گام‌های الگوریتم از یک مقدار خاص بیشتر شد (در این مسئله این مقدار برابر یک دهم اندازه جمعیت در نظر گرفته شده) این احتمال برابر با یک تقسیم بر اندازه پاسخ خواهد شد. دلیل این اتفاق این است که الگوریتم در یک استیت خاص گیر نکند و اگر پاسخ‌های یک جمعیت یکنواخت شد بتواند پاسخ‌های جدید تولید کند.

سوالات

پرسش اول: معیار تناسب باید نمودی از مناسب بودن یک پاسخ برای مسئله باشد. یعنی بر اساس آن بتوان بحث کرد که چه مقدار پاسخ مورد نظر برای مسئله خوب است و شرایط آن را ارضا میکند. همچنین بر اساس آن باید بتوان بین دو جواب مختلف مقایسه انجام داد و تعیین کرد کدامیک برای مسئله بهتر هستند. تعداد ردیف‌هایی که جواب مورد نظر برای آنها پاسخ مورد انتظار را می‌دهد شرایط یک معیار تناسب خوب را داراست.

پرسش دوم: همانطور که ذکر شد انتخاب افراد برای تولید جمعیت بعدی rank base است. نکته مثبت این انتخاب حفظ گوناگونی در جمعیت جدید است. فرض کنید انتخاب rank base نباشد. در این صورت اگر اختلاف fitnessها زیاد باشد همواره کروموزومی که fitness آن از همه بیشتر است انتخاب می‌شود. ممکن است این کروموزوم خوب باشد اما به پاسخ نرسد زیرا قسمتی از کروموزوم هدف قسمتی از کروموزومی باشد که fitness آن کمتر از سایرین است. در این صورت در حلقه گیر می‌کنیم و به پاسخ نمی‌رسیم. وقتی انتخاب بر اساس rank باشد در عین اینکه افرادی که fitness بهتری دارند شانس بیشتری برای حضور در جمعیت بعدی خواهند داشت، گوناگونی جمعیت بعدی نیز حفظ می‌شود زیرا احتمال حضور افراد با fitness کمتر به صفر میل نمی‌کند (در صورت وجود گپ بین fitnessها)

پرسش سوم: هر چه به پاسخ نزدیک‌تر می‌شویم احتمال crossover کمتر می‌شود. احتمال mutation اندک است ولی اگر بعد از تعدادی مرحله به پاسخ نرسیدیم نشان می‌دهد که در نقطه‌ای گیر کرده‌ایم و احتمال mutation بیشتر می‌شود. توضیحات کامل در قسمت قبل آورده شده است.

پرسش چهارم: ممکن است الگوریتم در نقطه‌ای گیر کند. زیرا کل جمعیت یکسان می‌شود و همواره یک سری کروموزوم مشخص انتخاب می‌شوند و حتی ممکن است به پاسخ نرسیم. زیرا همواره یک سری کروموزوم ثابت انتخاب می‌شوند. و یا حتی ممکن است به نقطه‌ای برسیم که نتوان از جمعیت موجود به هدف رسید. برای حل این مشکل باید گوناگونی هر جمعیت را بیشتر کرد که راهکارهای زیر در رابطه با همین هدف هستند:

(۱) همانطور که ذکر شد بعد از تعدادی گام احتمال mutation بیشتر شود که همین ایده در پروژه پیاده سازی شده است. قطعاً نمیتوان از ابتدا احتمال mutation را زیاد گذاشت زیرا هر mutation ممکن است به کلی یک کروموزوم را (که ممکن است یکی از کلیدی‌ترین کروموزوم‌ها برای رسیدن به هدف باشد) تغییر دهد.

۲) بعد از تعدادی مرحله احتمال crossover بیشتر شود. این راهکار بستگی به ذات مسئله دارد. در بعضی crossoverها ممکن است فرزندان تولید شده از هدف بسیار دور شوند. (مانند همین مسئله) در کل این روش برای مسائلی که crossover در آنها با احتمال زیادی جمعیت را به هدف نزدیک میکنند و یا باعث ایجاد گوناگونی در جمعیت می‌شوند (به طوری که از هدف فاصله زیادی نگیرند) مناسب است.

۳) می‌توان اندازه جمعیت را بیشتر کرد در نتیجه گوناگونی پاسخ نیز بیشتر می‌شود. البته این روش باعث می‌شود زمان اجرای الگوریتم بیشتر شود که مطلوب ما نیست.

اجرای برنامه

آدرس فایل ورودی به عنوان آرگومان داده می‌شود. با دستور زیر میتوان کد را اجرا کرد:

```
python3 codes.py truth_table.csv
```