

### هدف:

ما در این پروژه می خواهیم با الگوریتم های ژنتیک آشنا شویم. و با استفاده از روش های برگرفته از انتخاب طبیعی، مسئله ی طرح شده را مدل سازی کرده و جوابی برای مسئله بیابیم.

### کلیدواژه:

گیت - جمعیت اولیه - ژن - gate – fitness – selection – crossover – mutation - کروموزم

### خلاصه:

در ابتدا یک جمعیت اولیه باید تعریف کنیم. من این جمعیت را متناسب با تعداد ژن ها یا همان تعداد گیت ها که با ورودی داده می شود تعریف می کنم. (جمعیت اولیه برابر است با 2 به توان تعداد ژن ها). یعنی ژن ها همان گیت های ما هستند. این جمعیت اولیه رندوم و با ژن های رندوم ساخته می شوند. سپس فیتنس را برابر با تعداد بار هایی که می توان با ژن های آن کروموزم به جواب درست رسید تعریف می کنیم. پس برای هر کروموزم این جمعیت اولیه، فیتنس متناظر با آن را حساب می کنیم. سپس در یک حلقه ی بی نهایت تا وقتی به جواب برسیم عملیات های زیر را انجام می دهیم. و تعداد جمعیت را ثابت نگه می داریم.

پس داریم:

ژن: هر کدام از گیت های تعریف شده در مسئله را در حکم یک ژن تعریف می کنیم. ما در این مسئله 6 نوع ژن خواهیم داشت.

کروموزم: مجموعه ای از چینش ژن ها در کنار هم که ژن ها می توانند تکراری هم باشند. در هر کروموزم 9 تا ژن خواهیم داشت.

فیتنس: برابر با تعداد بار هایی که می توان با ژن های آن کروموزم به حالات مختلف جدول درستی، پاسخ درست داد تعریف می شود. به فیتنس، تابع تناسب هم می گوییم.

در ابتدا با سلکشن، از بین جمعیتی که داریم، کروموزم هایی را به صورت رندوم انتخاب می کنیم. با این شرط که هر کروموزم یک وزنی برای انتخاب شدن دارد و آن متناسب با رنک فیتنس آن کروموزم است. در نتیجه کروموزم ها می توانند چند بار هم انتخاب بشوند. ما برای اینکه بهتر و بهینه تر بتوانیم سلکت کنیم، وزن هر کروموزم را برابر با رنک فیتنس آن کروموزم به توان 2 تعریف می کنیم.

سپس این جمعیت سکلت شده را کراس اوور می کنیم. و یک احتمال برای انجام این عملیات تعریف می کنیم. من این احتمال را این گونه تعریف کردم که میانگین فیتنس های کل کروموزم های آن مرحله را حساب کردم، سپس یک عدد رندوم کوچکتر از مقدار هدف نهایی مسئله تولید کردم و این عدد رندوم را با میانگین مقایسه کردم و اگر عدد رندوم کمتر از میانگین باشد عملیات کراس اوور انجام می شود. در هر مرحله نیاز به حدود 60 تا 80 درصد کراس اوور برای کروموزم ها داریم.

از بین جمعیت دوتا والد به صورت رندوم و با شرط غیر تکراری بودن انتخاب می کنیم و اگر شرط احتمال کراس اوور برقرار بود، یک نقطه رندوم را انتخاب و عملیات را انجام می دهیم. و این دو بچه ی جدید را به جمعیت بعدی اضافه می کنیم. و اگر شرط این احتمال برقرار نبود، آن دو والد به صورت مستقیم به جمعیت بعدی اضافه می شوند.

سپس روی این جمعیت کراس اوور شده، عملیات میوتیشن را انجام می دهیم. و یک احتمال برای انجام این عملیات تعریف می کنیم. من این احتمال را این گونه تعریف کردم که مقدار احتمال برابر است با یک تقسیم بر تعداد ژن ها. با این شرط که اگر چند مرحله ماکسیمم فیتنس کروموزم های ما تکراری شد، این احتمال در یک ضربی ضرب می شود تا احتمال انجام گرفتن عملیات برای آن مرحله زیاد شود و سپس بعد از انجام مرحله، این احتمال به مقدار قبلی اش باز می گردد. (جلوگیری از سوگیری کروموزم ها). سپس یک عدد رندوم تولید کردم و این عدد رندوم را با این احتمال مقایسه کردم و اگر عدد رندوم کمتر از احتمال میوتیشن باشد عملیات میوتیشن انجام می شود. در هر مرحله ما نیاز به حدود 10 درصد میوتیشن کروموزم ها داریم.

روی تمام کروموزم ها و روی همه ی ژن های آن بررسی می کنیم که اگر شرط میوتیشن برقرار بود، آن ژن را با ژن جدید و رندوم جایگزین می کنیم و در غیر این صورت آن ژن ثابت می ماند.

سپس بررسی می کنیم که آیا هدف سوال پیدا شده است یا خیر. و اگر پیدا شده است، آن ژن ها را چاپ می کنیم.

هدف مسئله را یافتن کروموزم با چینی از ژن ها که همه ی حالات جدول درستی را پاسخ درست بدهد، تعریف می کنیم.

هایپرپارامتر های مسئله را جمعیت اولیه، فیتنس، احتمال کراس اوور و احتمال میوتیشن و رنک فیتنس ها تعریف می کنیم.

### سوالات:

#### پرسش اول:

تابع تناسب یا فیتنس ما نشان دهنده این است که با هدف تعیین شده برای مسئله، چقدر فاصله داریم. فیتنس ما تعداد بار هایی که کروموزم متناظر با آن، می تواند حالات جدول درستی بپذیرد را شامل می شود. هدف ما هم این است که کروموزمی پیدا کنیم که همه ی حالات جدول درستی را بپذیرد. پس با تابع تناسب خود می توانیم بفهمیم با جواب مسئله چقدر فاصله داریم.

#### پرسش دوم:

در مرحله ی سلکشن، ما از جمعیت بدست آمده قبلی، یک جمعیت رندوم و به دور از جهت گیری انتخاب می کنیم. همانطور که در بالا هم توضیح داده شد از روش وزن دادن به فیتنس ها بر اساس رنک آن فیتنس، و البته توان دادن به رنک ها به دلیل بیشتر شدن تاثیر رشد کردن فیتنس ها استفاده کرده ایم.

#### پرسش سوم:

عمل کراس اوور باعث می شود که کروموزم های جدید، از ژن های خوب کروموزم والدین بهره ببرد و ما را به جواب نزدیک تر کند. طبق توضیحات بالا این احتمال برابر با میانگین فیتنس های کل کروموزم های آن مرحله است. (برای بین صفر و یک شدن این احتمال می توان آن را در این بازه تناظر کرد ولی چون عدد رندوم تولیدی ما هم بین صفر تا مقدار هدف مسئله است در نتیجه اشکالی ایجاد نمی کند).

و عمل میوتیشن باعث می شود که اشکالات کوچک ژن های خاص یک کروموزم که هر چند ممکن است کروموزم خوبی هم باشد را برطرف کرده و با ژن های جدید جایگزین شده و ما را به جواب هدف نزدیک تر کند. طبق توضیحات بالا این احتمال برابر با یک تقسیم بر تعداد ژن ها است. ( در حدود کمتر از 10 درصد میوتیشن)

#### پرسش چهارم:

به دلیل اینکه احتمال رخداد کراس اوور و میوتیشن اگر ثابت بماند، جمعیت ما از یک حدی بیشتر بهتر نمی شود و باید این احتمالات تغییر کند. برای اینکار من احتمال کراس اوور را متغیر و متناسب با میانگین فیتنس

های آن مرحله تعریف کردم. و برای احتمال میوتیشن هم یک ضریب در نظر گرفتم که اگر به همچین مشکلی برخورد کردیم، این ضریب را بالا ببریم.

عکس های خروجی بعد از دو بار اجرای کد:

```
hamid@hamid-ZenBook-UX434FLC-UX434FL:~/Desktop/AI-CA2/Codes$ ./usr/bin/
1 960 1.5742451239999997 s
2 960 3.269154438 s
3 968 4.920569788 s
4 964 6.652189227 s
5 972 8.376805872999999 s
6 994 10.101426235 s
7 1004 11.820465493999999 s
8 1004 13.524545780999999 s
9 1008 15.258658070000001 s
10 1010 17.002955209 s
11 1010 18.783341884000002 s
12 1010 20.559660799 s
13 1012 22.334934723 s
14 1016 24.102654458 s
15 1012 25.894092033 s
16 1016 27.665594210000002 s
17 1020 29.424724336 s
18 1020 31.202669758 s
19 1016 32.956840577 s
20 1016 34.723805869 s
21 1020 36.485267887 s
22 1020 38.243738999 s
23 1024 40.005449086 s
['nand', 'xnor', 'nor', 'xor', 'and', 'or', 'nand', 'xnor', 'nor']
```

```
hamid@hamid-ZenBook-UX434FLC-UX434FL:~/Desktop/AI-CA2/Codes$ ./usr/bin/
1 918 1.339856803 s
2 928 2.725336536 s
3 922 4.134336162 s
4 952 5.581824575000001 s
5 960 7.058683654000001 s
6 1004 8.555976064000001 s
7 1004 10.05053024 s
8 1008 11.595630518 s
9 1012 13.150138659000001 s
10 1016 14.701898695 s
11 1012 16.250304599 s
12 1012 17.896151659 s
13 1012 19.627801196 s
14 1010 21.427178475 s
15 1012 23.235563792 s
16 1016 25.026590796 s
17 1020 26.817610831 s
18 1016 28.6145073 s
19 1016 30.409236199000002 s
20 1024 32.224993442999995 s
['nand', 'xnor', 'or', 'xnor', 'and', 'or', 'and', 'xor', 'nor']
['nand', 'xnor', 'nor', 'xor', 'and', 'or', 'nand', 'xnor', 'nor']
```