



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف اول درس هوش مصنوعی (بخش تئوری)

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۱/۱۴۰۰

مدرس: دکتر حسین فلسفین

دستیاران آموزشی: مجید فرهادی - علی ملاحسینی - آرش وشاق

۱ نمایش مسئله Hop-Constrained Minimum Spanning Tree

هر Candidate solution در این مسئله را می‌توان نظیر یک رشته‌ی باینری به طول تعداد یال‌های موجود در گراف نمایش داد به طوری که صفر (یک) بودن بیت i ام به معنی قرار داشتن (نداشتن) آن یال در مجموعه‌ی یال‌های نظیر آن Candidate solution می‌باشد. همچنین مجموعه‌ی همسایگی نظیر یک Candidate solution را می‌توان به شکل مجموعه‌ی همه‌ی رشته‌هایی که با رشته مذکور دقیقاً در یک بیت اختلاف دارند، تعریف کرد.

۲ نمایش مسئله Equal Piles

هر Candidate solution در این مسئله را می‌توان نظیر یک رشته به طول n از اعداد موجود در مجموعه‌ی G_l در نظر گرفت (نمایش اعداد صحیح) به طوری که i بودن کاراکتر m ام به معنی قرار گرفتن جسم m ام در ستون i ام است. همچنین مجموعه‌ی همسایگی نظیر یک Candidate solution را می‌توان به شکل مجموعه‌ی همه‌ی رشته‌هایی که با رشته مذکور دقیقاً در یک کاراکتر اختلاف دارند، تعریف کرد.

۳ ارزیابی افزایشی تابع هدف برای Quadratic Assignment Problem

هر Candidate solution در این مسئله را می‌توان به یک جایگشت از اعداد ۱ تا n نظیر کرد به طوری که i بودن جایگاه m ام به معنی نگاشت مرکز رفاهی i به مکان m است. مجموعه‌ی شامل همه‌ی جایگشت‌های اعداد ۱ تا n را با P_n نشان می‌دهیم. به ازای هر جایگشت در P_n ، تابع ϕ را تعریف می‌کنیم نگاشت از مرکز رفاهی به مکان نظیر آن مرکز رفاهی در آن جایگشت. همچنین مجموعه‌ی S_n را تعریف می‌کنیم مجموعه‌ی همه‌ی ϕ ها. تابع $d(i, j)$ فاصله‌ی بین دو مکان و تابع $f(i, j)$ جریان بین دو مرکز رفاهی را برمی‌گرداند. در واقع کمینه‌سازی تابع هدف به شکل زیر شکل می‌گیرد:

$$\min_{\phi \in S_n} (\sum_{i=1}^n \sum_{j=1}^n f(i, j) \times d(\phi(i), \phi(j)))$$

در جایگشت همسایگی‌ها به شکل 2-swap می‌توانند تعریف شوند که در این صورت تابع هدف نظیر همسایه‌ی یک Candidate solution تنها با تغییر $f(i, j) \times d(\phi(i), \phi(j))$ به ازای i و j های جایگزین شده میسر است و لازم به محاسبه مجدد همه‌ی عبارات نیست.

۴ الگوریتم ژنتیک برای مسئله Bin Packing

۱.۴ representation (definition of individuals)

نمایش جایگشت نمایش مناسبی برای این مسئله است. هر individual در این مسئله را می‌توان به $d!$ بردار $1 \times n$ نظیر کرد به طوری که n تعداد بسته‌ها و d تعداد جعبه‌های مورد نیاز برای یک individual است و نحوه‌ی قرار گرفتن تعدادی بسته در هر جعبه به این شکل است که جایگشت را از چپ به راست پیمایش می‌کنیم و به صورت حریصانه تا جایی که ممکن است بسته در جعبه‌ی فعلی قرار می‌دهیم و در صورتی که جعبه جا نداشت سراغ جعبه‌ی جدید می‌رویم. البته هر دو جایگشت مختلف الزاماً متناظر با دو individual متفاوت نیستند و می‌توانند نظیر یک individual یکتا باشند (جعبه‌ها یکسان‌اند و ترتیب قرارگیری جعبه‌ها اهمیتی ندارد و صرفاً قرارگیری یا عدم قرارگیری تعدادی بسته در یک جعبه باعث تمایز individualها می‌شود. پس تعداد حالات $d!$ می‌شود). مجموعه‌ی شامل همه‌ی جایگشت‌های اعداد ۱ تا n را با P_n نشان می‌دهیم.

۲.۴ evaluation function (or fitness function)

حال fitness function را می‌توانیم به شکل زیر تعریف کنیم:

$$f : P_n \rightarrow \mathbb{W}$$

$$x \in P_n$$

$$f(x) = \text{Number of required boxes for } x$$

۳.۴ population

جمعیت اولیه را می‌توان متشکل از تعدادی (مثلاً ۴) جایگشت تصادفی از P_n پنداشت.

۴.۴ parent selection mechanism

می‌توان متناسب با مقدار fitness هر عضو از جمعیت، یک احتمال والد شدن به آن عضو نسبت داد و والد را از اعضای جمعیت با یک احتمال خاص انتخاب کرد (مانند مسئله‌ی n وزیر).

۵.۴ variation operators, recombination and mutation**۱.۵.۴ recombination (crossover)**

یک موضع دلخواه انتخاب می‌کنیم و دو والد را از آن موضع یکتا به دو بخش تقسیم می‌کنیم. بخش اول فرزند اول (دوم) را به ترتیب از بخش اول والد اول (دوم) انتخاب می‌کنیم و برای بخش دوم فرزند اول (دوم) والد دوم (اول) را از چپ پیمایش می‌کنیم و هر درایه‌ای که در بخش اول ظاهر نشده بود را به همان ترتیبی که در والد واقع شده است وارد فرزند می‌کنیم تا حالت جایگشت بودن حفظ شود.

۲.۵.۴ mutation

2-swap می‌تواند یک mutation مناسب برای این مسئله باشد. در واقع با یک احتمال دو درایه در فرزند را باهم جابه‌جا می‌کنیم. به این شکل جایگشت حفظ می‌شود.

۶.۴ survivor selection mechanism (replacement)

مکانیسک انتخاب بازمانده می‌تواند به صورت fitness-based یا age-based باشد که در واقع در مکانیسم اول همه اعضای جمعیت را بر اساس fitnessشان رتبه بندی می‌کنیم و رتبه‌های بالا را به عنوان نسل بعد انتخاب می‌کنیم و در مکانیسم دوم می‌توان اعضای جوان‌تر (فرزندها) را صرفاً حفظ کرد و والدها را کلاً کنار بگذاریم.

۵ الگوریتم ژنتیک برای مسئله Magic Square**۱.۵ representation (definition of individuals)**

نمایش جایگشت نمایش مناسبی برای این مسئله است. هر individual در این مسئله را می‌توان به یک بردار $1 \times n^2$ نظیر کرد به طوری که درایه‌ی i ام آن، نظیر درایه‌ی سطر $\lceil \frac{i}{n} \rceil$ و ستون $i \equiv r \pmod{n}$ (که r باقی‌مانده است) در ماتریس Magic Square

است.

مجموعه‌ی شامل همه‌ی جایگشت‌های اعداد ۱ تا i را با P_i نشان می‌دهیم.

۲.۵ evaluation function (or fitness function)

حالت ایده‌آل زمانی است که مجموع همه‌ی درایه‌های سطرها (یا ستون‌ها یا قطرها) باهم برابر شوند. پس مجموع درایه‌های هر سطر، ستون یا قطر (S) برابر است با:

$$n \times S = \sum_{i=1}^{n^2} i = \frac{n^2 \times (1+n^2)}{2} \implies S = \frac{n \times (1+n^2)}{2}$$

حال به عنوان fitness function می‌توانیم مجموع فاصله‌های جمع درایه‌های هر سطر، ستون یا قطر را از ایده‌آل محاسبه کنیم. پس fitness function به شکل زیر تعریف می‌شود:

$$f : P_{n^2} \rightarrow \mathbb{Z}$$

$$f(x) = n(n+1)(n^2+1) - (\sum_{i=1}^n (\sum_{j=1}^n |S - x_{i,j}|) + \sum_{j=1}^n (\sum_{i=1}^n |S - x_{i,j}|) + \sum_{i=1}^n |S - x_{i,i}| + \sum_{i=1}^n |S - x_{i,n-i+1}|)$$

توجه شود چون معمولاً در الگوریتم‌های تکامل مسئله را بیشینه‌سازی در نظر می‌گیرند، در اینجا عبارت را از یک مقدار بیشینه کم کردیم تا با یک مسئله‌ی بیشینه‌سازی روبه‌رو باشیم. در واقع فیتنس بهترین individual برابر $n(n+1)(n^2+1)$

۳.۵ population

جمعیت اولیه را می‌توان متشکل از تعدادی (مثلاً ۴) جایگشت تصادفی از P_{n^2} پنداشت.

۴.۵ parent selection mechanism

می‌توان متناسب با مقدار fitness هر عضو از جمعیت، یک احتمال والد شدن به آن عضو نسبت داد و والد را از اعضای جمعیت با یک احتمال خاص انتخاب کرد (مانند مسئله‌ی n وزیر).

۵.۵ variation operators, recombination and mutation

۱.۵.۵ recombination (crossover)

یک موضع دلخواه انتخاب می‌کنیم و دو والد را از آن موضع یکتا به دو بخش تقسیم می‌کنیم. بخش اول فرزند اول (دوم) را به ترتیب از بخش اول والد اول (دوم) انتخاب می‌کنیم و برای بخش دوم فرزند اول (دوم) والد دوم (اول) را از چپ پیمایش می‌کنیم و هر درایه‌ای که در بخش اول ظاهر نشده بود را به همان ترتیبی که در والد واقع شده است وارد فرزند می‌کنیم تا حالت جایگشت بودن حفظ شود.

۲.۵.۵ mutation

2-swap می‌تواند یک mutation مناسب برای این مسئله باشد. در واقع با یک احتمال دو درایه در فرزند را باهم جابه‌جا می‌کنیم. به این شکل جایگشت حفظ می‌شود.

۶.۵ survivor selection mechanism (replacement)

مکانیسم انتخاب بازمانده می‌تواند به صورت fitness-based یا age-based باشد که در واقع در مکانیسم اول همه اعضای جمعیت را بر اساس fitnessشان رتبه بندی می‌کنیم و رتبه‌های بالا را به عنوان نسل بعد انتخاب می‌کنیم و در مکانیسم دوم می‌توان اعضای جوان‌تر (فرزندها) را صرفاً حفظ کرد و والدها را کلاً کنار بگذاریم.

منابع