

درس هوش مصنوعی

تمرین شماره ۲: بررسی الگوریتم ژنتیکی مرکب (hybrid) با جستجوی تپهنوردی

«بخش تکمیلی»

در بخش تکمیلی این تمرین دانشجویان باید علاوه بر آزمایش الگوریتم ژنتیکی مرکب پیاده‌سازی شده بر روی مسأله کوله‌پشتی آن را بر روی مسأله زمان‌بندی خودکار کلاس‌های درسی که در ادامه توضیح داده خواهد شد امتحان کنند. به این منظور دانشجویان می‌توانند با تبدیل نحوه نمایش حالت‌های مسأله جدید (متغیرها و مقادیرشان) به نمایش دودویی همچنان از رشته‌های بیتی برای نمایش افراد استفاده کنند یا امکان نمایش افراد جمعیت با استفاده از رشته‌های عددی را فراهم کنند که قطعاً منجر به جستجوی دقیق‌تری خواهد شد. در صورت انتخاب گزینه دوم عملگرهای جهش و تقطیع باید در صورت لزوم تغییر یابند تا با رشته‌های عددی (و نه لزوماً بیتی) کار کنند. برای توضیحات بیشتر به اسلایدهای درس مراجعه کنید.

سامانه زمان‌بندی خودکار کلاس‌های درسی: هدف پیاده‌سازی برنامه‌ای برای زمان‌بندی خودکار کلاس‌های درسی در طول هفته برای یک مؤسسه آموزشی به صورت یک مسأله بهینه‌سازی قید (COP – constraint optimization problem) – دسته‌ای خاص از مسائل ارضاء قید (CSP) – است. فرق اصلی مسائل بهینه‌سازی قید و مسائل ارضاء قید در داشتن برخی توابع هدف در کنار قیدهاست، به این معنی که در اینگونه از مسائل به دنبال انتساب مقادیری به متغیرهای مسأله هستیم که علاوه بر ارضاء کردن تمام قیدهای مسأله، تا حد ممکن توابع هدف را نیز بهبود دهند. از این رو به توابع هدف در اینگونه مسائل معمولاً قیدهای ترجیحی نیز می‌گویند.

برای پیاده‌سازی سامانه گفته شده، دانشجویان باید ابتدا با شناسایی متغیرها، دامنه‌های آنها، قیدهای اصلی و قیدهای ترجیحی، یک مسأله بهینه‌سازی قید تعریف کرده و سپس با بکارگیری الگوریتم ژنتیکی مرکب پیاده‌سازی شده اقدام به حل مسأله تعریف شده کنند. از آنجا که این الگوریتم جستجو تمام مقداردهی‌های ممکن را فارغ از اینکه قیدهای اصلی مسأله را ارضاء می‌کند یا خیر، مورد بررسی قرار می‌دهد، بکارگیری تکنیک‌هایی برای محدود کردن راه‌حل‌های پیدا شده توسط الگوریتم به راه‌حل‌های سازگار ضروری است. استفاده از تکنیک‌های استنتاج (انتشار قید) برای محدود کردن دامنه متغیرها پیش یا در حین جستجو، بکارگیری نمایشی از راه‌حل‌ها همراه با عملگرهای جهش و تقطیع خاص که فقط مقداردهی‌های سازگار به متغیرها را در نظر بگیرد، و اصلاح راه‌حل‌های ناسازگار تولید شده در حین جستجو می‌تواند برخی راهکارهای ممکن باشد.

توصیف مسأله

در یک مؤسسه آموزشی تعدادی کلاس درس (C) توسط اساتید خاصی (p) در زمان‌هایی در طول هفته (t) و در مکان‌های مشخصی (r) ارائه می‌شود. لیست دروس ارائه شده و تعداد دانشجویان ثبت‌نام کرده در آنها، استاد ارائه دهنده هر درس، مدت زمان لازم برای ارائه هر درس و مکان‌های موجود در مؤسسه برای ارائه کلاس‌ها به همراه ظرفیت هر یک از آنها به عنوان ورودی به سامانه داده می‌شود. خروجی سامانه باید ساعت و روز تشکیل کلاس و مکان آن را برای هر درس داده شده مشخص کند. در پیدا کردن زمان و مکان مناسب برای برگزاری کلاس درس باید نکات زیر حتماً رعایت شود (قیدهای اصلی):

- هر استاد در هر زمان فقط می‌تواند در یک کلاس درس حضور داشته باشد.

- در هر اتاق در هر زمان فقط یک درس تشکیل می‌شود.
- هیچ کلاس درسی نمی‌تواند پیش از مدت زمان تعیین شده برای آن تمام شود.
- تعداد دانشجویان ثبت‌نام کرده در درس باید حداکثر برابر با ظرفیت مکان تشکیل کلاس آن درس باشد.
- بازه زمانی تشکیل کلاس‌های درس در هر روز از هفته محدود بوده و به عنوان ورودی به برنامه داده می‌شود.

در کنار ضرورت‌های اجباری بالا برای تعیین زمان و مکان کلاس‌های درس، برخی ترجیحات نیز در مورد نحوه تشکیل کلاس‌ها به شرح زیر وجود دارد (قیدهای ترجیحی):

- 1. به دلیل بار مالی برای مؤسسه هر استاد روزهای کمتری مجبور به حضور در مؤسسه باشد.
- 2. تا حد ممکن کلاس‌های درس در طول روزهای هفته دارای پراکندگی یکنواخت باشند. برای بررسی یکنواختی پراکندگی می‌توان مجموع ساعت کلاس‌های درس در هر روز را در نظر گرفته و سعی در یکسان‌سازی آنها کرد.
- 3. کلاس‌های مختلف یک استاد در مکان یکسانی برگزار شود.
- 4. افزایش فاصله بین زمان تشکیل کلاس درس‌های خاصی که به عنوان ورودی به برنامه داده می‌شود. این ورودی به صورت تعدادی زوج داده می‌شود. مثلاً درس‌های «ریاضی و شیمی» و «ادبیات و عربی» که به معنی ترجیح به افزایش فاصله بین زمان برگزاری کلاس‌های ریاضی و شیمی و همچنین کلاس‌های ادبیات و عربی است. در صورتی که هیچ ورودی به این منظور به برنامه داده نشود به معنی عدم وجود ترجیح خاص در این مورد است.

یک پیشنهاد برای تابع هدف بهینه‌سازی

رابطه زیر مثالی از نحوه ترکیب قیدهای ترجیحی مختلف داده شده برای مسأله در قالب یک تابع هدف را نشان می‌دهد که باید **کمینه (minimize) شود**. در این رابطه تابع $numDays(p)$ تعداد روزهای مختلفی که استاد p در مؤسسه کلاس دارد را نشان می‌دهد. تابع $numRooms(p)$ تعداد مکان‌های مختلفی که کلاس‌های استاد p در آن برگزار می‌شود را نشان می‌دهد. تابع $totalTime(d)$ مجموع زمان کلاس‌ها در روز d و تابع $std(x)$ انحراف معیار مقادیر موجود در بردار x را محاسبه می‌کند. تابع $dist(i, j)$ فاصله بین زمان تشکیل کلاس‌های i و j را محاسبه می‌کند.

$$f(s) = \alpha_1 \sum_{p \in Profs(s)} numDays(p) + \alpha_2 \sum_{p \in Profs(s)} numRooms(p) + \alpha_3 \cdot std([totalTime(d_i) : i \in \{1, \dots, 7\}]) - \alpha_4 \sum_{\langle i, j \rangle \in Separate(s)} dist(i, j)$$

در این رابطه s نشان دهنده یک زمانبندی (راه‌حل) است. ضرایب $\alpha_1, \alpha_2, \alpha_3$ و α_4 که هر یک دارای مقداری در بازه $[0, 1]$ و مجموع آنها برابر با یک بوده، نشان دهنده میزان تأثیر هر یک از قیدهای ترجیحی در تابع هدف است و می‌توان مقادیر آنها را بسته به اهمیت هر یک از قیدهای ترجیحی مربوطه در نظر گرفت. به عنوان مثال: $\alpha_1 = 0.4, \alpha_2 = 0.2, \alpha_3 = 0.3$ و $\alpha_4 = 0.1$ (دانشجویان می‌توانند ترکیب مقادیر دیگری برای این ضرایب در نظر بگیرند). مجموعه $Profs(s)$ دربردارنده اساتید در نظر گرفته شده در زمان‌بندی s و مجموعه $Separate(s)$ حاوی زوج درس‌هایی در زمانبندی s است که باید با فاصله زمانی از هم تشکیل شوند.

تأکید می‌شود که این تابع هدف یک پیشنهاد است و دانشجویان می‌توانند از هر تابع هدف دیگری که منجر به محقق شدن اهداف مسأله شود استفاده کنند.

ورودی‌های برنامه

اطلاعات مورد نیاز برای یافتن یک زمان‌بندی می‌تواند به اشکال مختلف به برنامه داده شود. این اطلاعات می‌تواند در تعامل با کاربر برنامه (مثلاً در قالب واسط‌های گرافیکی – GUI) یا به صورت یک فایل و یا با هر روش دلخواه دیگر بدست آید. نکته مهم این است که این اطلاعات از قبل مشخص نبوده و برنامه باید بتواند برای مجموعه ورودی‌های متفاوت زمان‌بندی‌های متفاوتی تولید کند. برخی از ورودی‌های برنامه اختیاری هستند به این معنی که کاربر برنامه می‌تواند آنها را مشخص نکند. در این صورت برنامه باید مقادیر پیش‌فرضی برای این ورودی‌ها در نظر بگیرد. ورودی‌های اختیاری عبارتند از:

- تعداد دانشجویان ثبت نام کرده در هر درس
 - مقدار پیش‌فرض: ۳۰ نفر
- مدت زمان برای ارائه هر درس
 - مقدار پیش‌فرض: ۲ ساعت
- ظرفیت هر اتاق
 - مقدار پیش‌فرض: ۳۵ نفر
- بازه زمانی تشکیل کلاس‌های درس در هر روز
 - مقدار پیش‌فرض: ۸ صبح تا ۴ بعدازظهر

به عنوان مثال اگر ورودی‌ها توسط یک فایل به برنامه داده شود، می‌توان ساختاری مانند شکل ۱ برای آن در نظر گرفت که توسط برنامه شناخته شود.

```
#Course
'Math 1' 25
'Stat 1' 67
'Geometry'

#Prof
'Mr. A' 'Math 1' 'Stat 1'
'Mrs. B' 'Geometry'

#Time
'Geometry' 1.5
'Stat 1' 1

#Room
'Class 11' 25
'Class 12'
'Class 24' 40

#Span
8 - 18

#Separate
'Math 1' 'Stat 1'
```

شکل ۱: مثالی از یک فایل ورودی برای برنامه

که در آن بخش #Course شامل نام درس‌ها و تعداد دانشجویان ثبت‌نامی در آنها، بخش #Prof شامل نام اساتید و درس‌های ارائه شده توسط هر استاد، بخش #Time حاوی نام درس‌ها و مدت زمان مورد نیاز برای ارائه آنها، بخش #Room شامل نام اتاق‌های موجود و ظرفیت آنها، بخش #Span دربردارنده بازه زمانی برگزاری کلاس‌های درس در هر روز و بخش #Separate شامل زوج درس‌هایی است که باید ترجیحاً با فاصله زمانی از یکدیگر برگزار شوند.

دقت کنید تأکید خاصی بر این قالب برای ورودی اطلاعات نیست و دانشجویان می‌توانند هر قالب دیگری برای آن در نظر بگیرند. نکته مهم دریافت این اطلاعات از ورودی و امکان تغییر این مقادیر است.

خروجی‌های مورد انتظار

همانند بخش اول تمرین، سامانه پیاده‌سازی شده باید نتایج اجرای حداقل پنج بار الگوریتم ژنتیکی ترکیبی روی مسأله‌ای که به عنوان ورودی داده می‌شود را گزارش کند. این نتایج در هر اجرا باید شامل اطلاعات زیر باشد:

- بهترین راهحل بدست آمده و مقدار تابع هدف برای آن (برای نمایش هر راهحل جزئیات انتساب مقادیر به متغیرها باید نشان داده شود)
- اگر بهترین راهحل سازگار نیست (بسته به پیاده‌سازی صورت گرفته) باید قیدهای اصلی نقض شده توسط آن مشخص شده و در صورت وجود، بهترین راهحل سازگار پیدا شده و مقدار تابع هدف آن نیز نمایش داده شود
- متوسط مقدار تابع هدف بدست آمده (بر اساس جمعیت آخرین نسل الگوریتم)

در پایان تمام اجراها نیز باید اطلاعات زیر گزارش شود:

- بهترین راهحل بدست آمده در تمام اجراها و مقدار تابع هدف برای آن
- اگر بهترین راهحل سازگار نیست (بسته به پیاده‌سازی صورت گرفته) باید قیدهای اصلی نقض شده توسط آن مشخص شده و در صورت وجود، بهترین راهحل سازگار پیدا شده در تمام اجراها و مقدار تابع هدف آن نیز نمایش داده شود
- میانگین بهترین راهحل‌های بدست آمده (در همه اجراها)
- میانگین متوسط مقدار تابع هدف بدست آمده (بر اساس جمعیت آخرین نسل الگوریتم)
- میانگین دفعات فراخوانی تابع برازندگی

رفع ابهام در مورد برخی نکات

در صورت تمرین اولیه در مورد عملکرد الگوریتم ژنتیکی ترکیبی ابهاماتی ایجاد شده بود که نکات زیر در راستای رفع آن باید مورد توجه گیرد:

- لزوماً ارتباطی بین درصد والدین (#ParentPercent) و درصد فرزندان (#OffspringPercent) وجود ندارد.
- اگر تعداد فرزندان مورد نیاز فرد بود، فقط از یک فرزند یکی از جفت والدین (مثلاً آخرین جفت) استفاده می‌شود.
- در مرحله جفت‌سازی والدین که به صورت تصادفی انجام می‌گیرد امکان انتخاب والدین یکسان در یک جفت وجود دارد.

هر گونه ابهام دیگر در مورد بخش‌های مختلف این تمرین را از طریق ایمیل professor.karshenas@gmail.com مطرح کنید.

نکات قابل توجه

- انجام این تمرین تعیین کننده بخشی از نمره نهایی دانشجویان خواهد بود.
- این تمرین و بخش تکمیلی آن باید به صورت انفرادی انجام شود و هرگونه کپی برداری از کار دیگر دانشجویان به منزله تقلب است.
- تحویل تمرین به صورت حضوری خواهد بود. زمان و مکان تحویل تمرین به زودی اطلاع‌رسانی خواهد شد.
- در زمان تحویل، اجرای صحیح و کامل برنامه پیاده‌سازی شده در ارزشیابی نقش اساسی خواهد داشت و از دانشجو در مورد جزئیات پیاده‌سازی سوال خواهد شد.
 - دانشجویان باید تمام نرم‌افزارهای مورد نیاز برای اجرا شدن کامل برنامه خود را پیش از تحویل در کامپیوتر مورد استفاده نصب کرده باشند.
 - مسئولیت هر گونه عملکرد پیش‌بینی نشده و نقص در کارکرد مورد نظر برنامه در هنگام تحویل بر عهده دانشجو است.
 - با توجه به تعداد زیاد دانشجویان هر فرد فقط یکبار امکان تحویل تمرین خواهد داشت.
- لازم است دانشجویان قبل از تحویل تمرین تمام فایل‌های خواسته شده در صورت اولیه تمرین به همراه کدهای مرتبط با این بخش تکمیلی را در قالب یک فایل فشرده ارسال کرده باشند.
- همانگونه که قبلاً نیز گفته شد، دانشجویان می‌توانند جهت پیاده‌سازی بخش‌های مختلف این تمرین از کدها و بسته‌های نرم‌افزاری موجود استفاده کنند به شرطی که مرجع آن دقیقاً مشخص شده باشد و اطمینان داشته باشند که الگوریتم و مسائل توصیف شده در این تمرین (با تمام جزئیات آن) پیاده‌سازی و بررسی شود.

موفق باشید

کارشناس