

بسمه تعالی

تمرین سری دوم درس پردازش تکاملی دکتر اثنی عشری

لطفا در حل تمرین به موارد زیر توجه داشته باشید:

- تمرین را خودتان انجام بدهید و به کاری که ارائه می کنید، تسلط کامل داشته باشید.
- تمرین را با هر زبان برنامه نویسی می توانید انجام دهید.
- به طور کلی، استفاده از کتابخانه های آماده مجاز نیست.
- اسم فایل ارسالی به صورت HW2_Name_StudentNumber باشد. همچنین اسم و شماره دانشجویی باید در فایل PDF پاسخ ها و داخل کدها به صورت comment وجود داشته باشد.
- فایل نهایی به صورت فشرده شده، باید شامل ۱ فایل PDF (پاسخ سوالات و گزارش کدها) و یک فولدر شامل همه کدها (و فایل های لازم) باشد. از قرار دادن فایل های اضافی جدا خودداری فرمایید.
- گزارش باید کامل و در عین حال مختصر و مفید باشد. گزارش باید شامل توضیحات لازم (پاسخ سوالات پرسیده شده)، نکات پیاده سازی، مقادیر های پیرامترها و نتایج اجرای کد (به همراه مقایسه نتایج با هم در صورت نیاز) باشد.
- نمره هر تمرین از ۳ بخش اصلی تشکیل می شود: گزارش، کد و ارائه تمرین. پس از تحویل هر تمرین، زمانی برای ارائه آن هماهنگ خواهد شد. تسلط شما بر موضوع و کد، بسیار مهم تر از صرف پاسخ دادن و پیاده سازی است که با پاسخ گویی به سوالات پرسیده شده در جلسه ارائه تمرین، سنجیده می شود.
- تمرین به صورت متنوع و گلچین آماده شده است تا با چالش های جدید و حتی مطرح نشده در کلاس روبرو شوید. هدف، افزایش قدرت تحلیل و حل مسئله شماست.
- در صورت مشاهده پاسخ ها و کدهای مشابه، نمره کسر می گردد.

سوال (۱) زمان بندی شیفت های کاری کارکنان

سناریوی مسئله: یک شرکت نیاز دارد که کارکنان خود را برای شیفت های مختلف (مثلاً صبح، بعدازظهر، شب) در طول هفته برنامه ریزی کند. هر کارمند ترجیحات، محدودیت ها و حداکثر ساعات مجاز دارد. هدف، ایجاد یک برنامه زمانی است که این محدودیت ها را رعایت کند، تعادل کاری را حفظ نماید و نیازهای پوشش دهی کاری را تأمین کند.

کدام بازنمایی برای برنامه ی هفتگی شیفت های کارکنان در الگوریتم تکاملی مناسب تر است؟

- رشته دودویی (باینری): یک رشته ی بلند از ۰ و ۱ که در آن هر بیت نشان دهنده ی "حضور/عدم حضور" یک کارمند در یک بازه ی زمانی است.
- بردار اعداد اعشاری (مقادیر پیوسته): برداری از اعداد حقیقی که در آن هر ژن نشان دهنده ی تخصیص کسری (بین ۰ تا ۱) از کارکنان به شیفت ها است.
- بردار اعداد صحیح (هر عنصر نشان دهنده ی شناسه ی یک کارمند است): برنامه ی زمانی به صورت لیستی از اعداد صحیح است که هر موقعیت در لیست نشان دهنده ی یک بازه ی زمانی خاص بوده و مقدار آن شناسه ی کارمند اختصاص داده شده است.
- جایگشت روزهای هفته: جایگشتی از روزهای هفته که برای تعیین اولویت یا ترتیب تخصیص کارکنان به شیفت ها استفاده می شود.

سوال (۲) تخصیص زمین های کشاورزی

سناریوی مسئله: یک تعاونی کشاورزی دارای چندین قطعه زمین با شرایط خاک متفاوت و چندین نوع محصول (مانند گندم، ذرت، سویا) است که هر کدام بازده، مصرف منابع و سودآوری متفاوتی دارند. هدف این است که تصمیم بگیریم چه مقدار از هر محصول در هر قطعه زمین کاشته شود تا سود کلی (یا بازده) حداکثر شود، در حالی که محدودیت هایی مانند مصرف آب، نیروی کار و قوانین تناوب زراعی رعایت شود.

کدام بازنمایی برای برنامه ی تخصیص محصولات در یک الگوریتم تکاملی مناسب تر است؟

- جایگشت انواع محصولات: یک جایگشت که محصولات را به ترتیب مشخصی فهرست می کند (مثلاً ذرت، گندم، سویا، ...).
- بردار اعداد اعشاری (سهام هر محصول از زمین): برای هر قطعه زمین و نوع محصول، یک عدد حقیقی نشان دهنده‌ی مساحت (یا درصدی از کل زمین) اختصاص داده شده به آن محصول است.
- رشته‌ی دودویی (باینری): برای هر قطعه زمین و نوع محصول، یک بیت (۰ یا ۱) نشان دهنده‌ی این است که آیا محصول در آن قطعه کاشته شده است یا خیر.
- رمزگذاری عدد صحیح (یک محصول واحد برای هر قطعه): هر قطعه زمین دارای یک عدد صحیح است که نشان دهنده‌ی محصولی است که در آنجا کاشته می شود.

سوال ۳) بهینه سازی شکل آیرودینامیکی بال هواپیما

سناریوی مسئله: یک شرکت هوافضا قصد دارد شکل بال یک هواپیما را بهینه سازی کند تا نیروی پسار (Drag) را کاهش داده و نیروی بالابر (Lift) را افزایش دهد. بال را می توان با پارامترهای هندسی خاصی (مانند طول، ضخامت در نقاط خاص، زاویه‌ی اتصال) توصیف کرد که تحت محدودیت‌های فیزیکی و آیرودینامیکی قرار دارند.

کدام روش نمایش برای مسئله‌ی بهینه سازی آیرودینامیکی با الگوریتم ژنتیک مناسب تر است؟

- رشته‌ی دودویی (باینری): هر پارامتر هندسی به عنوان یک بخش از بیت‌ها (۰ و ۱) رمزگذاری شده و به مقادیر حقیقی تبدیل می شود.
- بردار اعداد اعشاری (مقادیر پیوسته): هر ژن مستقیماً نشان دهنده‌ی یک پارامتر هندسی پیوسته است.
- جایگشت شاخص‌ها: جایگشتی که ترتیب "قطعات بال" یا الگوهای طراحی از پیش تعریف شده را تغییر می دهد.
- رمزگذاری عدد صحیح (الگوهای بال از پیش تعریف شده): هر ژن یک عدد صحیح است که به یک شکل از پیش تعریف شده اشاره دارد (مثلاً ۱ برای بال باریک، ۲ برای بال پهن و ...).

سوال ۴) آیا استراتژی زیر در یک الگوریتم تکاملی معقول است؟ توضیح دهید.

" الگوریتم تکاملی را با یک اندازه تورنمنت کوچک (مثلاً ۲) برای انتخاب آغاز کنید تا تنوع بیشتری بین افراد حفظ شود. سپس، با افزایش تعداد نسل‌ها، به تدریج اندازه تورنمنت را افزایش دهید (مثلاً ۵ یا ۶) تا فشار انتخاب را افزایش داده و به راه‌حل‌های بهتر همگرا شوید. "

سوال ۵) عملگرهای تقاطع و جهش

فرض کنید دو کروموزوم زیر با بازنمایی جایگشت (permutation) موجود است:

$P1 = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]$

$P2 = [4 \ 1 \ 7 \ 6 \ 5 \ 2 \ 3 \ 8]$

حاصل اعمال عملگرهای زیر چیست؟ توضیح دهید و مثال بزنید.

- تقاطع مرتبه اول (Order 1) از ژن سوم تا پنجم
- تقاطع چرخه‌ای (Cycle)
- جهش جابجایی (Swap) ژن اول و آخر برای $P1$
- جهش درج کردن (Insert) ژن دوم و ششم برای $P2$

سوال ۶) پیاده‌سازی

ابتدا به تعداد کافی کروموزوم تصادفی در هر بازنمایی تولید نمایید. سپس برای هر یک از موارد زیر، یک تابع تعریف کنید و روی کروموزوم‌های نمونه، آن‌ها را اعمال کنید. (برای هر بخش، به تعداد کافی کروموزوم تولید و استفاده کنید)

۱. عملگر تقاطع چند نقطه‌ای برای بازنمایی مقدار عددی (integer)
 ۲. عملگر تقاطع یکنواخت (uniform) برای بازنمایی مقدار واقعی (floating point)
 ۳. عملگر تقاطع بازترکیب (Recombination) برای بازنمایی درختی (Tree)
 ۴. عملگر جهش تغییر بیت (bit flip) برای بازنمایی دودویی (binary)
 ۵. عملگر جهش جایگزینی (Replace) برای بازنمایی درختی (Tree)
 ۶. عملگر جهش گوسی برای بازنمایی مقدار واقعی (floating point)
 ۷. روش انتخاب مسابقه‌ای (Tournament) در بازنمایی دودویی (binary)
 ۸. روش انتخاب چرخ رولت (Roulette Wheel) در بازنمایی مقدار عددی (integer)
 ۹. روش انتخاب بریدنی (Truncation) در بازنمایی مقدار واقعی (floating point)
- اگر با موردی آشنا نیستید، در مورد آن تحقیق کنید. هدف آشنا شدن با انواع عملگرها و روش‌های انتخاب است. ورودی و خروجی متناظر هریک را داخل گزارش بیاورید و در مورد عملکرد هریک به اختصار توضیح دهید.

سوال (۷) پیاده‌سازی

وظیفه شما طراحی و پیاده‌سازی یک الگوریتم تکاملی برای کمینه‌سازی تابع Ackley در دو بعد است. این تابع یک معیار رایج در مسائل بهینه‌سازی است که به دلیل فضای جستجوی وسیع و وجود کمینه‌های محلی متعدد، یک چالش مناسب برای الگوریتم‌های تکاملی محسوب می‌شود. این تابع به صورت زیر تعریف می‌شود:

$$f(x, y) = -20 \exp \left[-0.2 \sqrt{0.5(x^2 + y^2)} \right] - \exp[0.5(\cos 2\pi x + \cos 2\pi y)] + e + 20$$

۱. تابع را به صورت ۳ بعدی رسم کنید. کمینه سراسری این تابع کجاست؟
۲. یک الگوریتم تکاملی برای یافتن کمینه محلی این تابع طراحی کنید.
 - بازنمایی: جفت عدد حقیقی (floating-point) به صورت (X,Y)
 - تعداد افراد جمعیت: ۱۰۰
 - حداکثر تعداد نسل: ۵۰
 - مقداردهی اولیه جمعیت: تصادفی
 - عملگر تقاطع: تقاطع تک نقطه‌ای با احتمال ۹۰٪
 - عملگر جهش: افزودن مقدار تصادفی به X یا Y با احتمال ۱۰٪
 - تابع شایستگی: مقدار خود تابع (که باید کمینه شود)
 - روش انتخاب: انتخاب مسابقه‌ای (Tournament)

- به ازای هر نسل، بدترین، بهترین و میانگین شایستگی افراد جمعیت را رسم کنید (محور افقی تعداد نسل‌ها و محور عمودی میزان شایستگی). همچنین، بهترین شایستگی یافت شده توسط الگوریتم را در انتها گزارش کنید.
- افراد جمعیت را در نسل اول و آخر اجرای الگوریتم روی نمودار ۳ بعدی نمایش دهید.

سوال (۸) پیاده‌سازی

زمان‌بندی کارخانه جریان (Flow Shop Scheduling Problem - FSSP)

شما مدیر یک کارخانه هستید که دارای تعدادی دستگاه (Machines) و تعدادی کار (Jobs) می‌باشد. هر کار باید به ترتیب از دستگاه اول تا دستگاه آخر عبور کرده و روی هر دستگاه دقیقاً یک عملیات انجام دهد. هر دستگاه تنها قادر است در هر لحظه یک عملیات را پردازش کند و نمی‌تواند همزمان دو عملیات مختلف را انجام دهد. همچنین یک کار خاص (کار شماره i ام) تا زمانی که در ماشین قبلی به انتها نرسیده باشد، نمی‌تواند در ماشین بعدی شروع به کار کند.

در واقع هدف شما در این مسئله یافتن ترتیب شروع کارها است، به طوری که زمان تکمیل کل عملیات‌ها (Makespan) به حداقل برسد. به عبارت دیگر تابع هدف شما در این مسئله به صورت زیر تعریف می‌شود.

$$Fitness = 1 / Makespan$$

فرمت ورودی:

ورودی شامل دو عدد n و m در یک سطر و همچنین یک ماتریس $n \times m$ در سطر بعدی می‌باشد به طوری که:

n : تعداد دستگاه‌ها ($50 \geq n \geq 1$)

m : تعداد کارها ($50 \geq m \geq 1$)

سپس n خط بعدی داده می‌شود به طوری که هر خط شامل m عدد صحیح مثبت است و زمان پردازش کارها روی هر دستگاه را بیان می‌کند. به عبارتی دیگر عنصر واقع در سطر i ام و ستون j ام، بیانگر مدت زمان مربوط به کار شماره j ام است که در دستگاه i ام انجام می‌شود.

فرمت خروجی:

خروجی یک آرایه شامل ترتیب انجام کارها به گونه‌ای است که makespan کمترین مقدار ممکن باشد. همچنین مدت زمان انجام کل عملیات نیز در ادامه نشان داده شود. به عبارتی دیگر fitness و کروموزوم نهایی نشان داده شود.

مثال:

ورودی:

3 3
3 2 4
4 3 1
2 5 3

خروجی:

[job2, job3, job1] or [job2, job1, job3]

تحلیل دقیق تمام حالات

1. [job1, job2, job3] - Makespan = 18

عملیات	Job1	Job2	Job3
دستگاه ۱	0-3	3-5	5-9
دستگاه ۲	3-7	7-10	10-11
دستگاه ۳	7-9	10-15	15-18

2. [job1, job3, job2] - Makespan = 17

عملیات	Job1	Job3	Job2
دستگاه ۱	0-3	3-7	7-9
دستگاه ۲	3-7	7-8	9-12
دستگاه ۳	7-9	9-12	12-17

3. [job2, job1, job3] - Makespan = 15

عملیات	Job2	Job1	Job3
دستگاه ۱	0-2	2-5	5-9

دستگاه ۲	2-5	5-9	9-10
دستگاه ۳	5-10	10-12	12-15

4. [job2, job3, job1] - Makespan = 15

عملیات	Job2	Job3	Job1
دستگاه ۱	0-2	2-6	6-9
دستگاه ۲	2-5	6-7	9-13
دستگاه ۳	5-10	10-13	13-15

5. [job3, job1, job2] - Makespan = 19

عملیات	Job3	Job1	Job2
دستگاه ۱	0-4	4-7	7-9
دستگاه ۲	4-5	7-11	11-14
دستگاه ۳	5-8	11-13	14-19

6. [job3, job2, job1] - Makespan = 16

عملیات	Job3	Job2	Job1
دستگاه ۱	0-4	4-6	6-9
دستگاه ۲	4-5	6-9	9-13
دستگاه ۳	5-8	9-14	14-16

محدودیت‌ها:

۱. محدودیت تعداد دستگاه‌ها و کارها:

- تعداد دستگاه‌ها (n) باید بین ۱ تا ۵۰ باشد
- تعداد کارها (m) باید بین ۲ تا ۵۰ باشد

۲. ترتیب پردازش عملیات‌های هر کار:

- هر کار باید به ترتیب ثابت از دستگاه اول به دستگاه آخر پردازش شود.

- هیچ کاری اجازه ندارد قبل از اتمام پردازش روی دستگاه قبلی، وارد دستگاه بعدی شود.
- ۳. پردازش همزمان عملیات‌ها:
 - هر دستگاه در هر زمان فقط می‌تواند یک کار را پردازش کند (هیچ هم‌زمانی در یک دستگاه مجاز نیست).
 - اما دستگاه‌های مختلف می‌توانند همزمان عملیات‌های مختلف را پردازش کنند.
- ۴. یکتایی عملیات‌ها:
 - هر کار دقیقاً یک بار بر روی هر دستگاه پردازش می‌شود.
 - هیچ تکرار یا حذف عملیاتی وجود ندارد.
- ۵. محدودیت در انتخاب روش حل مسئله:
 - تنها مجاز به استفاده از الگوریتم‌های تکاملی هستید، که شامل عملگرهای زیر می‌شوند و با تمام حالات زیر پیاده‌سازی و بررسی شود.
- Crossover: فقط از Cycle Crossover و Edge Recombination و Partially Map
- Mutation: فقط از Scramble Mutation و Insert Mutation

موفق باشید