به نام خالق بی‌همتا

مستند فاز اول پروژه – الگوریتم‌های تکاملی

درس

مبانی هوش محاسباتی

استاد

دکتر حسین کارشناس

دستیاران آموزشی

محمدامین نصیری

سید حسین حسینی دولت‌آبادی

دانشکده مهندسی کامپیوتر
دانشگاه اصفهان
زمستان ۱۴۰۴

فهرست

۱. مرحله‌ی اول - ژنتیک برای بهترین مدل (بهینه‌سازی وزن‌ها و ضرایب) ۳

۱.۱. توضیحات مجموعه داده ۳

۱.۲. تعریف مدل ٤

۱.۲.۱. تابع اول - هزینه‌ی انرژی ۴

۱.۲.۲. تابع دوم – شاخص رشد گیاه ۵

۱.۲.۳. نرمال‌سازی f1 و f2 ۶

۱.۲.۴. قید پخش‌شدگی ضرایب ۶

۱.۲.۵. مدل نهایی ۷

۱.۲.۶. امتیازی ۷

۱.۳. تعریف تابع برازندگی و محاسبه از روی دیتاست ۷

۱.٤. نمایش کروموزوم ۷

۱.۵. خروجی ۸

۲. مرحله‌ی دوم - ژنتیک برای بهترین ورودی سیستم در شرایط آزمایشگاهی ۹

۲.۱. قیود ۹

۲.۱.۱. مشکلات فیزیکی ۹

۲.۱.۲. قید واقع‌گرایی (اختیاری) ۱۰

۲.۲. ساختار کروموزوم ۱۰

۲.۳. تابع برازندگی ۱۰

۲.٤. خروجی ۱۰

۳. مرحله‌ی سوم - ژنتیک برای بهترین ورودی سیستم در محیط واقعی ۱۱

۳.۱. ساختار کروموزوم ۱۱

۳.۲. اجرا ۱۱

۳.۳. سناریوهای محیطی ۱۱

۳.٤. خروجی ۱۲

٤. آنچه از دانشجو انتظار می‌رود ۱۳

٤.۱. پیش‌پردازش داده‌ها ۱۳

٤.۲. تعریف اجزای الگوریتم ژنتیک ۱۳

٤.۳. روش‌های انتخاب (Selection) ۱۳

٤.٤. روش‌های تقطیع (Crossover) ۱۳

٤.۵. جهش (Mutation) ۱۳

٤.٦. معیار توقف ۱٤

٤.۷. طراحی آزمایش ۱٤

٤.۸. خروجی‌های مورد نیاز ۱٤

٤.٩. تحلیل مورد انتظار ١٤
٤.١٠. تحویل نهایی باید شامل:..... ١٥
٥. محدودیت‌های پیاده‌سازی..... ١٥
٦. سوالات تحلیلی برای گزارش ١٥

شرح کلی پروژه

یک شرکت طراحی سیستم‌های هوشمند گلخانه‌ای قصد دارد سیستمی طراحی کند که مصرف انرژی را کاهش دهد و همزمان شرایط مورد نیاز گیاهان را حفظ کند. داده‌های تاریخی محیطی در اختیار شما قرار گرفته است. شما باید:

۱. ابتدا پارامترهای یک مدل ارزیابی (Fitness Function) را بهینه کنید.
۲. سپس بهترین سناریوی ورودی را در شرایط آزمایشگاهی بیابید.
۳. در نهایت در شرایط واقعی، بهترین تنظیمات قابل‌کنترل را برای هر وضعیت محیطی محاسبه کنید.

۱. مرحله‌ی اول – ژنتیک برای بهترین مدل (بهینه‌سازی وزن‌ها و ضرایب)

هدف از این مرحله طراحی و تنظیم یک تابع برازندگی است که بتواند عملکرد سیستم را کمی‌سازی کند و بین کاهش مصرف انرژی و افزایش آسایش تعادل برقرار کند. این ایجاد تعادل براساس داده‌های موجود در دیتاست انجام می‌شود و تصور می‌شود این داده‌ها برای این سیستم معتبرترین می‌باشند.

در این مرحله:

- ورودی‌ها ثابت‌اند (دیتاست)
- انرژی واقعی در دیتاست وجود دارد که با دیگر ویژگی‌ها ارتباط دارد
- هدف، تنظیم وزن‌ها و ضرایب تابع ارزیابی است

نکته: در این بخش، دانشجو **مدل پیش‌بینی یا یادگیری نظارت‌شده** انجام نمی‌دهد. هیچ MSE و خروجی هدفی داده نمی‌شود. هدف صرفاً **طراحی/تنظیم تابع برازندگی** و استفاده از GA برای بهینه‌سازی پارامترهای آن است.

۱.۱. توضیحات مجموعه داده

هر رکورد دیتاست شامل:

- T_{in}: دمای داخل
- T_{out}: دمای بیرون
- H_{in}: رطوبت داخل
- H_{out}: رطوبت بیرون
- L: نور مصنوعی
- Solar: تابش خورشید
- CO_2: غلظت CO2
- Wind: سرعت باد
- N: تراکم گیاه
- W: وضعیت آب‌وهوا(categorical)

- E: مصرف انرژی

۱.۲. تعریف مدل

دو تابع هدف تعریف می‌کنیم که در بخش‌های ۱.۲.۱ و ۱.۲.۲ توضیحات مفصل مربوط به هریک مشخص شده است:

- f_1: هزینه و فشار عملکرد سیستم (کمینه‌سازی)
- f_2: شاخص کیفیت رشد گیاه (بیشینه‌سازی)

این دو تابع دارای ویژگی‌های مشترک هستند تا trade-off واقعی ایجاد شود. تابع نهایی ترکیبی از این دو تابع خواهد بود.

۱.۲.۱. تابع اول (f_1) - هزینه‌ی انرژی

باتوجه به اینکه یکی از اهداف سیستم کمینه کردن مصرف انرژی است، نیاز به تعریف یک تابع هدف برای این منظور داریم. اما مصرف بالای انرژی به تنهایی باعث ایجاد مشکل نیست. در بعضی شرایط اگر اختلاف دمای بیرون و درون زیاد باشد یا تعداد گیاهان در اتاق زیاد باشد، مصرف بالای انرژی تا حدی قابل قبول است. برای همین برای شرایط غیرقابل‌قبول یک سری مولفه‌ی جریمه تعریف می‌کنیم که به صورت زیر است:

میزان مصرف انرژی بیش از حد مجاز (۱۰۰) که فقط درصورت تجاوز از آستانه جریمه می‌شود.

$$P_E = \max(0, E - 100),$$

هرچه اختلاف دمای درون و بیرون بیشتر باشد، حفظ شرایط داخلی سخت‌تر و هزینه‌ی انرژی بیشتر خواهد بود.

$$P_{\Delta T} = |T_{in} - T_{out}|,$$

اختلاف زیاد رطوبت داخل و خارج نشان‌دهنده نیاز بیشتر به سیستم‌های کنترل رطوبت است.

$$P_{\Delta H} = |H_{in} - H_{out}|,$$

میزان کمبود نور نسبت به نیاز گیاه که باید توسط نور مصنوعی جبران شود. L_{req} نور مورد نیاز گیاه است که به طور میانگین، مقدار ثابت ۵۰۰ برای آن در نظر گرفته شود.

$$P_{LightDef} = \max\Big(0, L_{req} - (L + Solar)\Big),$$

ریسک تهویه ناشی از ترکیب تراکم گیاه و جریان هوا که می‌تواند باعث اختلال در شرایط پایدار شود.

$$P_{VentRisk} = \max(0, N \cdot Wind - \tau),$$

تعریف نهایی تابع:

$$f_1 = \widetilde{a_1}\log(1 + P_E^2) + \tilde{a}_2 P_{\Delta T} + \tilde{a}_3 P_{\Delta H} + \tilde{a}_4 P_{LightDef} + \widetilde{a_5}\sqrt{N} + \tilde{a}_6 P_{VentRisk}$$

هدف در این قسمت کمینه کردن تابع f_1 است.

$\tilde{a}_i$ها نرمال‌شده‌ی ضرایب هستند. اگر ضرایب جریمه‌ها آزاد باشند، الگوریتم می‌تواند آن‌ها را به بی‌نهایت ببرد. برای جلوگیری از این مشکل ضرایب را نرمال‌سازی می‌کنیم:

$$\tilde{a}_i = \frac{a_i + \epsilon}{\sum_{k=1}^{6}(a_k + \epsilon)}$$

که در آن:

$$\epsilon = 10^{-6}$$

۱.۲.۲. تابع دوم – شاخص رشد گیاه

از اهداف دیگر سیستم افزایش رشد گیاهان است. گیاهان دوست دارند در بازه دمایی و رطوبت مطلوبی قرار گیرد و شدت روشنایی اتاق به‌اندازه باشد. همه‌ی این نیازها درحالی است که انجام این اهداف در تضاد با مصرف کم انرژی است و پیدا کردن نقطه تعادل (trade-off) سخت است. با این توضیحات تابع دوم به شکل زیر تعریف می‌شود:

امتیاز دما: امتیاز کیفیت دمای داخلی براساس نزدیکی به بازه مطلوب رشد گیاه.

$$S_T = \begin{cases} 1, & 20 \le T_{in} \le 24 \\ 0.6, & 18 \le T_{in} < 20 \text{ or } 24 < T_{in} \le 26 \\ 0.2, & 16 \le T_{in} < 18 \text{ or } 26 < T_{in} \le 28 \\ 0, & \text{otherwise} \end{cases}$$

امتیاز رطوبت: امتیاز کیفیت رطوبت داخلی براساس فاصله از محدوده‌ی مناسب.

$$S_H = \begin{cases} 1, & 45 \le H_{in} \le 60 \\ 0.5, & 35 \le H_{in} < 45 \text{ or } 60 < H_{in} \le 70 \\ 0, & \text{otherwise} \end{cases}$$

امتیاز نور: امتیاز نور براساس کف نیاز گیاه و ناحیه‌ی اشباع آن که رفتار غیرخطی نور را مدل می‌کند.

$$S_L = \begin{cases} 0, & L + Solar < L_{min} \\ 1, & L_{min} \le L + Solar \le L_{sat} \\ 0.7, & L + Solar > L_{sat} \end{cases}$$

که در آن $L_{min} = 30$ و $L_{sat} = 900$ می‌باشد. خوب است بدانید این مقادیر به طور منطقی از دیتاست قابل استخراج اند؛ به طوری که: $L_{min} = Q_{25}(L + Solar)$ و $L_{sat} = Q_{75}(L + Solar)$.

امتیاز CO_2: کیفیت غلظت کربن دی‌اکسید برای فتوسنتز که در یک بازه‌ی بهینه بیشترین مقدار را دارد.

$$S_C = \begin{cases} 1, & 800 \le CO_2 \le 1200 \\ 0.4, & 700 \le CO_2 < 800 \text{ یا } 1200 < CO_2 \le 1400 \text{ (محدوده‌ی مرزی)} \\ & 0, otherwise \end{cases}$$

جریمه‌ی تراکم: افزایش بیش از حد تراکم منجر به کاهش کیفیت رشد می‌شود.

$$P_N = \max(0, N - N_{max})$$

تعریف نهایی تابع:

$$f_2 = \widetilde{b_1}S_T + \widetilde{b_2}S_H + \widetilde{b_3}S_L + \widetilde{b_4}S_C - \widetilde{b_5}P_N$$

که در آن $\tilde{b}$ها نرمال‌شده‌ی ضرایب هستند:

$$\tilde{b}_j = \frac{b_j + \epsilon}{\sum_{k=1}^{5}(b_k + \epsilon)}$$

هدف در این قسمت بیشینه کردن تابع f_2 است.

۱.۲.۳. نرمال‌سازی f_1 و f_2

از آنجایی که خروجی دو تابع طراحی شده و بازه‌های آن‌ها کیسان نمی‌باشد، قبل از ترکیب توابع نیاز به نرمال‌سازی داریم. ابتدا از روی دیتاست یک تخمین از بازه‌ی هر تابع و میانگین توابع می‌گیریم و سپس نرمال‌سازی را انجام می‌دهیم:

$$f_{1_{norm}} = \frac{\overline{f_1} - F_{1_{min}}}{F_{1_{max}} - F_{1_{min}} + \delta}$$

$$f_{2_{norm}} = \frac{\overline{f_2} - F_{2_{min}}}{F_{2_{max}} - F_{2_{min}} + \delta}$$

$$\delta = 10^{-8}$$

۱.۲.٤. قید پخش‌شدگی ضرایب

چون در عمل همه‌ی پارامترهای محیطی گلخانه (دما، رطوبت، نور و...) همزمان روی عملکرد سیستم اثر می‌گذارند، منطقی نیست مدل تقریباً تمام اهمیت را فقط به یک عامل بدهد؛ اما نرمال‌سازی ضرایب به‌تنهایی جلوی این تمرکز افراطی را نمی‌گیرد.

برای جلوگیری از این حالت، یک ترم مبتنی بر آنتروپی به تابع هدف نهایی(ترکیب $f_{1_{norm}}$ و $f_{2_{norm}}$) اضافه می‌کنیم که ضرایب بیش‌ازحد متمرکز را جریمه می‌کند، در حالی که همچنان اجازه می‌دهد عوامل مهم‌تر سهم بیشتری داشته باشند. این کار باعث می‌شود توزیع ضرایب واقعی‌تر و تفسیرپذیرتر باقی بماند.

$$H(\tilde{a}) = -\sum_{i=1}^{6} \tilde{a}_i \log(\tilde{a}_i) \quad , \qquad H(\tilde{b}) = -\sum_{j=1}^{5} \tilde{b}_i \log(\tilde{b}_i)$$

نرمال‌سازی آنتروپی (برای اینکه بین 0 و 1 باشد):

$$\hat{H}(\tilde{a}) = \frac{H(\tilde{a})}{\log 6} \quad , \qquad \hat{H}(\tilde{b}) = \frac{H(\tilde{b})}{\log 5}$$

ترم Regularization:

$$R(X) = \lambda\left(\widehat{H}(\tilde{a}) + \widehat{H}(\tilde{b})\right)$$

هرچه λ بزرگتر باشد پخش‌شدگی ضرایب بیشتر حفظ می شود. در این قسمت $\lambda = 0.1$ درنظر بگیرید.

۱.۲.۵. تابع نهایی

تابع نهایی باید ترکیبی از توابع تعریف شده باشد. برای مثال:

$$F(X) = w_1 f_{1_score} + w_2 f_{2_score} + R(X), \qquad w_1 + w_2 = 1$$

که در آن $f_{1_score} = 1 - f_{1_{norm}}$ و $f_{2_score} = f_{2_{norm}}$ است (چرا؟). با دانستن اینکه اهداف توابع یکسان نیستند (برخی نیاز به بیشینه‌سازی دارند و برخی نیاز به کمینه‌سازی) تابع بالا را با هدف بیشینه‌سازی استفاده کنید و یا از ترکیب دلخواه (و منطقی) خود برای تعیین مدل نهایی بهره ببرید.

۱.۲.۶. امتیازی

استفاده از الگوریتم‌های تکاملی چند هدفه (multi-objective) مثل SPEA2 یا NSGA-II به جای تعریف تابع ترکیبی.

۱.۳. تعریف تابع برازندگی و محاسبه از روی دیتاست

برای هر فرد (X):

۱. روی **هر ردیف دیتاست i**، $f_1(X_i)$ و $f_2(X_i)$ محاسبه می‌شود.

۲. تجمیع روی کل دیتاست (میانگین):

$$\bar{f}_1(X) = \frac{1}{n}\sum_{i=1}^{n} f_1(X_i), \qquad \bar{f}_2(X) = \frac{1}{n}\sum_{i=1}^{n} f_2(X_i)$$

۳. نرمال‌شده‌ی توابع طبق بخش ۱.۲.۳ پیدا می‌شود.

۴. طبق رابطه‌ی بخش ۱.۲.۵ Fitness نهایی محاسبه می‌شود:

$$\text{Fitness}(X) = F(X)$$

چرا استفاده از الگوریتم‌های تکاملی برای این مسئله مناسب‌تر از روش‌های مبتنی بر گرادیان است؟ توضیح دهید.

۱.٤. نمایش کروموزوم

یک نمونه نمایش کروموزوم می‌تواند به شکل زیر باشد:

$$X = [a_1, a_2, a_3, a_4, a_5, a_6, b_1, b_2, b_3, b_4, b_5, w_1, w_2]$$

قیود:

- $w_1 + w_2 = 1$
- $a_i, b_j \in (0, 10]$

پس از هر crossover/mutation ضرایب داخل هر تابع نرمال‌سازی می‌شوند.

❖ <u>نکته</u>: در توابع برازندگی چندهدفه‌ی ترکیبی، وارد کردن وزن‌ها به کروموزوم(قابل آموزش کردن وزن‌ها) و انجام فرآیند بهینه‌سازی منجر به نتایج خوبی نمی‌شود(چرا؟ - در گزارش پاسخ داده شود). برای جلوگیری از مشکل الگوریتم را با $w_1 = 0.4$ و $w_2 = 0.6$ اجرا کنید.

امتیازی:

- تنظیم و تست دستی وزن‌ها
- استفاده از وزن‌های تطبیقی

۱.۵. خروجی

- بهترین کروموزوم(X^*)
- نمودار همگرایی
- مقدار (f_1) و (f_2) جداگانه
- تحلیل حساسیت ضرایب: با ایجاد تغییرات کوچک در ضرایب و بررسی تغییر خروجی، میزان تاثیر هر پارامتر بر عملکرد سیستم ارزیابی می‌شود تا اهمیت نسبی پارامترها مشخص شود.
- تحلیل رفتار مدل در نقاط بحرانی دیتاست: عملکرد مدل را در شرایط خاص (مثلا وقتی ورودی‌ها خیلی بزرگ یا خیلی کوچک هستند) بررسی می کنیم تا ببینیم مدل در حالت‌های غیرعادی چطور رفتار می‌کند. با این کار قابلیت تعمیم مدل در شرایط خاص ارزیابی می گردد.

۲. مرحله‌ی دوم - ژنتیک برای بهترین ورودی سیستم در شرایط آزمایشگاهی

در این بخش با استفاده از تابع برازندگی بدست آمده می‌خواهیم ایده‌ال‌ترین حالت را برای سیستم درنظر بگیریم. در بخش قبل ورودی‌ها را داشتیم و می‌خواستیم ضرایب مدل را بهینه کنیم. در این بخش ورودی‌ها(دیتاست) را کنار گذاشته و با استفاده از مدل به دست آمده از قسمت قبل، به پیدا کردن بهترین ورودی برای سیستم می‌پردازیم. در این قسمت شرایط آزمایشگاهی است و تمامی ویژگی‌ها حتی دمای بیرون و آب‌وهوا نیز قابل دستکاری هستند. اما مصرف انرژی متغیر مستقلی نیست و وابسته به میزان امکاناتی است که در سیستم استفاده می‌کنیم. متخصصان ما در آزمایشگاه طی تحقیقاتی که روی مصرف انرژی انجام داده‌اند متوجه شدند مصرف انرژی تقریبا از رابطه‌ی زیر پیروی می کند:

$$E = k_w(W)(0.8(\Delta T)^2 + 12\log(1+N) + 0.02P_{hum}) + 6\log(1+L) + 2\log(1+CO_2)$$

که در آن:

$$\Delta T = |T_{in} - T_{out}|$$
$$P_{hum} = \max(0, H_{in} - 60)^2 + \max(0, 30 - H_{in})^2$$
$$k_w(W) = \{"sunny": 0.9, "cloudy": 1.0, "rainy": 1.15, "stormy": 1.25\}$$

- ضریب آب‌وهواست ($k_w(W)$) (مثلاً stormy پرهزینه‌تر از sunny)

هدف در این قسمت یافتن بهترین سناریوی ورودی است زمانی که همه متغیرها قابل کنترل هستند.

در این مرحله:

- وزن‌ها و ضرایب مدل ثابت‌اند.
- انرژی از رابطه تحلیلی محاسبه می‌شود.
- دیتاست فقط برای اعمال قید واقع‌گرایی می‌تواند استفاده شود(اختیاری).

چرا استفاده از الگوریتم‌های تکاملی برای این مسئله مناسب‌تر از روش‌های مبتنی بر گرادیان است؟ توضیح دهید.

۲.۱. قیود

۲.۱.۱. مشکلات فیزیکی

در شرایط آزمایشگاهی با وجود اینکه تمام ورودی‌ها برای ما قابل دستکاری هستند، این دستکاری کار ساده‌ای نیست و تجهیزات آزمایشگاه ما محدود است و محدودیت‌های زیر را برای ما بوجود می‌آورد:

- $T_{in} \in [10,35]$
- $T_{out} \in [-5,45]$
- $H_{in} \in [10,90]$
- $L \in [0,1000]$
- $N \in [1,30]$
- $CO_2 \in [300,1700]$
- ✓ N: اعداد گسسته است.

- ✓ W: انواع قابل قبول از دیتاست استخراج شود.

۲.۱.۲. قید واقع‌گرایی (اختیاری)

ایجاد شرایط آزمایشگاهی نزدیک به واقعیت همیشه کار ساده‌ای نیست و نیاز به تعیین قیدهایی برای کنترل روی ورودی دارد. یکی از این قیدها می‌تواند داده‌هایی که کمتر به واقعیت نزدیک هستند را جریمه کند. برای مثال می توان توزیع داده‌های واقعی‌مان را از روی دیتاست بدست آورد و یک نمونه قید را بصورت زیر تعریف کرد:

$$P_{real} = \sum_{j} \left(\frac{x_j - \mu_j}{\sigma_j} \right)^2$$

سپس مقدار آن را با ضریب دلخواه به‌عنوان جریمه به تابع برازندگی اضافه کرد:

$$F' = w_1 f_{1_score} + w_2 f_{2_score} + \lambda_{real} P_{real}$$

۲.۲. ساختار کروموزوم

با توجه به توضیحات مسئله ساختار کروموزوم برای حل مسئله را تعیین کنید و نوع هر ژن (اعداد حقیقی، دسته‌ای، اعداد صحیح و...) را مشخص کنید.

۲.۳. تابع برازندگی

مشابه قبل، همراه با محدود کردن ورودی به دامنه مجاز.

$$\text{Fitness} = F'$$

۲.٤. خروجی

- بهترین سناریو و کروموزوم برتر
- مصرف انرژی و میزان آسایش مربوطه
- اثر ضریب واقع‌گرایی (اختیاری)

۳. مرحله‌ی سوم - ژنتیک برای بهترین ورودی سیستم در محیط واقعی

در شرایط واقعی، برخی ویژگی‌ها قابل تنظیم نیستند.

فرض کنید:

$$z = [T_{out}, H_{out}, Solar, Wind, N, W]$$

داده شده‌اند.

و فقط:

$$u = [T_{in}, H_{in}, L, CO_2]$$

قابل تنظیم‌اند.

هدف:

$$u^*(z) = \arg\min F(u, z)$$

دقت شود مصرف انرژی همچنان از رابطه‌ی داده شده در مرحله‌ی دوم پیروی می کند.

۳.۱. ساختار کروموزوم

با توجه به توضیحات مسئله ساختار کروموزوم برای حل مسئله را تعیین کنید و نوع هر ژن (اعداد حقیقی، دسته‌ای، اعداد صحیح و...) را مشخص کنید.

۳.۲. اجرا

برای هر سناریوی محیطی(u) داده‌شده:

1. مقدارهای غیرقابل تنظیم ثابت نگه داشته شوند.
2. GA روی (u) اجرا شود.
3. بهترین تنظیم گزارش شود.

از آنجایی که الگوریتم‌های تکاملی تصادفی هستند و در دفعات مختلف اجرا، نتایج متفاوتی می دهند؛ برای هر سناریو مراحل بالا را چند بار انجام دهید و میانگین نتایج را نیز گزارش کنید.

۳.۳. سناریوهای محیطی

#	W	T_out	H_out	Solar	Wind	N	توضیح کوتاه
1	Sunny	32	45	900	3	120	ظهر تابستان، آفتابی و شلوغ
2	Sunny	38	30	1000	2	80	موج گرما، رطوبت کم
3	Cloudy	22	55	300	4	150	اتاق ،هوای معتدل پرتراکم
4	Rainy	18	85	80	6	60	بارانی و مرطوب
5	Stormy	16	75	50	12	90	بالا k_w ،طوفانی
6	Cold	8	60	200	5	40	زمستان سرد
7	Cold	3	70	100	8	110	سرد و بادی، شلوغ

8	Humid	28	90	600	2	130	شرجی، نیاز به کنترل رطوبت
9	Dry	26	20	700	5	30	کم تراکم ،خشک
10	Night	20	65	0	4	100	نیاز ،(Solar≈۰) شب نور مصنوعی

۳.٤. خروجی

برای هر سناریو:

$$z \to u^*(z)$$

به همراه:

- انرژی نهایی
- مقدار آسایش
- تحلیل شرایط سخت‌تر/آسان‌تر

گزارش شود.

٤. آنچه از دانشجو انتظار می‌رود

دانشجو باید:

۱) چند روش مختلف برای هر یک از مراحل زیر انتخاب کند.

۲) ترکیب‌های مختلف ایجاد کند.

۳) آن‌ها را روی یک سناریوی ثابت اجرا کند.

۴) نتایج را مقایسه و تحلیل کند.

شما باید الگوریتم ژنتیک را از ابتدا (from Scratch) کدنویسی کنید و از کتابخانه‌های آماده‌ی GA استفاده نکنید.

٤.١.پیش‌پردازش داده‌ها

قبل از اعمال الگوریتم ژنتیک، باید مجموعه داده را پردازش کنید و مراحل زیر را انجام دهید:

مدیریت داده‌های از دست‌رفته: مقداردهی به مقادیر گمشده یا حذف ردیف های دارای داده‌های گمشده.

حذف داده های پرت (Outliers)

رمزگذاری ویژگی‌های دسته‌ای: تبدیل ستون‌های دسته‌ای به فرمت عددی با استفاده از Label Encoding یا One-Hot Encoding

٤.٢.تعریف اجزای الگوریتم ژنتیک

نمایش کروموزوم: نوع نمایش خود را مشخص و تعریف کنید.

جمعیت اولیه: مجموعه‌ای تصادفی از کروموزوم‌ها را به عنوان جمعیت اولیه تولید کنید.

٤.٣.روش‌های انتخاب (Selection)

دانشجو باید حداقل دو استراتژی انتخاب را انتخاب و پیاده‌سازی کند. استراتژی انتخابی می تواند شامل موارد زیر اما نه محدود به آنها باشد:

- Roulette Wheel Selection
- Tournament Selection

٤.٤.روش‌های تقطیع (Crossover)

دانشجو باید حداقل دو استراتژی تقطیع را انتخاب و پیاده‌سازی کند. استراتژی انتخابی می تواند شامل موارد زیر اما نه محدود به آنها باشد:

- Single Point Crossover
- Multi-Point Crossover
- Uniform Crossover
- Arithmetic Crossover (برای کدگذاری حقیقی)

٤.٥.جهش (Mutation)

نرخ جهش را تعیین کنید. (مثلا ۱ تا ۵ درصد ژن‌ها به طور تصادفی معکوس شوند)

٤.٦.معیار توقف

الگوریتم باید در شرایط زیر متوقف شود:

- رسیدن به تعداد نسل های ثابت.
- همگرایی جمعیت (عدم بهبود قابل توجه).

٤.٧.طراحی آزمایش

دانشجو باید **در هر مرحله**:

1. یک سناریو مشخص را انتخاب کند.
2. حداقل ۴ ترکیب مختلف از عملگرها بسازد و نتایج را گزارش و مقایسه کند. مثال:

ترکیب	Selection	Crossover	Mutation
A	Roulette	Single Point	۰.۰۵
B	Tournament	Single Point	۰.۰۵
C	Tournament	Uniform	۰.۰۱
D	Roulette	Two Point	۰.۰۱

❖ منظور از مقایسه این است که نشان دهید الگوریتم طراحی شده گزینه مناسبی بوده است. مثلاً اگر از انتخاب رقابتی استفاده کرده‌اید باید نشان دهید که نتیجه آن بهتر از انتخاب متناسب با برازندگی بوده است.

٤.٨.خروجی‌های مورد نیاز

برای هر ترکیب در هر مرحله:

1. مقدار f1 و f2 نرمال شده و تابع هدف در هر نسل
2. بهترین مقدار تابع هدف
3. میانگین مقدار تابع هدف
4. تعداد نسل تا همگرایی
5. نمودار همگرایی (Best Fitness vs Generation)

٤.٩.تحلیل مورد انتظار

دانشجو باید پاسخ دهد:

- کدام روش انتخاب باعث همگرایی سریع‌تر شد؟
- کدام روش جهش از گیر افتادن در مینیمم محلی جلوگیری کرد؟
- آیا Uniform Crossover تنوع بیشتری ایجاد کرد؟
- آیا Tournament Selection باعث فشار انتخابی بیش از حد شد؟

- آیا برخی ترکیب‌ها ناپایدار بودند؟

۴.۱۰. تحویل نهایی باید شامل:

- توضیح پیاده‌سازی هر اپراتور
- جدول مقایسه نتایج
- نمودارهای همگرایی
- پاسخ سوالات پرسیده‌شده در هر مرحله
- تحلیل کلی (بدون استفاده از هوش‌مصنوعی نوشته شود)

نتیجه‌گیری نهایی: بهترین ترکیب برای این مسئله چیست و چرا؟

❖ هرگونه خلاقیت در پیاده‌سازی **(با هماهنگی دستیاران آموزشی پیش از پیاده‌سازی)** شامل نمره‌ی اضافه می‌شود

۵. محدودیت‌های پیاده‌سازی

از کتابخانه های آماده‌ی GA مانند DEAP یا PyGAD استفاده نکنید. شما باید الگوریتم ژنتیک را از ابتدا پیاده‌سازی کنید.

می‌توانید از **Scikit-learn**، **Pandas**، **Numpy** و **Matplotlib** برای پردازش داده‌ها، ارزیابی و نمایش نتایج استفاده کنید.

۶. سوالات تحلیلی برای گزارش

- آیا کاهش انرژی همیشه باعث افت آسایش شد؟
- حساسیت سیستم نسبت به وزن‌ها چقدر است؟
- کدام پارامتر GA بیشترین اثر را داشت؟
- آیا روش چندهدفه پایدارتر بود؟(اختیاری)
- آیا جواب نهایی یکتا است یا چند جواب تقریباً معادل وجود دارد؟