

## پروژه ۱

### تقریب تابع با کمک برنامه نویسی ژنتیک (GP)

#### شرح:

یکی از مسائل مطرح در علوم ریاضی و مهندسی تقریب تابع است. تقریب تابع یعنی با داشتن توانایی محاسبه مقدار یک تابع نامعلوم در نقاط دلخواه، ضابطه ریاضی تابع مذکور تقریب زده شود. در این پروژه برای تقریب تابع از برنامه نویسی ژنتیک (Genetic Programming - GP) استفاده می‌شود.

**هدف پروژه:** ارزیابی میزان تسلط به مفاهیم پایه الگوریتم‌های تکاملی (در اینجا برنامه نویسی ژنتیک)

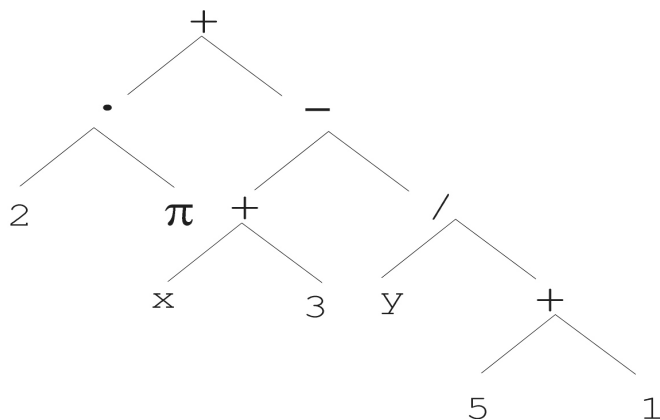
یکی از کاربرد های GP ، تقریب توابع است. در این کاربرد، هر فرد از جمعیت بیانگر یک عبارت ریاضی است که به شکل یک درخت نمایش داده می‌شود. عناصر میانی این درخت عملگرهای ریاضی، منطقی و یا هر گونه عملگر تعریف شده کاربر و برگ‌های درخت حاوی مقادیر ثابت و متغیرهای ورودی هستند. هدف از این کاربرد این است که فرمول ریاضیاتی یک تابع هدف را تخمین بزنیم. تابع هدف یک Black Box است و تنها چیزی که از آن میدانیم (یا قادریم بدست آوریم)، بدست آوردن مقدار تابع در نقاط دلخواه است. بدین ترتیب در تعدادی نقطه دلخواه مقدار تابع را یافته و از آنها به عنوان مجموعه آموزشی یاد می‌کنیم. برای هر درخت بر حسب مقادیری که برای نقاط آموزشی بدست می‌آورد و میزان اختلافش با مقادیر واقعی میتوان یک مقدار شایستگی نسبت داد و بر اساس آن الگوریتم تکاملی را دنبال کرد.

عملگرهایی که در درختان به کار برده می‌شود حداقل شامل موارد زیر باشد (موارد بیشتری نیز میتوانید اضافه کنید):

عملگرهای دو عملوندی شامل:  $+$  ،  $-$  ،  $\times$  ،  $/$  ،  $^$  (توان)

عملگرهای تک عملوندی شامل:  $\sin(x)$  ،  $\cos(x)$

یک نمونه از درخت:



$$2 \cdot \pi + \left( (x + 3) - \frac{y}{5 + 1} \right)$$

**ورودی برنامه:** تعدادی نقطه آموزشی و مقدار خروجی تابع هدف در آن نقاط (برای سادگی، تابع یک بعدی در نظر گرفته میشود).

**تبصره ۱:** در عمل برای تولید ورودی و ارزیابی عملکرد الگوریتم میتوان توابعی دلخواه (شامل هر عملگری که در الگوریتم پیشی بینی شده است و حتی عملگرهایی که در الگوریتم پیشی بینی نشده است) در نظر گرفت و در تعدادی نقطه دلخواه مقدار تابع را محاسبه کرد و این نقاط را به عنوان ورودی الگوریتم در نظر گرفت.

حتما در آزمایشها حالتی را نیز آزمایش کنید که در آن ورودی از روی تابعی تولید شده باشد که دارای عملگری است که در فهرست عملگرهای استفاده شده در GP وجود ندارد. مثلا در ضابطه تابع استفاده شده برای تولید نقاط آزمایشی از عملگر لگاریتم یا تانژانت (یا ...) استفاده کنید ولی در GP این عملگرها مجاز نباشند. یا عملگر ضرب و تقسیم و را در GP حذف کنید و نقاط را با تابعی که این عملگرها را دارند تولید کنید.

## تبصره ۲:

در یکی از آزمایشها، نقاط آزمایشی را به نمونه برداری از تابعی تخیلی به صورت خط خطی (مانند شکل زیر) و بدون داشتن ضابطه مشخص برای تابع تولید کنید.



## تبصره ۳:

در یکی از آزمایشها در تابع نقطه یا نقاط گسستگی ایجاد کنید و تفاوت رفتار GP را بررسی کنید. مثلا متوانید برای نقاط کوچکتر از عدد  $C1$  از ضابطه  $y=x^2$  و در نقاط بین  $C1$  تا  $C2$  از ضابطه  $y=-x-3$  و در نقاط بزرگتر از  $C2$  از تابع دیگری برای تولید نقاط آموزشی استفاده کنید.

## تبصره ۴:

در همه موارد، از مثالهای خیلی ساده شروع کنید و بعد از اطمینان از سلامت کد، و سلامت نحوه آزمایش، در حد امکان به سمت مثالهای خفن تر(!!) بروید.

## تبصره ۵:

تلاش شما نمره اصلی را خواهد داشت و حتی اگر نتایج دلچسب نباشند، امکان گرفتن نمره کامل در صورت صرف وقت و یادگرفتن مشکلات راه، وجود دارد. همچنین در صورتی که نتایج کاملی از همه بخشها به دست آورید، ممکن است باعث نمره مثبت (نمره اضافی) هم بشود.

**خروجی برنامه:** نمایش فرمول متناظر با بهترین درخت تولید شده توسط الگوریتم، مقدار شایستگی بهترین درخت، تعداد نسلها، تعداد محاسبه شایستگی، زمان اجرا.

آنچه تحویل داده میشود:

۱- کد اجرایی برنامه به همراه تست کردن چند ورودی متفاوت توسط دانشجو و نیز امکان تست کردن ورودی های دیگر در زمان تحویل برنامه

۲- گزارش از جزئیات اجرای پروژه که حداقل موارد زیر را در برمی گیرد (بدون ترتیب):

a. نحوه دقیق نگاشت مساله به درخت و محدودیت های احتمالی

i. عملگرها، اعداد ثابت، و ...

b. تابع شایستگی

c. تولید جمعیت اولیه

d. نحوه انتخاب والدین

e. نحوه تولید نسل بعد

i. نحوه ترکیب متقاطع و جهش

f. شرط خاتمه الگوریتم

g. چالش های مواجه شده و روش حل آنها

h. گزارشی از آزمایش های مختلف انجام شده (حداقل ۳ آزمایش برای ۳ تابع مختلف ورودی) و

ارزیابی و تحلیل عملکرد الگوریتم در آزمایش ها

i. جمع بندی و نتایج عملی که درباره GP در این پروژه به دست آوردید.