Sawtooth Method

Shubert-Piyavski

Global Maximum

روش Shubert-Piyavskii که اغلب روش دندان اره‌ای نامیده می شود، راهی برای یافتن ماکسیمم سراسری در یک بازه شناخته شده است. البته واضح است که با ضرب (-) در تابع f(x) می‌توان به مینیمم نیز دست یافت.

برای استفاده از این متود، تابع موردنظر باید دارای خاصیت پیوستگی لیپ‌شتیز باشد و عواملی مانند وجود اکسترمم های محلی و تک‌مدی نبودن مشکلی ایجاد نمی‌کنند.

ابزار لازم:

1. تابع
2. بازه‌ی جستجو
3. مقدار m

الگوریتم:

ابتدا نقطه وسط بازه را پیدا می‌کنیم. نقاط ابتدا، وسط و انتهای بازه را به ترتیب با a, c, b نشان می‌دهیم. سپس چهارخط زیر را پیدا می‌کنیم:

۱. خط گذرنده از نقطه ابتدای بازه با شیب m

۲. خط گذرنده از نقطه وسط بازه با شیب m-

۳. خط گذرنده از نقطه وسط بازه با شیب m

۴. خط گذرنده از نقطه انتهای بازه با شیب m-

نکته: در این الگوریتم بهترین m برابر با ماکسیمم شیب در بازه جستجو است، اما چون اینکار پر هزینه است، از یک مقدار تصادفی استفاده می‌کنیم.

مقدار x را در نقطه برخورد دو خط اول و دو خط دوم پیدا می‌کنیم. بدیهی است که دو نقطه در بازه [a,c] و [c,b] خواهد بود.

ماکسیمم دو نقطه را پیدا کرده، و مقدار x آن را در تابع اصلی پیدا می‌کنیم. نقطه بدست آمده تا اینجای کار، بهترین تقریب برای ماکسیمم سراسری است.

فرض می‌کنیم نقطه پیدا شده بین a و c باشد. از این نقطه، دو خط با شیب های m و m- رسم می‌کنیم. خط با شیب m

با خط (۴)، و خط با شیب m- با خط (۳) برخورد خواهند داشت. این نقاط برخورد، نقاط کاندید برای تکرار الگوریتم و یافتن مقدار ماکسیمم سراسری خواهند بود.

دوباره از بین سه نقطه ماکسیمم می‌گیریم و و روی تابع اصلی تصویر می‌کنیم و دو نقطه کاندید جدید بوجود می‌آوریم. این عملیات با تکرار دلخواه انجام می‌شود تا به تقریب بهتری برای مقدار ماکسیمم سراسری برسیم.

معمولا یک مقدار آستانه انتخاب می‌شود و هرگاه اختلاف دو نقطه پیدا شده کمتر از آستانه باشد، الگوریتم متوقف می‌شود.

توضیحات کلی برنامه:

برنامه از چهار بخش کلی تشکیل شده است.

بخش اول: دریافت تابع ورودی، بازه، مقدار m و تعداد تکرار.

بخش دوم: ساخت کلاس بازه و نقطه.

بخش سوم: تابع های کمکی برای پیدا کردن بلندترین نقطه در بازه‌های موجود،تقسیم یک بازه به دو بازه با دو نقطه کاندید جدید، پیدا کردن عرض از مبدا خط های رسم شده و رسم هر بازه.

بخش چهارم: ساخت دو نقطه کاندید ابتدایی و الگوریتم

اصلی.

توضیحات جزئی برنامه:

توابع و متغیرها

- **تابع هدف (Objective function)**:
  - در اینجا، تابعی به نام f تعریف شده است که تابع هدف ما را نشان می‌دهد. شما می‌توانید این تابع را براساس نیاز خود تغییر دهید. در این مثال، از یک تابع چندجمله‌ای درجهٔ پنجم استفاده شده است.
- **متغیرها**:
  - x0 و x1: حدود اولیهٔ بازهٔ جستجو برای بهینه‌سازی.
  - first و last: حداقل و حداکثر مقادیر برای محور x در نمودار.
  - m: شیب خطوطی که برای تقسیم بازه‌ها استفاده می‌شود.
  - iterations: تعداد تکرارهای الگوریتم.
  - xs: آرایه‌ای از مقادیر x برای تولید نمودار.

کلاس‌ها و توابع

- **کلاس‌ها**:
  - Interval: یک کلاس برای نمایش بازه‌ها استفاده

می‌شود. هر بازه شامل سه نقطهٔ ابتدایی، میانی و انتهایی است، همچنین شامل مقدار y نقطه کاندید در بازه است.

- Point: یک کلاس برای نمایش نقاط استفاده می‌شود. هر نقطه دارای مختصات x و y است.

- **توابع:**
  - findHighestPoint: تابعی برای یافتن بازه‌ای با بیشترین مقدار نقطه کاندید در لیست بازه‌ها استفاده می‌شود.
  - splitIntervals: تابعی که برای تقسیم یک بازه با بیشترین مقدار y نقطه کاندید به دو بازهٔ کوچکتر و ساخت دو نقطه کاندید دیگر استفاده می‌شود.
  - Intercepts: تابعی برای یافتن عرض از مبدا خطوط صاف.
  - plotIntervals: تابعی برای رسم نمودار تابع و بازه‌های موجود.

نحوه به کارگیری

تنظیمات اولیه:

- مقادیر اولیه برای بازهٔ جستجو (x0 و x1)، تعداد تکرارها (iterations) و سایر پارامترها را تنظیم کنید.

تولید بازه‌های اولیه:

- با استفاده از بازهٔ اولیه (initial_interval)، دو بازهٔ اولیه برای جستجو تولید شده و به لیست بازه‌ها (intervals) اضافه می‌شوند.

اجرای الگوریتم:

- با تکرار انجام دادن الگوریتم برای تعداد تعیین شدهٔ تکرارها، بازهٔ با بیشترین مقدار تابع انتخاب شده، تقسیم می‌شود و دو بازهٔ جدید به لیست اضافه می‌شود.

نمایش نتایج:

- پس از پایان اجرای الگوریتم، نمودار نهایی با تمام بازه‌های تولید شده نمایش داده می‌شود، و نقطهٔ با بیشترین مقدار تابع روی نمودار مشخص می‌شود.

نکته: برای ویرایش قاب نمایش داده شده در انتهای برنامه می‌توانید مقدار دلخواهی به plt.xlim و plt.ylim بدهید.