گزارش تکلیف دوم مبانی هوش محاسباتی

سجاد پیشوائیان – 9531015

بازنمایی مسئله:

در این مسئله، تمام نقاط ورودی در دو لیست x و y ذخیره می‌شوند پس از آن لیستی از کروموزوم‌ها طبق قالب تعریف پروژه ساخته می‌شود. اندازه لیست به اندازه طول ورودی است. در هر کروموزوم ۲ ژن a و b وجود دارد که ضرایب خط z را معلوم می‌کنند. هر z[i] به کمک نقطه متناظر آن در آرایه z محاسبه می‌شود. هر کروموزم یک score برای ذخیره انحراف معیار خود دارد

Read_from_file

در ابتدای اجرا برنامه نقاط ورودی را به صورت لیستی از جفت ضرایب برمی‌گرداند. [x_raw,y_raw]

class Chromosome

این کلاس برای نگهداری اطلاعات هر کروموزوم، شامل ضرایب و انحراف معیار است. برای محاسبه شایستگی(انحراف معیار) در تابع evaluate از کتابخانه statistics که جزو کتابخانه های داخلی پایتون است برای محاسبه انحراف معیار استفاده شده است. همچنین نحوه نرمال سازی در کد مشخص شده است

الگوریتم تکاملی در این قسمت اجرا می‌شود. مراحل آن را در زیر بررسی می‌کنیم.

generate_initial_population: به تعداد نقاط ورودی به مسئله، کروموزوم ساخته و آن‌ها را در list_of_chromosomes ذخیره می‌کنیم.

مراحل پایین در حلقه‌ای و به تعداد نسل انجام می‌شوند.

generate_new_seed: والدین جدیدی به صورت تصادفی به شکل جفت و به تعداد لامبدا انتخاب می‌شوند. نتیجه آن لیستی از جفت والدین خواهد شد.

cross: والدین انتخاب شده با احتمال (1- crossover_probability)که از پارامتر های مسئله است، با هم کراس اور می‌شوند. به این صورت که به شکل تصادفی ضرایب اول یا دوم والد اول به عنوان ضریب اول فرزند و ضریب دوم هم به صورت تصادفی از واالد دوم انتخاب می‌شود. اگر احتمال کراس اور در محدوده (1- crossover_probability) وجود نداشت، کراس اور صورت نمی‌گیرد. نتیجه این کار، لیستی از فرزندان جدید است.

mutation: جمعیت این نسل و فرزندان تولید شده با استفاده از تابع نرمال گوسی جهش می‌یابند. به این صورت که ضرایب هر کدام با مقدار noise که خروجی تابع نرمال است جمع می‌شود. پارامتر های sigma, mutation_rate قابل کنترل است و احتمال جهش و قدم جهش را تغییر می‌دهد.

evaluate_new_generation: شایستگی جمعیت جهش یافته که شامل جمعیت کنونی و فرزندان است، محاسبه می‌شود. شایستگی رابطه مستقیم با انجراف معیار هر کروموزوم از داده‌های ورودی دارد.

choose_new_generation: روش انتخاب شده Mu + Lambda خواهد بود. به این صورت که هر دو گروه جمعیت کنونی و فرزندان در یک لیست بر حسب شایستگی سورت شده و سپس، به اندازه نصف Mu از بالای لیست و نیمی دیگر از پایین لیست برگردانده می‌شود. لیست جدید همان جمعیت نسل بعد است.

نکات:

- در هر نسل، شایستگی بهترین کروموزوم، بدترین آن و میانگین شایستگی چاپ می‌شود.
- شرط خاتمه الگوریتم تکاملی پایان یافتن تعداد نسل خواهد بود. در نسل‌های آخر شاهد همگرایی شایستگی میانگین هستیم.

پارامترهای مسئله به صورت زیر هستند:

generations: تعداد نسل ها

TOURNAMENT_SIZE: اندازه تورنمنت

crossover_probability: احتمال ترکیب

- تعداد نسل 70 در نظر گرفته شده است.
- اگر احتمال کراس اور را افزایش دهیم مسئله دیرتر همگرا می‌شود و اگر احتمال آن را پایین بیاوریم حرکتی نیز تغییر نسل‌ها نسبت به هم کم شده و ممکن است باعث همگرایی زودرس شود.
- اگر تعداد فرزندان هر نسل کمتر شود حرکتی صورت نگرفته و ممکن است به جواب نرسیم.
- اگر احتمال جهش بالا باشد حرکت بیشتری صورت می‌گیرد و فضای بیشتری جستجو می‌شود.
- جستجوی بهینه با مقدار میانه‌ای از ضرایب بالا انجام می‌گردد.

ضرایب در نظر گرفته شده به شکل زیر است:

generations = 70

score = 0

Mu = 80

Lambda = 2 * Mu

crossover_probability = 0.4

TOURNAMENT_SIZE = 4

نتایج آزمایش **Dataset1:**

نسل اول:

best score: 22.716603700250825

worst score: 5.389961729291503

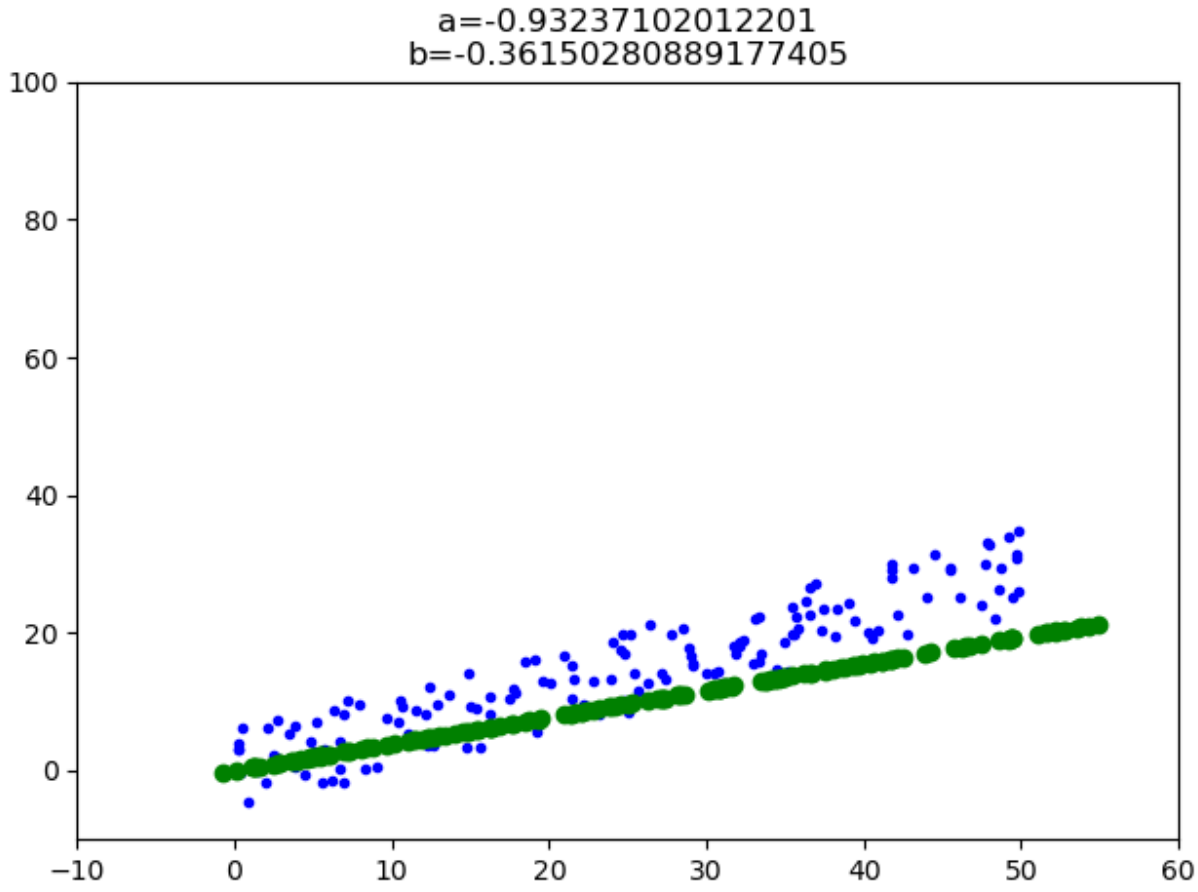
average: 16.578939095732885

نسل آخر (نسل 70):

best score: 7.096419152877146

worst score: 7.096419152877146

average: 7.096419152877146



نتایج آزمایش **Dataset2:**

نسل اول:

best score: 37.17327957503909

worst score: 16.08271430598159

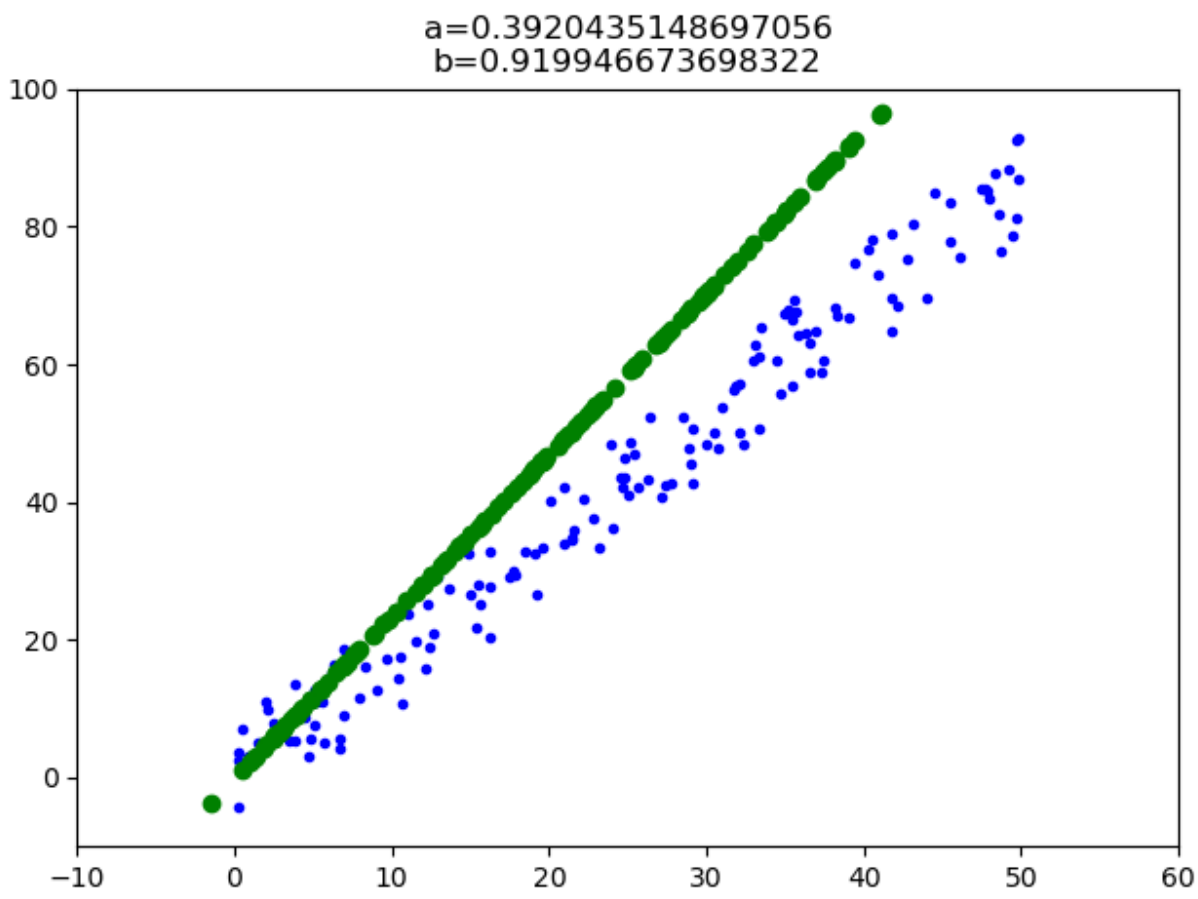
average: 29.208098336624573

نسل آخر (نسل 70):

best score: 29.794067470351624

worst score: 29.794067470351624

average: 29.79406747035164


a=0.3920435148697056
b=0.919946673698322
100
80
60
40
20
0
−10
0
10
20
30
40
50
60