

ابتدا من با یک تابع اطلاعات را از فایل های دیتا خواندم.
ولی دیتا رو خودم دستی تمیز کردم یعنی اون اوایل فایل که نوشته بود را حذف کردم.

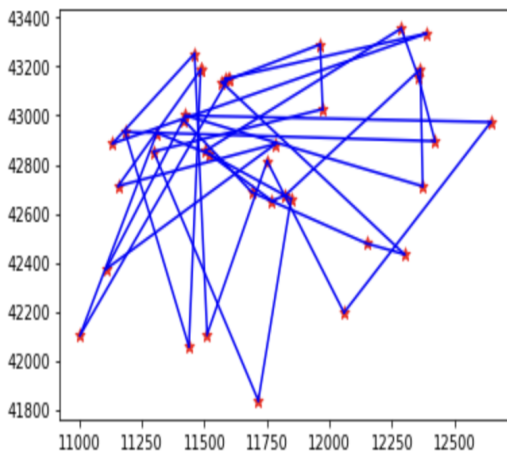
چون میشد یک تور n تایی را به یک جایگشت n تایی تناظر یک به یک داد بنده در تمامی این بخش یک جواب که تور n تایی است را یک لیست n تایی از رئوس ساختم.

ابتدا چند تا تابع کاربردی تعریف کردم مانند فاصله بین دو راس یا همسایه یک تور (همسایه توری میشود که فقط در دو تا عنصر با جایگشت ما اختلاف داشته باشد) را رندوم بدهد و محاسبه هزینه یک تور ...

الگوریتم hill climbing :

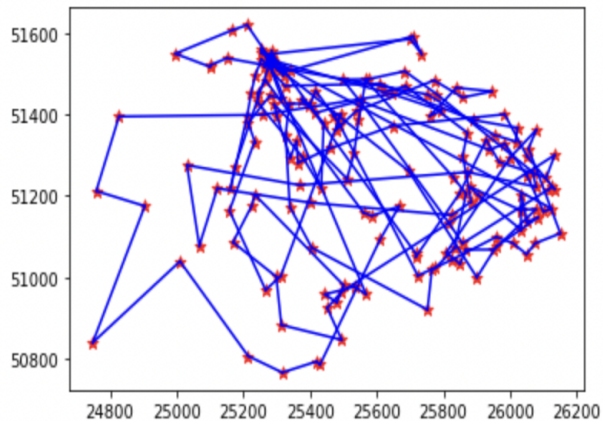
در این الگوریتم ما از یک استتیت که یک تور رندوم است شروع میکنیم و هر مرحله جای دو عنصر جابجا میکنیم به عبارت دیگر به همسایه این استتیت میرویم اگر این جاب جایی هزینه ما رو کاهش دهد.
ولی این الگوریتم ممکن است در یک جواب مینیمال بماند و جواب همه همسایه های آن بدتر از آن باشد در صورتی که این جواب مینیم نیست . برای همین من از ۲۰ نقطه متفاوت شروع کردم که جواب بهتری پیدا کنم.

19727.566794958628



جواب دیبوجی

27583.641953800303

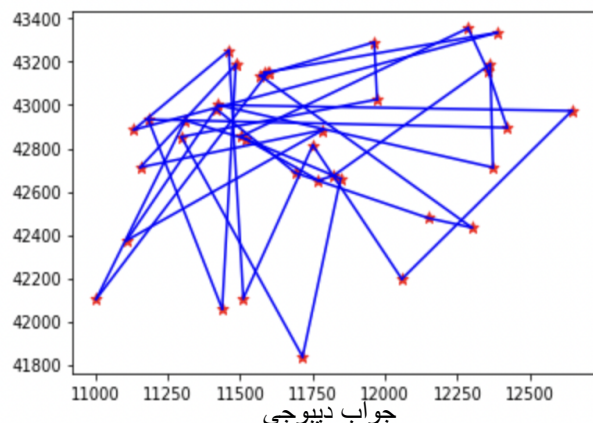


جواب قطر

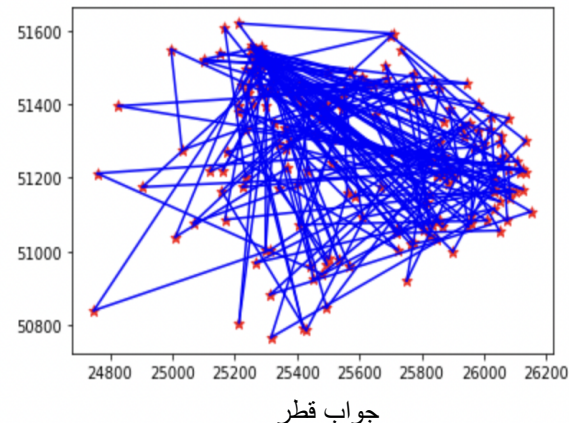
الگوریتم simulated annealing :

در این الگوریتم نیز از یک جایگشت رندوم شروع میکنیم و جواب آن را محاسبه میکنیم و سپس یک همسایه رندوم یعنی جای دو عنصر در این جایگشت را عوض میکنیم اگر جواب بهتر شد به آن می رویم آگه جواب بدتر شد به احتمال یک بر روی دو برابر گام به آن جواب بد وارد میشویم .
به عبارت دیگر الگوریتم در ابتدا راحت تر به جواب بد می رود هرچه جلو تر برویم این احتمال به همسایه بد رفتن کم تر میشود.

19727.566794958628



80371.17985722372

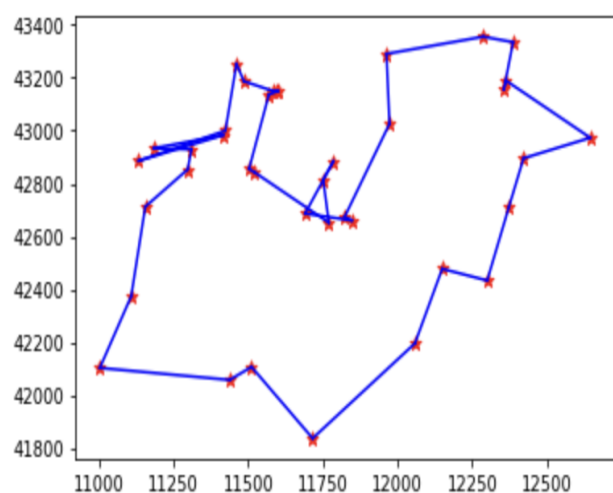


جواب قطر به نظر خوب نماید و گویا این الگوریتم روی تعداد شهر زیاد کار نمیکند و زیادی رندوم عمل میکند به عبارت دیگر روی یک جواب مینیمال گیر میکند و هوشمندی کمی دارد. ولی برای دیبوجی که شهر کمی دارد جواب منطقی تری دارد.

الگوریتم genetic :

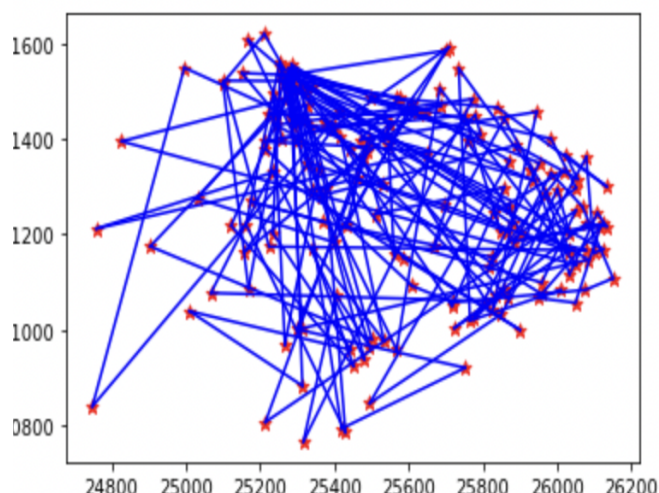
در این الگوریتم من ابتدا تعداد قابل توجهی جایگشت ساختم به عنوان جمعیت اولیه و از همان ها خوب هاشونو یعنی اونایی که هزینه کمتری دارند را انتخاب کردم .
در هر مرحله نسل برتر را انتخاب کرده و بین هر دو تایی از نسل برتر من از آن ها بچه ساختم به این صورت که مثلا از یک تا i که i عددی رندوم است از پدر انتخاب شود مابقی اعدادی که وارد نشده اند به ترتیب مادر وارد شوند .
در این صورت نسل بعدی ساخته میشود و بین آن ها باز تسل برتر انتخاب میشود .

7816.638534270056



جواب دیبوجی

7297.87314251684



جواب قطر

این الگوریتم جواب دیبوجی را خیلی بهتر کرد نزدیک به مینیمم و جواب قطر هم مقداری بهتر کرد این نشان میدهد که این الگوریتم برای کشور های که شهر های کمی دارند خیلی خوب جواب میدهد و در کل از الگوریتم های قبلی خیلی بهینه تر است.