

به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه ۲

سید نوید کرمی نژاد

۹۴۳۱۰۷۰

شرح پروژه:

در این پروژه، مجموعه‌ای از الگوریتم‌های جستجوی غیر کلاسیک که فهرست آن‌ها در زیر آورده شده‌است پیاده‌سازی و از آن‌ها برای حل چند مسئله جستجوی مختلف که در قالب یک واسط مشخص پیاده‌سازی می‌شوند، استفاده شده است.

الگوریتم‌ها:

الگوریتم‌هایی که باید پیاده‌سازی شوند، عبارت‌اند از:

- گرم و سرد کردن تدریجی (Simulated Annealing)
- تپه‌نوردی (ساده، تصادفی، اولین انتخاب و شروع مجدد تصادفی)
- الگوریتم ژنتیکی

مسائل:

به ازای هر یک از مسائل باید یک کلاس Problem تعیین شود که در آن توابع لازم برای تعریف مسئله (تابع حالت اولیه، تابع همسایگان هر حالت و تابع شایستگی(هدف)) پیاده‌سازی شده باشند.

مسئله اول : هشت وزیر

در این مسئله هدف این است که ۸ وزیر را طوری در صفحه شطرنج قرار دهیم که هیچ کدام از وزیرها همدیگر را نزنند.

این مسئله به کمک الگوریتم‌های تپه‌نوردی حل و نتایج آن به صورت زیر است:

- تپه‌نوردی ساده

```
[5, 6, 5, 5, 5, 5, 3, 1]
Distance From Goal: 14.0
[4, 6, 5, 5, 5, 5, 3, 1]
Distance From Goal: 9.0
[4, 6, 0, 5, 5, 5, 3, 1]
Distance From Goal: 5.0
[4, 6, 0, 2, 5, 5, 3, 1]
Distance From Goal: 2.0
[4, 6, 0, 2, 7, 5, 3, 1]
Distance From Goal: 0.0
Visited Nodes: 4
Expanded Nodes 4
```

```
We Are In Trap!!!
```

```
Process finished with exit code 0
```

این الگوریتم، کامل نیست زیرا همانطور که در بالا مشاهده می‌کنید ممکن است در اکستریم‌های محلی گیر کند. (از ۱۰ بار اجرای الگوریتم تنها ۱ بار به هدف رسید)

- تپهنوردی تصادفی

```
[2, 2, 7, 4, 7, 5, 0, 7]  
Distance From Goal: 6.0  
[2, 2, 6, 4, 7, 5, 0, 7]  
Distance From Goal: 4.0  
[2, 2, 6, 4, 7, 5, 0, 3]  
Distance From Goal: 3.0  
[2, 2, 6, 1, 7, 5, 0, 3]  
Distance From Goal: 2.0  
[2, 2, 6, 1, 7, 4, 0, 3]  
Distance From Goal: 1.0  
[5, 2, 6, 1, 7, 4, 0, 3]  
Distance From Goal: 0.0  
Visited Nodes: 5  
Expanded Nodes 5
```

```
We Are In Trap!!!
```

```
Process finished with exit code 0
```

این روش نیز همانند الگوریتم قبل کامل نیست و از ۱۰ بار اجرا تنها ۱ بار به هدف رسید. از نظر تعداد گره‌های گسترش یافته نیز در حالت نشان داده شده ضعیف‌تر عمل کرده است اما نمی‌توان گفت این الگوریتم نسبت به حالت ساده همواره از نظر زمان اجرا و حافظه‌ی مصرفی بدتر یا بهتر عمل می‌کند.

- تپهنوردی اولین انتخاب

```
[1, 0, 1, 1, 5, 6, 3, 5]  
Distance From Goal: 11.0  
[0, 0, 1, 1, 5, 6, 3, 5]  
Distance From Goal: 7.0  
[2, 0, 1, 1, 5, 6, 3, 5]  
Distance From Goal: 6.0  
[2, 4, 1, 1, 5, 6, 3, 5]  
Distance From Goal: 5.0  
[2, 4, 1, 3, 5, 6, 3, 5]  
Distance From Goal: 4.0  
[2, 4, 1, 7, 5, 6, 3, 5]  
Distance From Goal: 3.0  
[2, 4, 1, 7, 0, 6, 3, 5]  
Distance From Goal: 0.0  
Visited Nodes: 6  
Expanded Nodes 6
```

```
We Are In Trap!!!
```

```
Process finished with exit code 0
```

همانند الگوریتم‌های بالا کامل نیست اما از نظر درصد موفقیت در رسیدن به حالت هدف بهتر عمل می‌کند (از ۱۰ بار اجرای الگوریتم ۴ بار به هدف رسید) و مشابه قبل از نظر زمان اجرا و حافظه مصرفی نمی‌توان مقایسه‌ای بین آن‌ها انجام داد.

- تپهنوردی شروع مجدد تصادفی (برای این الگوریتم سقف ۱۰ بارهی شروع مجدد در نظر گرفته شده است اما می توان به دلخواه آن را تغییر داد)

```
[1, 4, 1, 4, 5, 1, 6, 6]
Distance From Goal: 9.0
[2, 4, 1, 4, 5, 1, 6, 6]
Distance From Goal: 5.0
[2, 4, 1, 7, 5, 1, 6, 6]
Distance From Goal: 3.0
[2, 4, 1, 7, 5, 1, 6, 0]
Distance From Goal: 1.0
[2, 4, 1, 7, 5, 3, 6, 0]
Distance From Goal: 0.0
Visited Nodes: 4
Expanded Nodes 4
```

```
We Are In Trap!!!

Process finished with exit code 0
```

این الگوریتم نیز کامل نیست اما هرچه تعداد دفعاتی که از نقطه ای جدید شروع به حرکت کنیم بیشتر باشد احتمال رسیدن به هدف بیشتر می شود. با انتخاب ۱۰ به عنوان تعداد تلاش برای شروع مجدد از ۱۰ بار اجرای الگوریتم در ۹ اجرا به هدف رسیدیم.

و نیز از نظر حافظه ی مصرفی و زمان اجرا نمی توان به یقین مقایسه ای با سایر الگوریتم های تپهنوردی انجام داد زیرا این مقادیر به حالت اولیه بستگی دارند و چون این حالت به صورت تصادفی انتخاب می شود مقایسه امکان پذیر نیست.

مسئله دوم : پارتیشن بندی گراف

هدف این مسئله آن است که یک گراف همبند ورودی را طوری به دو زیر گراف تقسیم کنیم که تعداد گره ها در دو طرف تقریباً برابر و تعداد یال هایی که این دو زیر گراف را به یکدیگر متصل می کنند کمترین باشد.

برای این مسئله تابع هدف (شایستگی) زیر در نظر گرفته شده است:

$$\text{میزان شایستگی} = \frac{\text{تعداد گره های موجود در زیر گراف کوچکتر}}{\text{تعداد گره های موجود در زیر گراف بزرگتر}} + \frac{\text{یال های متصل کننده دو گراف}}{\text{کل یال های موجود}}$$

همچنین از سه روش زیر برای سرد کردن استفاده کرده ایم:

۱- دمای هر مرحله ضریبی از دمای مرحله قبل

$$T_i = T_0 + i \frac{T_0 - T_N}{N} \quad \text{۲. استفاده از فرمول}$$

$$T_i = T_0 * \left(\frac{T_N}{T_0}\right)^{i/N} \quad \text{۳. استفاده از فرمول}$$

که در آن T_0 دمای اولیه، N تعداد دفعات سرد کردن، T_N دمای پایانی است.

نتایج به دست آمده از هر یک از روش های بالا به شرح زیر است:

۱.

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] [0]
[1, 3, 4, 5, 6, 7, 8, 9, 10, 11] [0, 2]
[1, 4, 5, 6, 7, 8, 9, 10, 11] [0, 2, 3]
[4, 5, 6, 7, 8, 9, 10, 11] [0, 2, 3, 1]
[5, 6, 7, 8, 9, 10, 11] [0, 2, 3, 1, 4]
[5, 6, 8, 9, 10, 11] [0, 2, 3, 1, 4, 7]
Counter: 7
Visited Nodes: 7
Extended Nodes: 6
```

۲.

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] [0]
[1, 3, 4, 5, 6, 7, 8, 9, 10, 11] [0, 2]
[3, 4, 5, 6, 7, 8, 9, 10, 11] [0, 2, 1]
[3, 5, 6, 7, 8, 9, 10, 11] [0, 2, 1, 4]
[3, 6, 7, 8, 9, 10, 11] [0, 2, 1, 4, 5]
[3, 6, 8, 9, 10, 11] [0, 2, 1, 4, 5, 7]
Counter: 7
Visited Nodes: 7
Extended Nodes: 6
```

۳.

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] [0]
[1, 3, 4, 5, 6, 7, 8, 9, 10, 11] [0, 2]
[3, 4, 5, 6, 7, 8, 9, 10, 11] [0, 2, 1]
[3, 4, 6, 7, 8, 9, 10, 11] [0, 2, 1, 5]
[4, 6, 7, 8, 9, 10, 11] [0, 2, 1, 5, 3]
[6, 7, 8, 9, 10, 11] [0, 2, 1, 5, 3, 4]
Counter: 7
Visited Nodes: 7
Extended Nodes: 6

Process finished with exit code 0
```

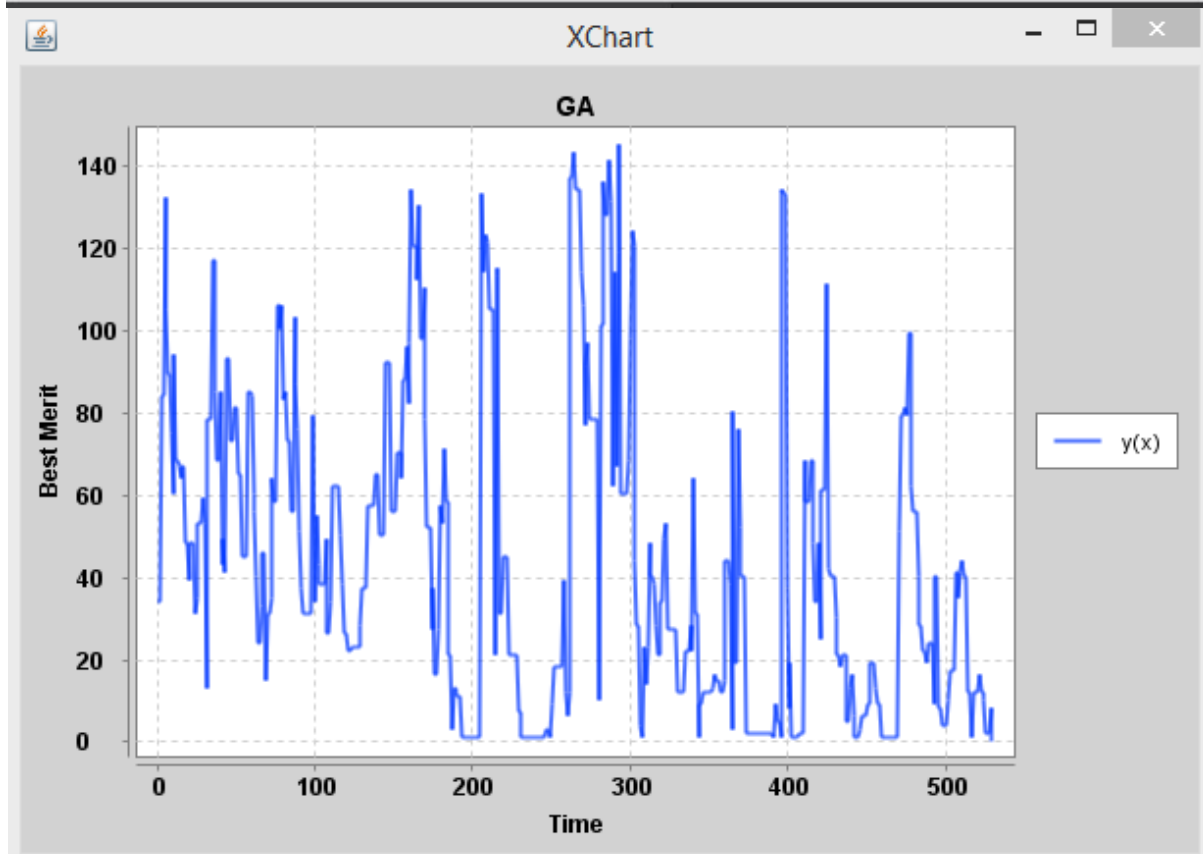
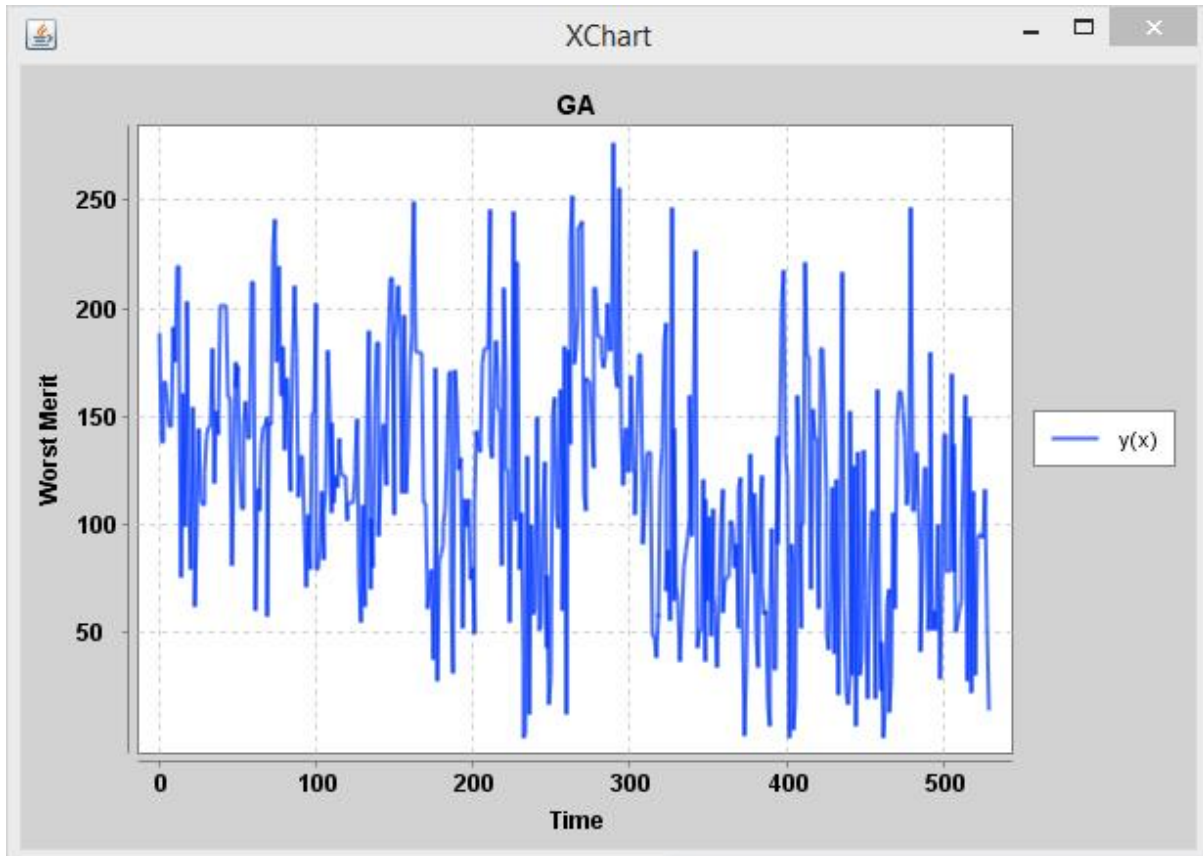
از نظر تعداد گره‌های گسترش داده شده همه‌ی الگوریتم‌ها با هم برابراند و تفاوت مشاهده شده در تقسیم‌بندی زیر گراف‌ها ناشی از تصادفی انتخاب شدن حالت بعدی در الگوریتم سرد کردن تدریجی است.

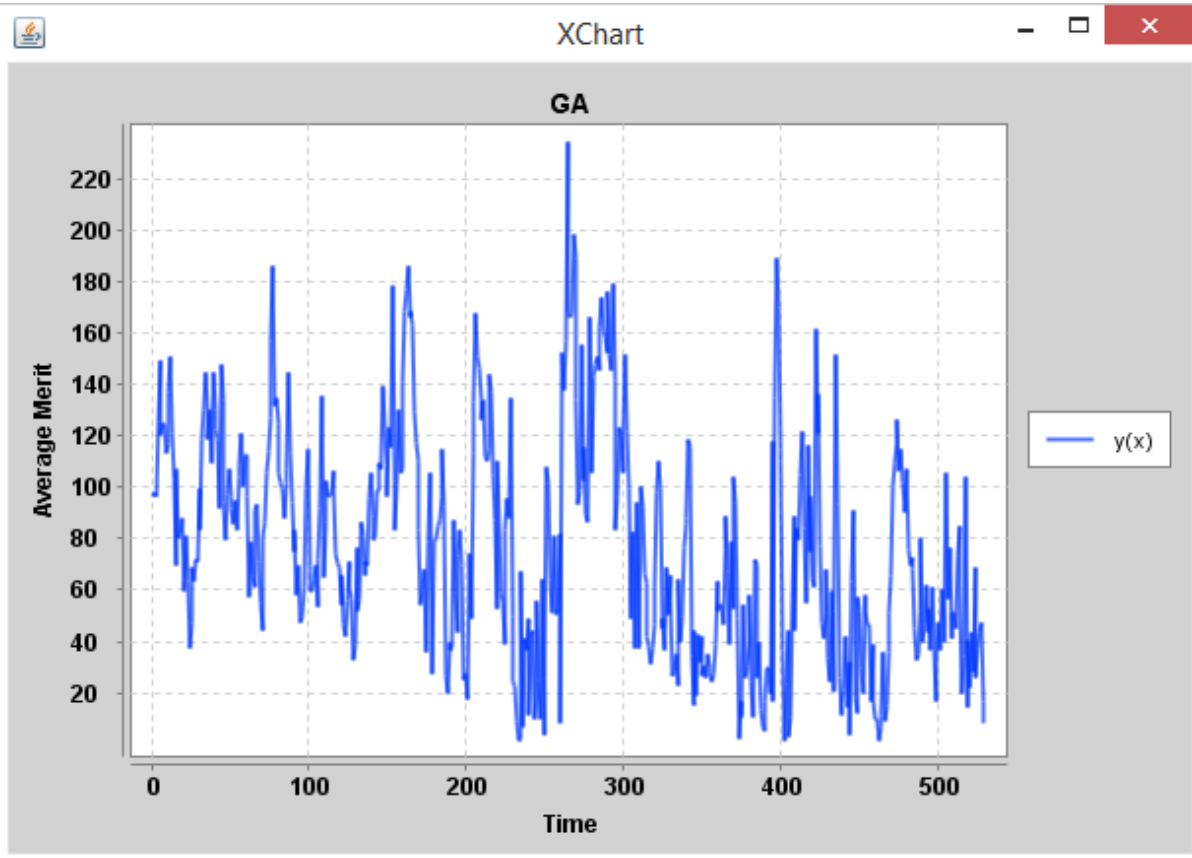
مسئله سوم : حل معادله

در این مسئله هدف حل معادله به کمک الگوریتم ژنتیکی است.
ابتدا با اندازه جمعیت ۶ نتایج به دست آمده به شرح زیر است:

```
0 14 0 3
529

Process finished with exit code 0
```





نتایج دیگر به دست آمده با این اندازه جمعیت:

```
3 7 5 2
4
Process finished with exit code 0
```

```
16 12 0 0
332
Process finished with exit code 0
```

این مسئله نشان می‌دهد که با توجه به نحوه انتخاب والدین و نوع جهش می‌توان نتایج متفاوتی از اجرای الگوریتم بگیریم زیرا انتخاب والدین و میزان جهش به صورت تصادفی است.

حال نتایج به دست آمده با انتخاب اندازه جمعیت برابر ۲۰ را مشاهده می‌کنیم.

```
0 6 8 1
67
Process finished with exit code 0
```

```
21 5 3 0
51

Process finished with exit code 0
```

```
9 8 1 3
33

Process finished with exit code 0
```

همانطور که مشخص است با افزایش جمعیت میزان متوسط تکرار فرایند به نسبت کاهش می‌یابد. پس هرچه اندازه جمعیت بزرگتری انتخاب کنیم احتمال اینکه با انجام فرایند کمتری به جواب برسیم بیشتر خواهد شد.