

به نام وی

گزارش پروژه میانترم مبانی هوش مصنوعی

نام : سید سروش مرتضوی مقدم

شماره دانشجویی : 9631068

## سوال اول)

### شرح مدل سازی مسئله:

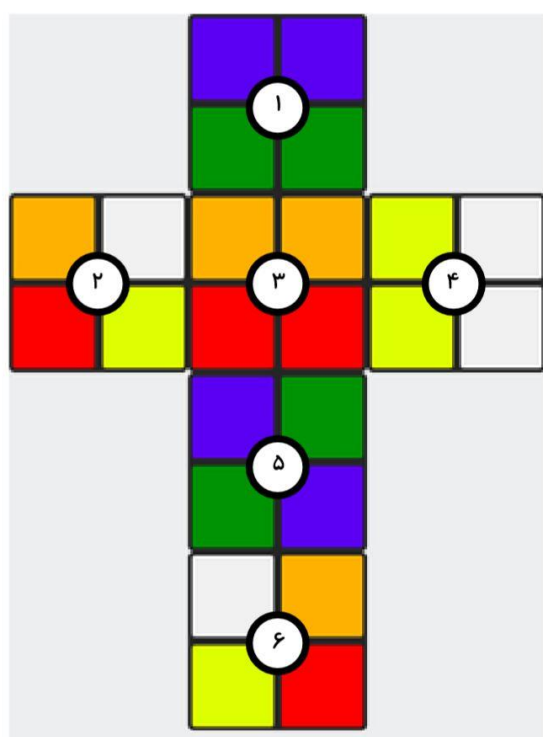
مکعب ما ۶ وجه دارد ، و هر وجه می تواند ساعت گرد و یا پاد ساعت گرد بچرخد. (= ۱۲ حرکت)

برای مدل سازی ، به هر یک از این حرکات یک عدد نسبت داده شده است که از طریق رابطه زیر وجه و جهت چرخش وجه به دست می آید: ( نام گذاری وجه ها طبق تصویر موجود در فایل پروژه است )

- شماره وجه =  $1 + [(move-1) / 2]$  (move شماره حرکت است)

- اگر عدد حرکت زوج باشد ، حرکت پاد ساعت گرد است و اگر فرد باشد ، ساعت گرد.

( مثلا ۶ یعنی : وجه ۳ پادساعت گردد بچرخد. یا ۱۱ یعنی وجه ۶ ساعت گرد بچرخد )



برای اعمال این چرخش ها ابتدا مشخص شده است که با چرخش هر وجه ، کدامین وجه های همسایه و کدامین خانه های آن وجه ها تحت تاثیر قرار خواهند گرفت. برای مثال (طبق تصویر مقابل) در صورت چرخش وجه ۳ ، وجوه ۱ و ۲ و ۴ و ۵ تغییر می کنند. و به ترتیب خانه های (۳و۲) و (۳و۱) و (۲و۰) و (۱و۰) آنها جابجا می شوند. ( به فرضی که شماره خانه های هر وجه از صفر شروع شود و سطر هارا از بالا به پایین و در هر سطر از سمت چپ به راست خانه هارا بشماریم )

این نوع تغییرات برای همه وجوه نوشته شده است و در تابع

change\_state با گرفتن یک پیکر بندی از مکعب و عدد

حرکت ، طبق دانش قبلی از روی نوع تغییرات به ازای هر وجه ، با

توجه به ساعت گرد یا پاد ساعت گرد بودن حرکت ، تغییرات را

اعمال کرده و خانه هارا جابجا می کند. خروجی آن هم پیکر بندی

جدید است.

## الگوریتم IDS

با گرفتن یک عمق اولیه و نهایت عمق قابل بررسی ، الگوریتم با عمق اولیه آغاز می شود و در هر مرحله در صورتی که به جواب نرسیم ، یک واحد به عمق جست و جو اضافه شده و دوباره جست و جو آغاز می شود. تا زمانی که یا به جواب برسیم و یا به عمق حداکثر ( که یعنی به جواب نرسیدیم ) جست و جو در هر مرحله به باز کردن حالات طبق حرکات ممکن می پردازد و با الگوریتم dfs تا عمق حداکثر آن مرحله پیش می رود.

در ادامه نتایج این الگوریتم بر روی دو مورد را مشاهده می کنیم ( اولی با عمق ۵ و دومی با عمق ۷ )

Answer: [6, 4, 5, 5, 1]  
side 3 anti-clockwise  
side 2 anti-clockwise  
side 3 clockwise  
side 3 clockwise  
side 1 clockwise

nodes created : 10623  
opened nodes : 10617  
goal depth : 5  
max nodes in mem : 5

Answer: [1, 3, 6, 4, 5, 5, 1]  
side 1 clockwise  
side 2 clockwise  
side 3 anti-clockwise  
side 2 anti-clockwise  
side 3 clockwise  
side 3 clockwise  
side 1 clockwise

nodes created : 94605  
opened nodes : 94597  
goal depth : 7  
max nodes in mem : 7

## الگوریتم (bidirectional)

در این الگوریتم از سمت گره هدف و از سمت گره شروع ، همزمان به بسط گره ها می پردازیم و در صورتی که یک گره در مجموعه frontier هر یک مشترک بود پس یعنی یک مسیر از شروع تا هدف پیدا شده است .  
افزایش سرعت و کاهش استفاده از حافظه برتری این الگوریتم است.

در ادامه نتایج این الگوریتم بر روی دو مورد را مشاهده می کنیم ( اولی با عمق ۵ و دومی با عمق ۷ )

Answer: [[6, 4, 6], [6, 1]]  
side 3 anti-clockwise  
side 2 anti-clockwise  
side 3 anti-clockwise  
side 3 anti-clockwise  
side 1 clockwise

goal depth : 5  
nodes created : 1392  
opened nodes : 254  
max nodes in Mem : 1139

Answer: [[1, 3, 6, 4], [6, 6, 1]]  
side 1 clockwise  
side 2 clockwise  
side 3 anti-clockwise  
side 2 anti-clockwise  
side 3 anti-clockwise  
side 3 anti-clockwise  
side 1 clockwise

goal depth : 7  
nodes created : 2454  
opened nodes : 447  
max nodes in Mem : 2008

## الگوریتم (UCS)

از انجایی که حرکات در بسط دادن گره ها هزینه یکسان دارند ، UCS مشابه BFS خواهد بود.

یک صف اولویت از گره هایی که باید باز شوند نگه می داریم ، به بسط گره های صف می پردازیم و بچه های آن را به انتهای صف اضافه می کنیم. تست هدف در هنگام تولید است. همچنین طبق تعریف الگوریتم در هر مرحله اگر به گره تکراری رسیدیم ، در صورتی که هزینه کمتری برای رسیدن داشته باشد ( از آنی که در صف است ) جایگزین آن می کنیم . اما در اینجا این مرحله اضافی است چرا که با هزینه ثابت هر حرکت و جست و جوی اول سطح ما مسیر بهینه را خواهیم داشت.

در ادامه نتایج این الگوریتم بر روی دو مورد را مشاهده می کنیم ( اولی با عمق ۵ و دومی با عمق ۷ )

Answer: [6, 4, 5, 5, 1]  
side 3 anti-clockwise  
side 2 anti-clockwise  
side 3 clockwise  
side 3 clockwise  
side 1 clockwise

nodes created : 4531  
opened nodes : 907  
goal depth : 5  
max nodes in mem : 3625

Answer: [1, 3, 6, 4, 5, 5, 1]  
side 1 clockwise  
side 2 clockwise  
side 3 anti-clockwise  
side 2 anti-clockwise  
side 3 clockwise  
side 3 clockwise  
side 1 clockwise

nodes created : 29531  
opened nodes : 5907  
goal depth : 7  
max nodes in mem : 23625

### مقایسه نتایج (

– از نظر عمق جواب همه الگوریتم ها مشابه هم عمل می کنند چرا که کوتاه ترین مسیر را می دهند.

– از نظر تعداد گره های تولید شده : **IDS > UCS > Bidirectional** دلیل آن هم این است که اولاً در الگوریتم IDS ما با افزایش مرحله ای عمق ، تعداد زیادی حالت تکراری خواهیم داشت که باعث می شود بیشتر از همه گره تولید شود. ثانیاً الگوریتم دوجهته با نصف کردن مرتبه زمانی و حجمی گره های کمتری تولید می کند.

– از نظر گره های بست داده شده: **IDS > UCS > Bidirectional** دلیل آن هم مشابه بالا است چرا که به ازای هر گره ای که تولید شده ، گره های بیشتر باز خواهند شد.

– از نظر حداکثر گره در حافظه : **UCS > Bidirectional > IDS** . الگوریتم IDS به این خاطر که ذات dfs دارد از نظر نگه داری گره ها در حافظه خیلی خوب عمل می کند چرا که فقط همان رشته ای که در آن جست و جو می کند را نگه می دارد. در باره دو مورد دیگر هم به همان دلیل نصف شدن مرتبه فضایی در الگوریتم دوجهته ، گره های کمتری در حافظه خواهیم داشت.

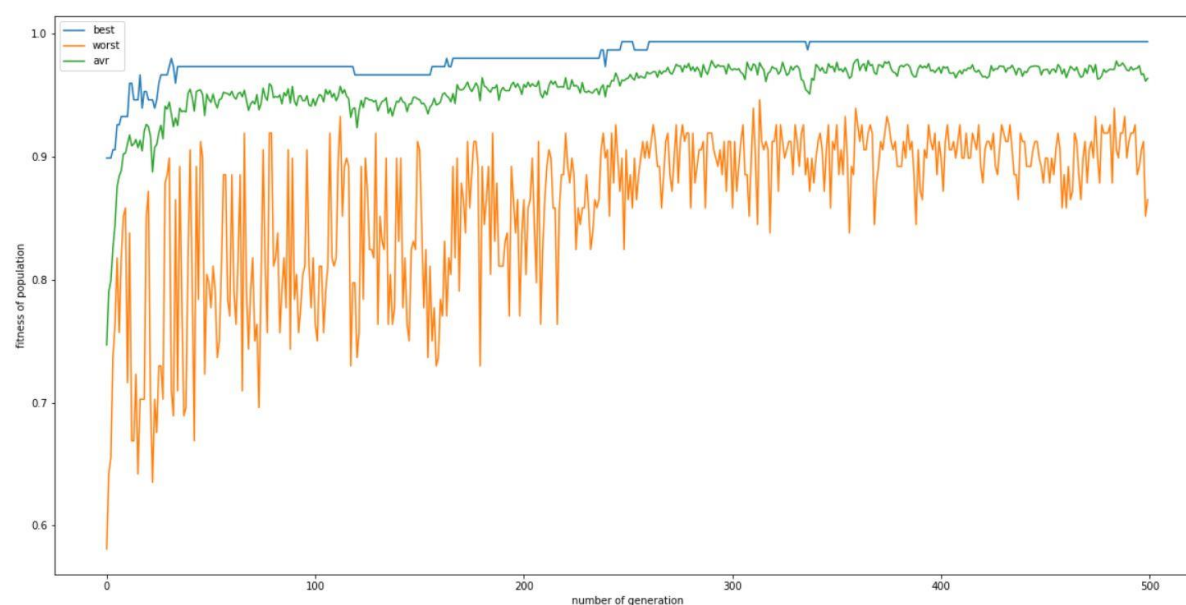
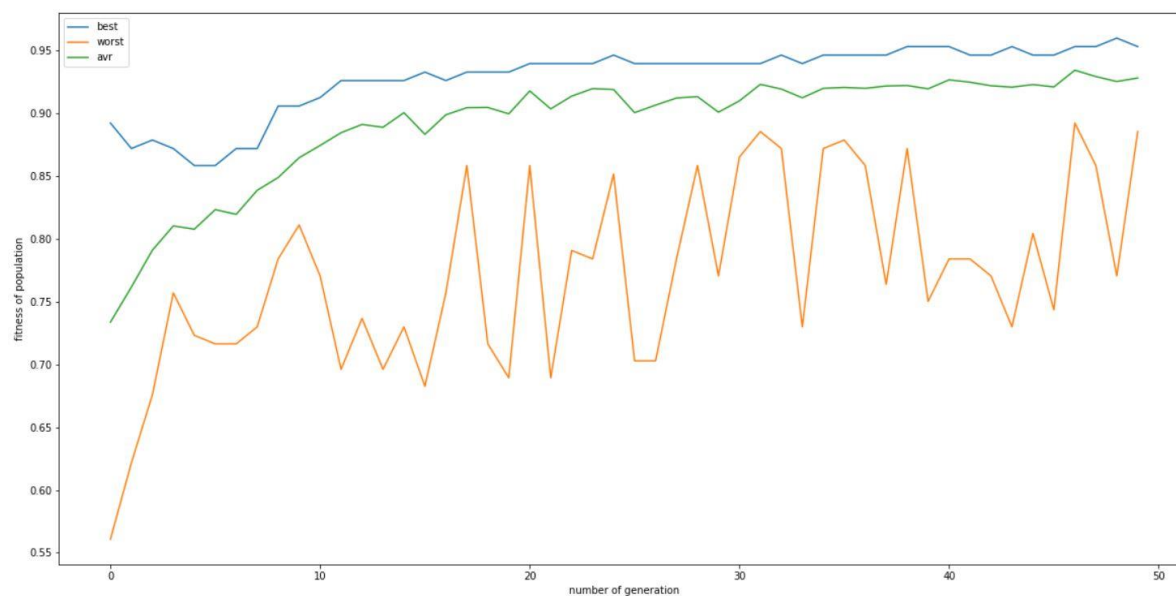
سوال ۲)

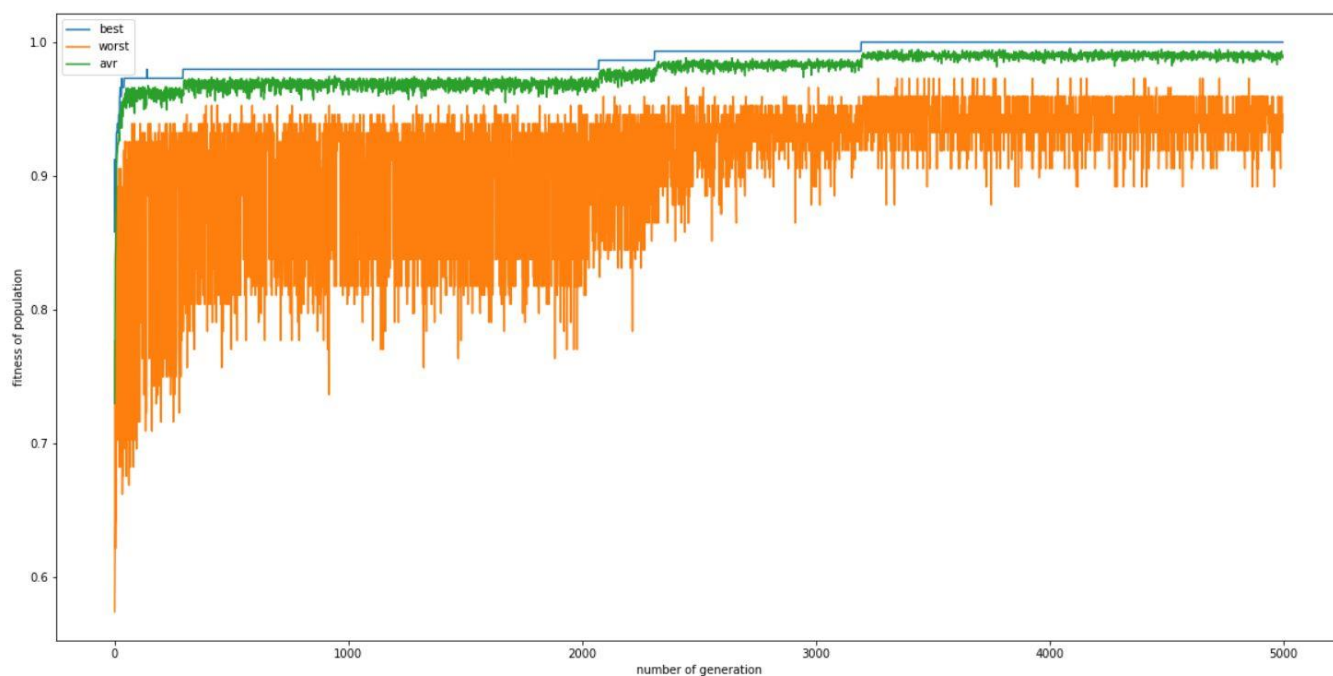
بخش اول - الگوریتم ژنتیک (

با استفاده از الگوریتم ژنتیک و پیاده سازی مراحل ذکر شده در تعریف پروژه ، توانستیم نقشه ایران را رنگ کنیم.

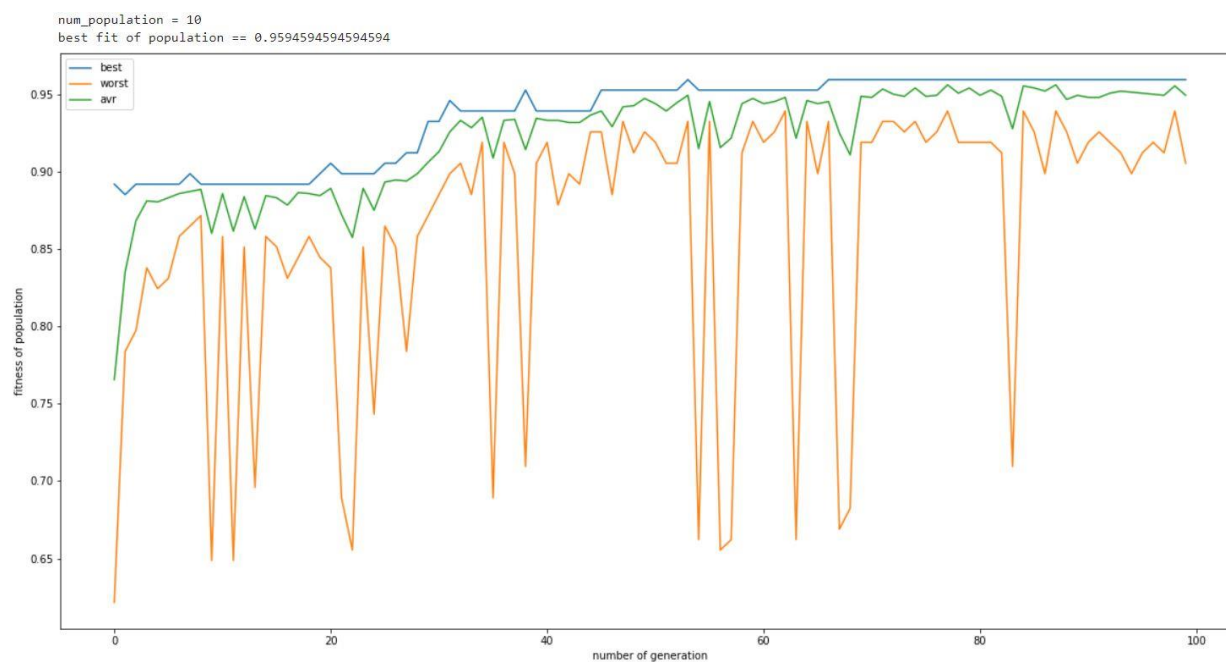
موارد خواسته شده در صورت تعریف پروژه را در این قسمت نشان می دهیم :

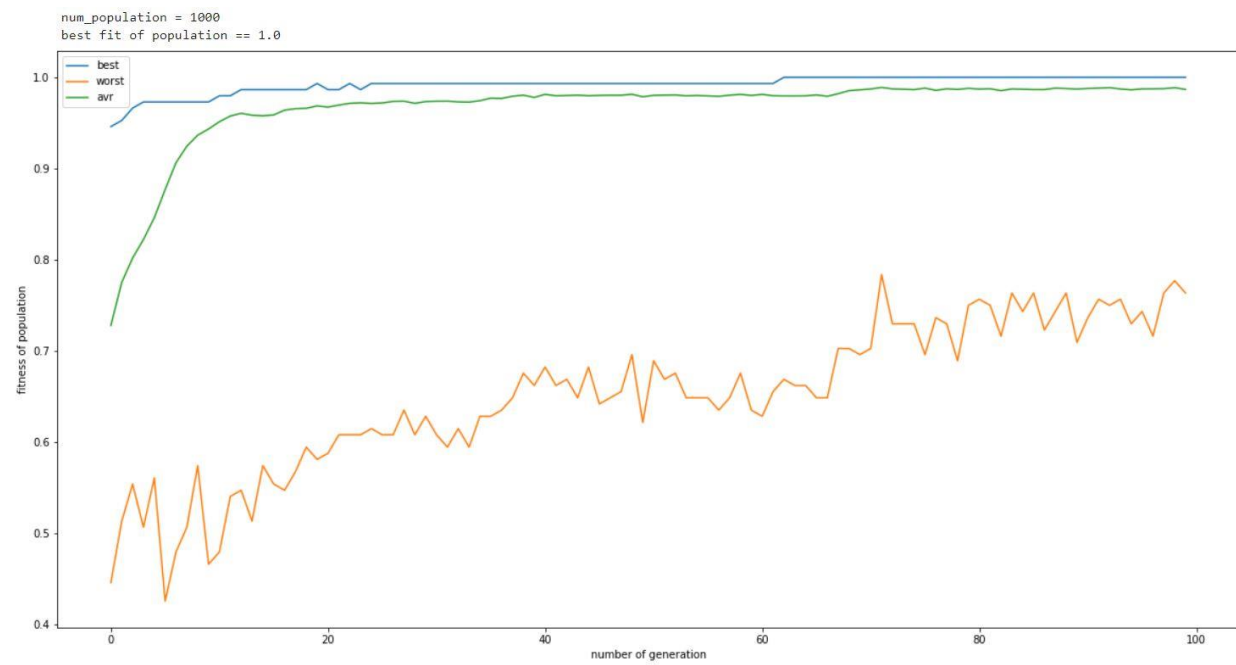
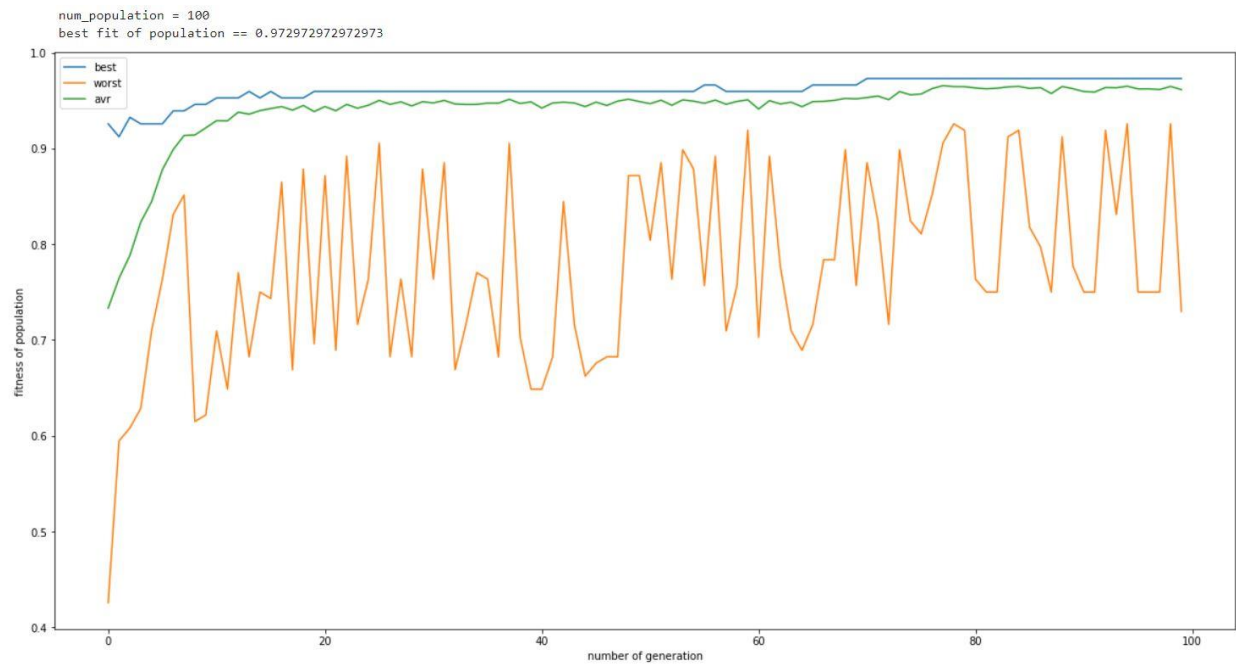
۱) تعداد نسل ها= ۵۰ و ۵۰۰ و ۵۰۰۰ ( جمعیت برابر ۱۰۰ و مرتبطه جهش برابر ۰,۰۲ و ضریب انتخاب ۴)





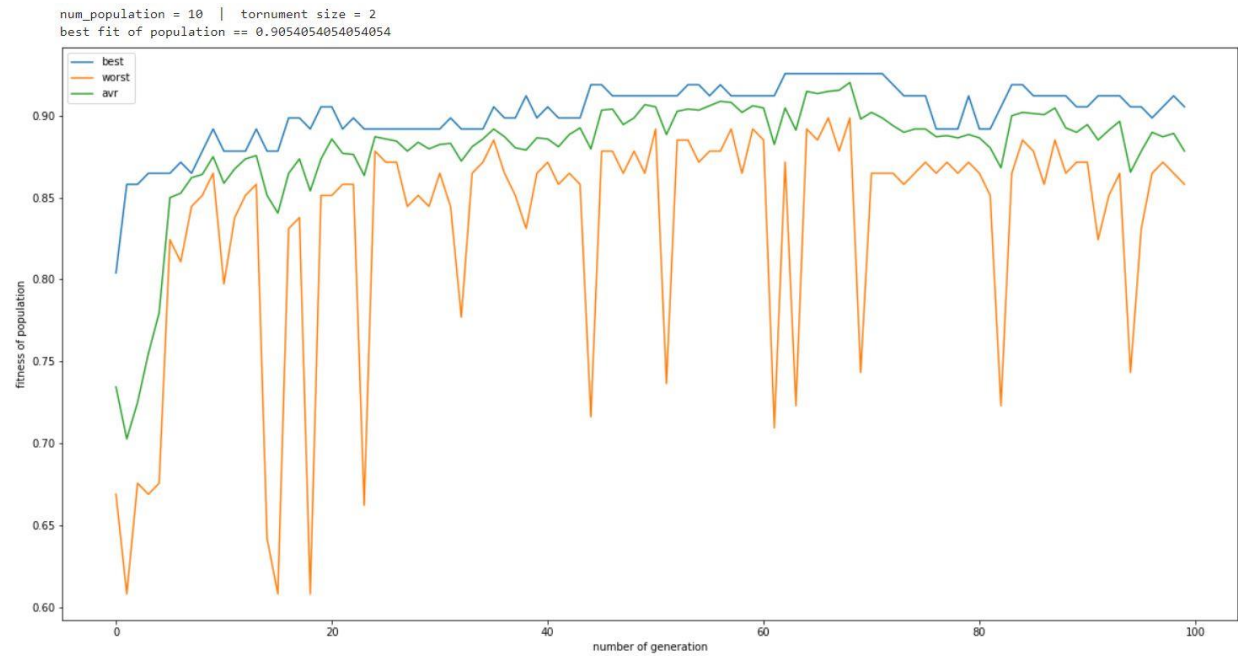
(۲) جمعیت ۱۰ و ۱۰۰ و ۱۰۰۰ (با تعداد ۱۰۰ نسل و نرخ جهش ۰,۰۲ و ضریب انتخاب ۴)



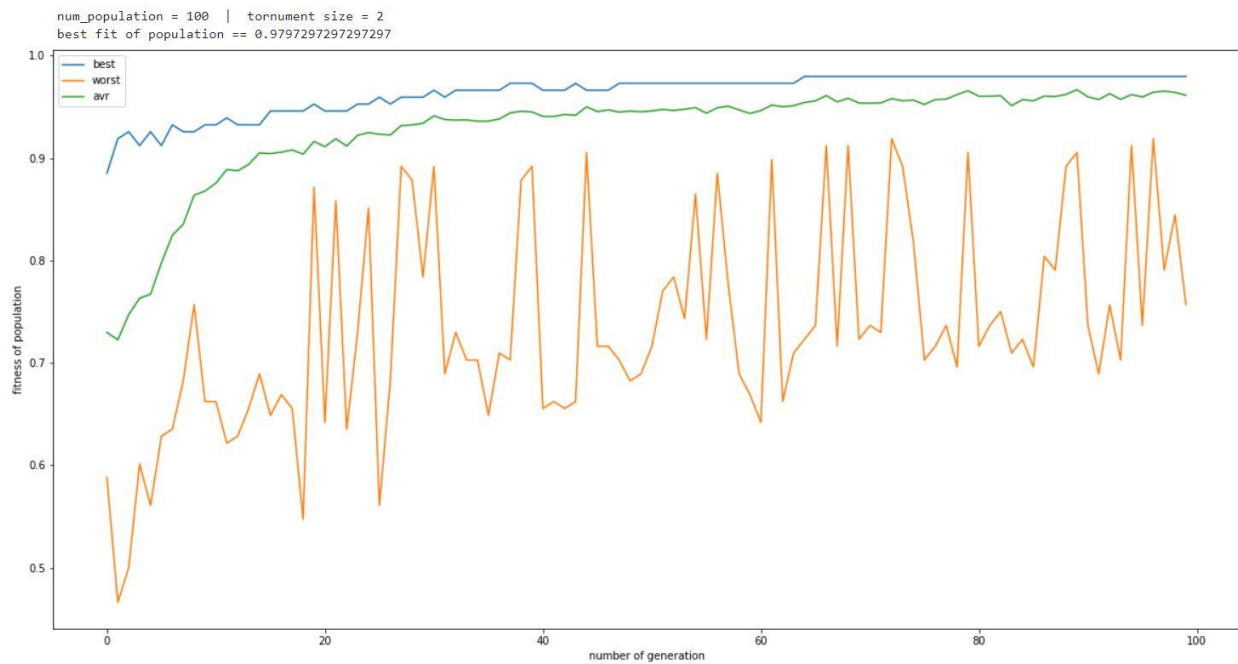


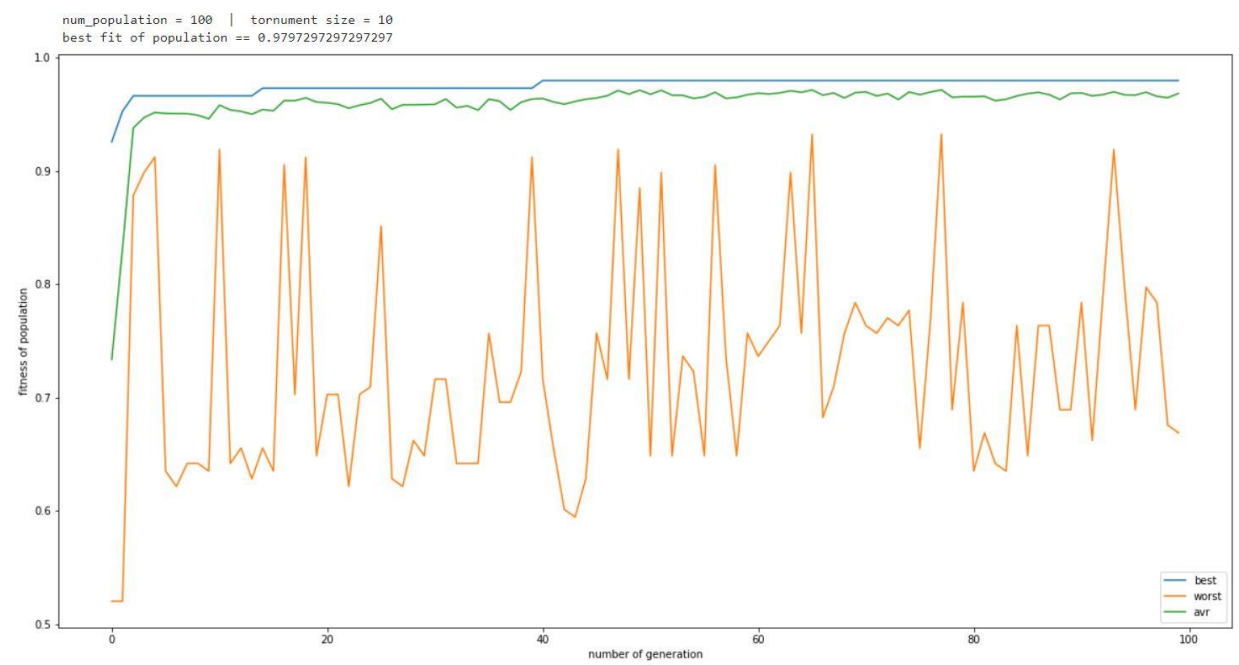
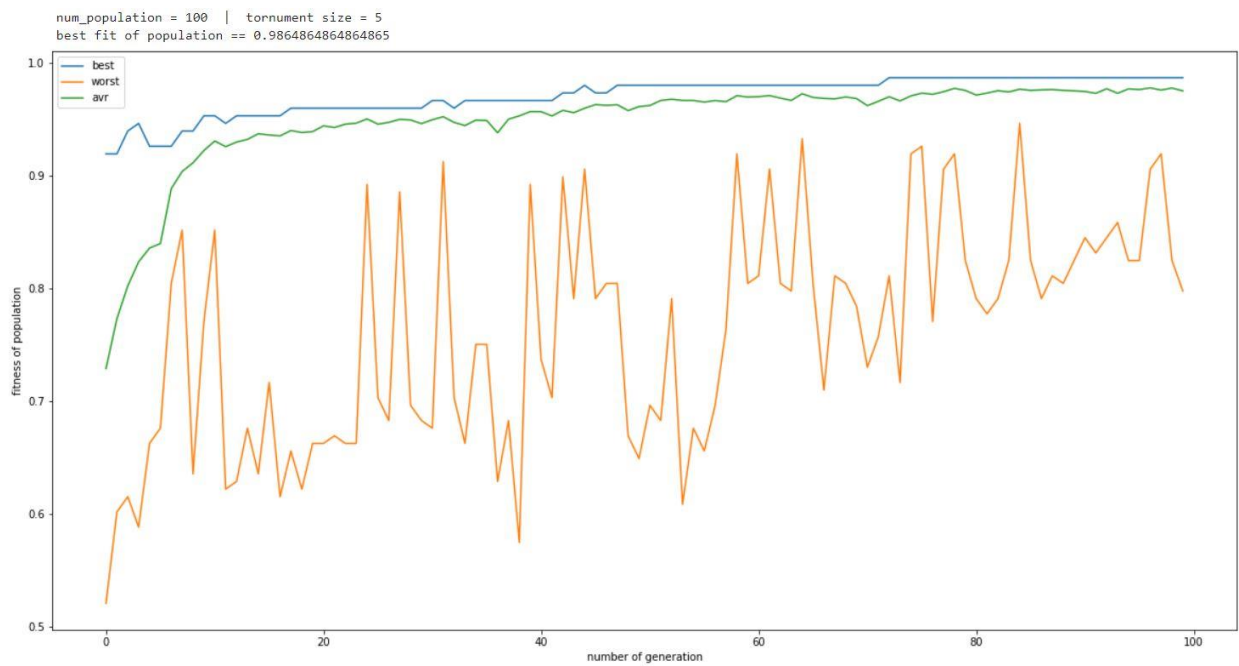


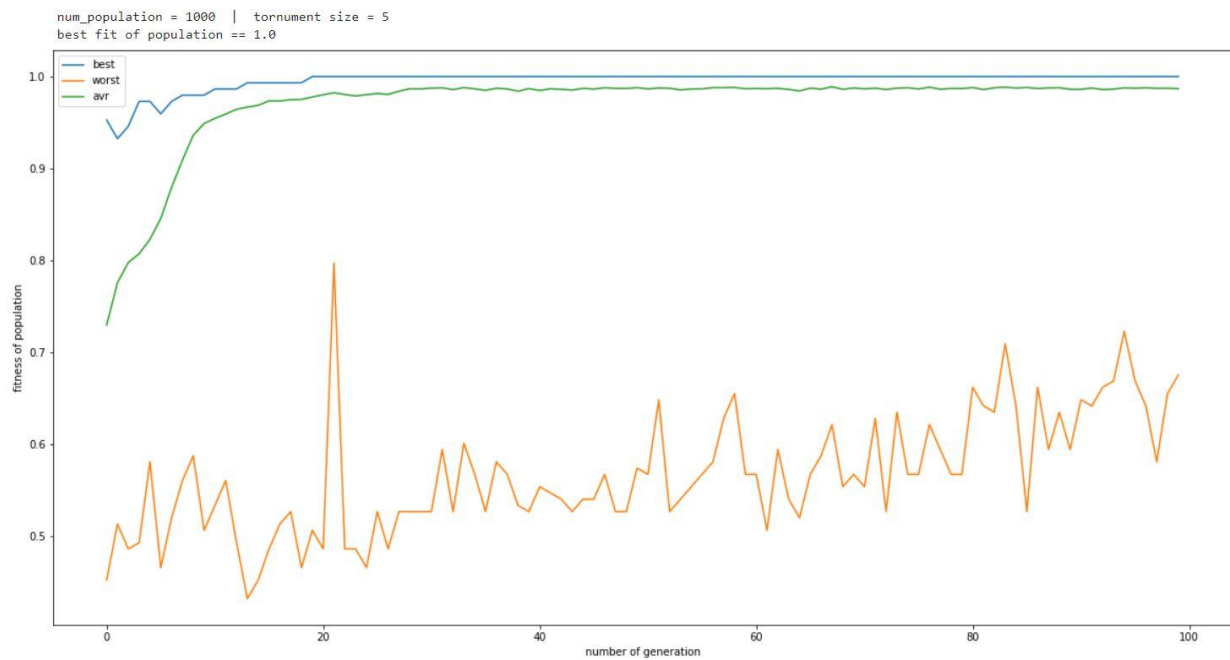
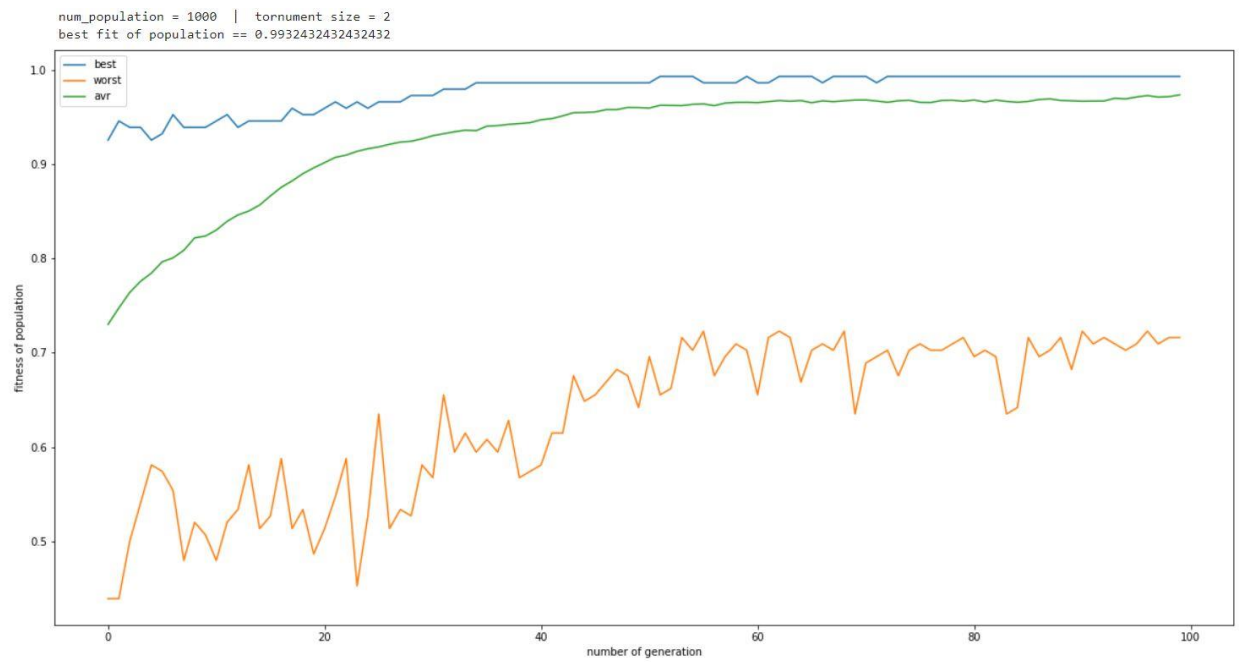
۳) ضریب انتخاب ۲ برای جمعیت ۱۰ (نرخ جهش ۰,۰۲)

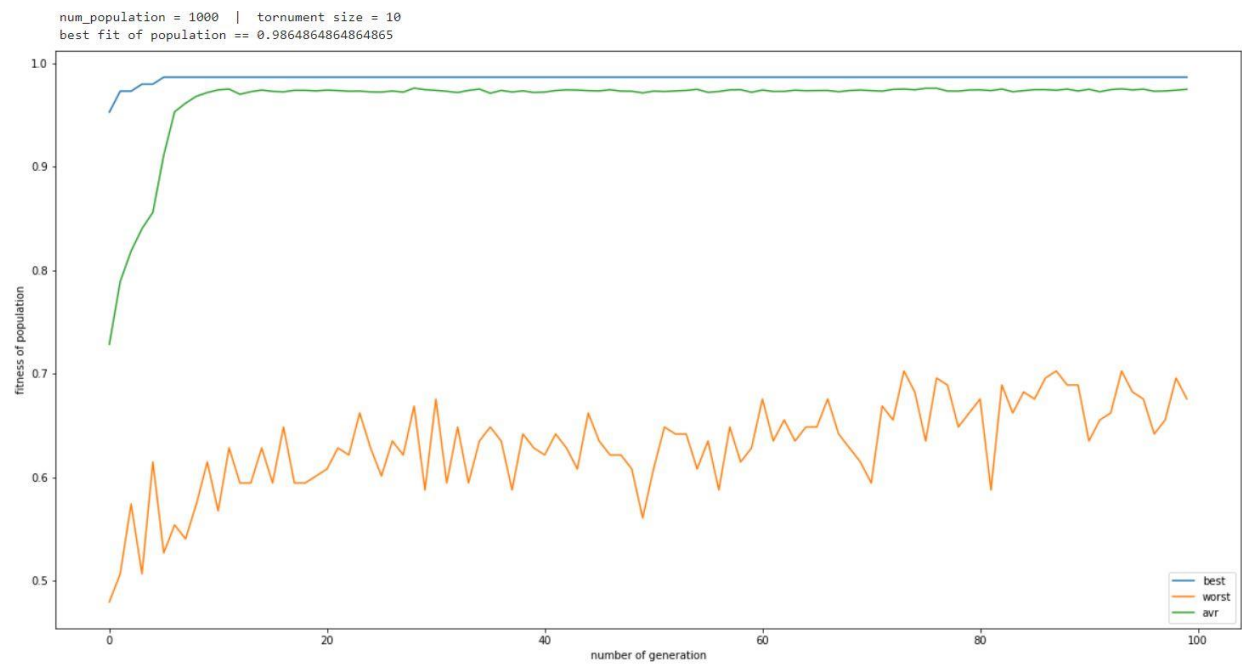


۴) ضریب انتخاب ۲ و ۵ و ۱۰ برای جمعیت ۱۰۰ و ۱۰۰۰ (نرخ جهش ۰,۰۲)

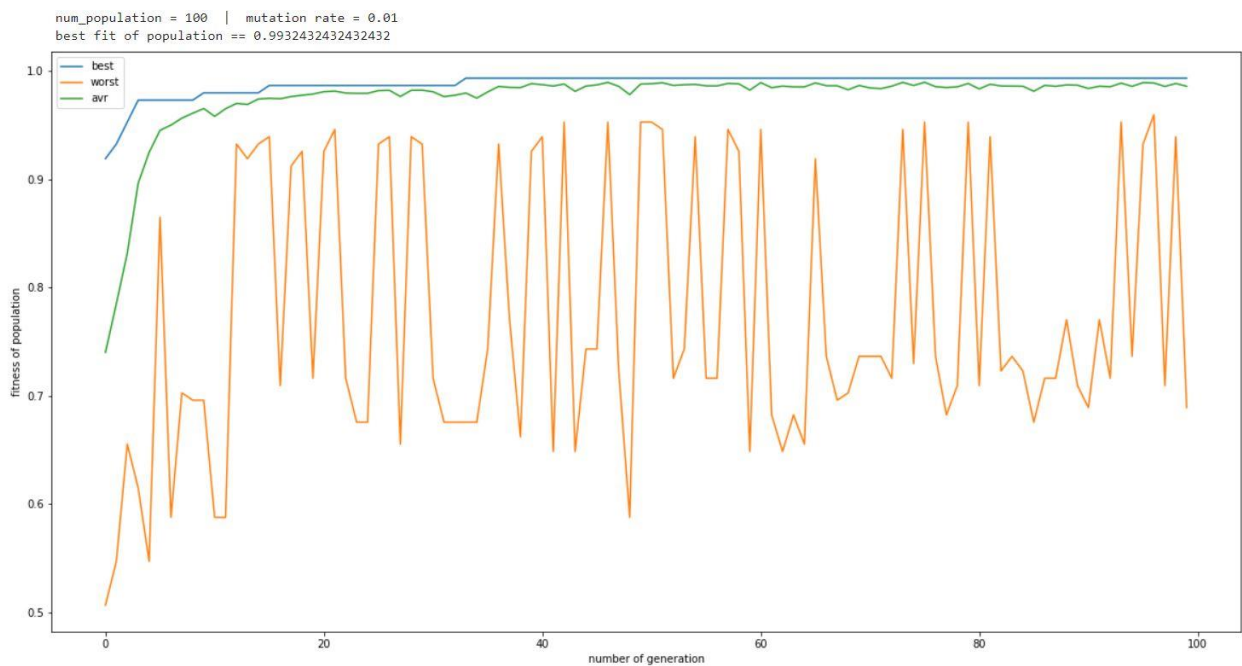




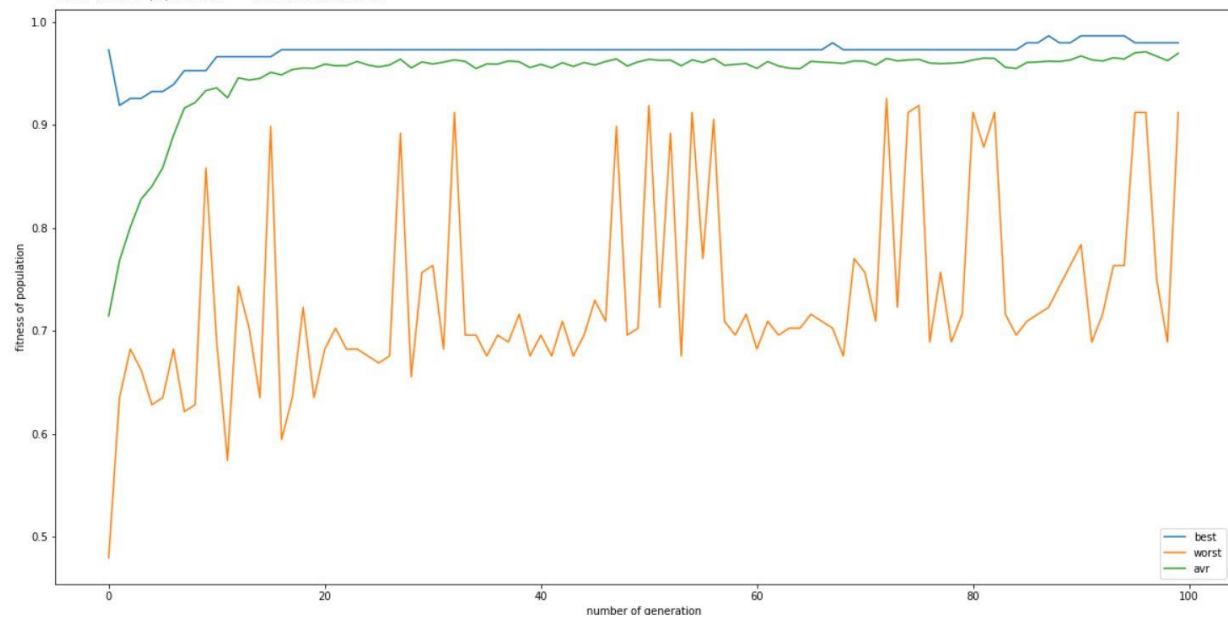




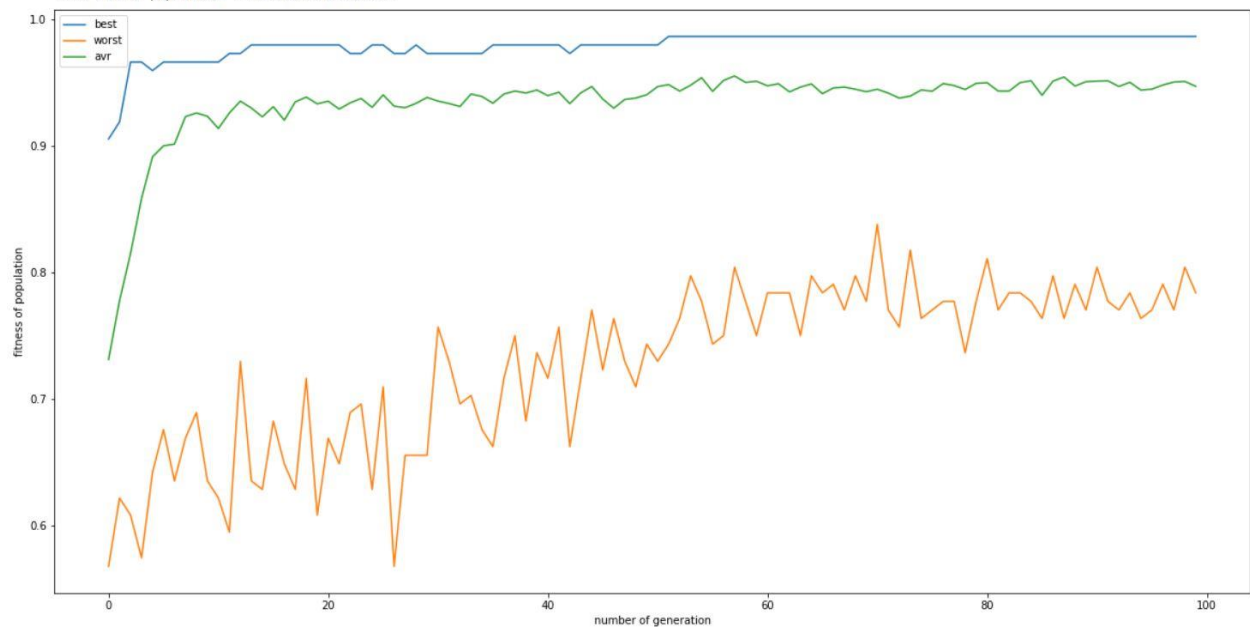
(۵) نرخ جهش ۰,۰۱ و ۰,۰۲ و ۰,۰۵ و ۰,۱ (جمعیت ۱۰۰)

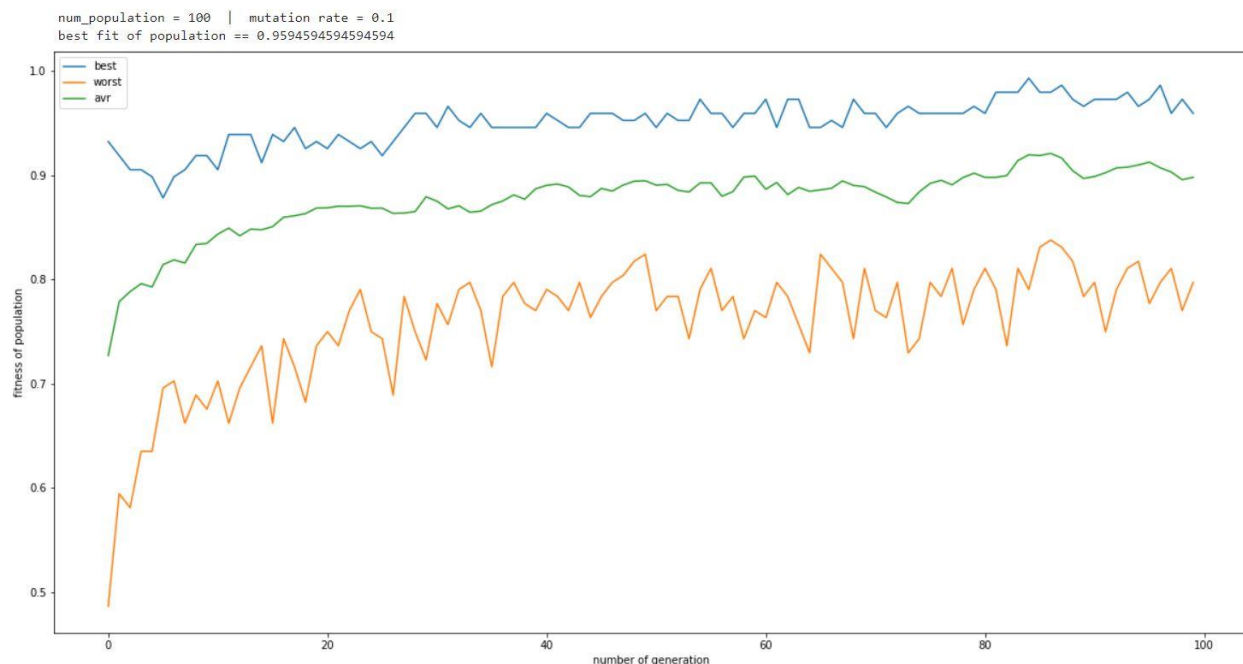


num\_population = 100 | mutation rate = 0.02  
best fit of population == 0.9797297297297297



num\_population = 100 | mutation rate = 0.05  
best fit of population == 0.9864864864864865





نتیجه گیری در مورد نحوه اثر گذاری پارامتر های فوق بر روی همگرایی به هدف:

۱- **شمار جمعیت** : با افزایش شمار جمعیت دو اتفاق کل رخ می دهد . اول اینکه به خاطر افزایش احتمال ایجاد یک کروموزوم شایسته ، احتمال رسیدن به نسلی شایسته تر بالا می رود پس سرعت همگرایی بیشتر می شود. یعنی در انتهای T دوره از جهش ، با احتمال بالاتری به هدف رسیده ایم. دوم اینکه اگر به نمودار ها دقت کنیم متوجه می شویم با زیاد شدن جمعیت ، شمار کروموزوم های ناشایست به نسبت کل جمعیت کاهش میابد. یعنی نمودار میانگین شایستگی به نمودار بهترین شایستگی نزدیک تر می شود تا به بدترین. این اتفاق ناشی از آن است که فضای بیشتر برای تولید نسل بعد ، احتمال افزایش نسبت نسل بهتر را بالا می برد.

۲- **تعداد نسل ها** (دفعات تولید نسل بعد) : تاثیر مهمی که این پارامتر به روی مسئله می گذارد آن است که با دادن فرصت بیشتر برای رسیدن به هدف ، شانس رسیدن به آن را افزایش می دهد. هر چه فرصت بیشتری برای تولید نسل های بعدی داشته باشیم ، احتمال همگرایی بالاتر می رود. مثلا در نمودار ۵۰۰۰ نسل مشاهده می شود که از بین نسل ۳۰۰ تا حدود ۲۰۰۰ ، برتر های جامعه تغییری نکردند . اما بعد از آن یک جهش باعث ایجاد نسل شایسته تری شده که بدون وجود فرصت ، فراهم نمی شد.

۳- **نرخ جهش** : این عامل به نحوه پیچیده تری بروی همگرایی جامعه تاثیر می گذارد. نکته قابل توجه این است که طبق نمودار ها با افزایش نرخ جهش ، شمار افراد با شایستگی کمتر هم بیشتر می شود ، یعنی نسبت کروموزم های نامطلوب افزایش می یابد. این نتیجه جهش های زیاد خواهد بود . همچنین تغییرات نمودار ها

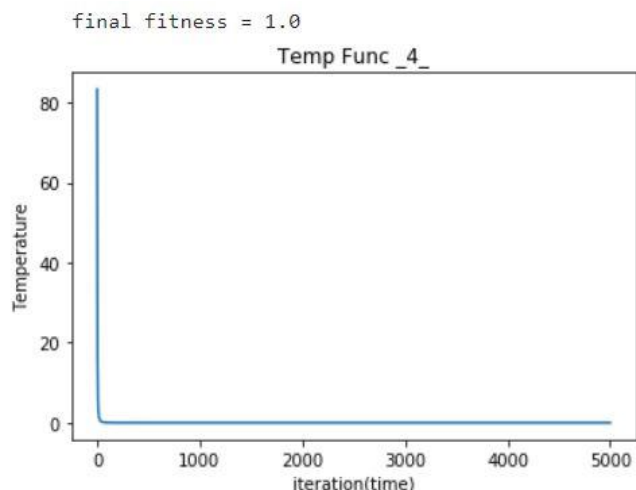
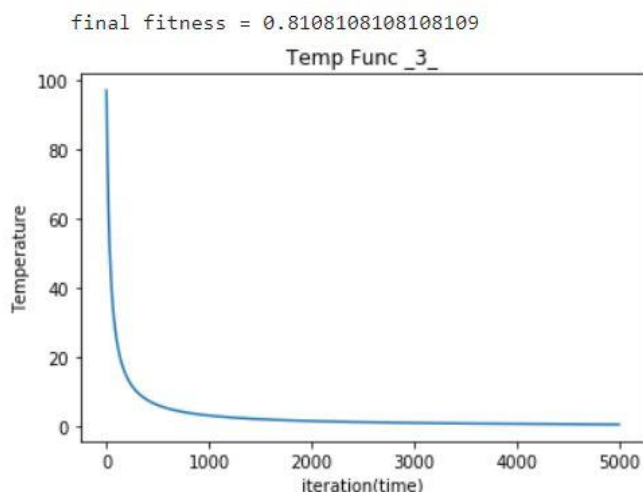
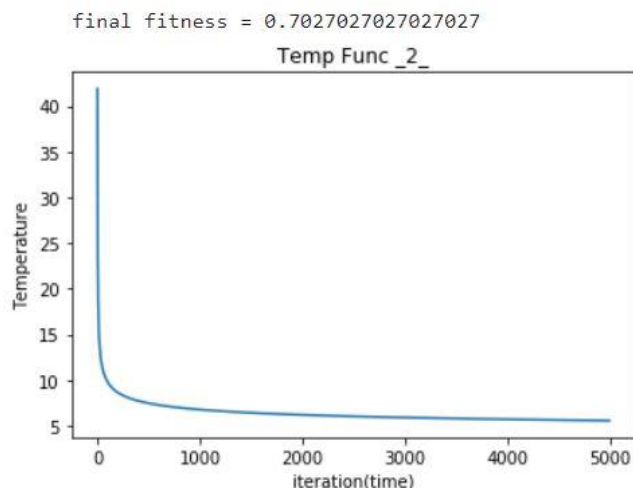
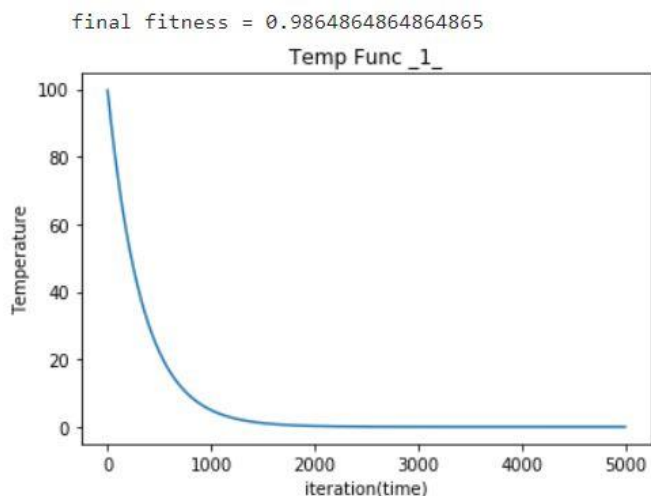
افزایش بیشتری داشته است. و برعکس در جمعیت های با نرخ جهش کمتر ، نمودار ها با ثبات تر بوده و اکثر افراد ژن های بهتری داشته اند (کروموزم های نامطلوب کمتر اند) .

طبق نتایج این افزایش نرخ جهش ابتدا احتمال همگرایی را بالا می برد اما بلافاصله (و در صورت زیاد شدن مداوم) سرعت همگرایی را پایین می آورد. لذا از پارامتر هایی است که باید با شبیه سازی های فراوان به بهترین نقطه آن رسید.

۴- **ضریب انتخاب (tornument size)** : این ضریب در واقع میزان رقابت بین افراد جامعه را تعیین می کند. هرچقدر این ضریب بیشتر ، نسبت افراد برتر هم بیشتر می شود (به نمودار دقت شود) به طوری که در ضریب ۱۰ نمودار میانگین بسیار نزدیک نمودار بهترین های شده. این نرخ باعث کاهش احتمال وجود کروموزوم نامطلوب می شود. جالب تر آنکه این عمل لزوما منجر به افزایش احتمال همگرایی نمی شود . طبق نتایج بین ضرایب ۲ و ۵ و ۱۰ بهترین نتایج مربوط به ضریب ۵ است . این بدین معنا است که برخی از کروموزوم ها که در نسل فعلی عدد شایستگی کمتر دارند میتوانند با یک جهش خیلی بهتر شوند و افزایش ضریب انتخاب شانس باقی ماندن آنها را در نسل بعد کاهش می دهد ، باعث بقای افراد محدودی شده و ممکن است در فرصت کم به همگرایی کامل نرسیم. پس این پارامتر همانند نرخ جهش باید هوشمندانه انتخاب شود و خیلی زیاد در نظر گرفته نشود.

## قسمت ب) الگوریتم شبیه سازی ذوب فلزات :

در ابتدا نمودار تغییرات دما به ازای گذشت زمان را برای هر ۴ تابع داده شده ملاحظه می کنیم:



### بهترین fitness دست یافته شده بالای هر نمودار نوشته شده است

نتایج قابل برداشت : ۱- تابع چهارم از همه بهتر جواب داده . این به این معنی است که احتمالاً اکسترمم های محلی زیادی نداریم و برای همین حرکت به سمت نقطه بهتر با احتمال خوبی ما را به هدف می رساند. ۲- بعد از نمودار چهارم ، نمودار اول بهترین نتیجه را دارد ، بیانگر این که در صورتی که بنا به حرکت تصادفی باشد ، اگر روند سرد کردن آن به اندازه کافی آرام آرام رخ دهد میتوان به نتایج خوبی رسید. (که بعد ازان این نکته در نمودار ۳ هم قابل ملاحظه است چون نسبت به نمودار ۲ اندکی آرام تر سرد شده است.)