به نام خدا

موضوع ارائه: الگوریتم ژنتیک

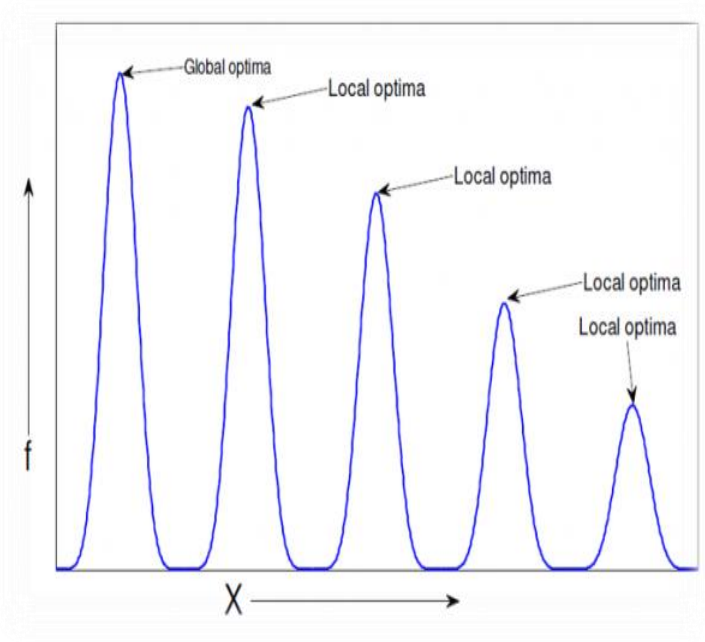
ارائه دهندگان: زهرا رجایی، نیرالسادات اعلائی، مرضیه صالحی

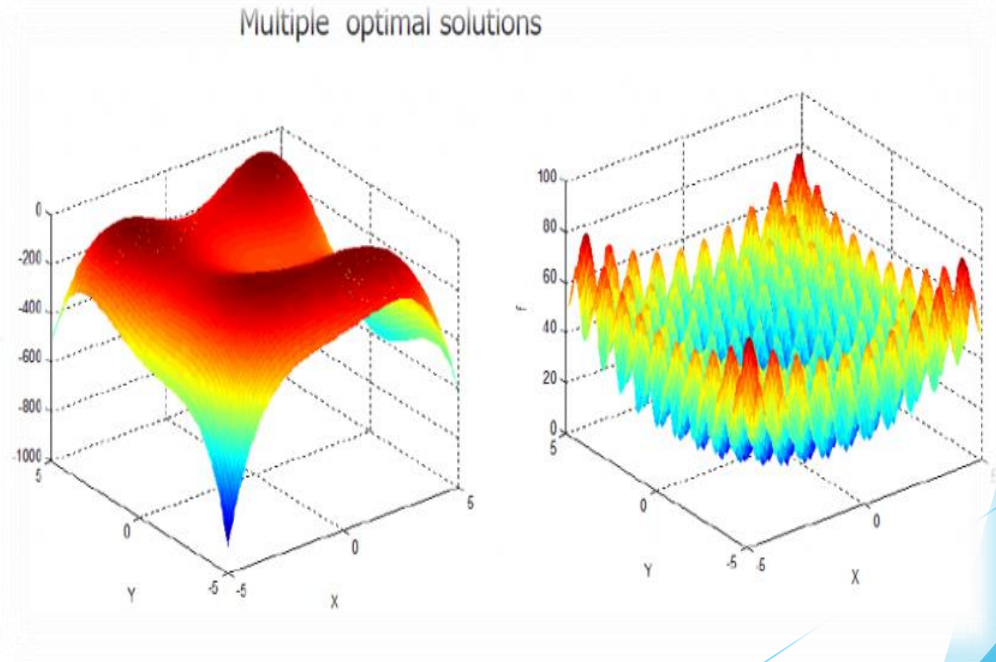
استاد مربوطه: جناب آقای دکتر ستوده بحرینی

دانشگاه صنعتی سیرجان-۱۴۰۰

هدف در بهینه سازی

در بهینه سازی ها هدف یافتن نقاط ماکزیمم و مینیمم کلی است.





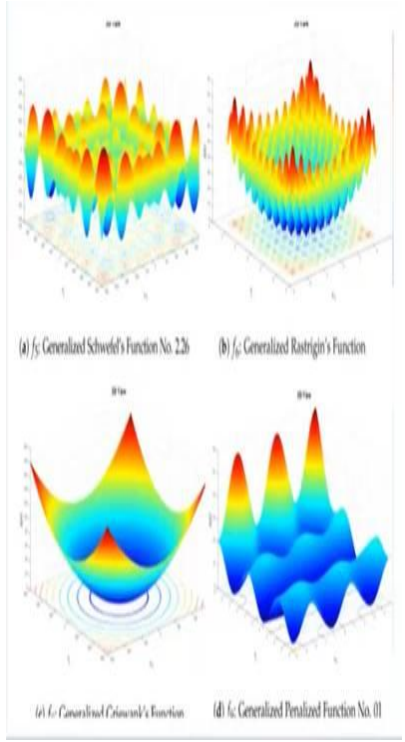
انواع روش های بهینه سازی

روش های بهینه سازی ریاضیاتی مثل روش های خطی و روش لاگرانژ

روش جستجوی سراسری (جستجوی تمام فضای سرچ)

روش های فراابتکاری مثل الگوریتم ژنتیک و PSO و ....

چرا الگوریتم های فرا ابتکاری؟



عوامل تاثیرگذار در پیچیدگی محاسباتی مدل های تصمیم گیری:

۱- غیرخطی بودن توابع هدف یا محدودیت ها

۲- گسسته بودن فضای حل مسئله

۳- اندازه مسئله

روش های دقیق:

در شرایط فوق قادر به پیدا کردن جواب بهینه در زمان قابل قبولی نیستند.

هدف روش های فراابتکاری:

پیدا کردن بهترین جواب ممکن نزدیک به بهینه (در زمان قابل قبول)

الگوریتم های فراابتکاری

در سی سال گذشته، نوع جدیدی از الگوریتم های تقریب ظهور یافته اند که اساساً هدف از آنها ترکیب روش های ابتکاری در چارچوبهای کلان تر به منظور کاوش کارا و اثربخش فضای جستجو می باشد.

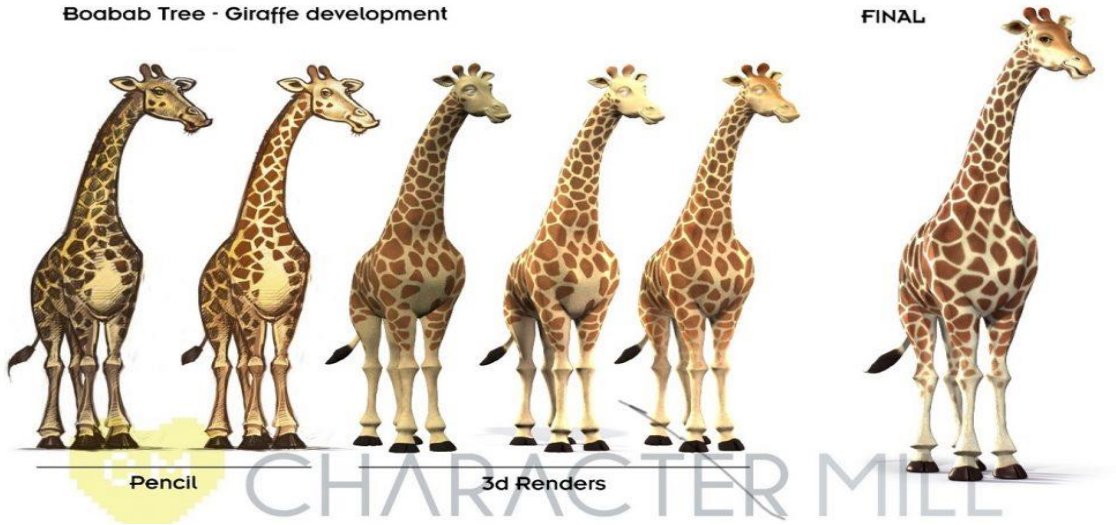
معرفی الگوریتم ژنتیک

الگوریتم ژنتیک، الهامی از علم ژنتیک و نظریهٔ تکامل داروین است و بر اساس بقای برترینها یا انتخاب طبیعی استوار است. یک کاربرد متداول الگوریتم ژنتیک، استفاده از آن بعنوان تابع بهینه کننده است.

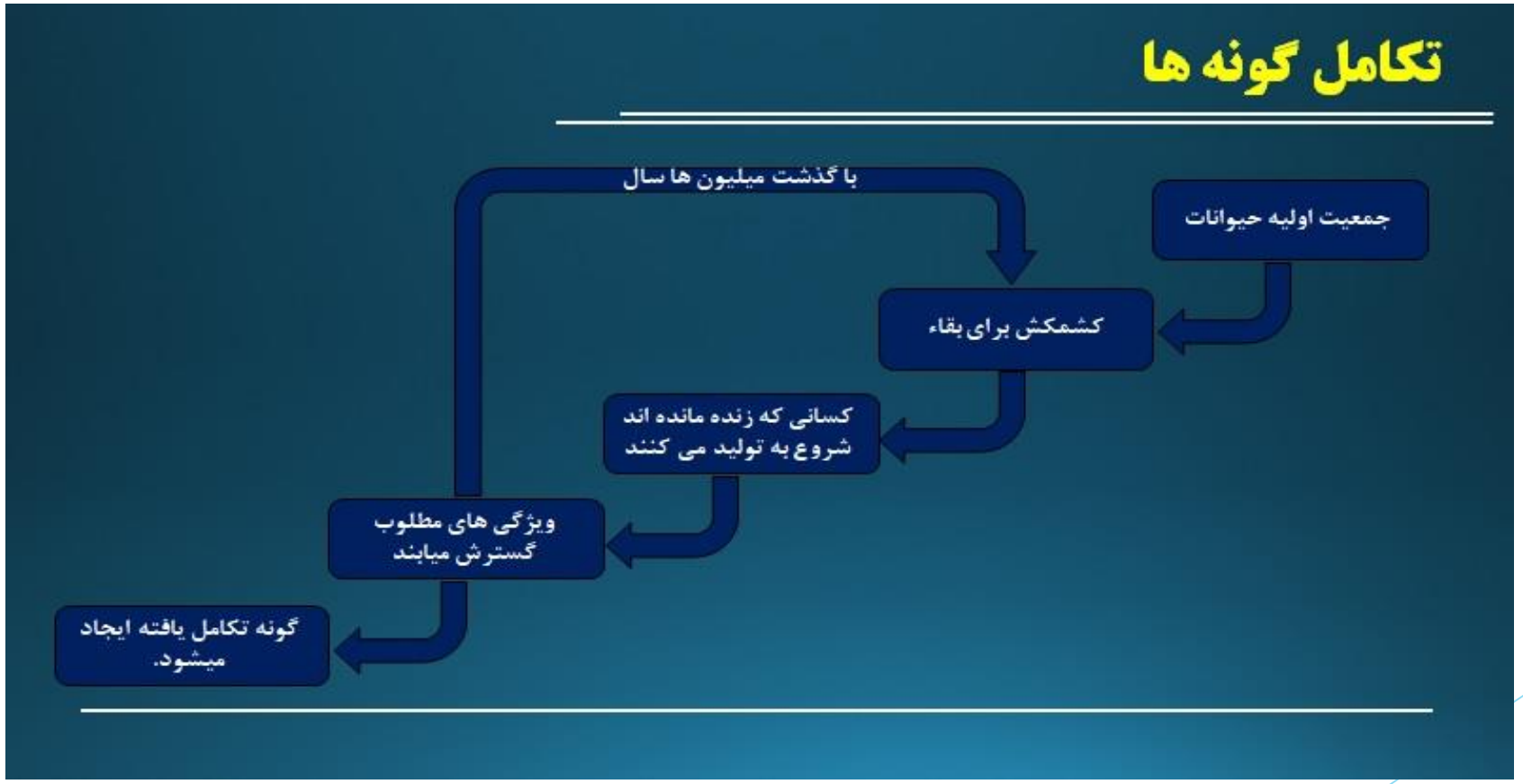
مثال: تکامل زرافه ها

زرافه ها گردن بلندی دارند:

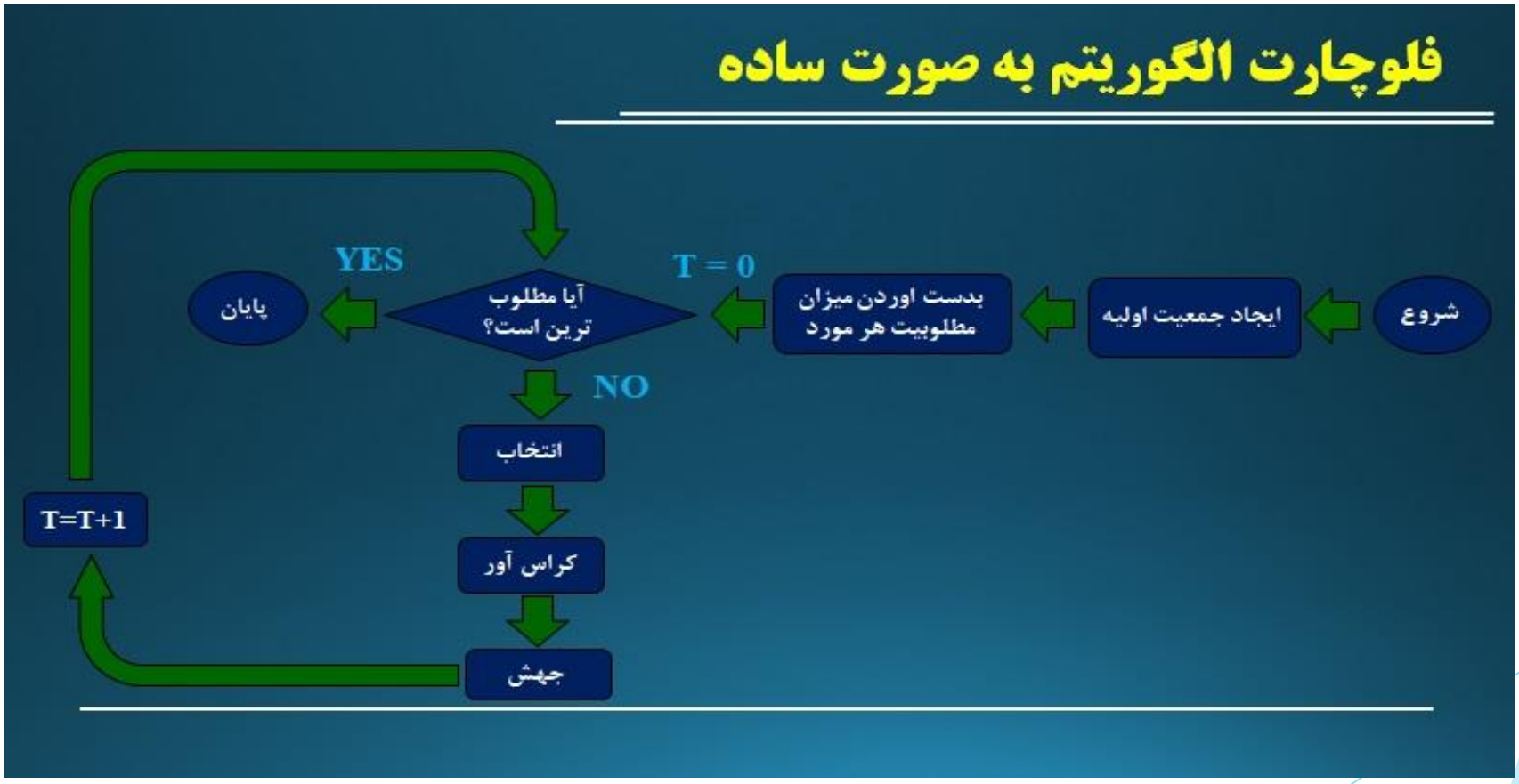
در گذشته در بین زرافه ها آن هایی که کمی گردنشان بلندتر بوده توانسته اند از شاخه های بلندتر گیاهان تغزیه کنند، درحالی که آن هایی که گردن کوتاه تری داشته اند غذای کمتری خورده اند.



فلوچارت تکامل گونه ها در الگوریتم ژنتیک

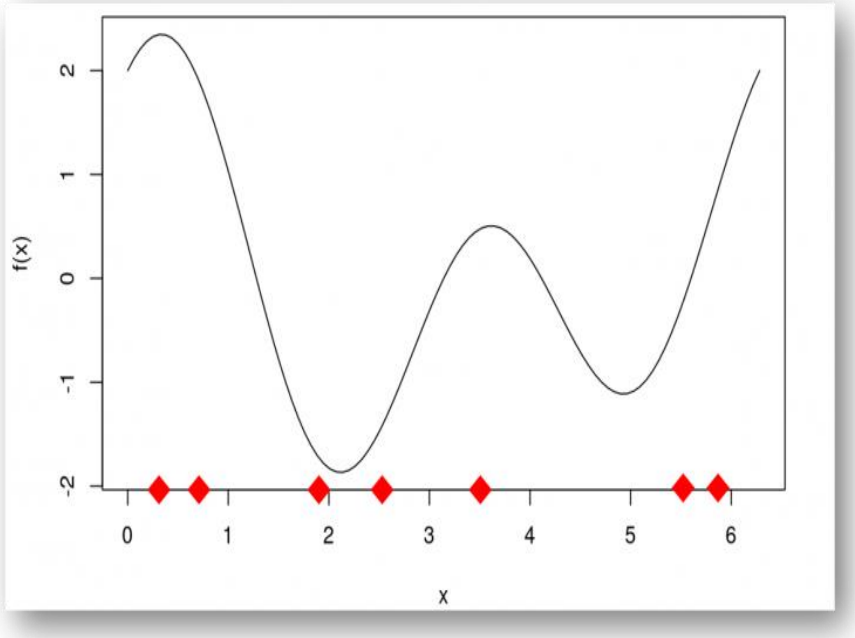


فلوچارت کلی الگوریتیم بهینه سازی ژنتیک



ایجاد جمعیت اولیه

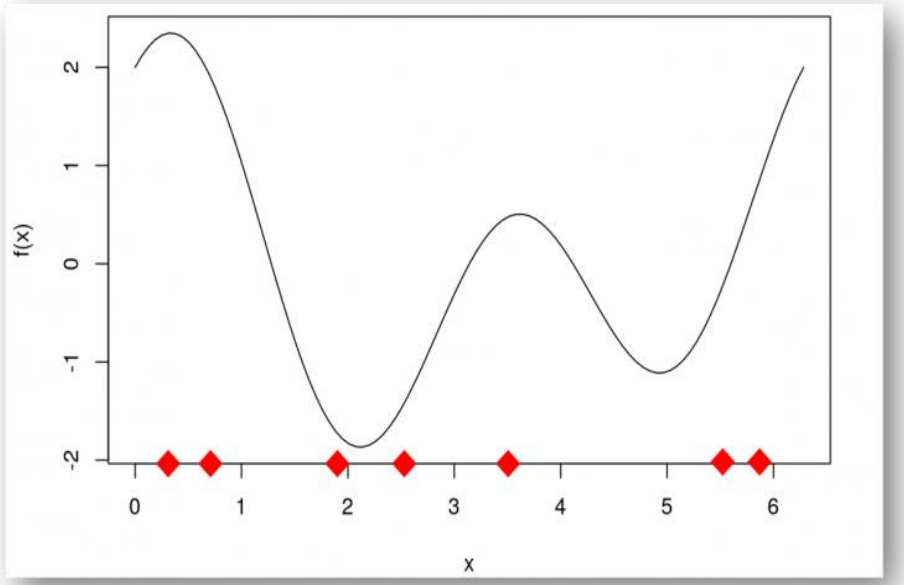
در ابتدا به صورت رندوم در کل فضای سرچ یک سری جمعیت اولیه ایجاد می شود.

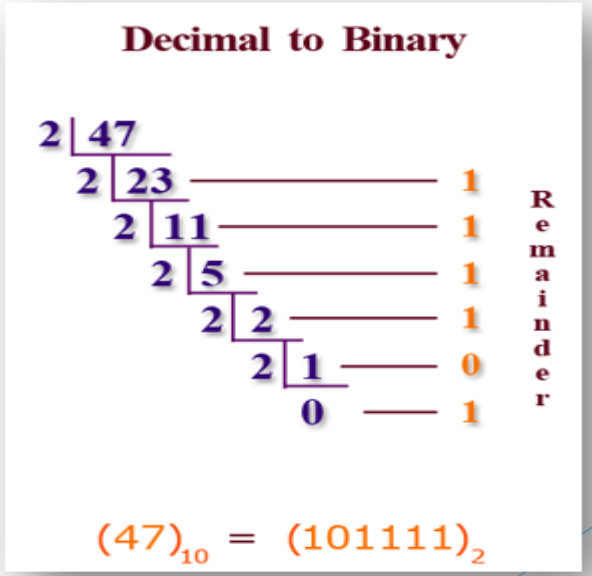


ایجاد جمعیت اولیه

تبدیل اعداد

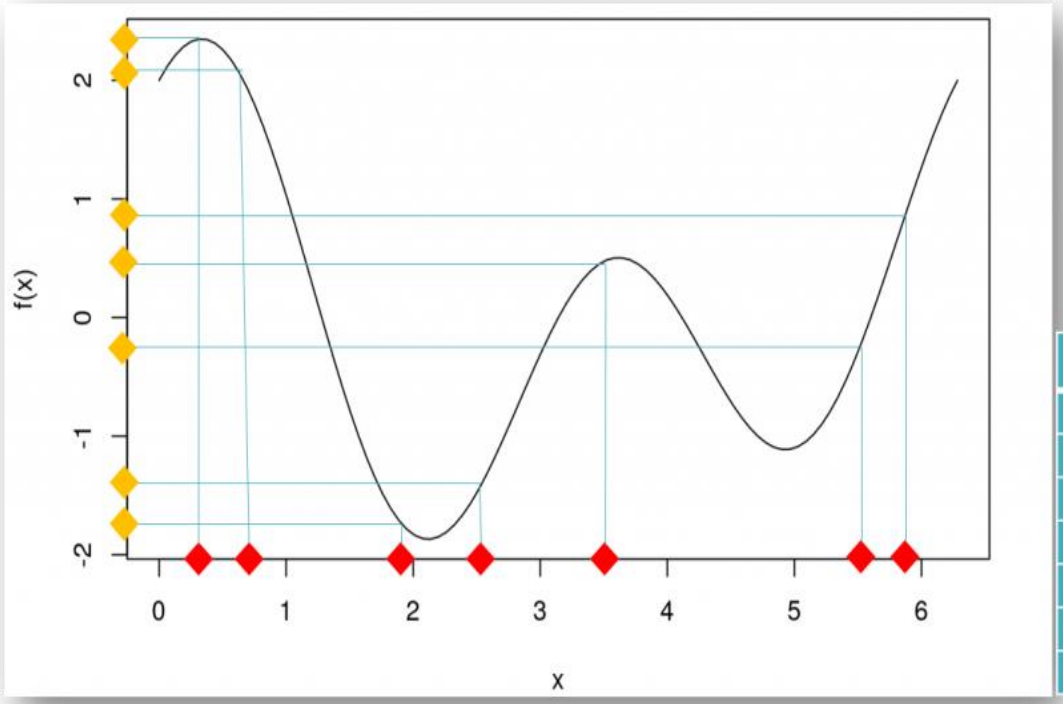
در این قسمت اعداد که به صورت دهدهی هستند تبدیل به اعداد در مبنای ۲ میشوند.





بدست آوردن ارزش هر تابع

با استفاده از تابع موجود یابد مقادیر ارزش هر تابع بدست آورده شود.



x	F(x)
0.4	2.2
0.7	2.05
1.9	-1.7
2.5	-1.3
3.5	0.5
5.6	-0.2
5.9	0.9

نحوه ی انتخاب بهترین ها برای نسل بعدی

انواع انتخاب ها در الگوریتم ژنتیک :

۱- انتخاب قطعی Deterministic Selection

۲- انتخاب رنک Rank Selection

۳- انتخاب یکنواخت Uniform Selection

۴- انتخاب ترنومنت Selection Tournament

۵- انتخاب Roulette wheel Selection

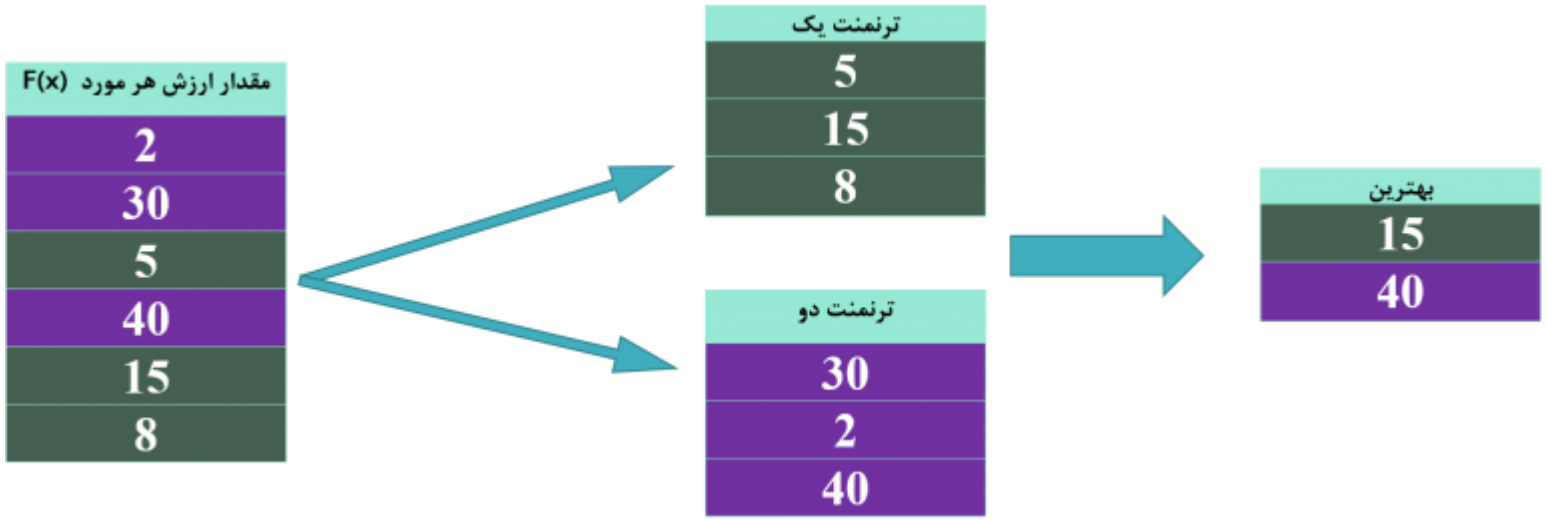
مقدار ارزش هر مورد F(x)
8
30
2
40
15
5

Deterministic Selection انتخاب قطعی

در این انتخاب ابتدا مقادیر تابع بر اساس بهترین به بدترین مرتب و بهترین ها برای نسل بعد انتخاب میشوند.

انتخاب ترنومنت

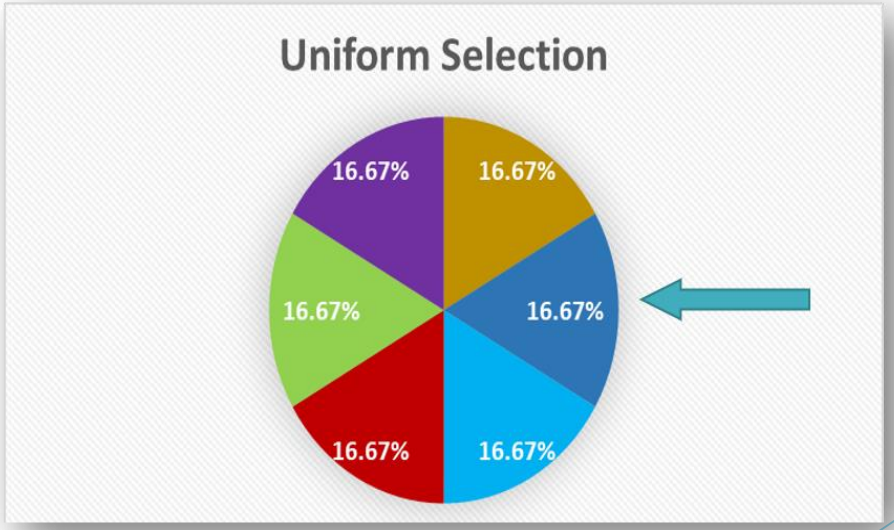
در این انتخاب ابتدا داده ها به صورت رندوم به گروه های مختلفی تقسیم میشوند، بعد در هر گروه بهترین انتخاب میشود.



انتخاب یکنواخت

در این نوع انتخاب شانس انتخاب تمامی داده ها برای نسل بعدی به یک نسبت است. این نوع انتخاب سرعت همگرایی را بسیار کم میکند ولی شانس جستجوی تمام فضا را بالا میبرد. معمولا از این نوع انتخاب در مواردی که تابع دارای تعداد لوکال خیلی زیادی است و ممکن است داده ها در لوکال ها گیر کند استفاده میشود.

مقدار ارزش هر مورد F(x)	درصد شانس هر داده
8	16.67%
30	16.67%
2	16.67%
40	16.67%
15	16.67%
5	16.67%
مجموع امتیاز ها	100%



انتخاب رنک

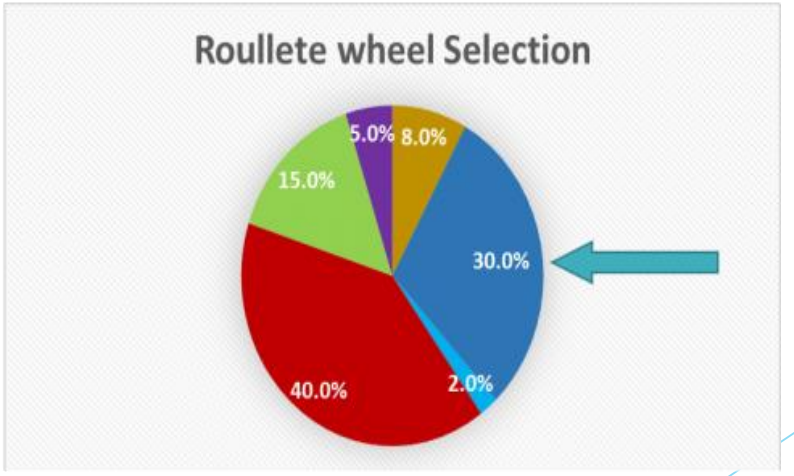
در این روش ابتدا مقادیر از بدترین به بهترین امتیاز ۱تا N را میگیرند. یعنی آنکه بهترین است بالاترین امتیاز و بعد آنکه کمترین است امتیاز ۱ را میگیرد و در هنگام انتخاب شانس هر داده که امتیاز بالاتری دارد بیشتر است.

مقدار ارزش هر مورد F(x)	امتیاز	شانس هر داده	درصد شانس هر داده
8	3	3/21	14.3%
30	5	5/21	23.8%
2	1	1/21	4.8%
40	6	6/21	28.6%
15	4	4/21	19%
5	2	2/21	9.5%
مجموع امتیاز ها	21	1	100%

انتخاب چرخ رولت

در این روش ابتدا مقادیر از بدترین به بهترین امتیاز از ۱تا N را میگیرند. یعنی آنکه بهترین است بالاترین امتیاز و بعد آنکه کمترین است امتیاز ۱ را میگیرد و در هنگام انتخاب شانس هر داده که امتیاز بالاتری دارد بیشتر است.

مقدار ارزش هر مورد F(x)	درصد هر تابع نسبت به کل
8	8%
30	30%
2	2%
40	40%
15	15%
5	5%
مجموع امتیاز ها	100%

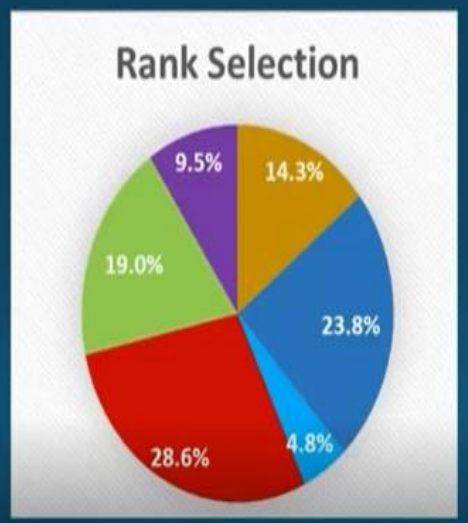


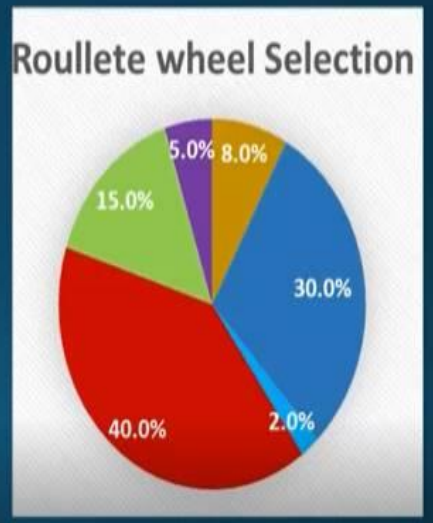
چرخ می چرخد و بهترین انتخاب می شود.

تفاوت انتخاب رنک و چرخ رولت

در روش چرخ رولت شانس انتخاب موارد بهتر بیشتر از رنک است چرا که بر اساس مقدار تابع درصد بندی میشود نه رنک تابع.

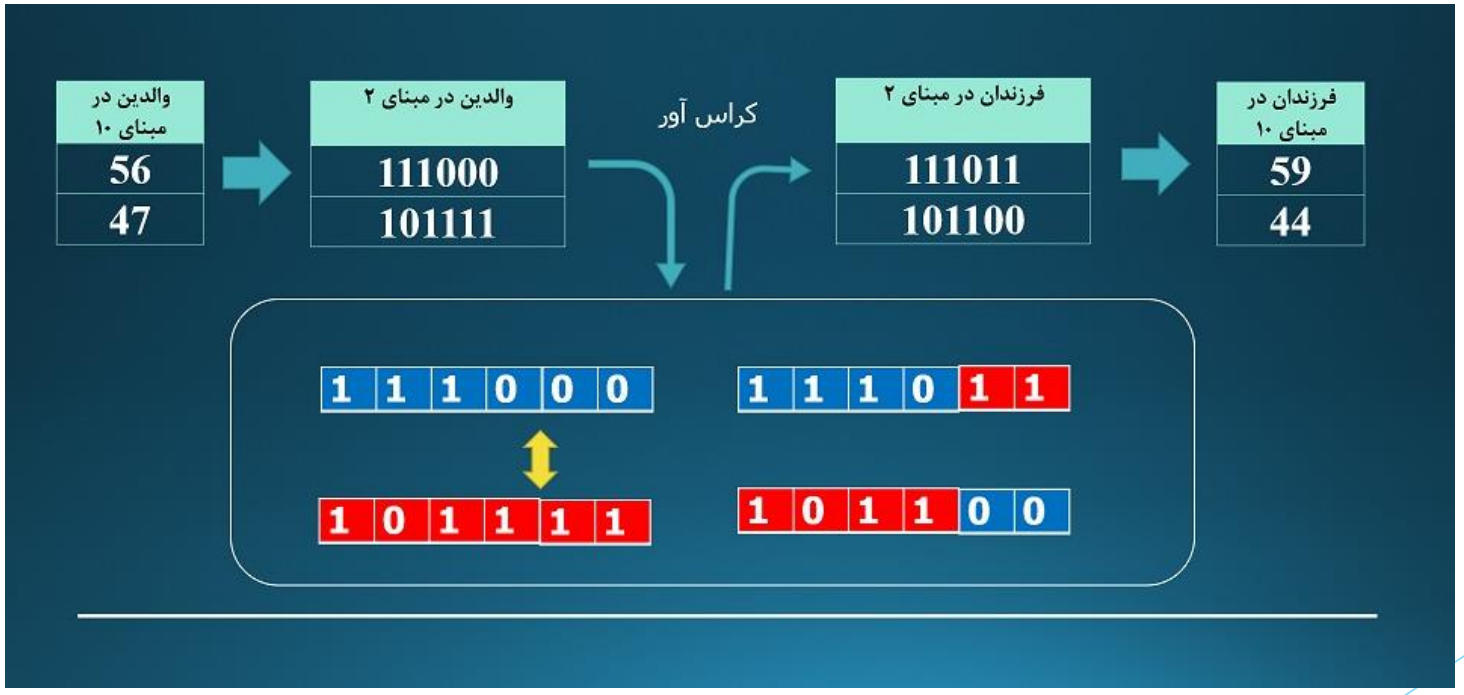
مقدار ارزش هر مورد F(x)	چرخ رولت	انتخاب رنک
8	8%	14.3%
30	30%	23.8%
2	2%	4.8%
40	40%	28.6%
15	15%	19%
5	5%	9.5%
مجموع امتیاز ها	100%	100%





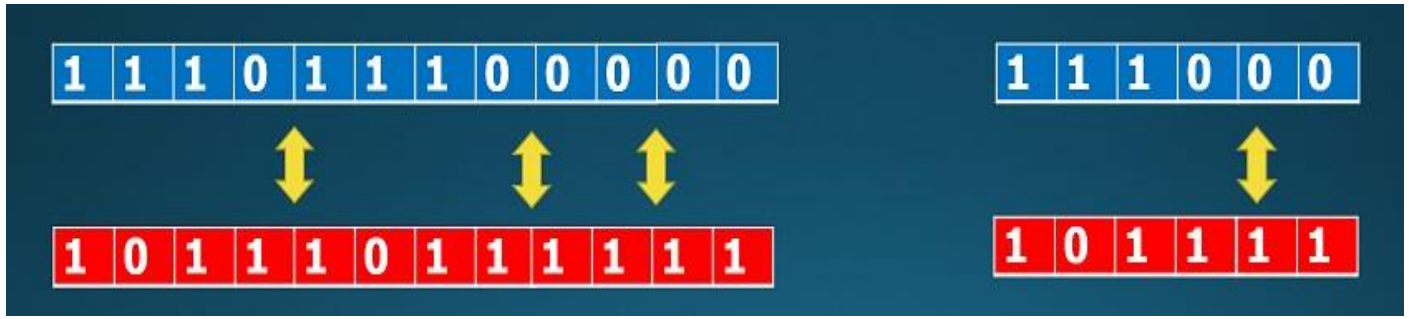
عملیات کراس آور

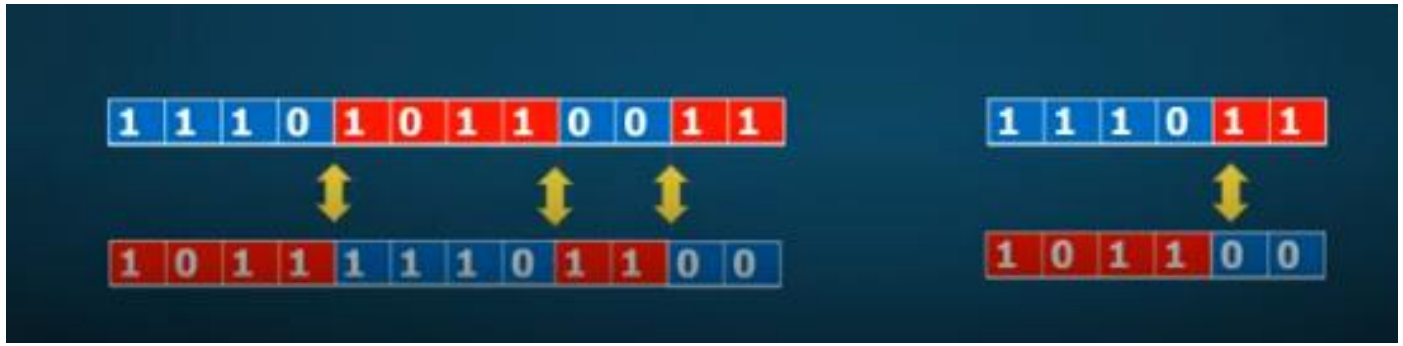
اعداد در مبنای ده بعد از تبدیل به مبنای دو به صورت زیر به صورت تک نقطه ای عملیات کراس آور روی آن ها انجام می شود. سپس خروجی مجددا به منبای ده بر میگردد تا تابع هدف آن محاسبه شود.



انواع کراس آور

در کراس اور یک نقطه ای یک شماره ی رندوم انتخاب و از همانجا کروموزوم بریده شده و با هم تعویض میشود.در کراس اور چند نقطه ای ، چند نقطه ی رندوم ایجاد و از محل شماره های رندوم کروموزوم ها بریده و با هم تعویض می شود.



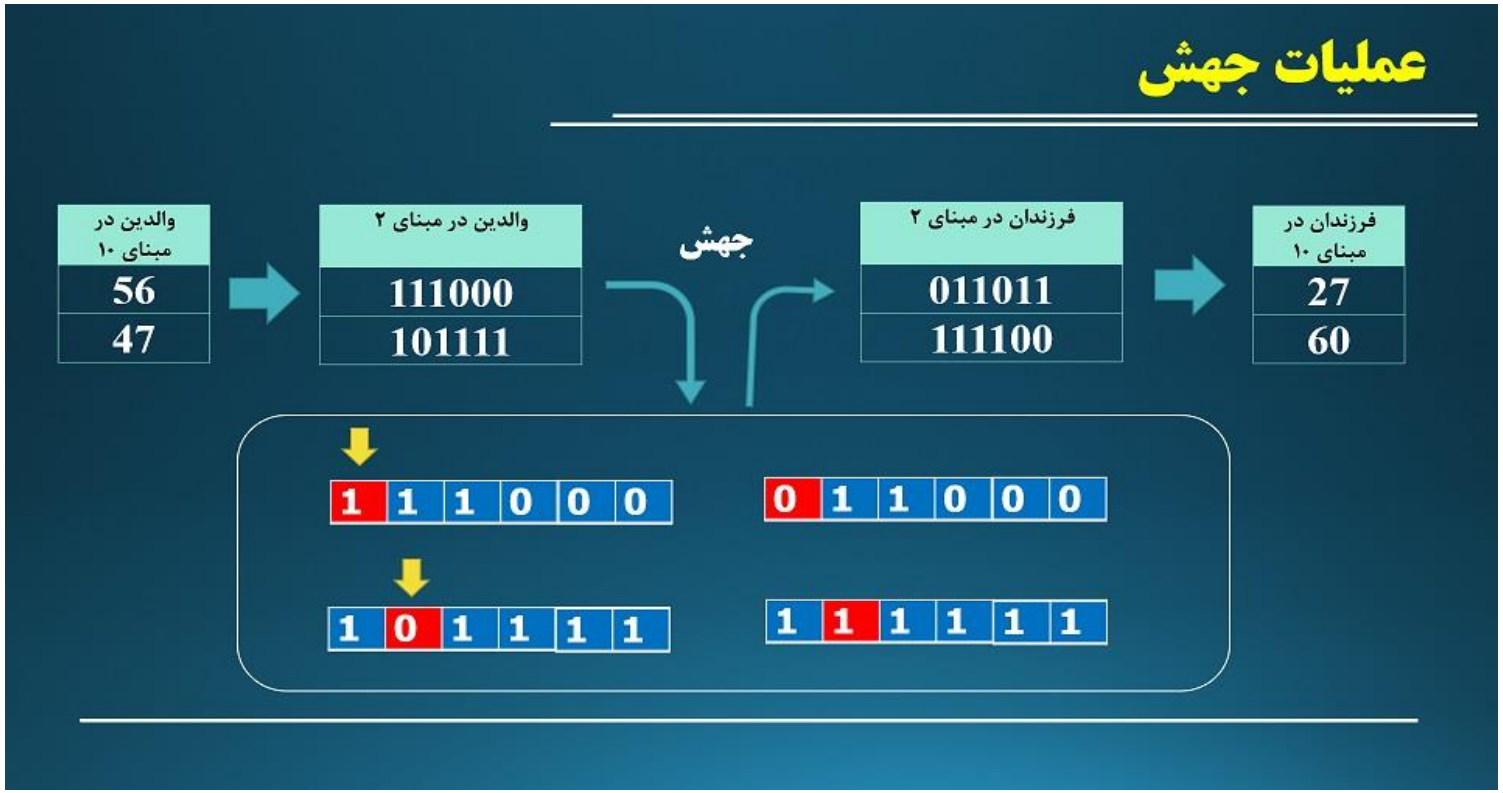


چرا کراس آور؟

در حقیقت کراس آور حکم لوکال سرچ را دارد چون ماهیت کروموزوم ها را عوض نمی کند بلکه کروموزوم ها را ترکیب می کند بنابراین در اکثر مواقع وقتی کراس اور بین دو والد اتفاق می افتد فرزندانی که ایجاد می شوند شبیه دو والد هستند شاید کمی بهتر باشند. در مسائل بهینه سازی کراس آور به معنی جستجوی اطراف نقاط رندوم قبلی است.

عملیات جهش

در عملیات جهش کروموزوم مورد نظر که رشته ای از صفر و یک هست، یکی از ژن ها به طور رندوم اگر یک است به صفر و اگر صفر است به یک تبدیل می شود.

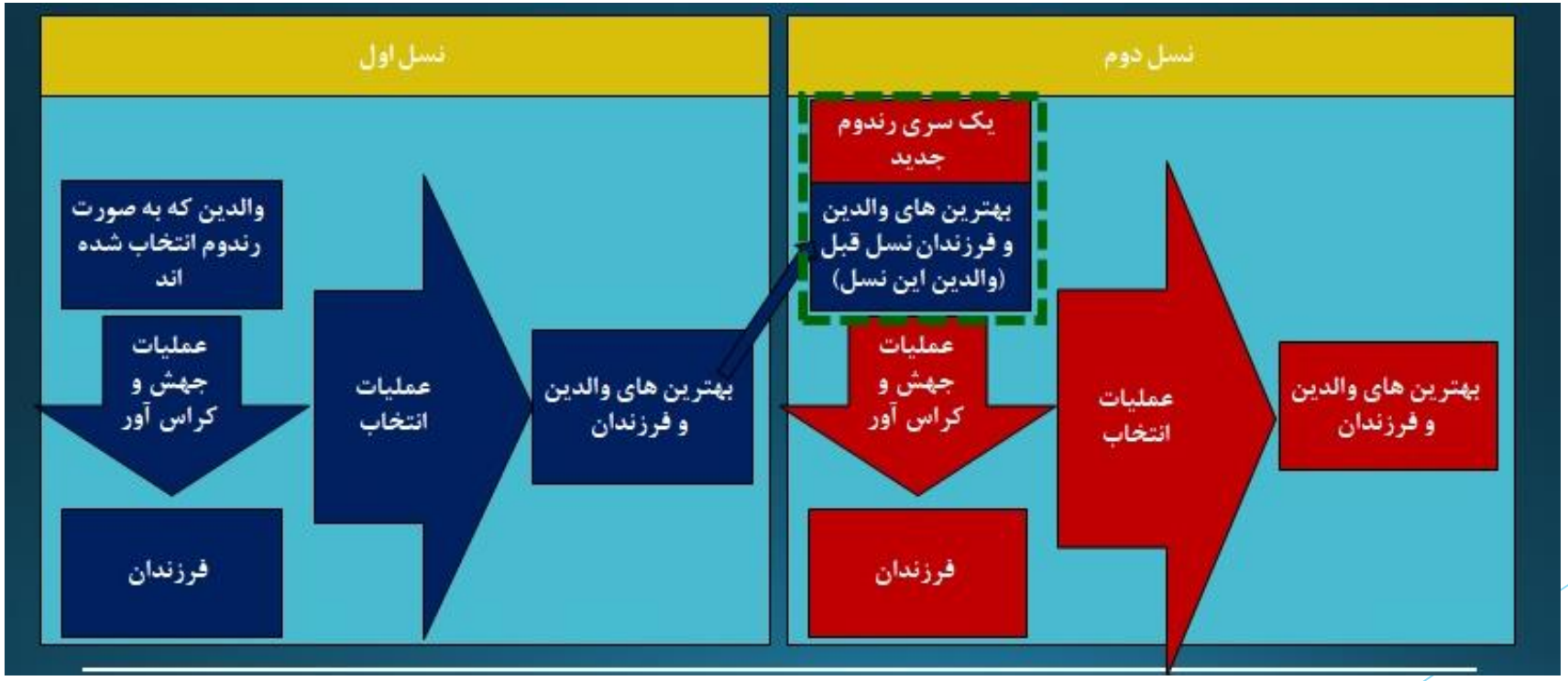


چرا جهش؟

در جهش به صورت رندوم یکی از سلول های کروموزم اگر یک باشد به صفر و اگر صفر باشد به یک تبدیل می شود. عملیات جهش اگر در سلول های بالایی کروموزوم اتفاق بیفتد ماهیت عدد را عوض کرده و حکم گلوبال سرچ را دارد. در بهینه سازی معمولا بعد از جهش روی کروموزوم ها سرچ از یک ناحیه به یک ناحیه ی دیگر جابجا می شود. در اصل عملیات جهش باعث می شود که فضای سرچ بیشتری جستجو شود و داده ها در فضای کلی بیشتر جستجو کنند.

نحوه ی انتخاب والدین نسل بعدی

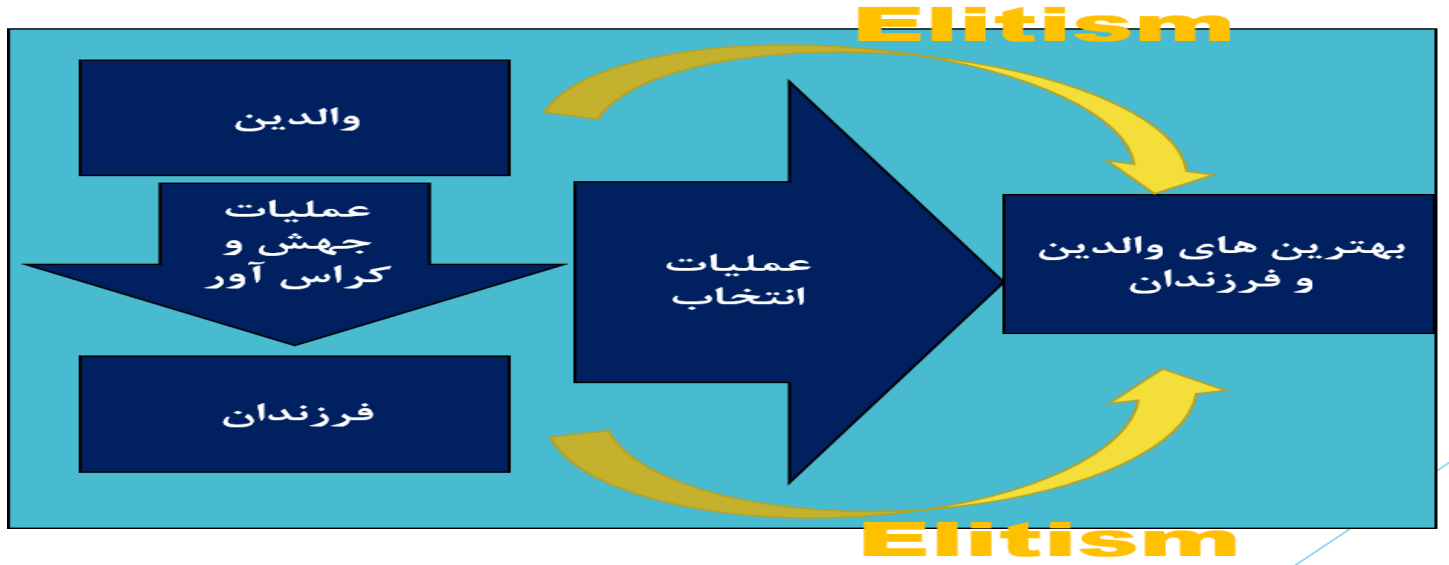
والدین نسل بعدی از انتخاب بهترین های والدین و فرزندان در نسل قبلی بدست می آیند.
این فرایند در شکل به طور کامل نشان داده شده است.



الیتیزم (نخبه گرایی)

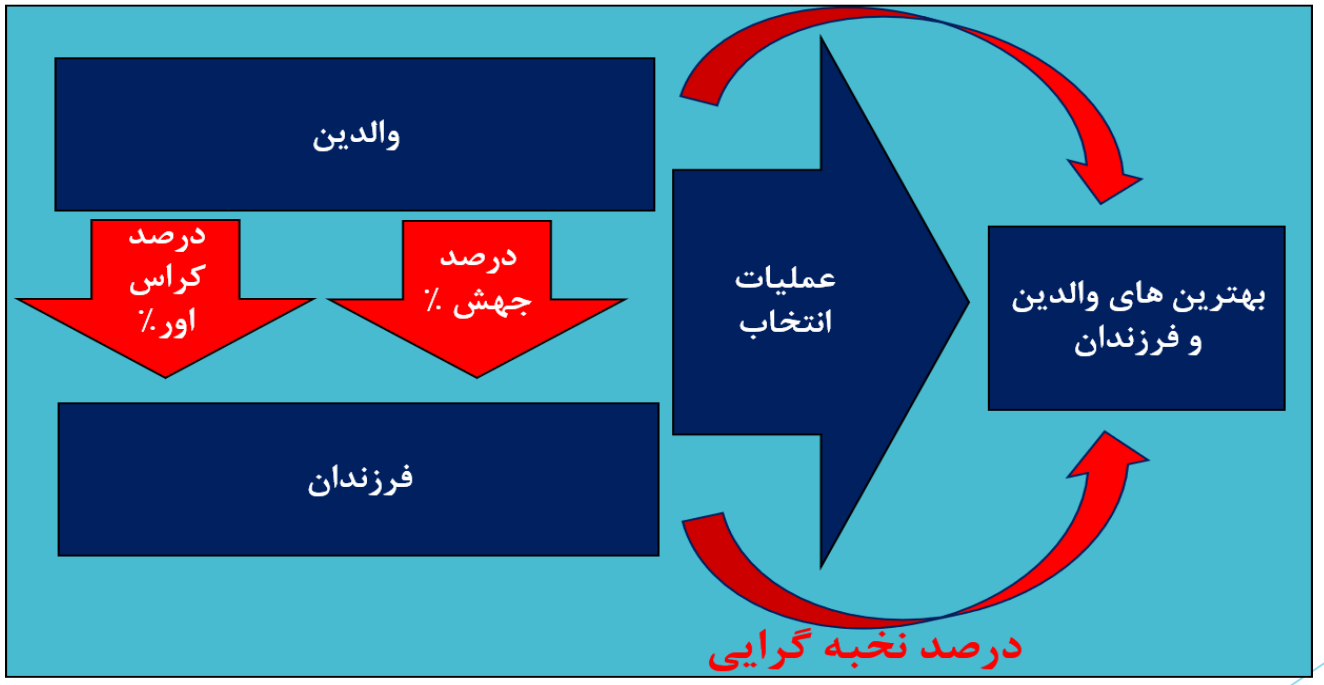
فرستادن بهترین ها به نسل بعد بدون شرکت در مرحله انتخاب

در برخی از موارد در الگوریتم ژنتیک برای اینکه سرعت همگرایی بالا برود، بهترین های هر نسل بدون اینکه در عملیات انتخاب شرکت کنند مستقیم به نسل بعدی راه پیدا می کنند. این کار باعث می شود که سرعت همگرایی الگوریتم افزایش یابد. اما عیبی که دارد این است که گاهی نخبه ها بهترین نیستند و حکم مقادیر لوکال خوب را دارند نه گلوبال خوب و الگوریتم توی لوکال ها ترپ می شود.



تعیین پارامترهای تغییر نسل

در الگوریتم ژنتیک باید پارامترهای مربوط به کراس آور و جهش و نخبه گرایی تعیین شود. مثلا از کل والدین چند درصد جهش ایجاد شود. چند درصد کراس آور ایجاد شود و چند درصد بدون طی مراحل انتخاب (نخبه گرایی) به نسل بعد باید راه پیدا کنند.



مسئله ی فروشنده دوره گرد

یک فروشنده میخواهد از ۸ شهر مختلف عبور کند بدون این که از یک شهر دوبار عبور کند والبته کمترین مسیر ممکن را طی کند.از این ۸ شهر به چه شکل عبور کند که شرایط بالا فراهم شود.

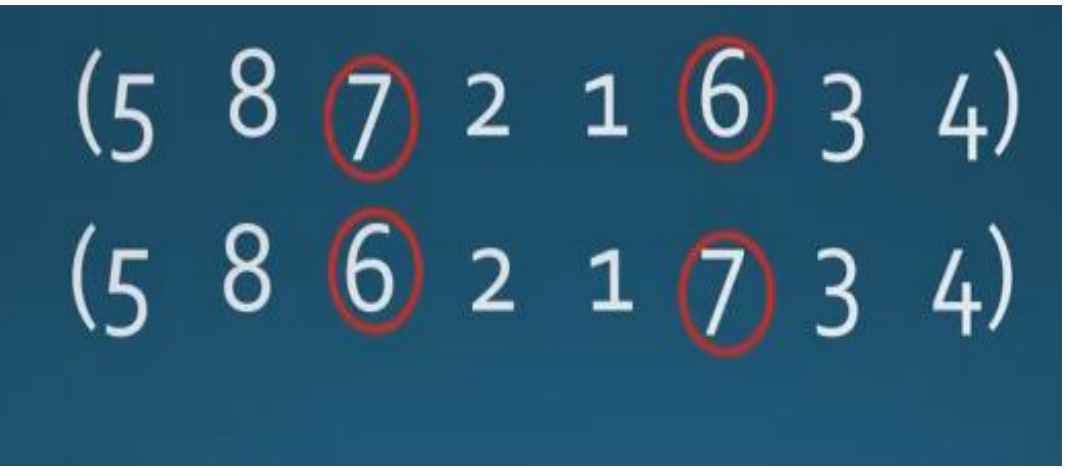
برای سادگی و نمایش ۸ شهر زیر را در نظر گرفته ایم

1) London 3) Dunedin 5) Beijing 7) Tokyo
2) Venice 4) Singapore 6) Phoenix 8) Victoria

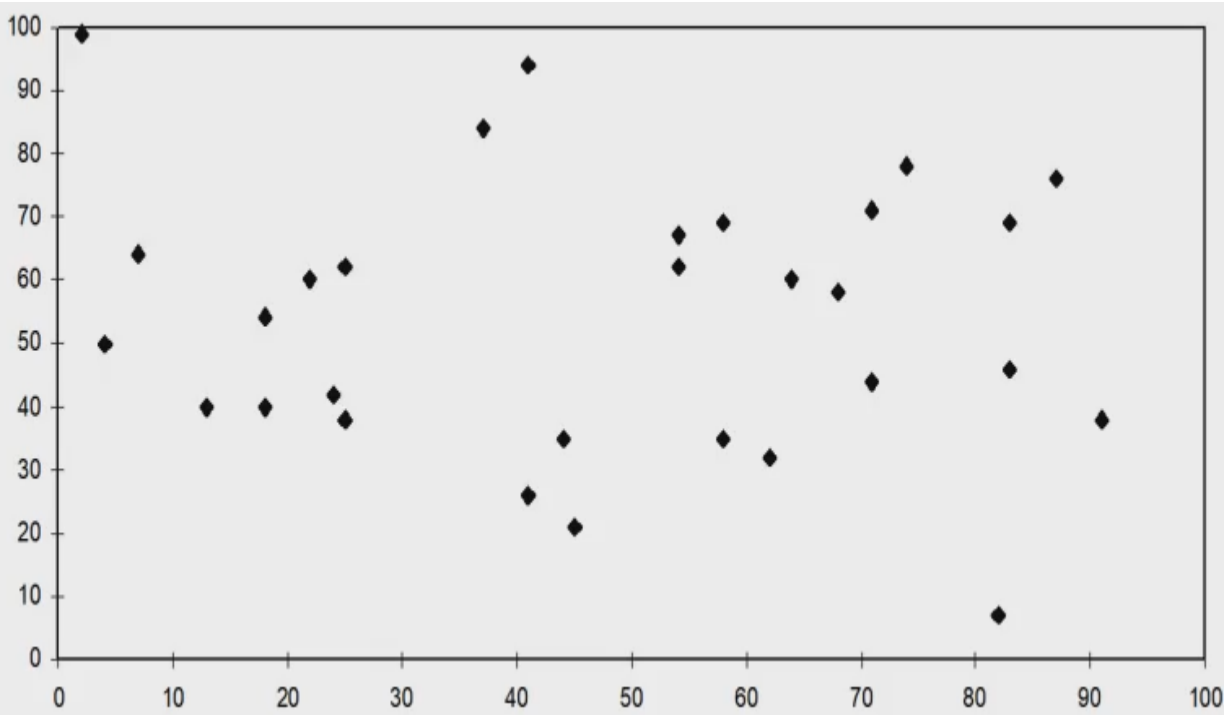
شهر ها را با ترتیب متفاوت می توان کنارهم قرار داد که دو ترتیبی که به صورت اتفاقی به عنوان والد انتخاب شده اند به این صورت است.

CityList1 (3 5 7 2 1 6 4 8)
CityList2 (2 5 7 6 8 1 3 4)

جهش :

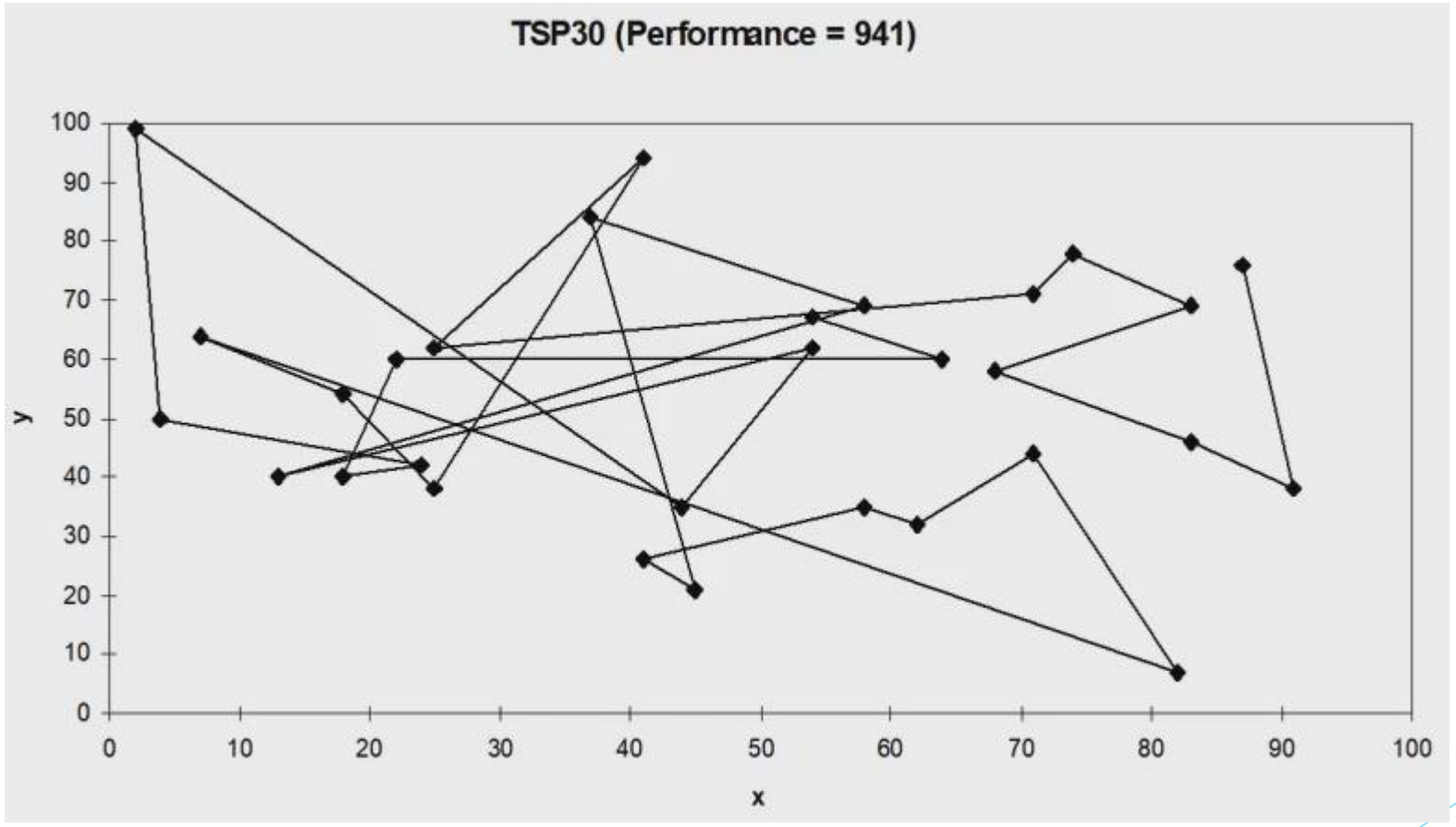


به این صورت است که باید در هر جایی که جهش اتفاق می افتد جای شهر ها عوض بشود.

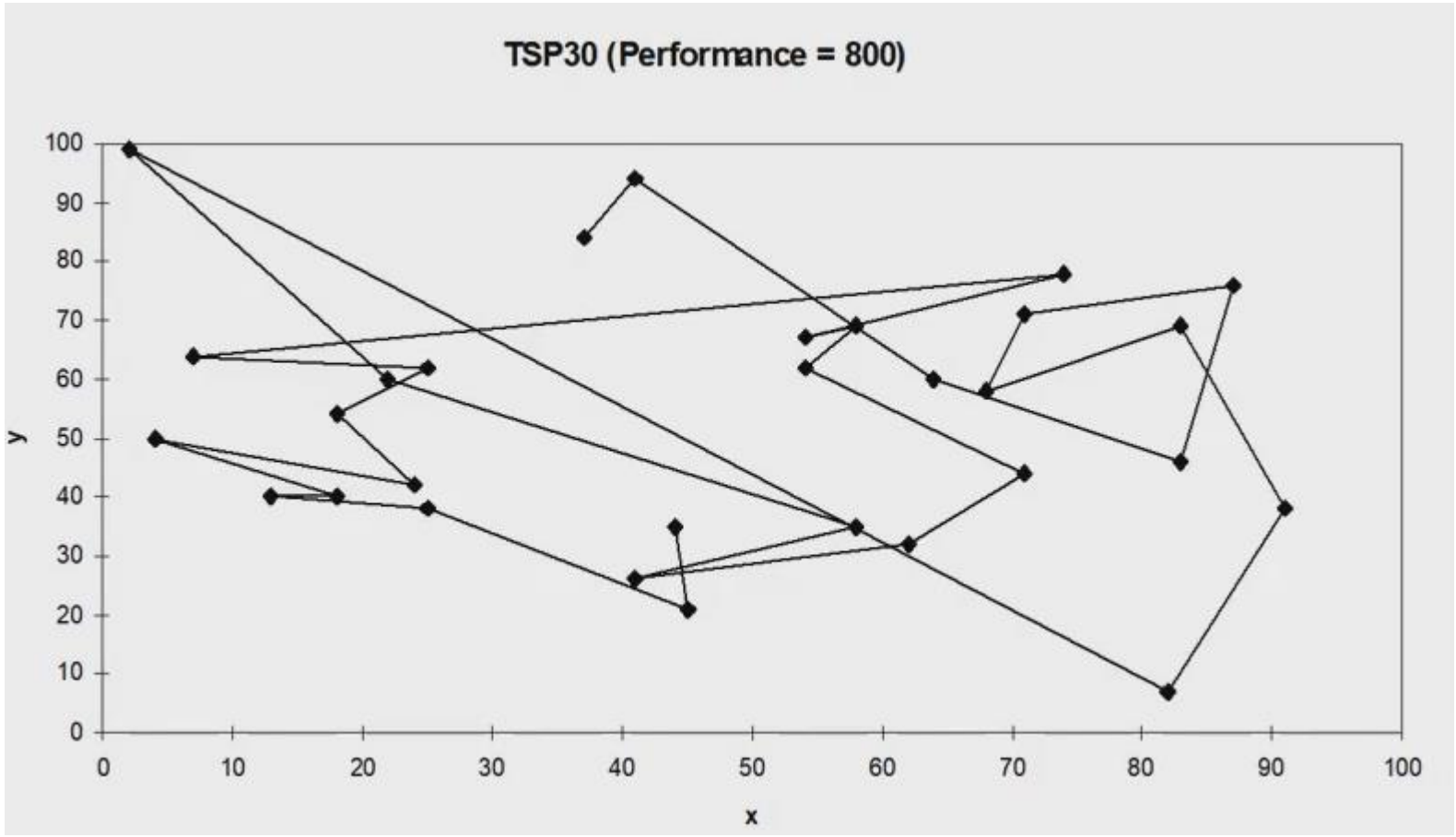


موقعیت ۳۰ شهر مختلف را می بینید.

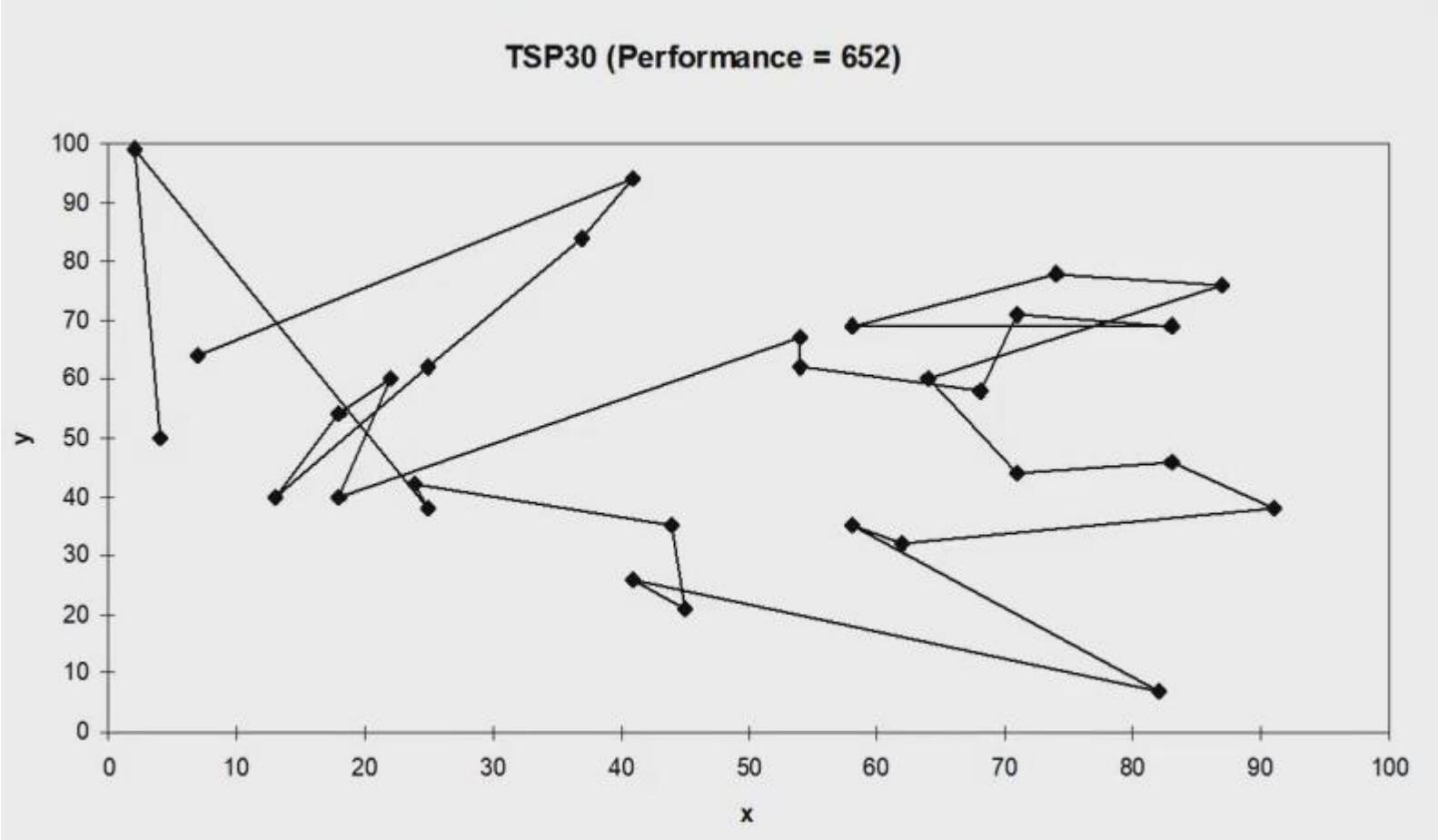
در این نسل فاصله بین شهر ها ۹۴۱ است.



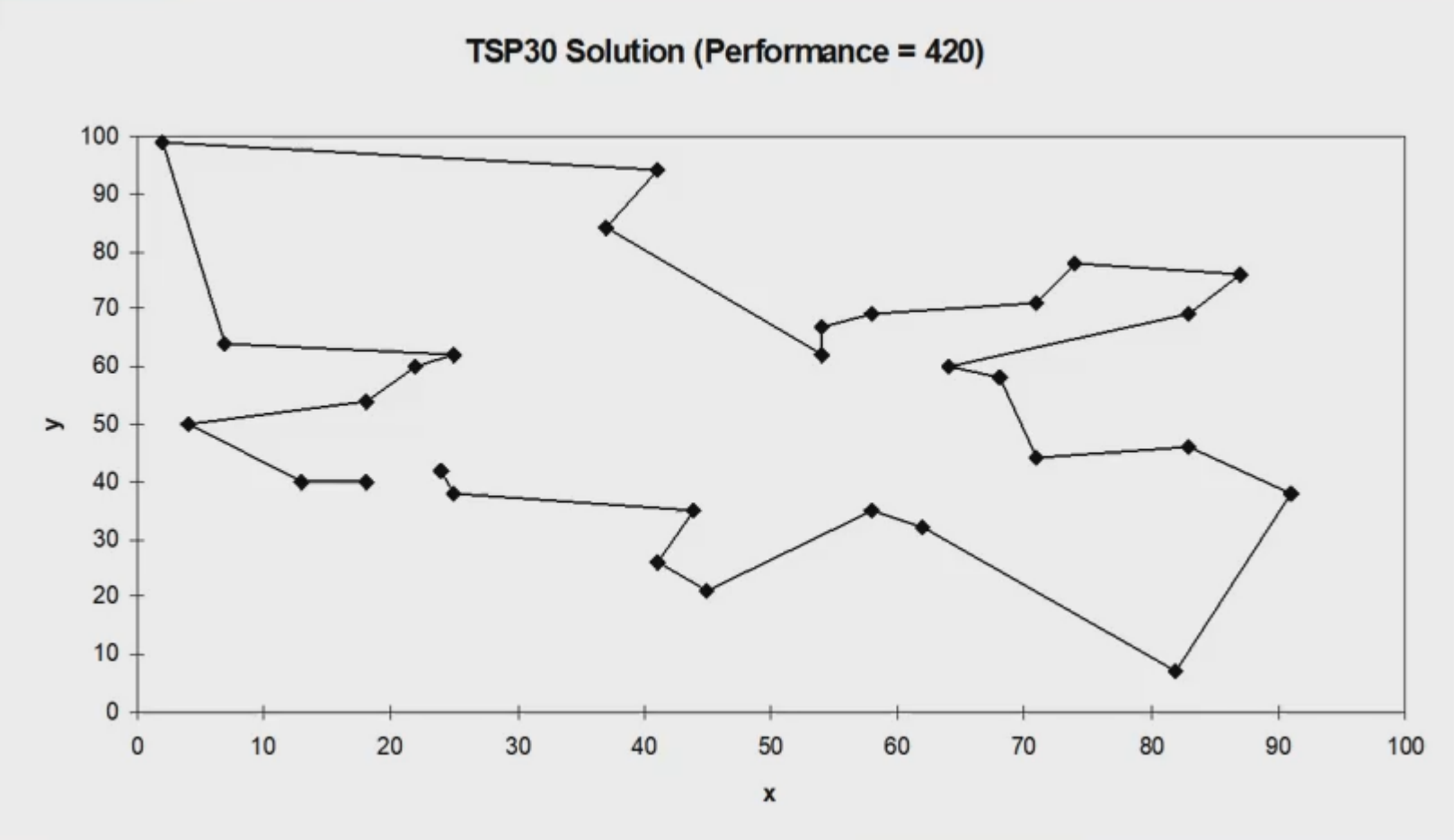
در این نسل فاصله بین شهر ها ۸۰۰ است.



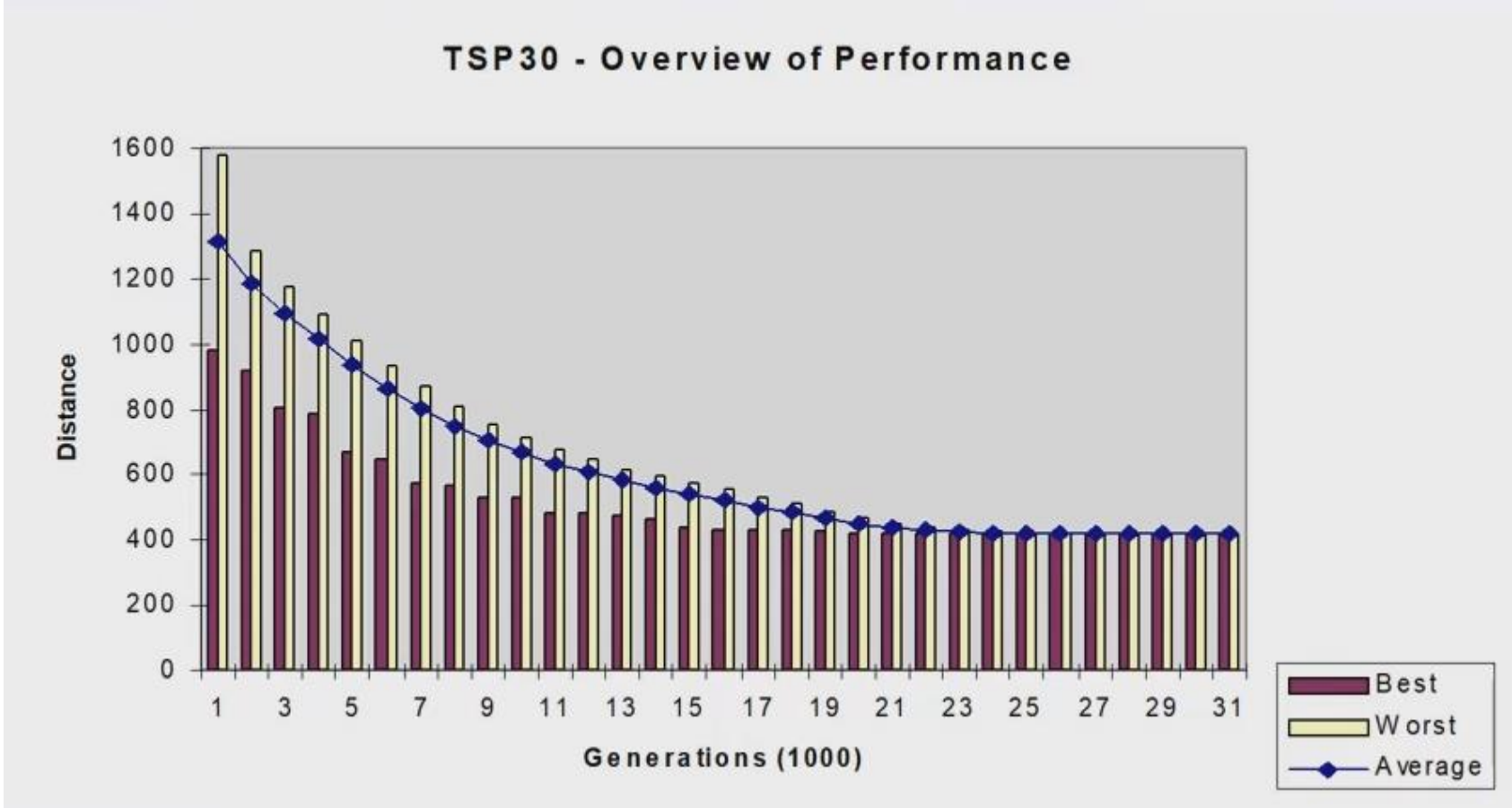
در این نسل فاصله بین شهر ها ۶۵۲ است.

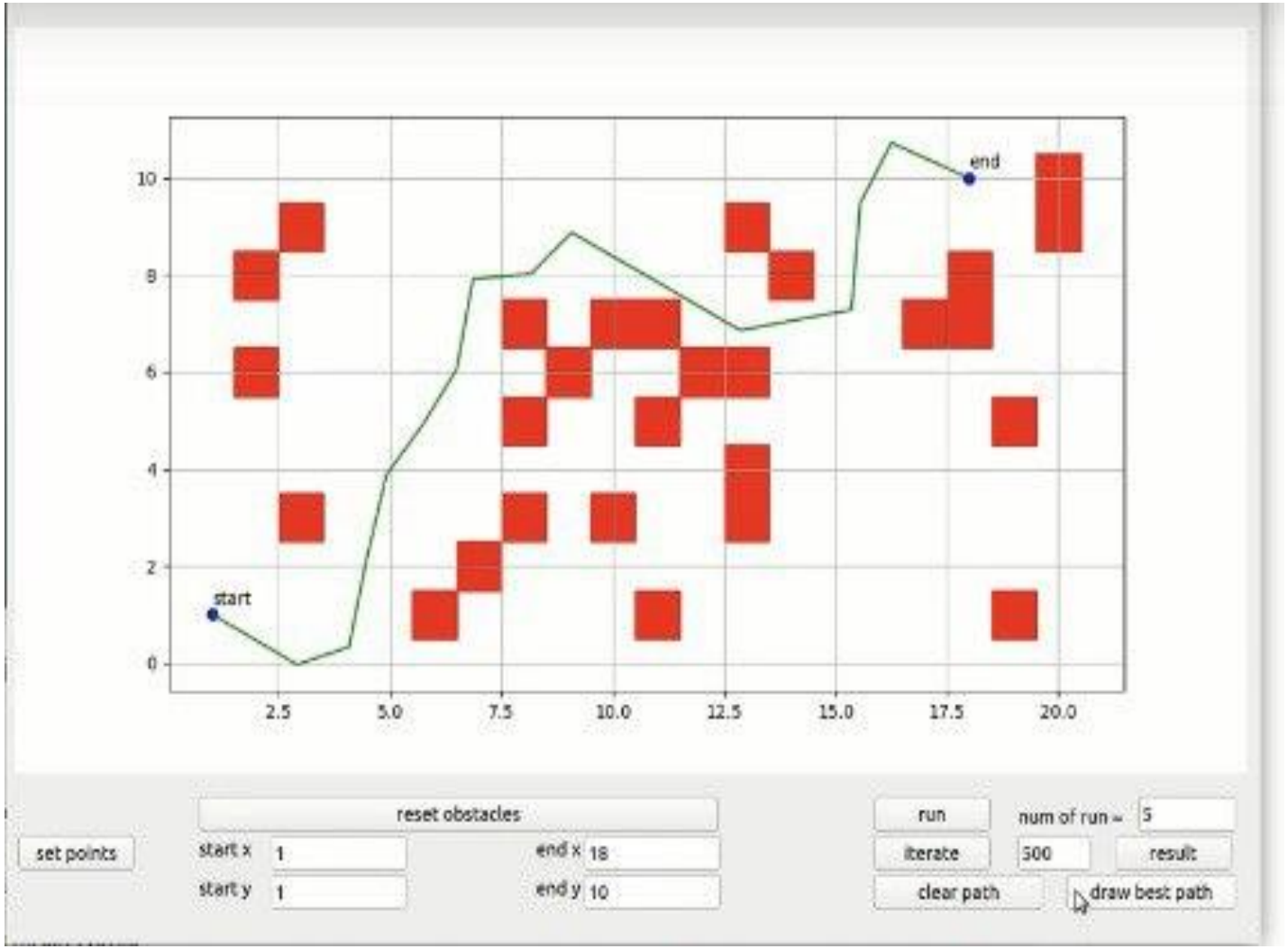


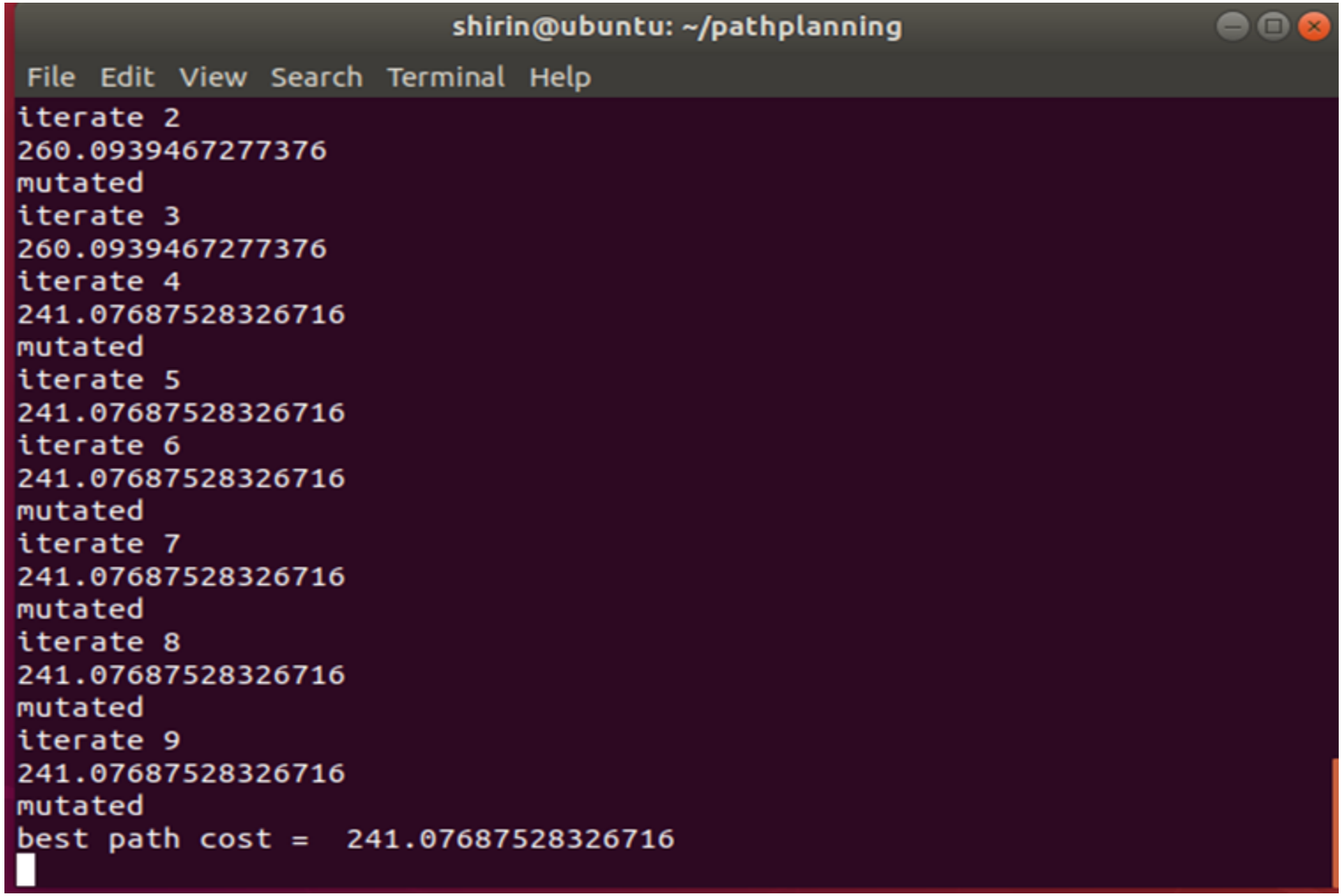
آخرین نسل : در این نسل فاصله بین شهر ها ۴۲۰ است که کمترین مقدار ممکن است.

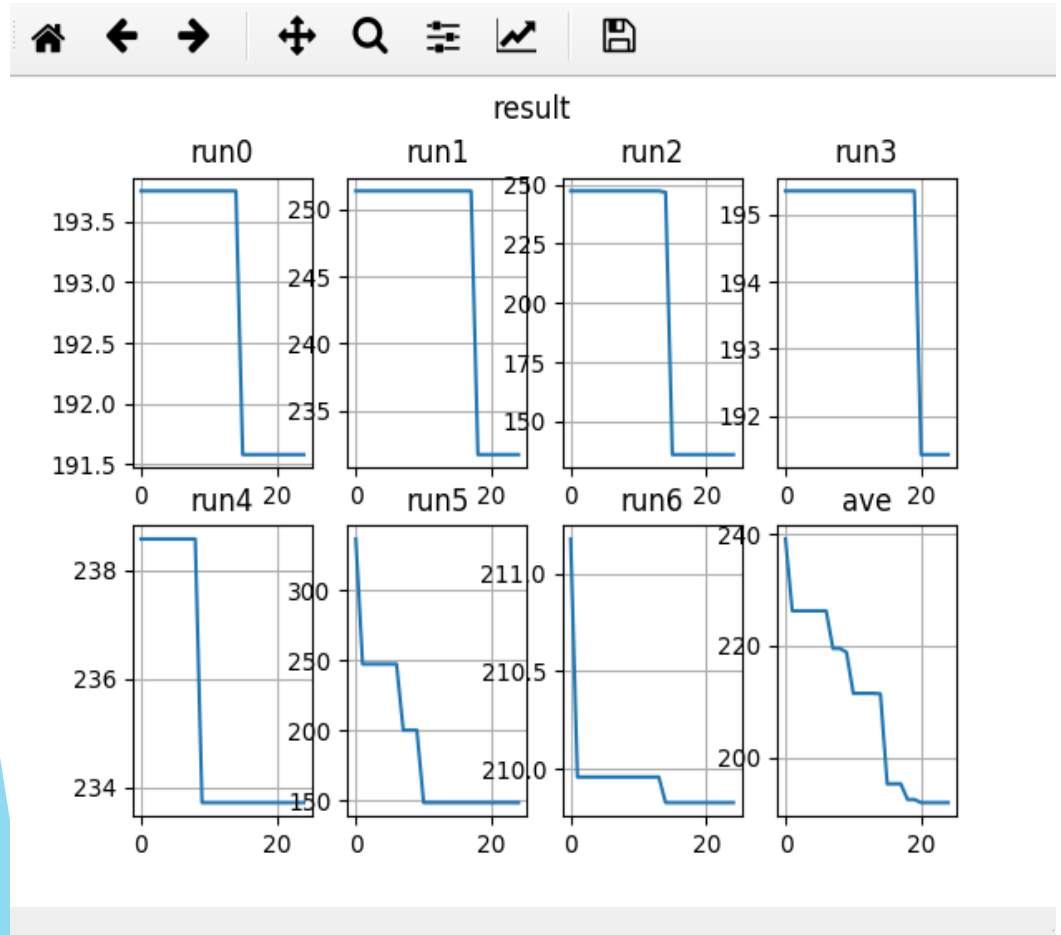


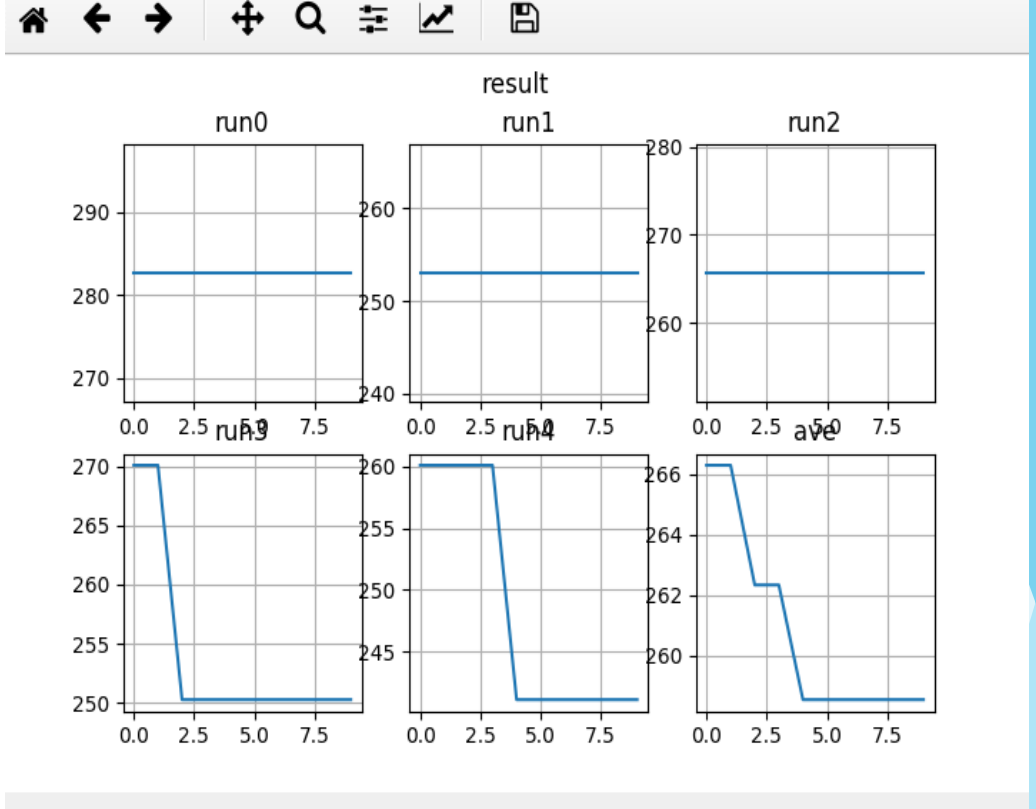
نمودار عملکرد مسئله فروشنده دوره گرد




end
start
10
8
6
4
2
0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
reset obstacles
set points
start x
1
start y
1
end x
18
end y
10
run
num of run =
5
iterate
500
result
clear path
draw best path


shirin@ubuntu: ~/pathplanning
File Edit View Search Terminal Help
iterate 2
260.0939467277376
mutated
iterate 3
260.0939467277376
iterate 4
241.07687528326716
mutated
iterate 5
241.07687528326716
iterate 6
241.07687528326716
mutated
iterate 7
241.07687528326716
mutated
iterate 8
241.07687528326716
mutated
iterate 9
241.07687528326716
mutated
best path cost = 241.07687528326716


result
run0
run1
run2
run3
run4
run5
run6
ave


result
run0
run1
run2
run3
run4
ave

خسته نباشید