ابتدا یک جمعیت کلی در نظر میگیریم مانند 500 عدد جفت عدد برای مقادیر x,y

سپس برای همه مقادیر مقدار fitness را بدست میاریم این مقدار fitness در واقع مقدار فاصله جواب ما از جواب واقعی است

در هر مرحله 100 تا 200 جفت از نسل قبلی را برای به ادامه نسل بعدی انتخاب میکردم که دقیقا جفت هایی بودند که کمترین مقدار fitness را در نسل خود داشته اند.

بقیه مقادیر نسل را به کمک crossover و mutartion بدست میاریم

در ابتدا مدل جواب های خوبی نمیداد و خب متوجه شدم که مشکل از مقادیری هست که برای پارامتر ها انتخاب کردم مثلا تعدادی که نسل ادامه یابد تا تعداد جمعیت یا تعداد جفت هایی که از نسل قبلی باقی بمانند

با فیکس کردن این مقادیر به دقت خوبی رسیدم و مدل معادلات را حل میکرد

بعد از به ان سراغ 3 معادله 3 مجهول رفتم و با خود فکر میکردم که نه تنها 3 معادله 3 مجهول بلکه برای هر تعداد معادله و به همان تعداد مجهول میتوان با همین طرز تفکر پیش رفت

برای همین یک تابع کلی در نظر گرفتم تحت عنوان الگوریتم ژنتیک و صرفا برای هر کدام از سوالات توابع fitness جدا گانه تعریف کردم و مدل برای همه ان ها با دقت خوبی کار میکرد

در ابتدا از numpy استفاده نکرده بودم به همین دلیل محاسبات خیلی طول میکشید و ممکن بود برای به نتیجه رسیدن یک نسل 1 دقیقه زمان لازم باشد.
اما بعد از استفاده از نام پای و یک سری بهینه سازی مدل در عرض 10 الی 20 ثانیه تمامی معادلات را حل میکرد.

جواب های مدل برای مثال هایی که در داک پروژه بود :

سوال اول :

Gen 1: x=-180173.1681, y=208012.7633 Loss=347194.73928852

Gen 2: x=-6529.9427, y=3450.6134 Loss=12696.60114938

Gen 3: x=-1585.9026, y=1197.1545 Loss=2371.39864807

Gen 4: x=24.2093, y=3.4067 Loss=125.48630178

Gen 5: x=12.6557, y=-8.0695 Loss=13.82791926

Gen 6: x=0.2744, y=2.7372 Loss=1.79491345

Gen 7: x=2.0888, y=0.9166 Loss=0.09951096

Gen 8: x=1.9785, y=1.0168 Loss=0.03099343

Gen 9: x=1.9968, y=1.0025 Loss=0.00455926

Gen 10: x=2.0001, y=1.0000 Loss=0.00039023

Gen 11: x=2.0000, y=1.0000 Loss=0.00002903

🧬 Stopped after 12 generations

✅ Best solution found:

x = 2.00

y = 1.00

🔍 Total error: 0.00

در اینجا مدل بعد از 12 نسل به جواب رسید .

سوال دوم :

Gen 1: x=-328.9693, y=-3009758.8324, z=-385672.2711, Loss=1019417679.48363471

Gen 2: x=987.2847, y=1238410.7571, z=8707.1608, Loss=74268962.21313876

Gen 3: x=2605.1106, y=-63175.8730, z=-99.7816, Loss=2378646.22021623

Gen 4: x=18.4812, y=64197.3946, z=-3757.2716, Loss=743526.44572994

Gen 5: x=-18.4426, y=-3275.3162, z=-172.9552, Loss=32086.42841737

Gen 6: x=6.4963, y=2682.5370, z=190.2967, Loss=20766.37976584

Gen 7: x=-10.5638, y=-67.6852, z=-1.7938, Loss=228.04013822

Gen 8: x=1.1406, y=4.9199, z=2.4485, Loss=58.80935359

Gen 9: x=0.5244, y=-5.4160, z=1.4099, Loss=24.89634512

Gen 10: x=0.7224, y=-4.2012, z=0.5486, Loss=6.09506548

Gen 11: x=0.6241, y=-5.1998, z=0.7379, Loss=0.95156673

Gen 12: x=0.6655, y=-5.0522, z=0.7575, Loss=0.25861253

Gen 13: x=0.6646, y=-5.0025, z=0.7506, Loss=0.06309570

Gen 14: x=0.6669, y=-5.0017, z=0.7501, Loss=0.01201481

Gen 15: x=0.6668, y=-4.9998, z=0.7500, Loss=0.00199653

Gen 16: x=0.6667, y=-4.9998, z=0.7500, Loss=0.00082309

Gen 17: x=0.6667, y=-5.0000, z=0.7500, Loss=0.00015512

Gen 18: x=0.6667, y=-5.0000, z=0.7500, Loss=0.00002583

🧬 Stopped after 19 generations

✅ Best solution found:

x = 0.67

y = -5.00

z = 0.75

🔍 Total error: 0.000

در اینجا مدل بعد از 19 نسل به جواب رسید البته هرزگاهی نیز جواب هایی را به دست می آورد که اصلا به جواب های درون داک نزدیک نبود اما وقتی ان ها را در معادلات جایگذاری میکردم

با دقت خوبی جواب ها را به دست می اورد اما دقیقا همان جواب نبود.

سوال سوم :

Gen 1: x=258356.6714, y=4706650.8778, z=-50730.1753, t=-6369435.0896 Loss=10881408.23203554

Gen 2: x=3125.3662, y=-731424.6574, z=-7626.3487, t=1053455.0679 Loss=1933304.73754572

Gen 3: x=-38898.5327, y=-433241.0959, z=-1974.6105, t=993579.9658 Loss=598675.63561746

Gen 4: x=-2581.2984, y=-54792.5608, z=-5040.8958, t=116856.1094 Loss=175890.78382332

Gen 5: x=-889.9916, y=-26982.5077, z=883.5603, t=44957.1587 Loss=60161.92168787

Gen 6: x=-325.5320, y=-3594.2393, z=202.9396, t=6359.4448 Loss=8044.50321844

Gen 7: x=-144.1166, y=-302.0264, z=91.3411, t=529.3523 Loss=3996.40263073

Gen 8: x=36.0762, y=230.8901, z=19.5035, t=-543.6233 Loss=989.17874573

Gen 9: x=0.4758, y=-62.2556, z=0.6136, t=48.1570 Loss=269.70175718

Gen 10: x=2.6631, y=21.2513, z=0.1882, t=-83.5291 Loss=104.16052680

Gen 11: x=-2.2365, y=-7.7660, z=0.0033, t=15.4752
Loss=25.94171678

Gen 12: x=-1.6080, y=-2.7458, z=0.1971, t=-1.9181
Loss=6.63772347

Gen 13: x=-1.5768, y=-4.5029, z=0.3960, t=0.3506
Loss=2.53084272

Gen 14: x=-1.5049, y=-3.7191, z=0.3329, t=-1.1425
Loss=0.74976728

Gen 15: x=-1.4935, y=-3.4774, z=0.3270, t=-1.3788
Loss=0.26352332

Gen 16: x=-1.4978, y=-3.5072, z=0.3328, t=-1.3557
Loss=0.06981716

Gen 17: x=-1.5016, y=-3.5044, z=0.3333, t=-1.3685
Loss=0.02868433

Gen 18: x=-1.5003, y=-3.4993, z=0.3334, t=-1.3752
Loss=0.00735438

Gen 19: x=-1.5000, y=-3.5008, z=0.3334, t=-1.3733
Loss=0.00191460

Gen 20: x=-1.5000, y=-3.5001, z=0.3333, t=-1.3748
Loss=0.00037790

Gen 21: x=-1.5000, y=-3.5001, z=0.3333, t=-1.3749
Loss=0.00020089

Gen 22: x=-1.5000, y=-3.5000, z=0.3333, t=-1.3750 Loss=0.00001992

Gen 23: x=-1.5000, y=-3.5000, z=0.3333, t=-1.3750 Loss=0.00001177

🧬 Stopped after 24 generations

✅ Best solution found:

x = -1.50

y = -3.50

z = 0.33

t = -1.37

🔍 Total error: 0.00

در اینجا مدل بعد از 24 نسل به جواب رسید که در این سوال نیز هرزگاهی جواب هایی را بدست می اورد که تقریب بسیار خوبی داشتند اما دقیق نبودند.

بعد از انجام این پروژه نتیجه گرفتم که نه تنها این معادلات بلکه هر معادله ای را میتوان با الگوریتم ژنتیک حل کرد فقط کافی است برا اساس نوع معادله تابع fitness مناسب تعریف کرد و انگاه میتوان هر نوع معادله ای را حل کرد

در واقع انتخاب تابع fitness مناسب 50 درصد راه است و بعد از ان میتوان هر معادله ای را با هر نسبت پیچیدگی با دقت و تقریب بسیار خوبی حل کرد .