CENTRAL WASHINGTON UNIVERSITY, DEPARTMENT OF COMPUTER SCIENCE

CS471 Project4

Jason Stearns

May 22, 2018

1 INTRODUCTION

The purpose of this lab was to code two optimization algorithms that use biomimicry, specifically swarms, one being the particle swarm optimization (PSO) and the other being the fire-fly algorithm. The particle swarm optimization algorithm uses a two-dimensional array known as a swarm, similar to the population in genetic algorithm from the previous experiment, the swarm consists of arrays that represent each particle in the Swarm. Another two-dimensional array is used to store the velocity of each particle in the swarm. During each iteration of the algorithm each particles' velocity us updated based both on randomness and the best solution found by the particle and the best solution found by the swarm. This emulates the travel of swarms by having the swarm move based on the best path found through older generations, the best path found through each members individual experience and random changes made in each movement. If a better solution is found by a member of the swarm every member of the swarm improves their solution based on this discovery. Then through many iterations the entire swarm benefits and a optimal solution is found. This means that particle swarm optimization solutions are highly dependent on iterations so if few iterations are ran the algorithm will not generate a very accurate solution. But this also means past a certain point there is a diminishing return on investment in runtime vs optimization of the solution space.

The second algorithm (Fire-Fly) uses another property of swarms which is attractiveness between members of the swarm, which is a biological property of fire-flies when they are trying to mate. Fire-flies create a bio-luminescence in order to attract a mate. The visibility/luminosity of a fire-fly is dependent on brightness and distance from nearby fire-flies in the swarm. In order to apply this to search space optimization the the algorithms uses a matrix that holds the vectors corresponding to the fireflies. The the fitness of each firefly is used to determine the brightness of each firefly. Using the brightness of each firefly, the luminosity is then calculated based on each fireflies position relative to nearby fireflies. Since fireflies are attracted to the brightness of other fireflies they will chose to move towards the brightest firefly they can see. So using an equation and a sorted list of each fireflies brightness, the fireflies are moved towards each other, which causes them to congregate around the firefly that has the best solution which results in a action similar to local search that makes improvements to the global best solution. Because this algorithm is based on attractiveness between particles and less on randomness introduced to the swarm the fire-fly algorithm is far less dependent on the number of iterations to achieve an optimal result.

2 Results

2.0.1 Particle Swarm Algorithm

Running particle swarm with 100 iterations and up to 30 dimensions results in data that is not very close to the known optimum for functions such as f1. For f1 the global optimum is known to be -419 x dimensions. In this case the results are -135 per dimension which is only 32 percent of the optimum which means the iterations should probably be increased in future experiments to achieve a closer optimum. Another way of increasing the accuracy of the algorithm would be to tune the values of c1 and c2 for each function since there is no two c1 and c2 that is optimal for all 15 functions. For example the 0.8 and 1.2 used for c1 and c2 respectively is fairly optimal for functions 5-7 but for functions 1-3 it is far from optimal.

2.0.2 Fire-fly Algorithm

Running the Fire-fly algorithms for 100 iterations resulted in as much as a 50 percent increase in accuracy in relation to the Particle Swarm algorithm. Also the results of the Fire-fly algorithm are much more tightly grouped when it comes to standard deviation and range, with an average decrease in both of 35 percent when compared to particle swarm. The firefly algorithm accuracy was also much less dependent on dimension size with changes in range and standard deviation only changing by 8 percent between 10, 20 and 30 dimensions. However further improvements could be made by increasing the number of iteration the algorithm is allowed to run for.

3 Conclusion

The data for particle swarm and fire-fly algorithm show that both are highly tied to the number of iterations the algorithm is run for. However this result is not unexpected, since in nature that is how swarms evolve. Swarms in the natural world evolve and improve over thousands and millions of iterations. The data also shows that fire-fly has more of a linear improvement in relation to iterations. This is evident by the relatively static range and standard deviation between the algorithm being run for 10, 20 and 30 dimensions. For particle swarm the relationship between the number of iterations and optimization of the result is much less linear as shown by the much larger and fluctuating range and standard deviation for the same number of dimensions when compared to fire-fly algorithm. This could be in part due to the random component used in particle swarm, in particle swarm a randomness is introduced into the movement of each particle which is in contrast with the fire-fly algorithm which does not use randomness in each particle but instead only in the best particle of the swarm which minimizes its effect but still keeps the swarm from stagnating.

Table 3.1: Computation PSO 10 dimensions

Problem	PSO				
	Avg	Median	Range	SD	T(s)
f_1	-1097.39446666667	-1166.295	2753.23	541.870410618303	0.04
f_2	12809.031	11722.8	22117.8	5403.06947379379	0.031
f_3	2652740000	1540525000	14665624000	2973731807.87312	0.055
f_4	4412.13284960898	18558.45	23525.56	5232.9727011096	0.042
f_5	71.0843433333334	62.25695	129.4338	30.0546625752997	0.052
f_6	-8.42830166666667	-8.49677	3.98927	0.856585608724496	0.069
f_7	43.1937633333333	41.82125	24.3119	6.06631241180794	0.073
f_8	92.3571433333333	90.0509	72.2679	18.4348171426007	0.064
f_9	133.910133333333	132.7675	70.143	14.4008158813621	0.08
f_{10}	-1394.02596666667	-1474.265	2531.34	655.067215705507	0.052
f_{11}	-967.767033333334	-974.865	1437.908	360.081243292906	0.061
f_{12}	3.900706	3.977565	1.02428	0.219524312512305	0.043
f_{13}	-2.35789966666667	-2.348495	2.82296	0.673977931752879	0.046
f_{14}	-4.0486449	-4.13552	6.362093	1.16831723337013	0.045
f_{15}	-0.0331033566666667	-0.0318006	0.0386051	0.00820181951352601	0.042

[1] , 3.4GHz Intel core i5-3570K, 8 GB RAM

Table 3.2: Computation Fire Fly 10 dimensions

Problem	Fire-Fly				
	Avg	Median	Range	SD	T(s)
f_1	-1350.5121	-1335.8	1458.619	335.627233050432	0.348
f_2	14288.9933333333	15203.95	11853.21	3313.57507273071	0.141
f_3	3158953233.33333	2665120000	5913166000	1566213143.86699	0.463
f_4	25781.55	26089.6	17456.1	4076.50561655854	0.328
f_5	91.73259	88.9622	91.0851	23.6737752982542	0.454
f_6	-8.32168	-8.25323	1.64471	0.467504717608996	0.62
f_7	46.3310833333333	46.4995	14.0973	3.38900966961179	0.714
f_8	108.376436666667	110.5545	55.2028	14.0000880645916	0.569
f_9	138.5931	137.8605	31.096	8.03183078486261	0.816
f_{10}	-2030.54433333333	-1965.035	1643.67	378.640588920267	0.459
f_{11}	-1355.52056666667	-1354.98	1403.061	293.930455623513	0.557
f_{12}	3.86106933333333	3.938385	0.77443	0.204032280516807	0.324
f_{13}	-2.82473233333333	-2.780115	1.43431	0.348067676433604	0.25
f_{14}	-4.86129366666667	-4.52524	4.19419	1.0536823023204	0.366
f_{15}	-0.0432904033333333	-0.0416815	0.0419843	0.00787040314568806	0.207

[1] , 3.4GHz Intel core i5-3570K, 8 GB RAM

Table 3.3: Computation PSO 20 dimensions

Problem	PSO				
	Avg	Median	Range	SD	T(s)
f_1	-1789.2043	-1735.06	2471.119	548.015349733634	0.069
f_2	29483.6833333333	28995.75	21847.8	6052.54644416068	0.043
f_3	11028637333.3333	9783330000	24941980000	4821915772.0792	0.112
f_4	108077.406666667	105999.5	110720.6	23251.6399763534	0.076
f_5	197.226796666667	202.6465	206.7301	43.2090933741845	0.094
f_6	-16.7597666666667	-16.88175	4.0725	1.04157416805312	0.128
f_7	102.7049	101.0035	33.14	8.54923283443218	0.139
f_8	246.563633333333	241.063	255.676	48.318882806127	0.115
f_9	301.899933333333	305.212	74.57	14.3007528564835	0.151
f_{10}	-2162.6803	-2115.28	3289.374	733.549077831023	0.097
f_{11}	-1805.996	-1778.77	1227.45	363.168185478851	0.135
f_{12}	8.80654533333333	8.779925	1.25944	0.274575801904603	0.075
f_{13}	-3.85824233333333	-3.980405	4.62681	1.01008962227017	0.083
f_{14}	-6.00012533333333	-5.83565	7.50254	1.89251377574684	0.079
f_{15}	-0.03494101	-0.0344889	0.0243618	0.00606666945022002	0.057

[1] , 3.4GHz Intel core i5-3570K, 8 GB RAM

Table 3.4: Computation Fire Fly 20 dimensions

Problem	Fire-Fly				
	Avg	Median	Range	SD	T(s)
f_1	-1935.09866666667	-1960.27	1637.15	383.70023589371	0.594
f_2	38046.1966666667	38330.6	23110.9	4995.61597319645	0.254
f_3	16468501333.3333	16064300000	14091100000	3537157612.40362	0.951
f_4	140086.326666667	143525	82278.2	17742.9996205441	0.632
f_5	244.280833333333	249.923	163.877	32.0436350238066	0.891
f_6	-16.43645	-16.28225	1.9675	0.553321345301866	1.289
f_7	107.127313333333	107.7275	19.5246	4.58283015982725	1.486
f_8	279.680266666667	279.6745	63.587	18.8247498291183	1.173
f_9	316.912166666667	318.951	37.897	8.5038592222721	1.622
f_{10}	-2812.14466666667	-2799.39	3350.16	685.704866006423	0.927
f_{11}	-1946.24366666667	-2005.91	1355.04	350.743583105791	1.138
f_{12}	8.69697566666667	8.70973	0.959070000000001	0.18206191673317	0.635
f_{13}	-4.638469	-4.62027	3.03301	0.640547897950656	0.504
f_{14}	-6.46386533333333	-6.383425	6.09015	1.28274123950684	0.72
f_{15}	-0.0423362733333333	-0.0404486	0.0223903	0.00593711464309802	0.214

[1] , 3.4GHz Intel core i5-3570K, 8 GB RAM

Table 3.5: Computation PSO 30 dimensions

Problem	PSO				
	Avg	Median	Range	SD	T(s)
f_1	-2168.92326666667	-2425.74	3474.553	819.167597582038	0.095
f_2	56275.94	54775.25	67075.2	12973.6958936559	0.054
f_3	21393973666.6667	17451250000	38317490000	9651341052.76792	0.143
f_4	276063.066666667	273950	329281	64163.3319593745	0.101
f_5	319.669066666667	320.939	217.136	46.8736849927072	0.135
f_6	-24.7463	-24.9089	3.9625	0.997802267652932	0.182
f_7	163.2719	162.7745	35.029	7.9102369553636	0.207
f_8	391.5284	392.3395	254.722	54.9397986087806	0.178
f_9	475.501333333333	474.685	126.647	23.4613621504711	0.227
f_{10}	-2822.5491	-3024.23	3640.766	1008.34635554378	0.139
f_{11}	-1823.9327	-1908.62	4701.657	791.230551952553	0.165
f_{12}	13.6556433333333	13.5837	1.4158	0.312426932986827	0.111
f_{13}	-5.38417866666667	-5.30011	4.63789	1.01581620527775	0.114
f_{14}	-6.14838933333333	-6.06092	9.19269	1.93075197035237	0.113
f_{15}	-0.03551273	-0.03445725	0.0232018	0.00543919041587073	0.072

[1] , 3.4GHz Intel core i5-3570K, 8 GB RAM

Table 3.6: Computation Fire Fly 30 dimensions

Problem	Fire-Fly				
	Avg	Median	Range	SD	T(s)
f_1	-2329.988	-2369.625	1762.96	433.937781272848	0.875
f_2	66018.0233333333	65609.5	23732.8	6748.38586096371	0.374
f_3	29386666666.6667	28300000000	16600000000	3918310293.42439	1.431
f_4	343870.2	343248	184764	37770.0682078636	0.934
f_5	406.9031	410.217	223.309	47.3547675229924	1.329
f_6	-24.19631	-23.9723	4.6656	0.848861651212965	1.958
f_7	171.853	173.0115	24.542	5.77795980140626	2.24
f_8	439.605	438.9245	133.846	29.3286269095572	1.766
f_9	494.822233333333	495.555	70.779	16.0672118669945	2.49
f_{10}	-3587.63166666667	-3511.215	3671.17	766.30894036754	1.388
f_{11}	-2389.30133333333	-2398.585	1762.1	466.997723033232	1.776
f_{12}	13.5324933333333	13.612	1.3456	0.302328191356494	0.987
f_{13}	-6.43057433333333	-6.188385	4.6176	1.00653766794123	0.761
f_{14}	-8.75811833333333	-8.13763	6.17544	1.52437256395341	1.086
f_{15}	-0.0423700833333333	-0.0415836	0.0226193	0.00537054524135016	0.227

[1] , 3.4GHz Intel core i5-3570K, 8 GB RAM