CENTRAL WASHINGTON UNIVERSITY, DEPARTMENT OF COMPUTER SCIENCE

CS471 Project 4

Taylor Apple

May 20, 2019

1 SWARM OPTIMIZATION

In this project we were required to expand upon our first three projects implementing 18 mathematical functions with multi-dimensional vectors, in simulation of NP-Hard problems. The problem set we were presented with was the following 18 functions: Schwefel's, 1st De Jong's sphere, Rosenbrock's Saddle, Rastrgrin's, Griewangk's, Sine Envelope Sine Wave, Stretched V Sine Wave, Ackley's One, Ackley's Two, Egg Holder, Rana, Pathological, Michalewicz's, Master's Cosine Wave, a Quartic function, Levy, a Step function, and Alpine.

Using these equations as our objective functions, we were required to implement three different swarm optimization algorithms in order to find local minimums for each equation in the search space ranges provided. These algorithms were Particle Swarm Optimization(PSO), Firefly Algorithm (FA), and Harmonic Search Algorithm (Harmonic/HS). Each of these ranges differed for each function and we had to develop a modular program which would read an input file and perform different types of testing based on the users input. This included them setting parameters for each equation which differed for the three different algorithms.

For each algorithm there were three adjustable parameters. PSO utilized a velocity dampener, a C_1 modification value, and a C_2 modification value. FA depended on a gamma for light absorption, a beta for attractiveness, and an alpha for scaling in the range. Finally, Harmonic used a Harmonic Consideration Rate (HMCR), Pitch Adjustment Rate (PAR), and bandwidth. Each of these parameters will be explained in more detail in the corresponding algorithm's explanation.

1.1 PARTICLE SWARM OPTIMIZATION (PSO)

Before, in Genetic Algorithm, vectors, or particles, were being selected at random and then dimensions were swapped and dimensions were mutated but the mutation was static and the crossover was only for a random number of dimensions and could sometimes even produce no effect depending on the position and number of those crossovers. With Particle Swarm Optimization a new process for manipulating dimensions is introduced. Each dimension of each particle keeps track of it's previous "velocity", a direction and speed by which the dimension was changing. This velocity is based on the global best and the personal best for this population and particle respectively and is added to the previous velocity.

$$v_{t+1,i} = k(v_{t,i} + C_1 * (pBest_{t,i} - x_{t,i}) + C_2 * (gBest_{t,i} - x_{t,i})) \tag{1.1}$$

By the equation above the new velocity is calculated and stored for each dimension. K is the dampener value and is used to make sure the velocity does not grow too fast and the bounds of the search space are not exceeded too quickly. $v_{t,i}$ is the current velocity of the dimension, C_1 is the size of step taken towards the personal best of the current vector, and C_2 is the size of step taken towards the global best dimension. The particles in question are $pBest$, the personal best particle at the current position, $gBest$, the global best particle of the entire population, and x, the current particle.

Once the new velocities for all dimensions are produced, a new particle is produced by adding the new velocity to the old value of this dimension.

$$x_{t+1,i} = x_{t,i} + v_{t+1,i} \tag{1.2}$$

Then the new particle's fitness is evaluated, and stored at the current position and then validated against the personal and global bests, replacing each if necessary. For this algorithm, the number of times the fitness is calculated is always once for the entire initial population, then once per new particle per iteration. Therefore, if there are M iterations and N particles in the population there will be $(M * N) + N$ fitness evaluations. So for our experiments of 500 iterations with a population of 500 particles we calculated the fitness of an individual particle 250,500 times.

1.2 Firefly Algorithm (FA)

The Firefly algorithm is the most intensive computationally. For each iteration, the entire population of fireflies is iterated through, and each firefly is compared to every other firefly. First, the light intensity or fitness of the current firefly is compared to the newly selected firefly. If the newly selected firefly has a better fitness than the current firefly is drawn to that firefly and the movement is calculated.

$$x_{t+1,i} = x_{t,i} + B * e^{-Gr^2} + A * (rand - 0.5) * (U - L) \tag{1.3}$$

Each dimension is calculated with the above formula. B is beta and represents the the size of step taken in the range of the distance between these two fireflies. This term also is calculated based on the inverse square principle where r is the distance between the fireflies, and Gamma, G is a scalar on the distance. A is the alpha parameter and is multiplied by a random number between 0 and 1, $rand$ and the range of values in the search space where U is the upper bound and L is the lower bound.

Once this new firefly is calculated, it is evaluated for its fitness, and if it is better than the worst firefly it is stored in place of the worst firefly. If it is better than the best firefly this is noted and stored. In both cases a new worst firefly is calculated and the process continues. So, with M iterations and a population of N fireflies we can prove in the worst case, a purely diverse population, how may fitness calculations will be present. We can assume in a population of 500 that the best firefly will require 0 evaluations, second best 1 evaluation, third best 2 evaluations, etc. until the worst has 499 evaluations. Therefore we can derive the following equation.

$$numEvals = M * \sum_{i=0}^{N-1} i \tag{1.4}$$

By the proof of the sum of Triangular Numbers we know that $\sum_{i=0}^{N-1} i = N(N-1)/2$. Therefore we can calculate that there would be 124,750 fitness evaluations per iteration and with 500 iterations there were 62,375,000 total fitness evaluations per experiment. Occasionally there were fewer, when the population would have fireflies with the same dimensions, and this would range from 60,000,000 - 62,375,000. However in the worst case there are $M * (N(N-1)/2)$.

1.3 Harmonic Search (Harmonic/HS)

Harmonic Search is the most basic of the algorithms implemented, with the smallest number of evaluations, changes, and improvement of the results. What it is good for though is narrowing the potential search space for the best answers. With only one new vector created per iteration, Harmonic Search relies on a high amount of stochasticity to find dimensional values. First, a random value is developed to check whether harmonic tuning should be considered. This check is against the Harmonic Consideration Rate (HMCR) and if it is being considered, a random dimensional value from this dimension among all the vectors in the current population is selected. Then, another check against the Pitch Adjustment Rate (PAR), determines whether the pitch of the vector should be adjusted or not.

$$x_{t+1,i} = x_{randT,i} + rand * bandwidth \tag{1.5}$$

For HS the bandwidth is the size of steps to be taken when tuning the vector and rand is a random number between -1 and 1 which controls the direction the tuning is done in. This is applied to the random dimensional value achieved in the previous step and a new dimension is produced. If tuning is not considered, then a new random number in the range of the search space is generated and this is used. As the final steps of the iteration, the new vector's fitness is evaluated and if it is better than the worst vector, that vector is replaced, and if it is better than the best fitness than this value is recorded. Then the worst value in the population is validated and the next iteration begins. This means that with M iterations and a population size of N there are only ever $M+N$ fitness evaluations and so there were 1000 fitness evaluations in these 500 iterations of a 500 population size.

2 Graphs and Data Tables

2.1 Graphs of Population Standard Deviation by Dimension

Below are 18 graphs, one for each equation, with lines which represent the different dimensions, 30 for the populations in our experiments, for each of the algorithms in this experiment. These charts will show the change in the standard deviation of each dimension which helps with analyzing the stagnation of a population. As the standard deviation of a dimension drops it is beginning to stagnate as there is no diversity in the dimension. When enough of the dimensions have stagnated, the population will be considered stagnated and results cannot be expected to improve much further. This is a random experiment among the initial 30 experiments and only represents one experiment's population.

Parameters for these experiments were 500 vectors/particles/fireflies and 30 dimensions per member of the population over 500 iterations. For PSO a dampener value of 0.9 was used, c1 modification value of 0.8, and c2 mod value of 1.2. For the Firefly Algorithm a gamma of 0.00001, alpha of 0.5, and beta of 0.2 were used. Finally, for the Harmonic Search Algorithm the HMCR value was 0.85, PAR value was 0.5, and bandwidth was 0.2.

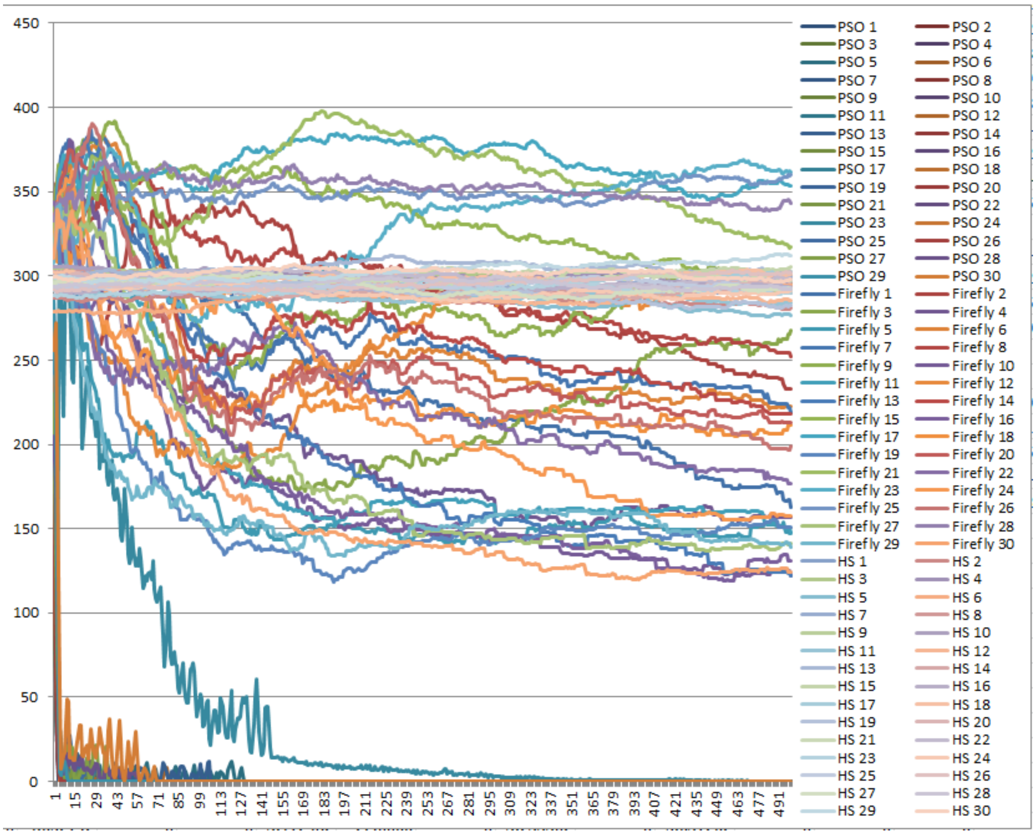

Figure 2.1: Graph of Population's Standard Deviation on Schwefel's Function

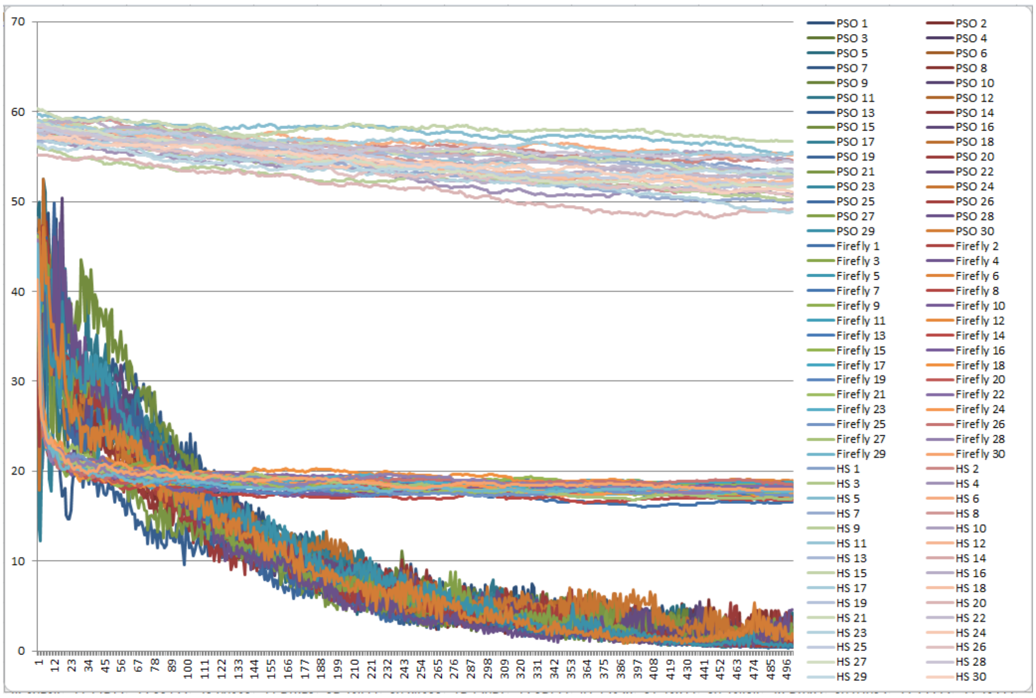

Figure 2.2: Graph of Population's Standard Deviation on DeJong's Sphere Function

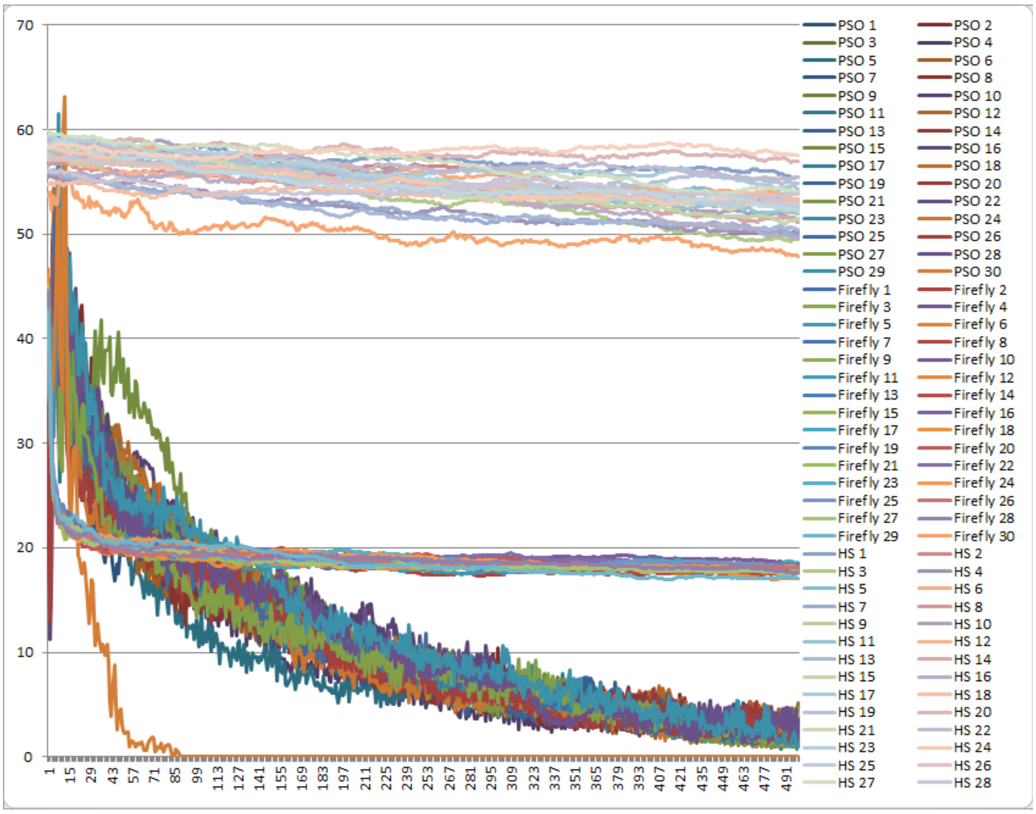

Figure 2.3: Graph of Population's Standard Deviation on the Rosenbrock's Saddle Function

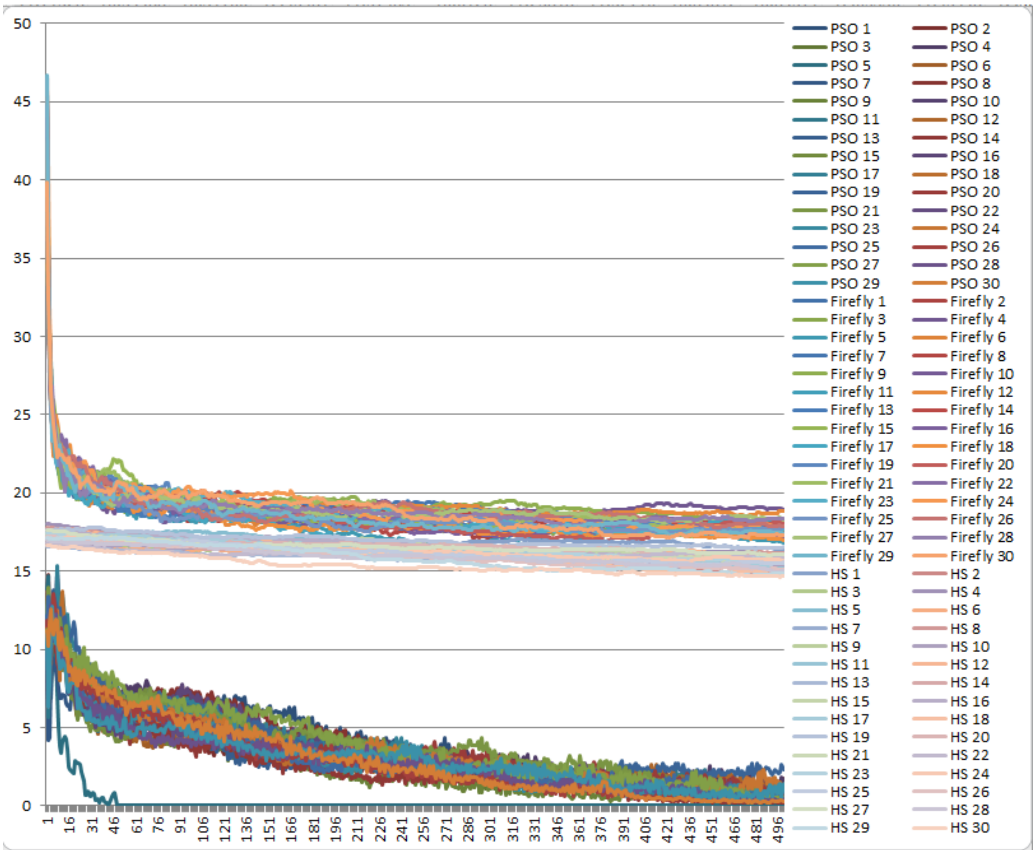

Figure 2.4: Graph of Population's Standard Deviation on Rastgrin's Function

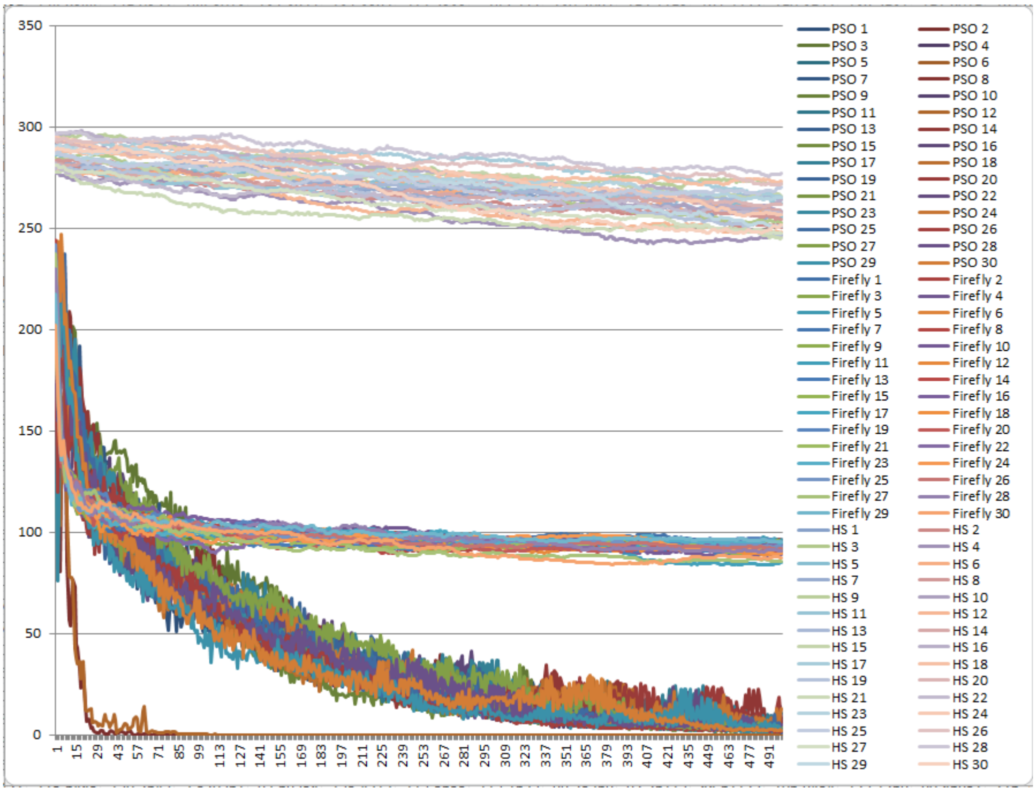

Figure 2.5: Graph of Population's Standard Deviation on Griewangk's Function

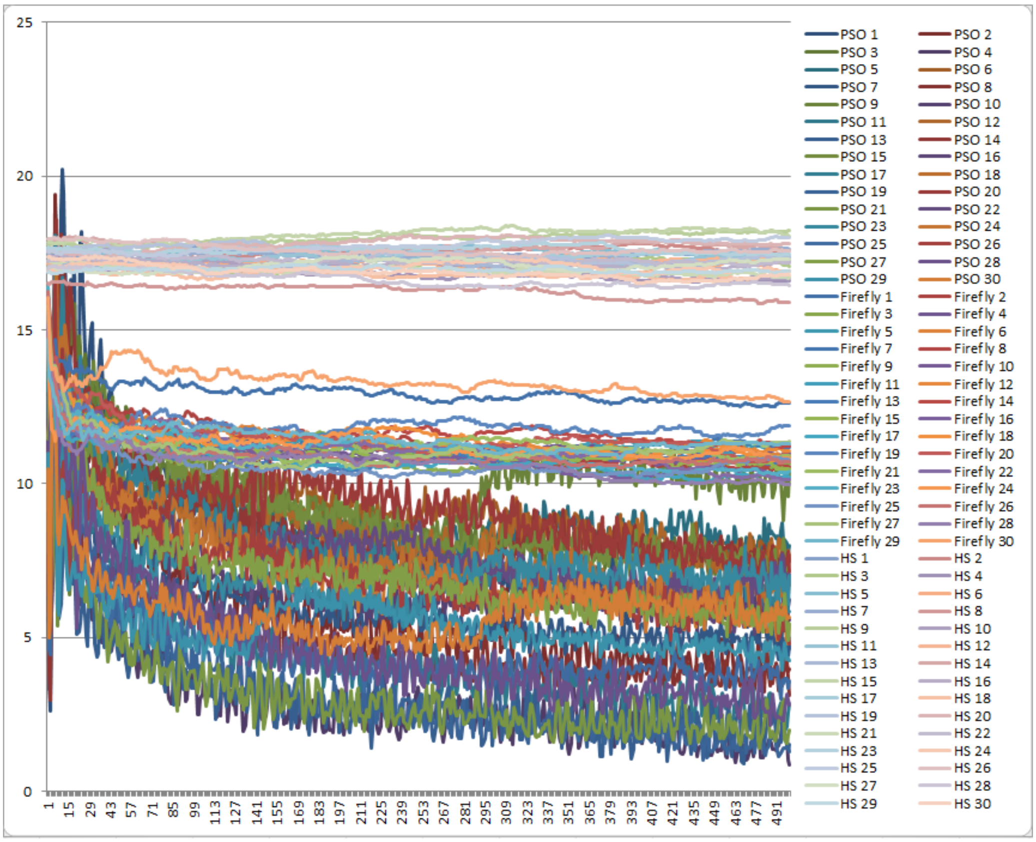

Figure 2.6: Graph of Population's Standard Deviation on the Sine Envelope Sine Wave Function

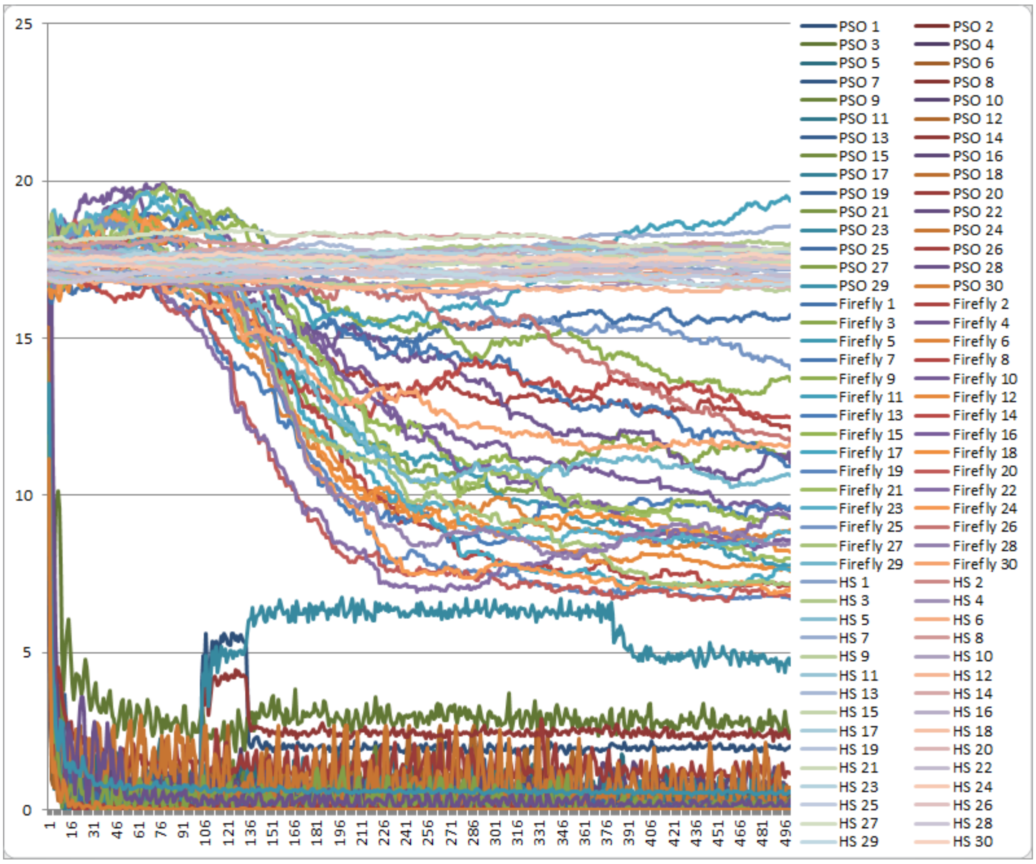

Figure 2.7: Graph of Population's Standard Deviation on the Stretch V Sine Wave Function

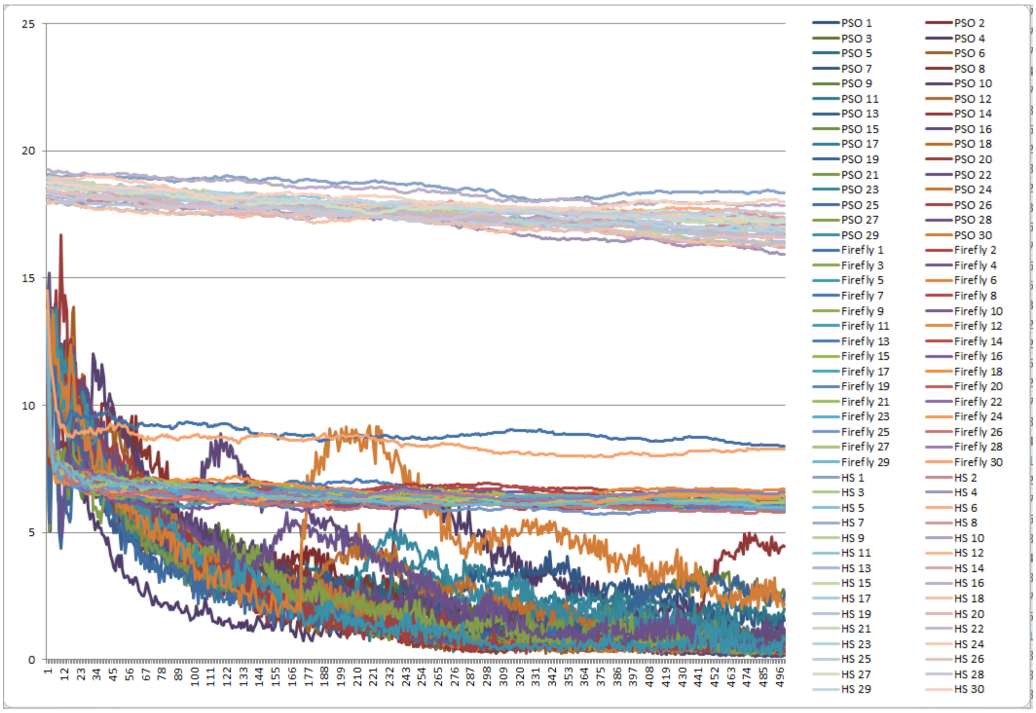

Figure 2.8: Graph of Population's Standard Deviation on the Ackley One Function

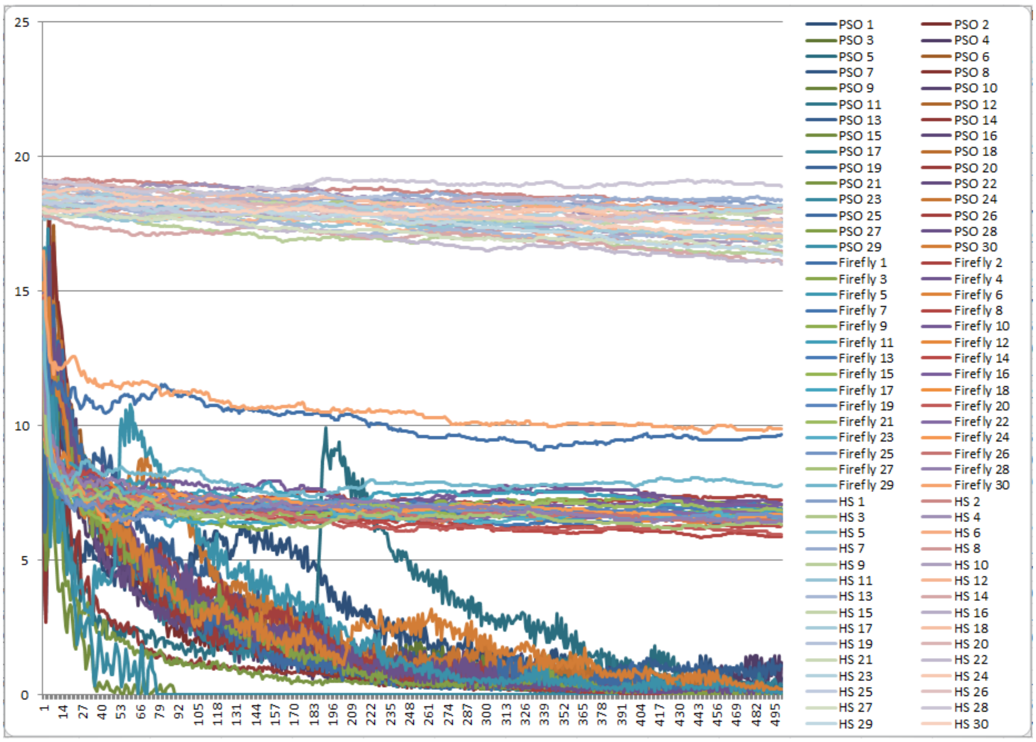

Figure 2.9: Graph of Population's Standard Deviation on the Ackley Two Function

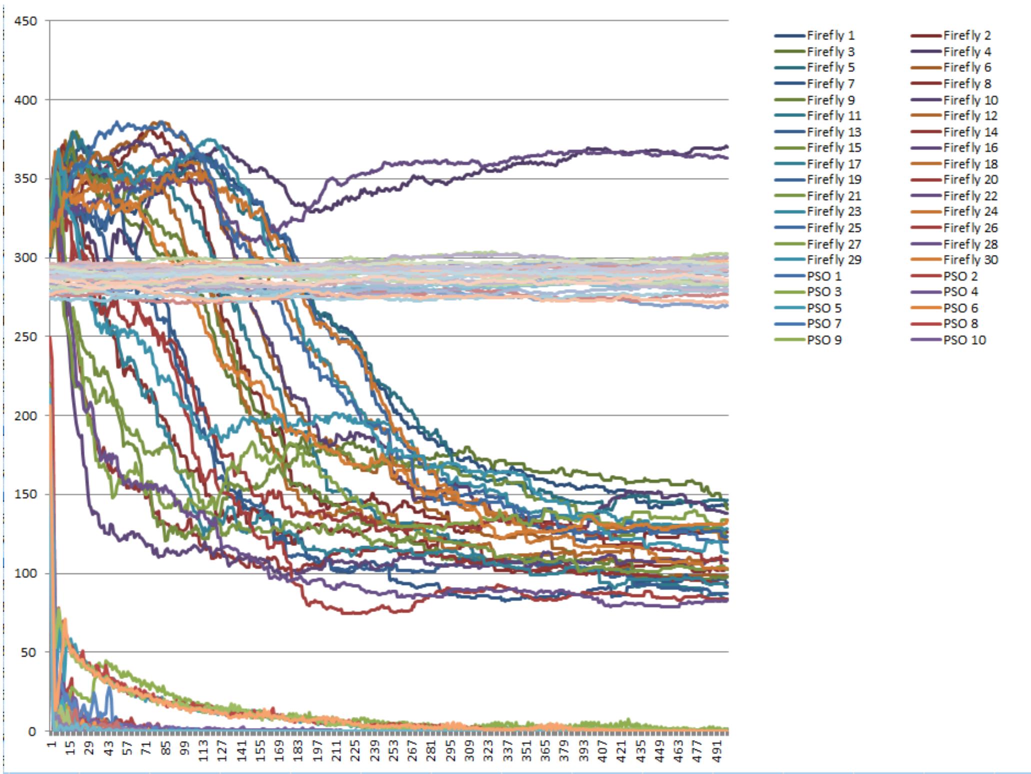

Figure 2.10: Graph of Population's Standard Deviation on the Egg Holder Function

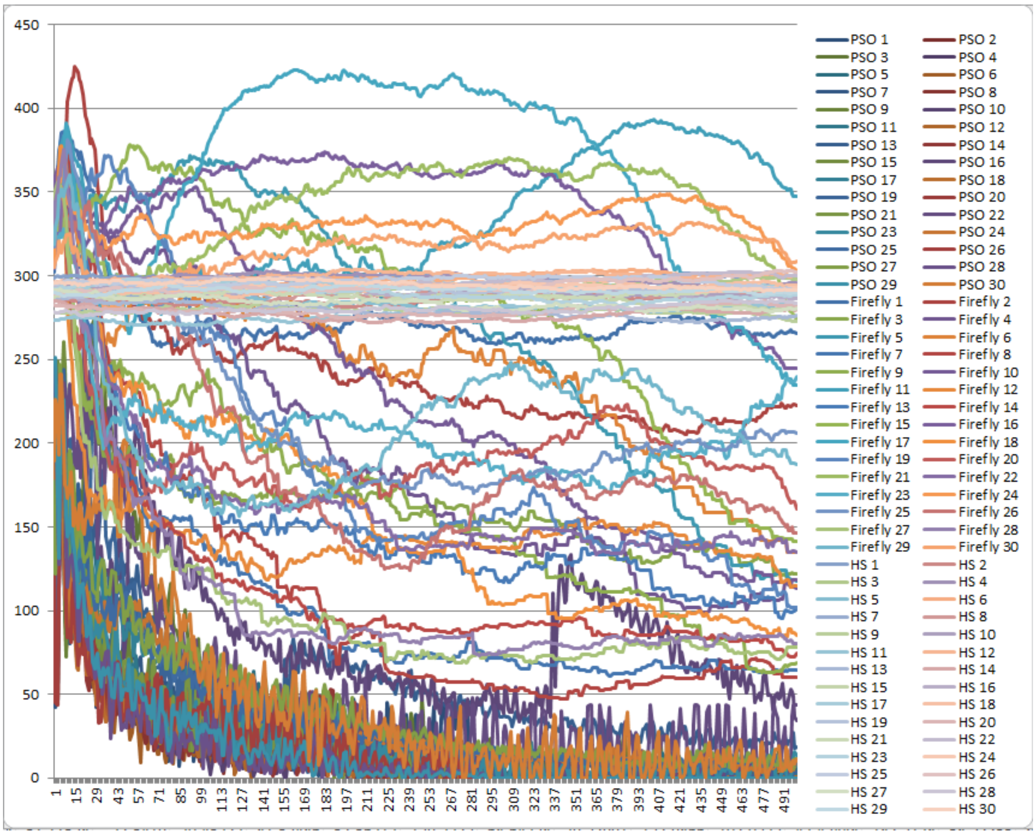

Figure 2.11: Graph of Population's Standard Deviation on Rana's Function

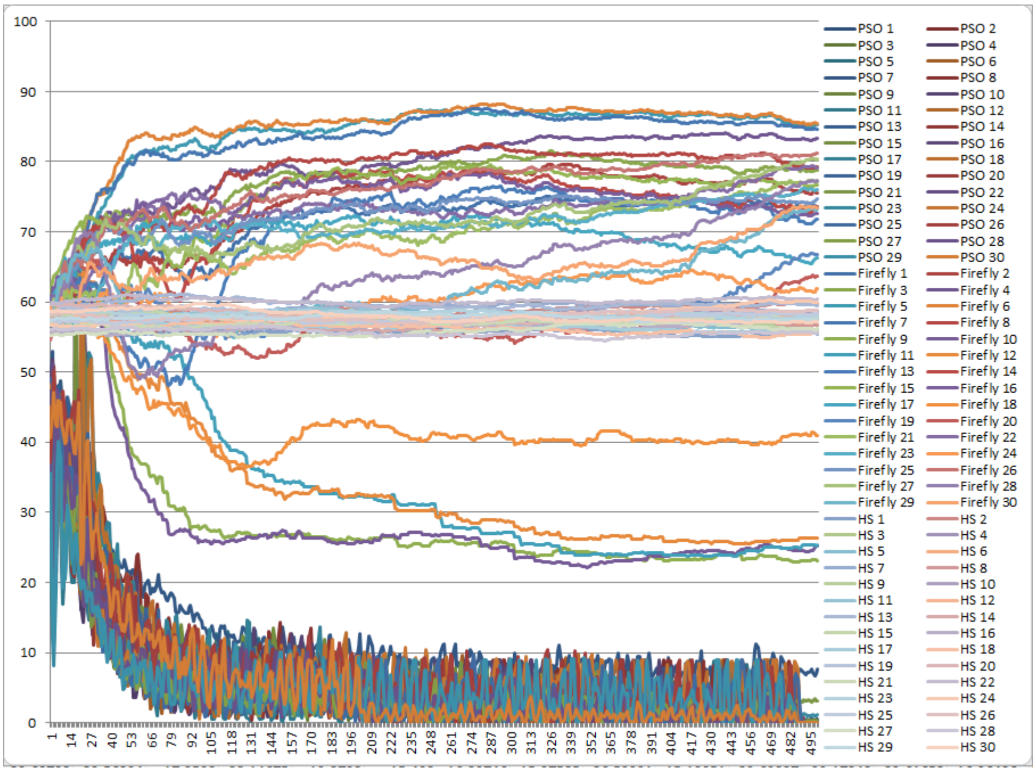

Figure 2.12: Graph of Population's Standard Deviation on a Pathological Function

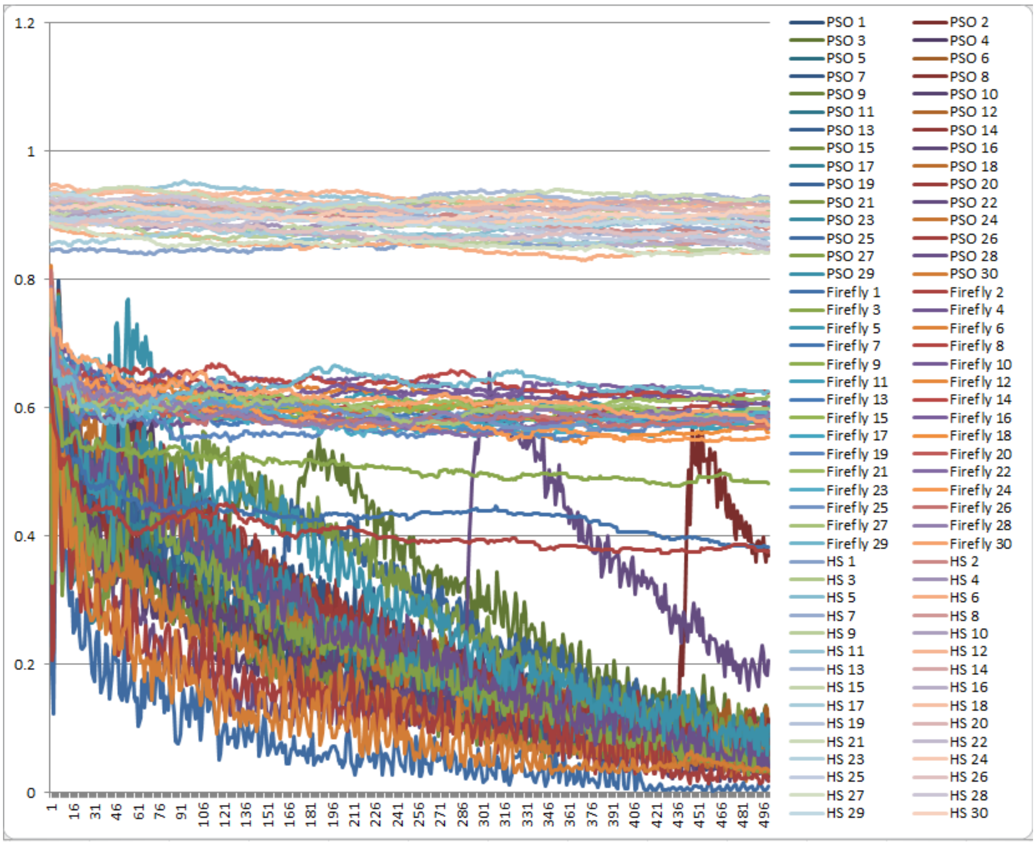

Figure 2.13: Graph of Population's Standard Deviation on the Michalewicz's Function

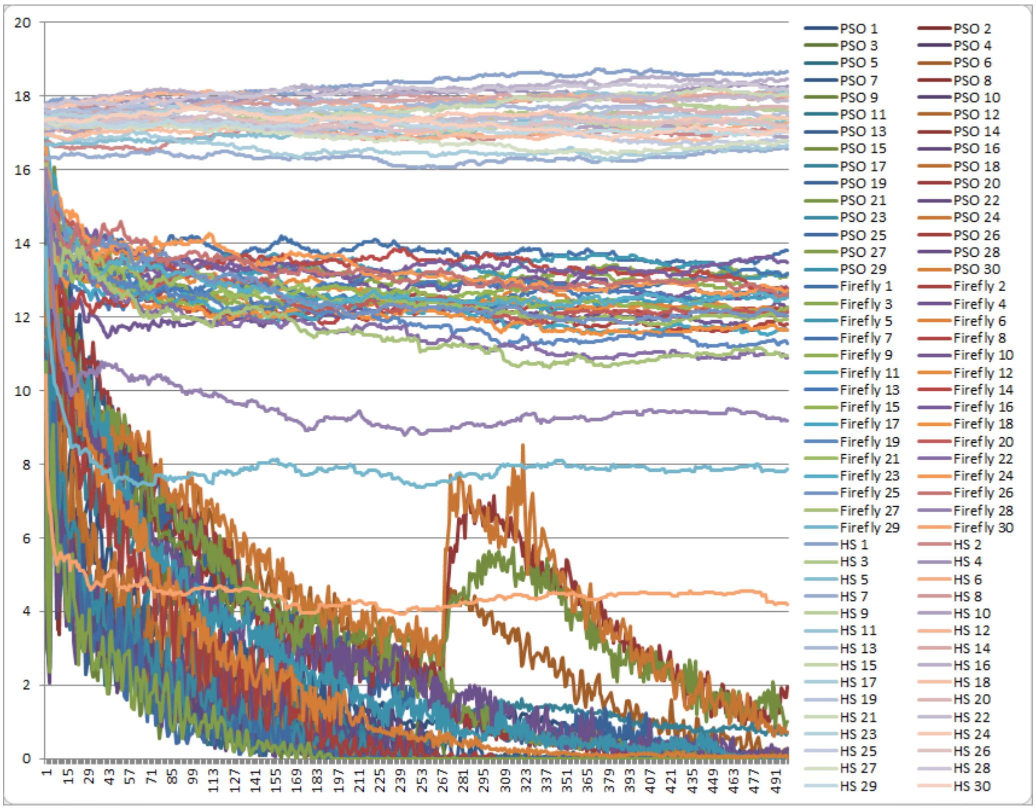

Figure 2.14: Graph of Population's Standard Deviation on the Masters' Cosine Wave Function

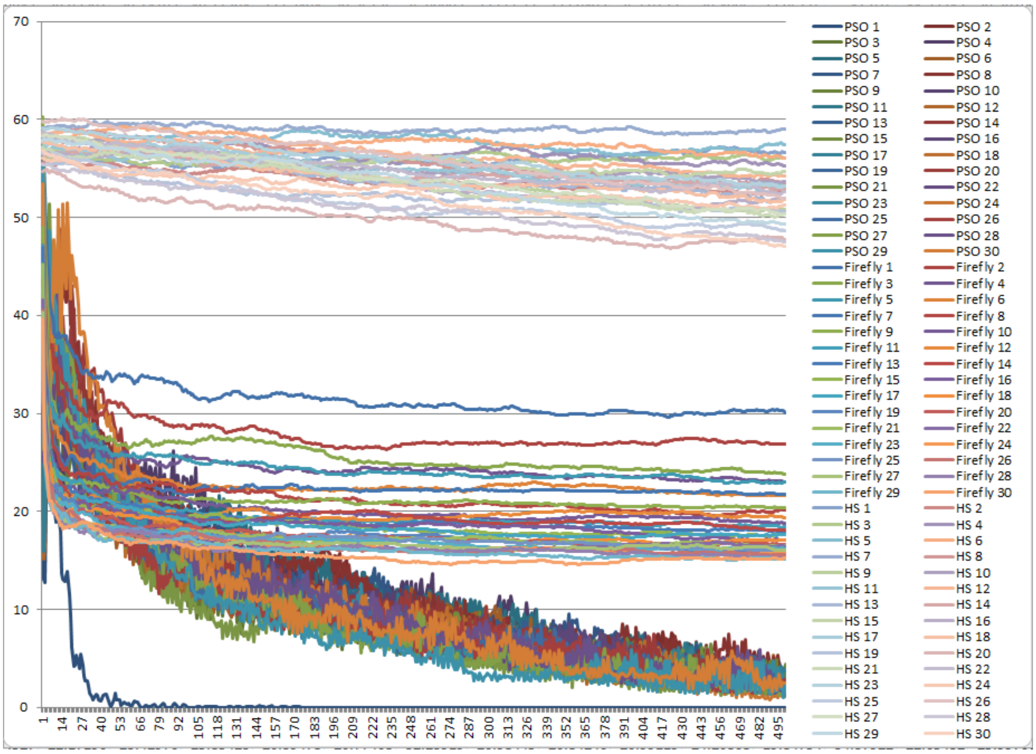

Figure 2.15: Graph of Population's Standard Deviation on the Quartic Function

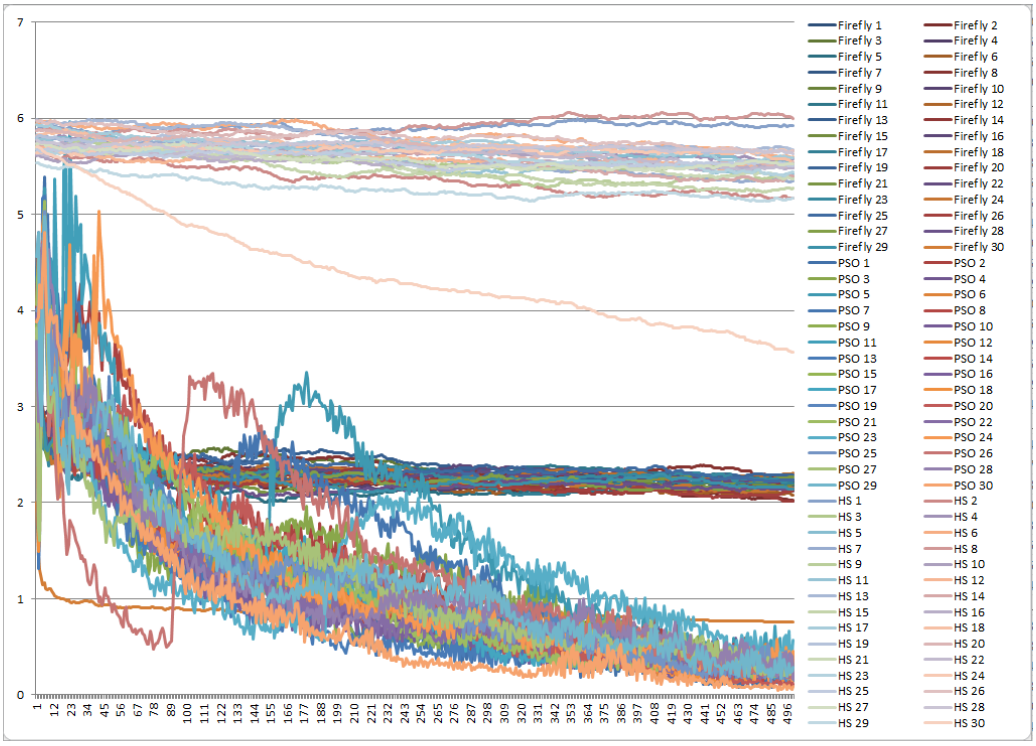

Figure 2.16: Graph of Population's Standard Deviation on the Levy Function

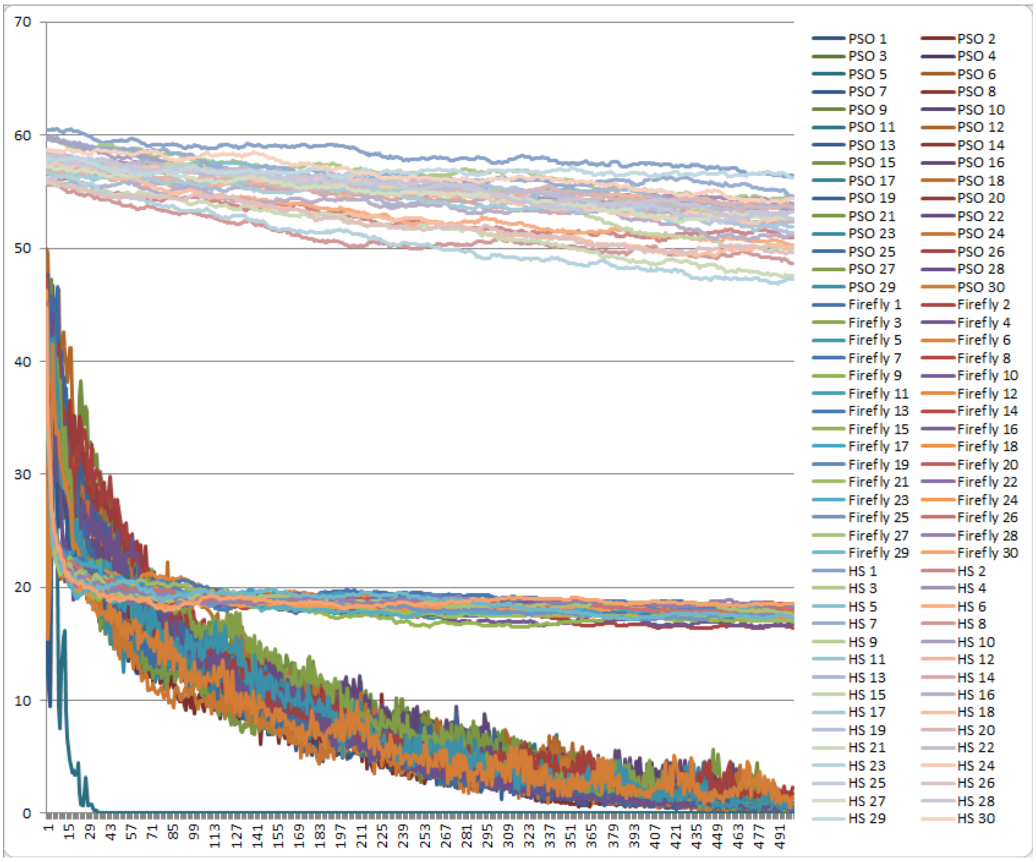

Figure 2.17: Graph of Population's Standard Deviation on the Step Function

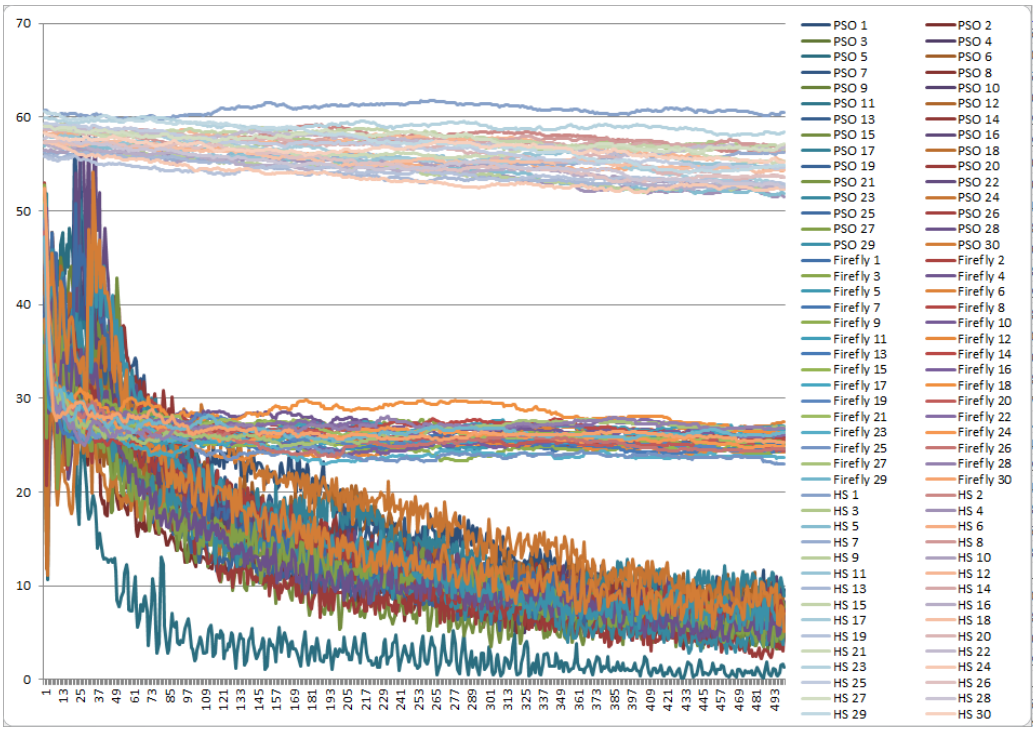

Figure 2.18: Graph of Population's Standard Deviation on the Alpine Function

2.2 Graphs of the Best and Worst Results

Below are 18 graphs, one graph for each equation, with each line representing either the best or worst results for each algorithm. For PSO there will be no worst results as the value is not used or needed to be tracked, whereas Firefly and Harmonic Search will both include a best and worst solution at each iteration as these are important to the algorithms. These graphs represent the results over one experiment, and it is the same experiment as the above population standard deviation graphs.

Parameters for these experiments were 500 vectors/particles/fireflies and 30 dimensions per member of the population over 500 iterations. For PSO a dampener value of 0.9 was used, c1 modification value of 0.8, and c2 mod value of 1.2. For the Firefly Algorithm a gamma of 0.00001, alpha of 0.5, and beta of 0.2 were used. Finally, for the Harmonic Search Algorithm the HMCR value was 0.85, PAR value was 0.5, and bandwidth was 0.2.

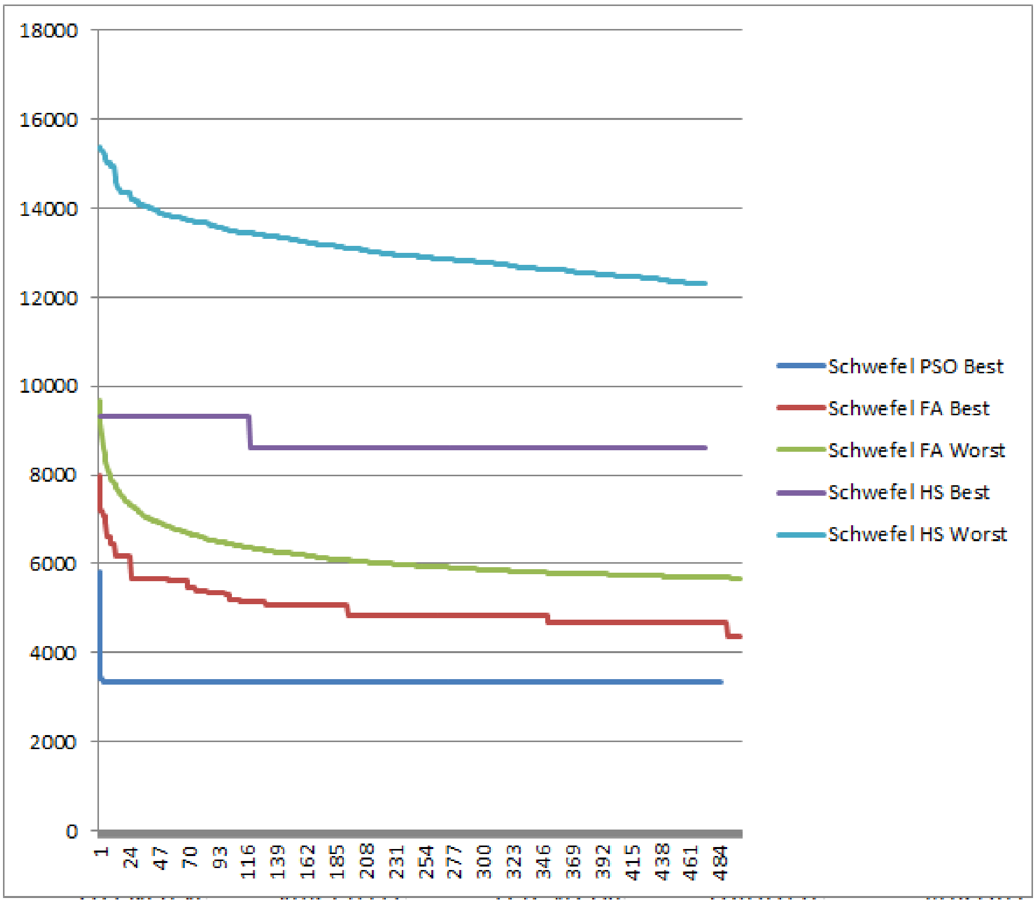

Figure 2.19: Graph of Best and Worst Results on Schwefel's Function

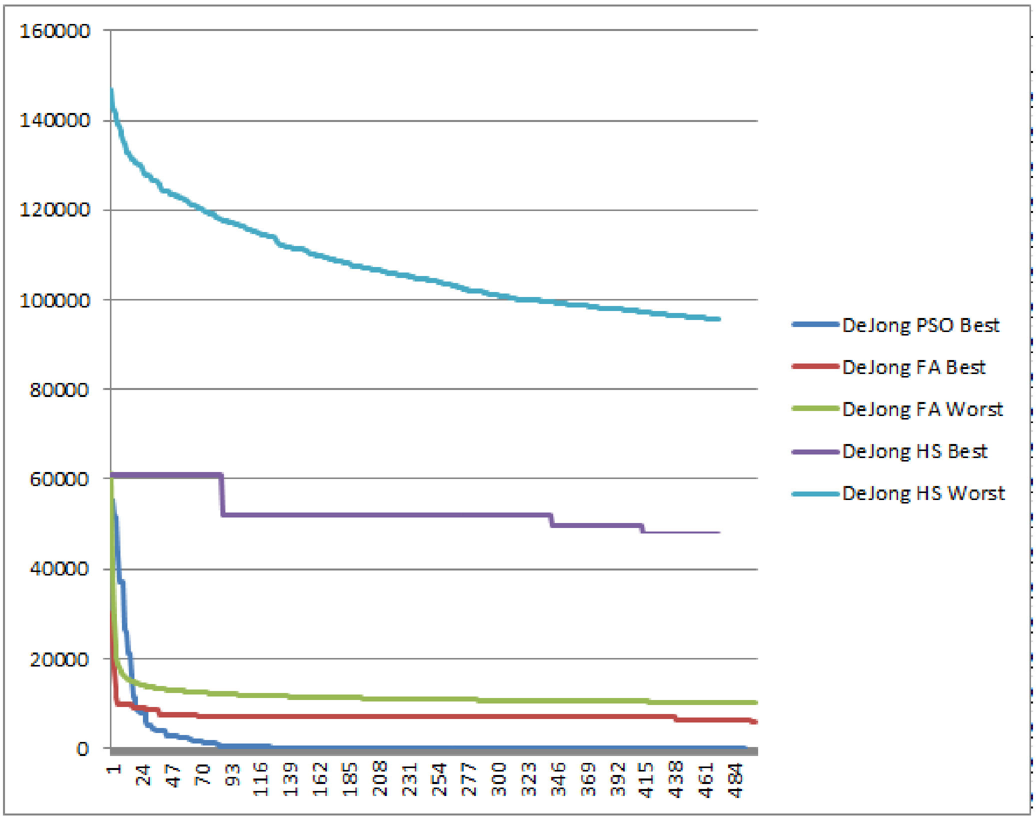

Figure 2.20: Graph of Best and Worst Results on DeJong's Sphere Function

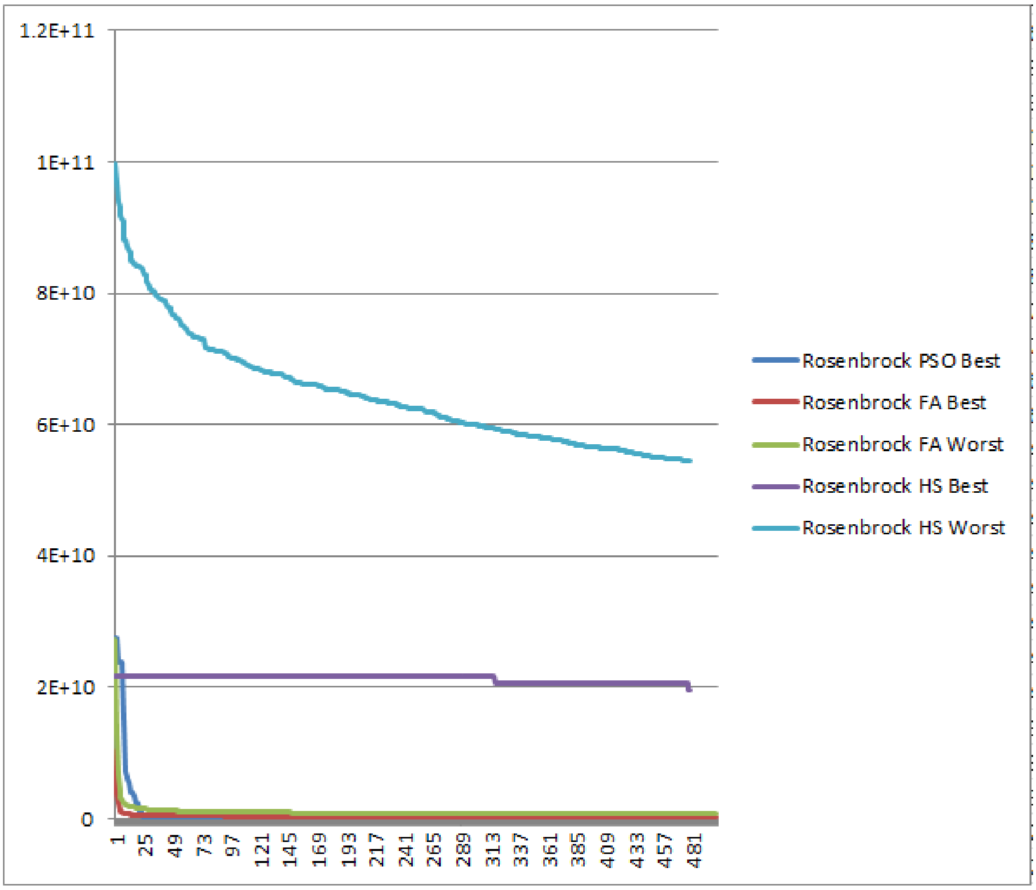

Figure 2.21: Graph of Best and Worst Results on the Rosenbrock's Saddle Function

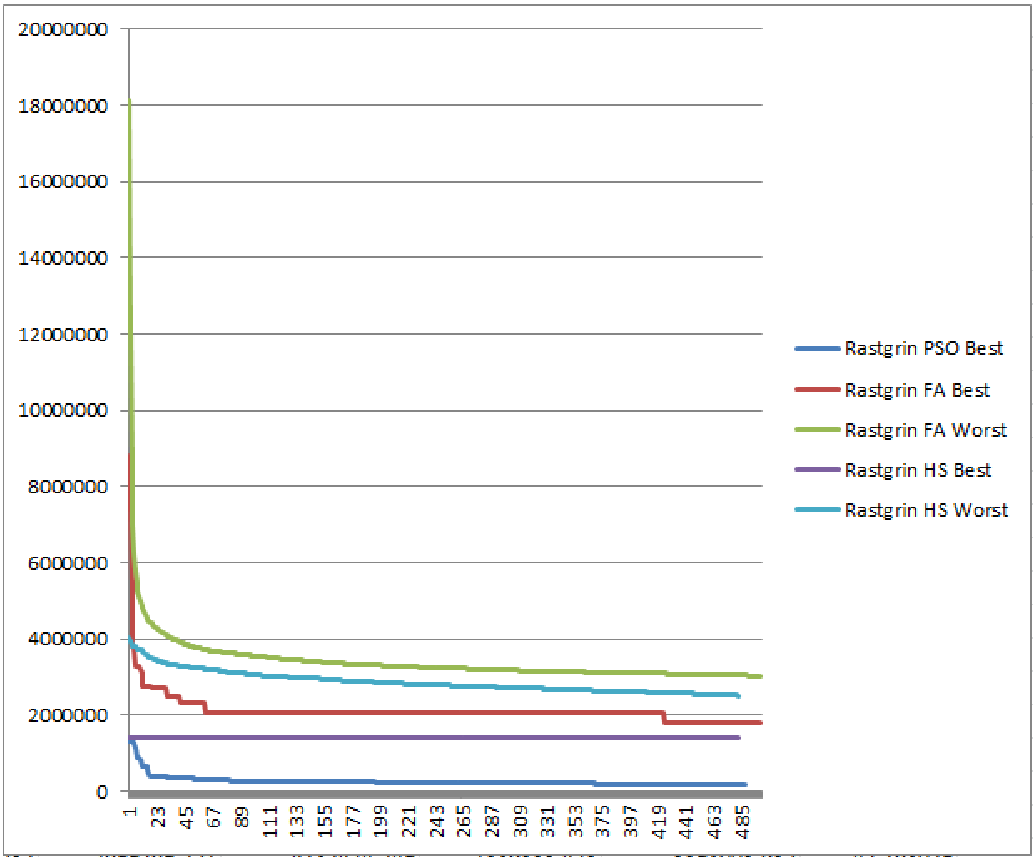

Figure 2.22: Graph of Best and Worst Results on Rastgrin's Function

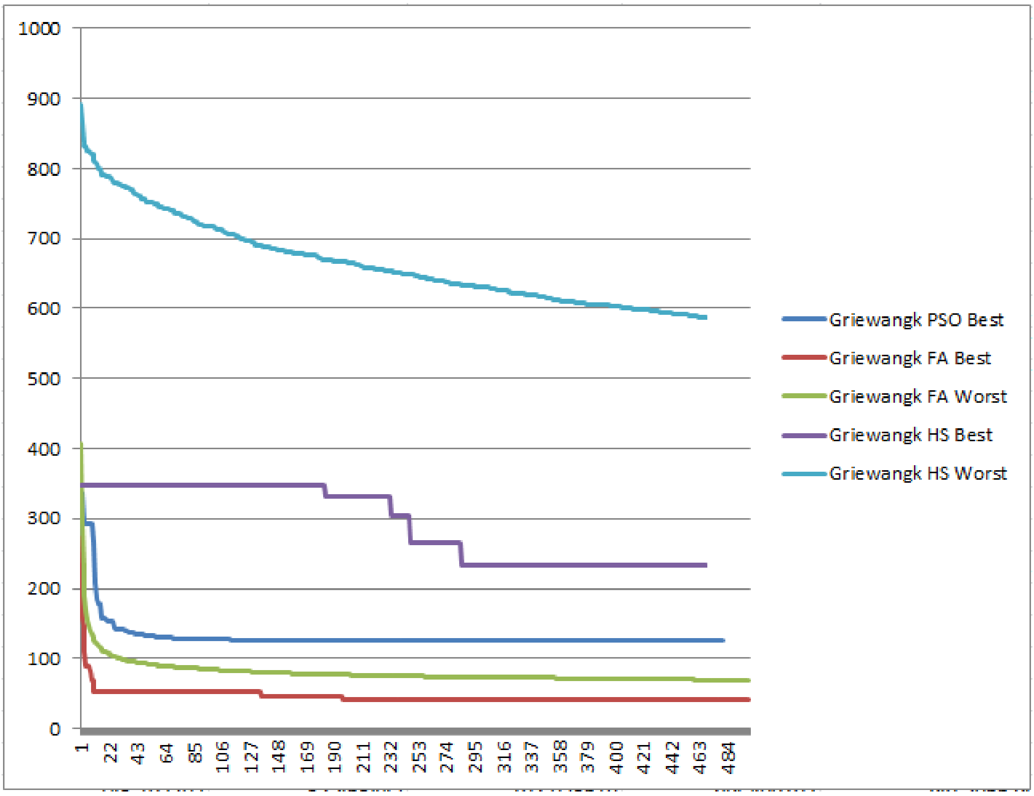

Figure 2.23: Graph of Best and Worst Results on Griewangk's Function

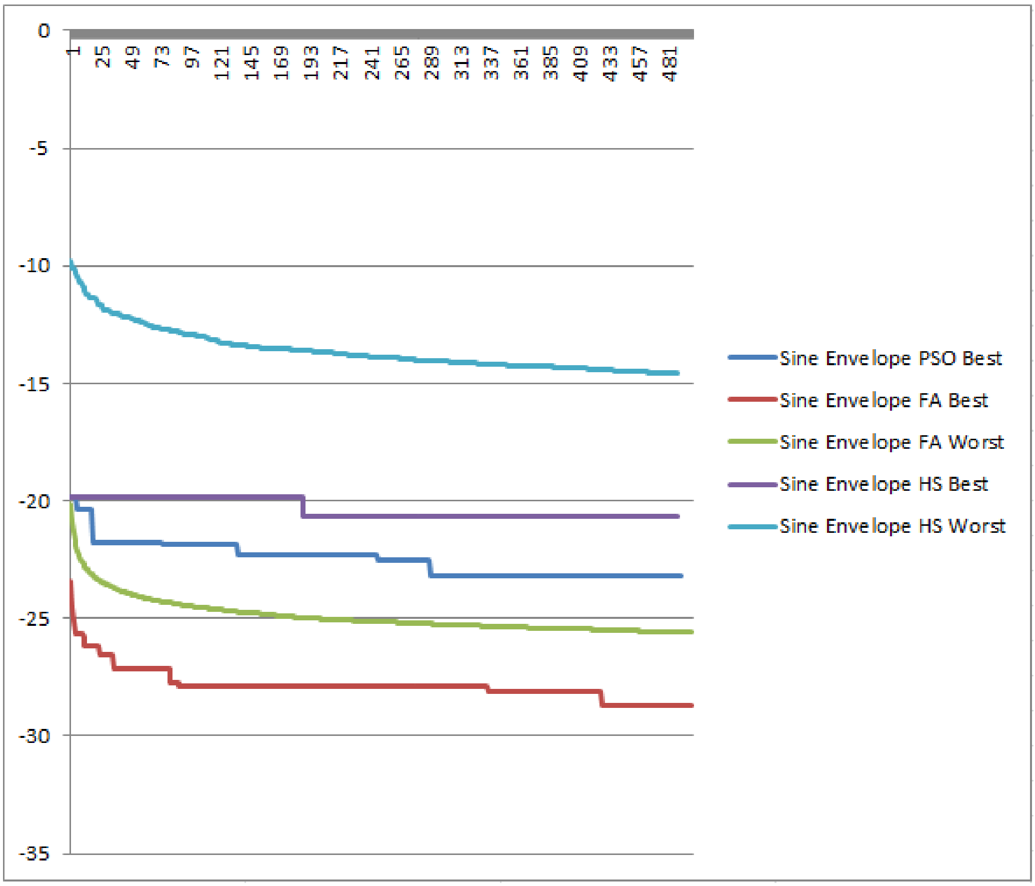

Figure 2.24: Graph of Best and Worst Results on the Sine Envelope Sine Wave Function

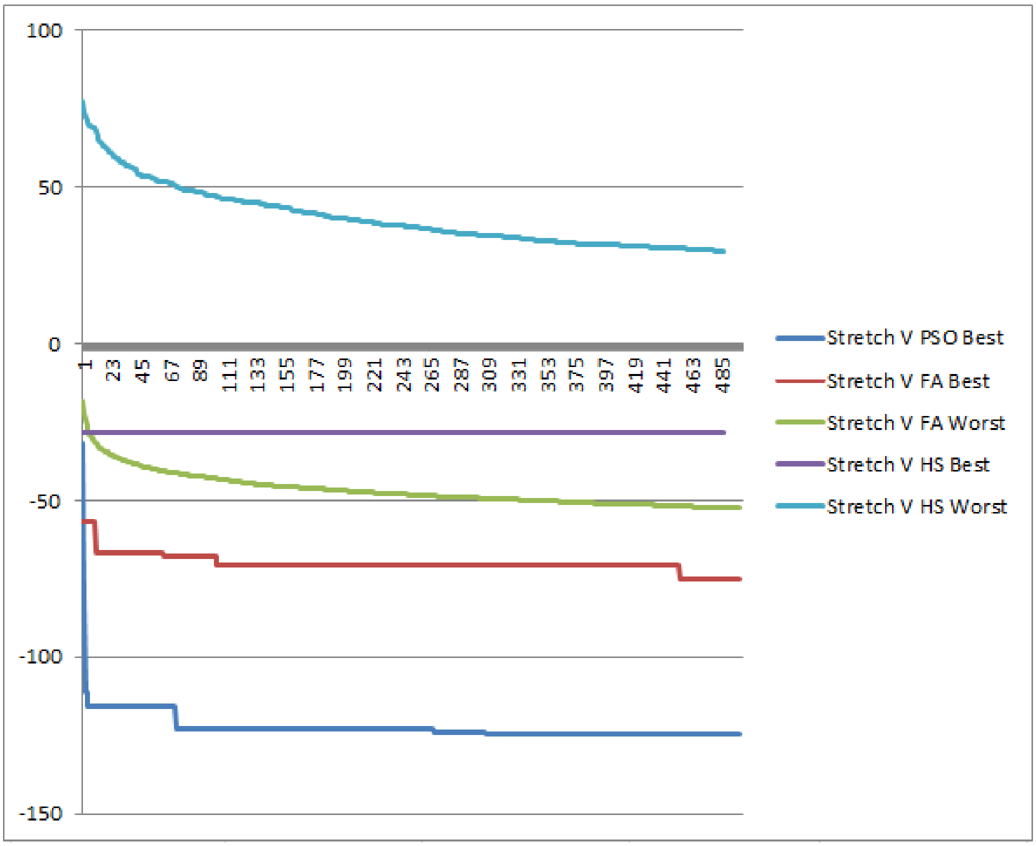

Figure 2.25: Graph of Best and Worst Results on the Stretch V Sine Wave Function

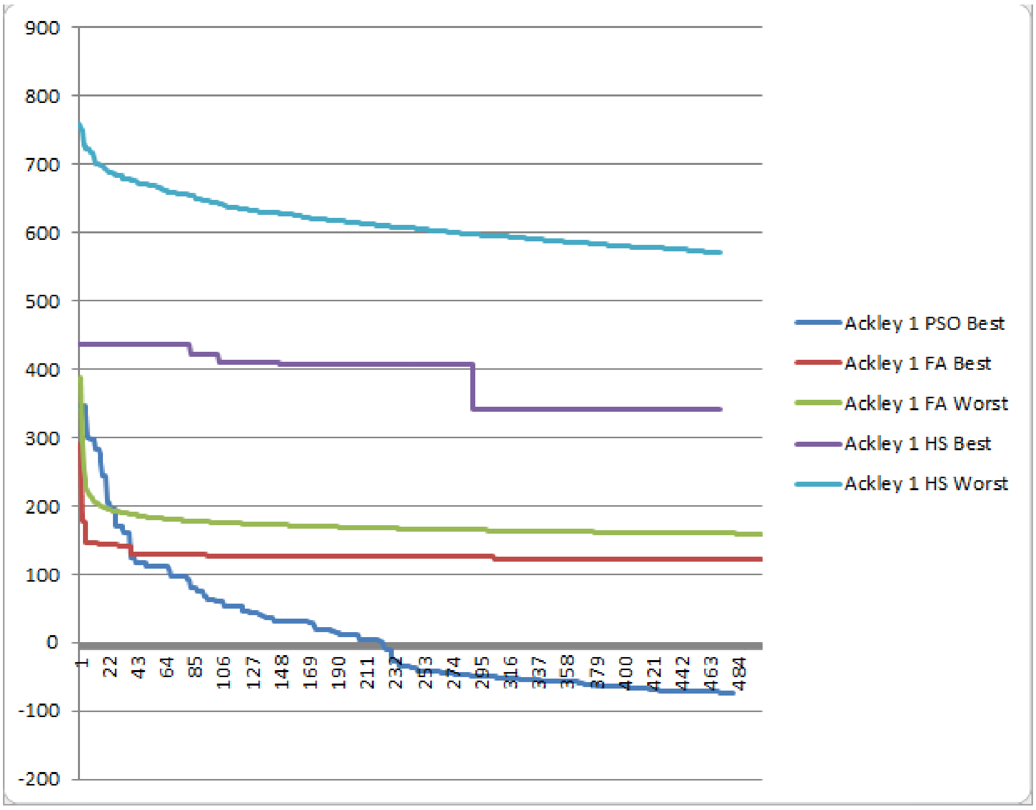

Figure 2.26: Graph of Population's Standard Deviation on the Ackley One Function

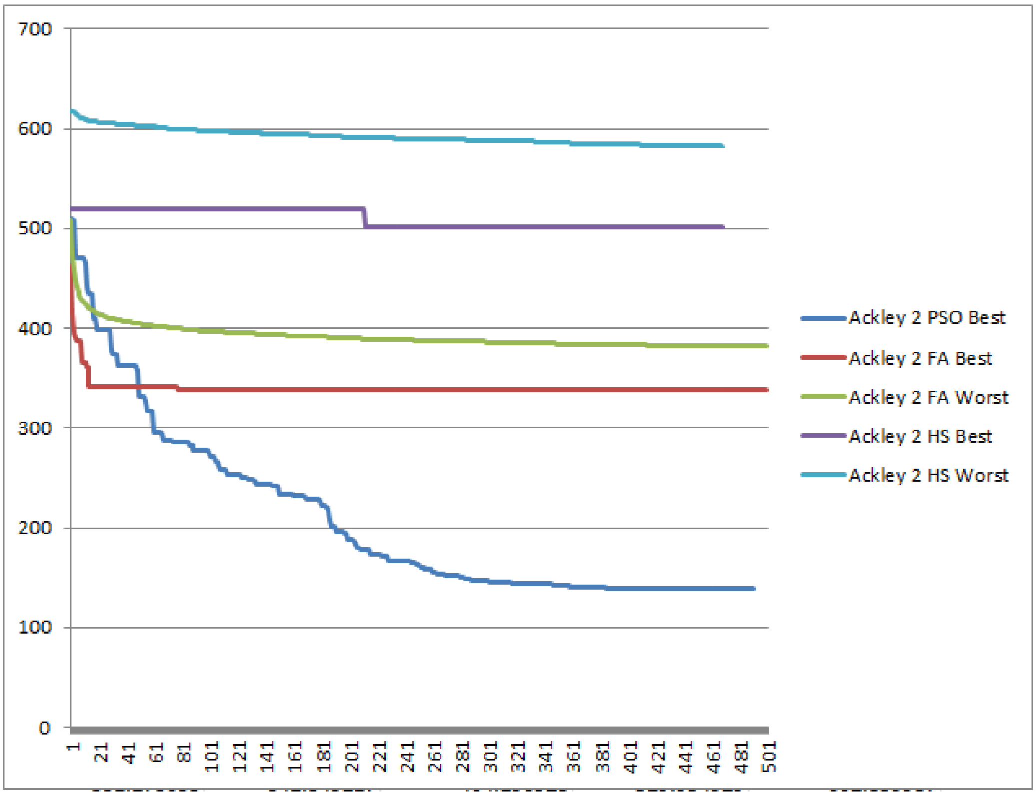

Figure 2.27: Graph of Best and Worst Results on the Ackley Two Function

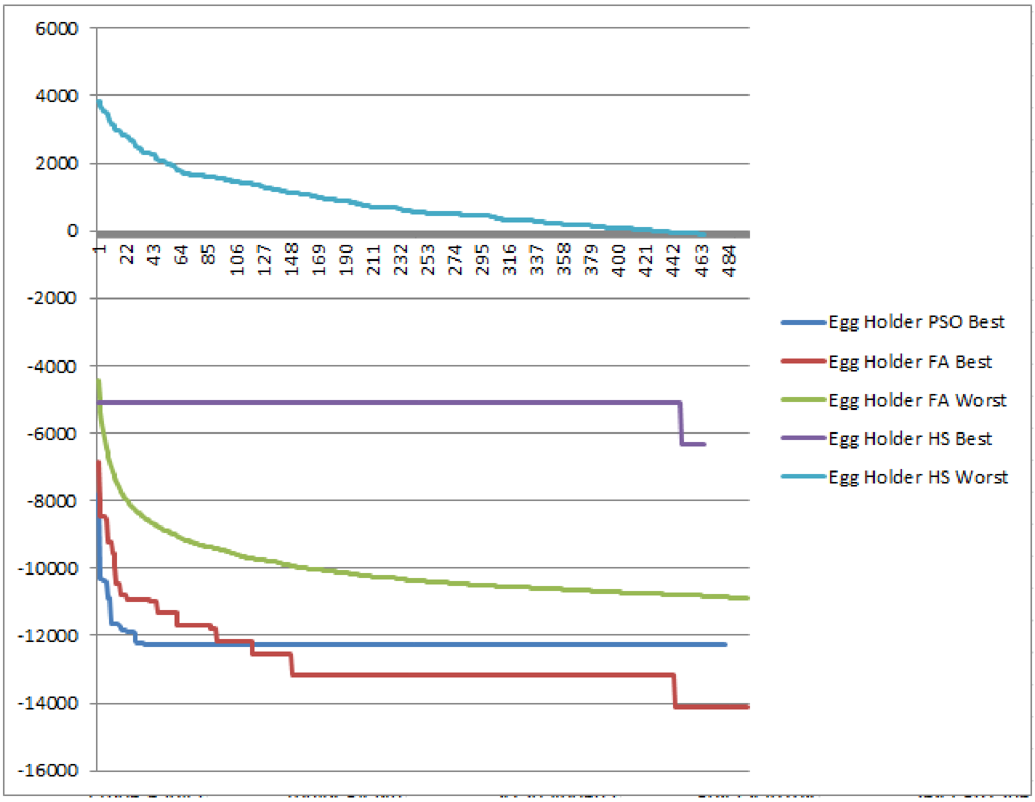

Figure 2.28: Graph of Best and Worst Results on the Egg Holder Function

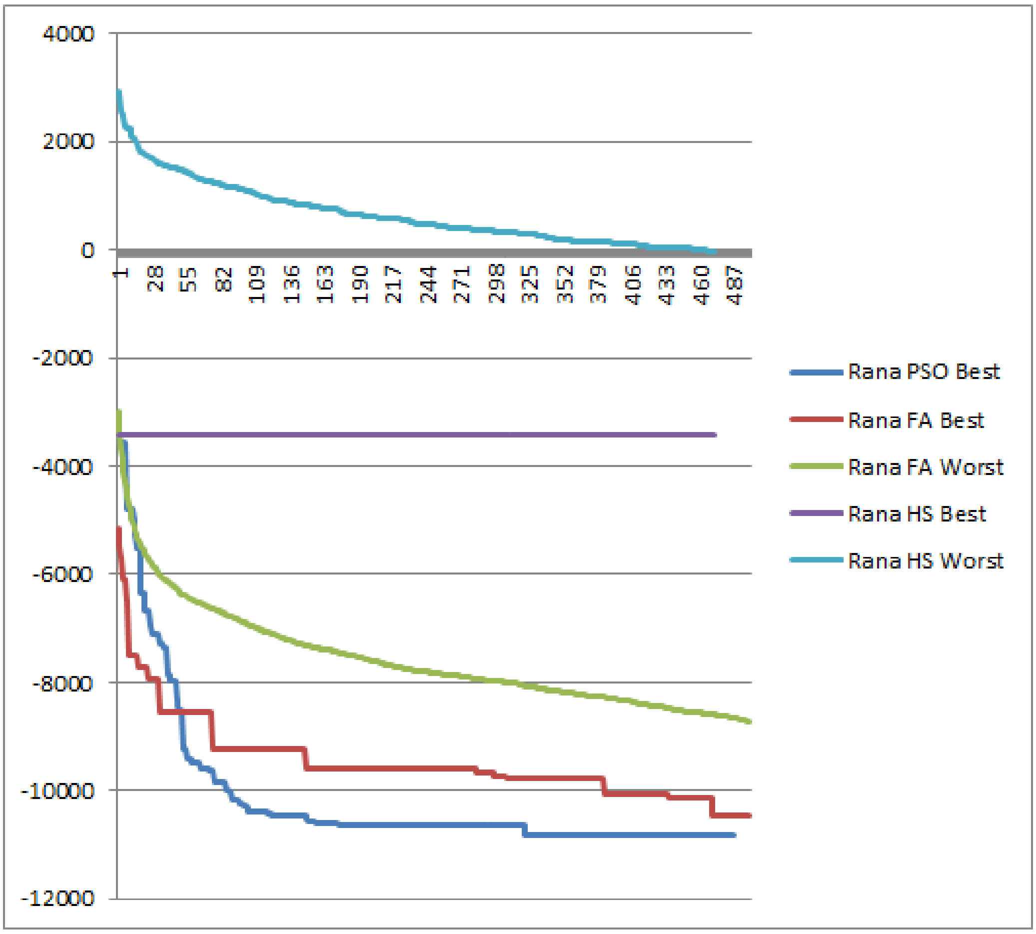

Figure 2.29: Graph of Best and Worst Results on Rana's Function

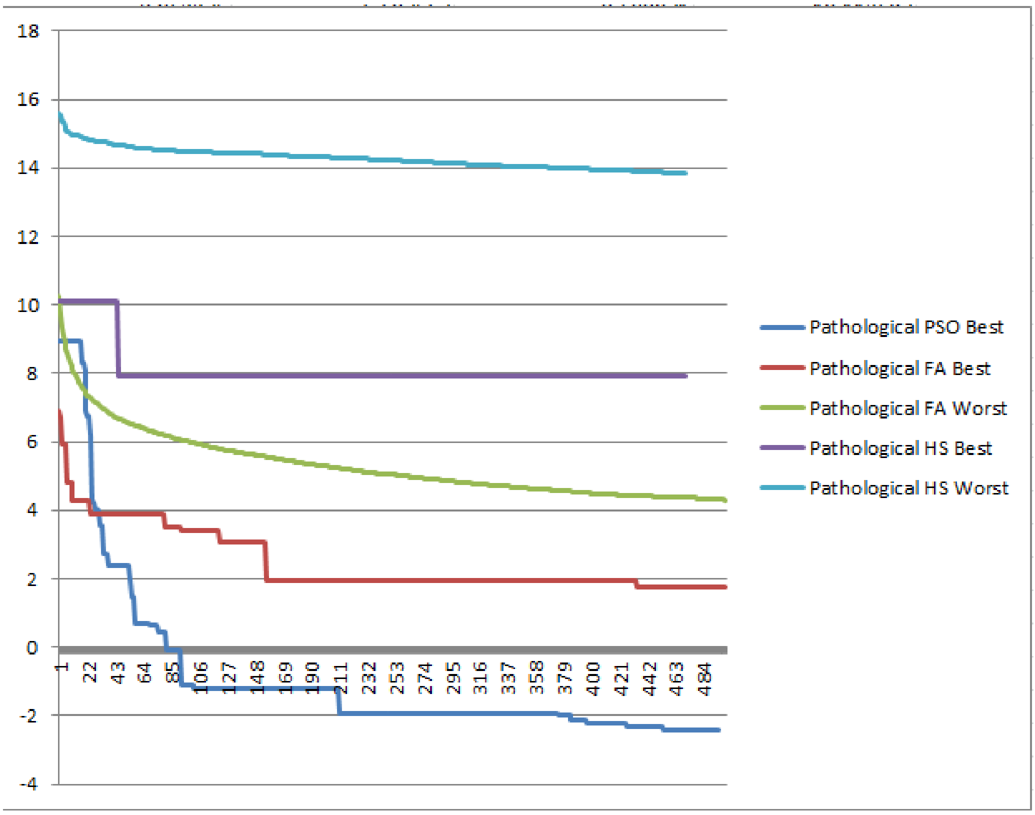

Figure 2.30: Graph of Best and Worst Results on a Pathological Function

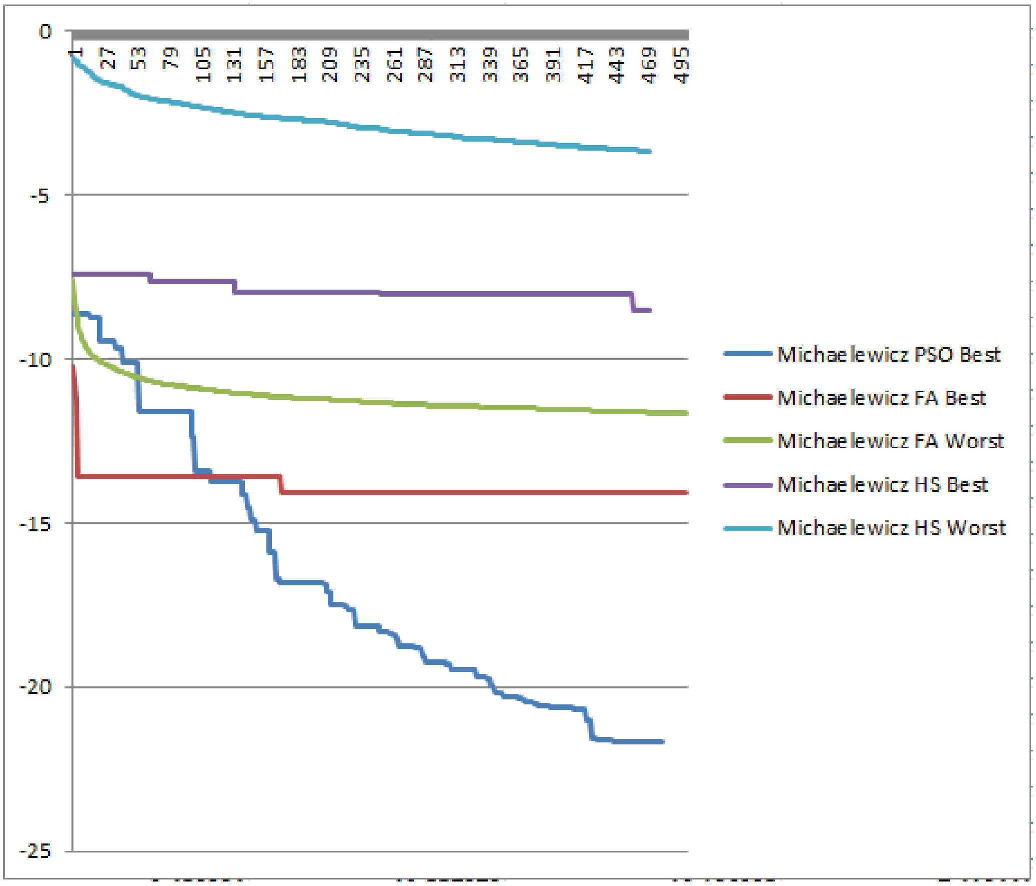

Figure 2.31: Graph of Best and Worst Results on the Michalewicz's Function

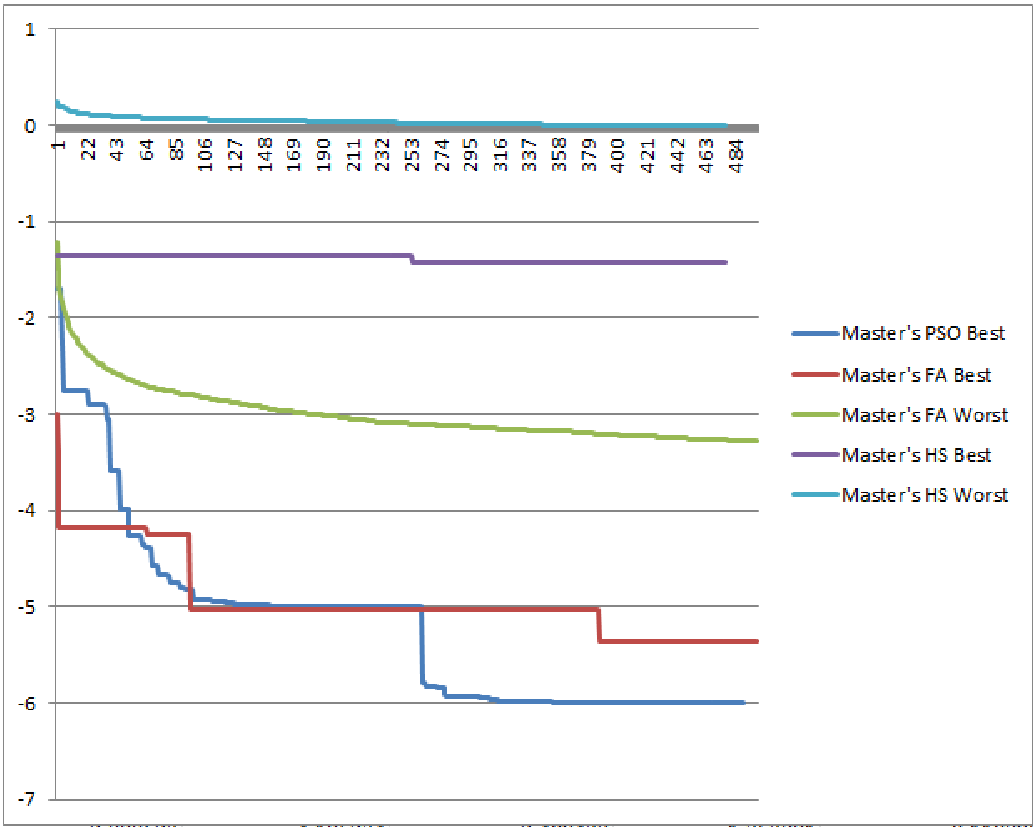

Figure 2.32: Graph of Best and Worst Results on the Masters' Cosine Wave Function

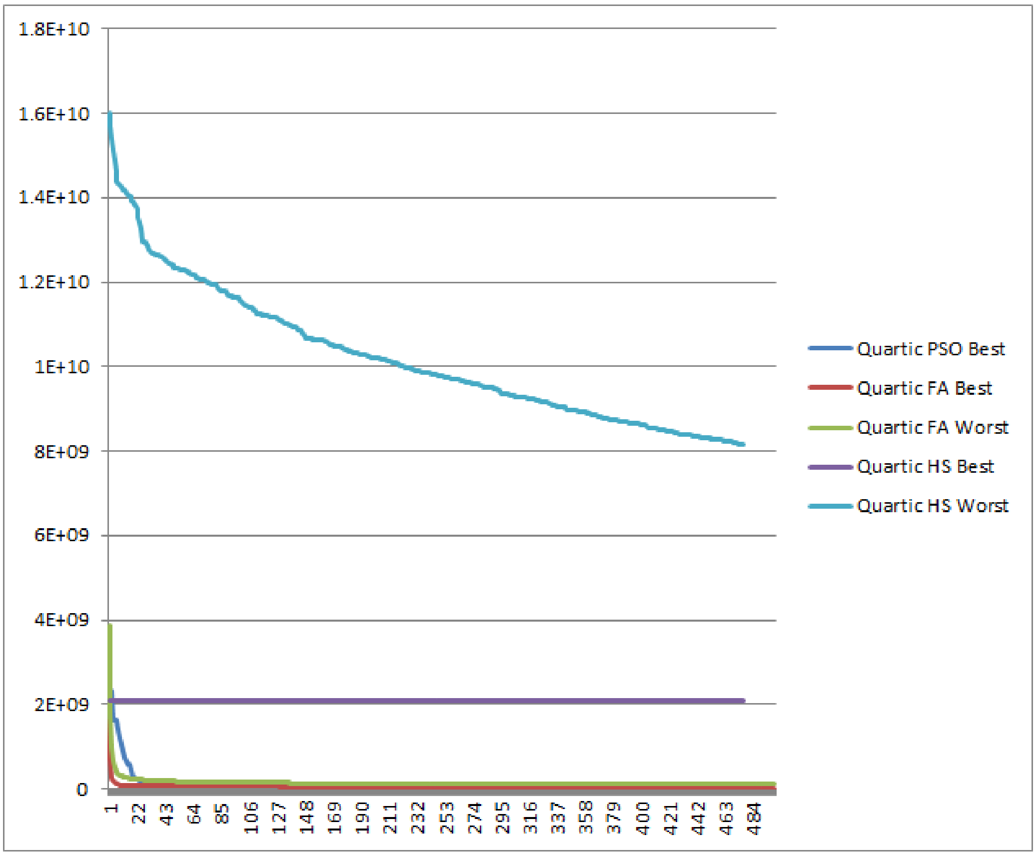

Figure 2.33: Graph of Best and Worst Results on the Quartic Function

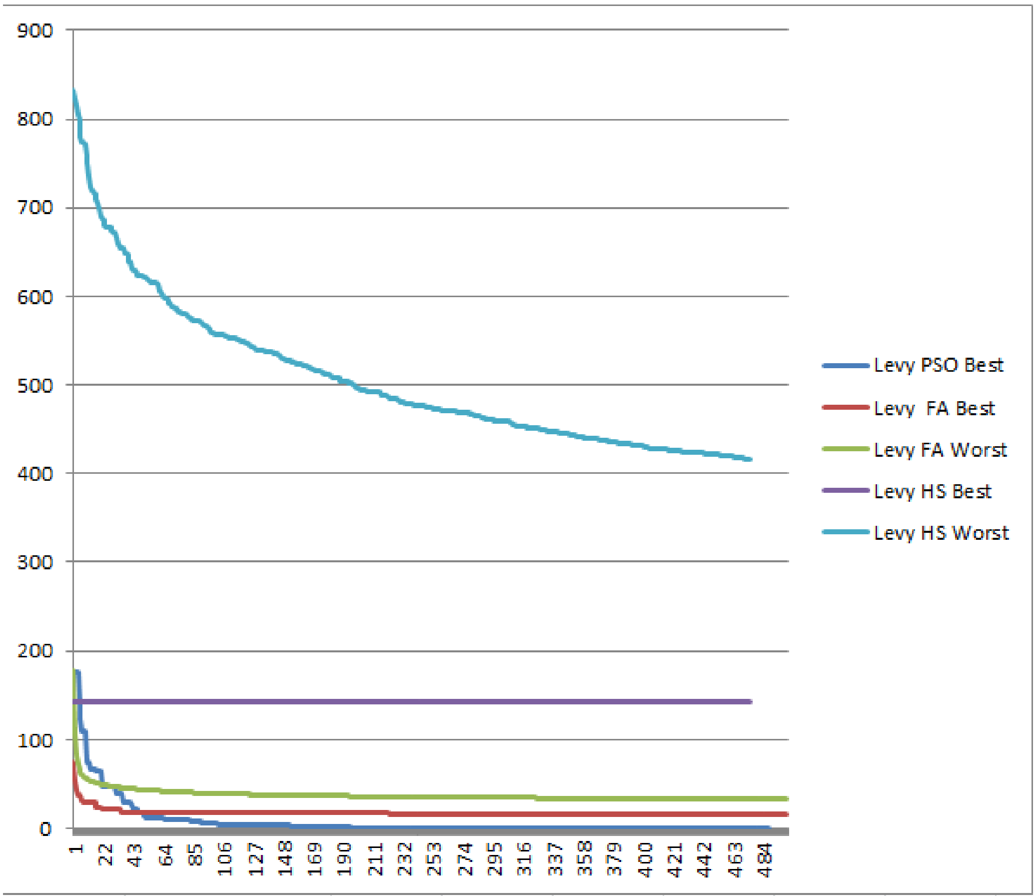

Figure 2.34: Graph of Best and Worst Results on the Levy Function

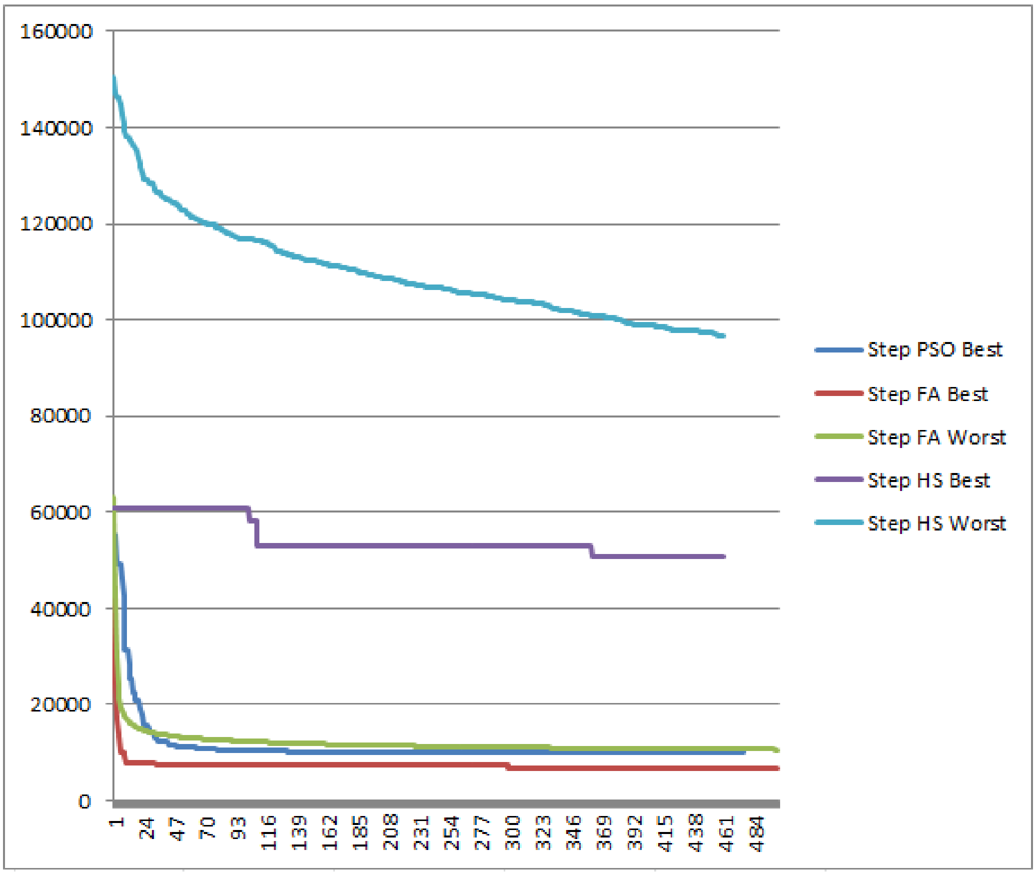

Figure 2.35: Graph of Best and Worst Results on the Step Function

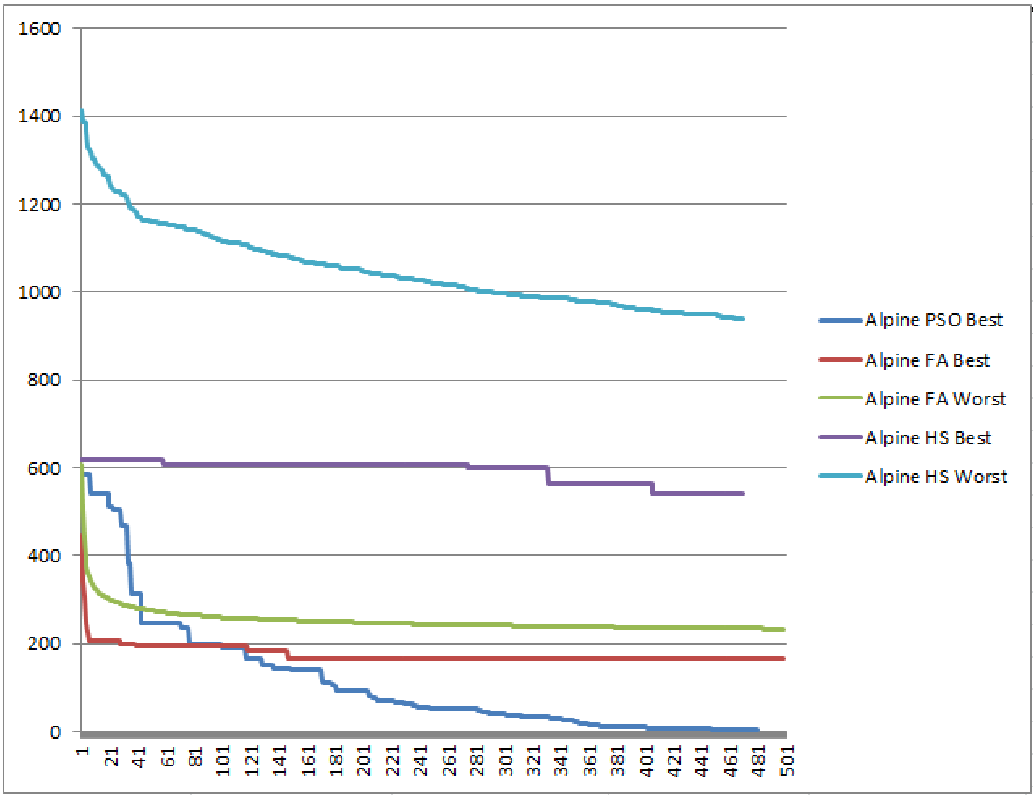

Figure 2.36: Graph of Best and Worst Results on the Alpine Function

2.3 Data Tables for the Different Algorithms

The following tables represent the calculated statistics for the best results over the course of 30 experiments utilizing the 3 different algorithms included in this exercise. Tables for Genetic Algorithm from the previous experimentation set have been included for later comparison.

Parameters for the larger experiments were 500 vectors/particles/fireflies and 30 dimensions per member of the population over 500 iterations. For PSO a dampener value of 0.9 was used, c1 modification value of 0.8, and c2 mod value of 1.2. For the Firefly Algorithm a gamma of 0.00001, alpha of 0.5, and beta of 0.2 were used. Finally, for the Harmonic Search Algorithm the HMCR value was 0.85, PAR value was 0.5, and bandwidth was 0.2.

Table 2.1: Genetic Algorithm: Roulette Selection

Problem	Name	Avg	Median	SD[1]	MIN	MAX	T(ms)[2]
f_1	Schwefel	7745.208601	7821.598315	585.198527	6542.93324	9060.140119	5477.247124
f_2	DeJong	38496.90217	38905.92924	5808.677609	22305.63986	50405.27316	5395.117498
f_3	Rosenbrock	12389233471	12567950535	3198662198	6575845331	18807847531	5512.483954
f_4	Rastgrin	1016428.958	1016973.581	138592.4632	719399.0649	1412256.443	5512.41334
f_5	Griewangk	233.200059	234.493827	36.51397882	166.51781	309.43273	5575.290934
f_6	Sine Envelope	-22.83768532	-22.4910415	1.323387222	-27.065237	-20.744985	5636.88921
f_7	Stretch V	-46.58762486	-44.8184085	10.19520764	-71.598765	-29.565998	5773.137068
f_8	Ackley One	315.7318962	321.883895	33.9078621	215.773062	384.174314	5651.828802
f_9	Ackley Two	461.4083916	461.6662015	17.09454301	425.875129	505.175334	5737.326944
f_{10}	Egg Holder	-7063.380974	-6931.535477	753.420965	-9064.174757	-5491.432159	5607.63025
f_{11}	Rana	-51336827919	-4466.500434	3.63006E+11	-2.56684E+12	-3370.638173	5726.845072
f_{12}	Pathological	7.1396394	7.3063085	1.086677879	4.523383	9.642612	5602.102622
f_{13}	Michalewicz	-10.14559394	-10.0425395	0.895675855	-12.714478	-8.203945	5727.53786
f_{14}	Master's	-2.77469218	-2.6153055	0.666197481	-4.711265	-1.558031	5742.770092
f_{15}	Quartic	1651069585	1667874575	457652121.4	602168026.9	2959596644	5651.218224
f_{16}	Levy	122.0204549	119.1675365	21.85124427	75.280712	177.171476	5633.387432
f_{17}	Step	38573.39704	39852.1505	5972.835031	23533.89446	49133.32123	5413.764398
f_{18}	Alpine	432.5746177	429.3380295	53.17849617	305.440251	551.162807	5514.516172

[1] Standard Deviation

[2] Time in milliseconds

Table 2.2: Genetic Algorithm: Tournament Selection

Problem	Name	Avg	Median	SD[1]	MIN	MAX	T(ms)[2]
f_1	Schwefel	-5551384325	3504.238328	39254240013	-2.77569E+11	4602.713178	3029.173118
f_2	DeJong	1886.106357	1842.173892	797.9654336	595.255232	3949.414946	2956.303308
f_3	Rosenbrock	81253898.58	59284594.76	87181476.39	10736929.48	560601965.2	2991.929045
f_4	Rastgrin	161005.1972	149900.4538	56038.32774	49782.26836	284470.6074	3065.888641
f_5	Griewangk	12.11860867	11.39462	4.908838396	5.623645	33.829199	3084.233227
f_6	Sine Envelope	-29.89528918	-30.1379845	1.410649189	-32.916683	-26.363354	3090.152692
f_7	Stretch V	-70421.91994	-93.2078585	444713.7155	-3128596.869	-74.111994	3223.479584
f_8	Ackley One	100.8461479	97.44202	31.93182992	46.556566	175.287111	3116.127759
f_9	Ackley Two	252.0672195	251.7592415	29.40687298	193.279309	331.58181	3156.999962
f_{10}	Egg Holder	-20345285630	-13156.38587	1.34204E+11	-9.47838E+11	-10007.75505	3136.912786
f_{11}	Rana	-1.78754E+11	-7844.805398	1.05187E+12	-7.28592E+12	-6833.853254	3140.059454
f_{12}	Pathological	-0.84991084	-0.8489985	1.832626302	-4.791914	2.949278	3149.572782
f_{13}	Michalewicz	-18.2258054	-18.277636	1.651568326	-21.848146	-15.071968	3201.07295
f_{14}	Master's	-6.15621514	-6.070449	1.508754213	-10.312894	-3.698482	3180.676742
f_{15}	Quartic	13227339.31	11252370.99	8924880.276	1742117.26	38654794.95	3145.908516
f_{16}	Levy	19.33085432	19.05874	6.840897914	7.362337	42.598749	3179.684068
f_{17}	Step	4737.798545	1445.175496	21469.58545	628.43733	153431.3177	3041.207088
f_{18}	Alpine	168.7708095	163.697625	34.59361242	88.035111	238.031935	3060.335514

[1] Standard Deviation

[2] Time in milliseconds

Table 2.3: Particle Swarm Optimization

Problem	Name	Avg	Median	SD[1]	MIN	MAX
f_1	Schwefel	3348.340797	3327.849117	112.8400762	2983.588903	3442.602522
f_2	DeJong	2333.852103	0.0233555	4302.753991	0.002048	10014.1931
f_3	Rosenbrock	435039.4556	10218.54511	502577.8949	93.753496	1000192.747
f_4	Rastgrin	-26195.42957	-71093.35533	102758.9392	-79278.26182	198996.1851
f_5	Griewangk	17.42105238	0.039538	37.16188334	0.006726	125.880769
f_6	Sine Envelope	-23.8506897	-23.6915685	0.93697931	-25.951107	-22.633675
f_7	Stretch V	-120.4546195	-119.659941	2.915991143	-126.004617	-115.470843
f_8	Ackley One	-8.043244759	-10.288174	47.19259189	-76.151902	108.935494
f_9	Ackley Two	286.9081571	298.1187895	62.33486186	138.306124	378.050961
f_{10}	Egg Holder	-13705.2482	-13526.00843	1341.134895	-17178.22144	-11646.96818
f_{11}	Rana	-11967.95165	-11849.7438	768.77457	-13499.61036	-10837.96074
f_{12}	Pathological	-6.867101345	-7.392273	2.582428828	-12.147095	0.796715
f_{13}	Michalewicz	-19.01532007	-19.544876	1.624418499	-21.687531	-14.777885
f_{14}	Master's	-7.0468629	-6.999843	2.440578762	-11.993983	-3
f_{15}	Quartic	63333336.13	2.0610135	106619959.5	0.074897	400000000.7
f_{16}	Levy	6.830728679	5.369005	6.168956481	0.001839	29.463954
f_{17}	Step	3375.623053	8.0825095	4841.768503	7.660513	10108.12574
f_{18}	Alpine	118.7867939	112.776548	64.76804571	4.47306	308.479323

[1] Standard Deviation

Table 2.4: Firefly Algorithm

Problem	Name	Avg	Median	SD[1]	MIN	MAX
f_1	Schwefel	4475.811525	4449.304972	186.570916	4124.511937	4819.86945
f_2	DeJong	6212.103386	6225.878103	532.7854076	4781.480173	7151.047208
f_3	Rosenbrock	276036229.9	272820253.6	41940562.25	138378984	340356038.3
f_4	Rastgrin	1845412.569	1857714.618	150883.9619	1506615.346	2158028.369
f_5	Griewangk	41.40228967	40.7080715	3.542409896	34.682773	47.910547
f_6	Sine Envelope	-28.9496863	-28.7699295	0.616619494	-30.369427	-28.115244
f_7	Stretch V	-90.59592667	-89.3298995	7.853855037	-104.506179	-75.005153
f_8	Ackley One	114.2570529	115.339449	6.897989698	92.796064	122.549045
f_9	Ackley Two	333.1675513	334.480037	8.27843333	306.558624	347.278637
f_{10}	Egg Holder	-13526.05704	-13537.85167	585.5624455	-15099.76994	-12416.25589
f_{11}	Rana	-10674.38514	-10664.26561	453.40773	-11563.49368	-9959.922455
f_{12}	Pathological	0.5261133	0.5219245	0.612357617	-0.438013	1.770668
f_{13}	Michalewicz	-14.1479153	-14.0976695	0.345548317	-15.054818	-13.64719
f_{14}	Master's	-5.100589333	-5.0801685	0.308486325	-5.832743	-4.627831
f_{15}	Quartic	37808788.47	38471454.74	7378328.773	19992534.38	50490252.69
f_{16}	Levy	16.727392	16.5802025	2.080217875	11.953868	19.942368
f_{17}	Step	6690.649959	6624.910492	506.6035863	5823.076181	7499.093095
f_{18}	Alpine	165.9993939	165.722814	7.290377298	145.560375	179.425633

[1] Standard Deviation

Table 2.5: Harmonic Search Algorithm

Problem	Name	Avg	Median	SD[1]	MIN	MAX
f_1	Schwefel	8699.998266	8710.967303	356.8864217	7821.546249	9284.022249
f_2	DeJong	45946.56328	47153.93992	3491.479926	39493.78002	51717.06845
f_3	Rosenbrock	16639955352	16531859315	2664777747	12329551125	21285725881
f_4	Rastgrin	1279646.247	1304890.764	99360.47959	1093779.577	1408388.273
f_5	Griewangk	290.0026059	289.938123	31.89936533	212.019808	335.416919
f_6	Sine Envelope	-20.7114769	-20.5445765	0.671184558	-22.163271	-19.748321
f_7	Stretch V	-30.19123693	-29.942635	5.567231571	-42.928823	-18.136013
f_8	Ackley One	362.7486429	365.6517655	21.26584982	296.125381	391.628829
f_9	Ackley Two	501.611883	504.213054	11.82022296	473.659507	520.880913
f_{10}	Egg Holder	-5499.038967	-5487.663365	544.7878981	-7016.731541	-4564.922182
f_{11}	Rana	-3524.753566	-3477.657349	412.3813306	-4545.092943	-2953.01799
f_{12}	Pathological	8.934109207	8.958444	0.698482676	7.76862	9.900386
f_{13}	Michalewicz	-8.976763034	-8.992635	0.683655661	-10.466247	-7.325708
f_{14}	Master's	-1.679045897	-1.701838	0.283054813	-2.162381	-1.098165
f_{15}	Quartic	2231901068	2218037090	393663420.4	1360792138	3105148139
f_{16}	Levy	159.3011938	161.358064	16.25737617	124.682432	190.358724
f_{17}	Step	46128.95071	47530.29187	5150.012865	35600.20527	52752.81387
f_{18}	Alpine	490.6515822	487.305424	40.28485754	425.393233	587.053306

[1] Standard Deviation

3 Conclusion

3.1 Standard Deviation of the Population

Beginning the summary of our results is a review of the standard deviations across populations in PSO, FA, and HS. For this summary please reference Figures 2.1 - 2.18. In terms of stagnation, PSO would stagnate the most out of all algorithms, oftentimes reaching 0 or almost 0 standard deviation in the population for all dimensions. However, occasionally there were large spikes upwards in a few dimensions which corresponded to steps in the best worst graphs of the related functions. This implies that certain dimensions could have very influential effects on the results. Firefly would generally have the second lowest standard deviation though it would often start with a high standard of deviation then quickly drop and settle into a mid range. There were some occurrences, see Figures 2.1, 2.7, 2.10, 2.11, and 2.12, where the standard deviation would wildly fluctuate however and some dimensions would get more diverse while others would get smaller followed by wide swings larger and smaller across the iterations.

Generally this would occur later in the iterations and as can be seen from Figures 2.19, 2.26, 2.29, 2.30, and 2.31 this would coincide with large periods of static best results. The intermediary fireflies would be pulled in so many random directions from good solutions that a wider variety of clustered search spaces were being explored. Finally, Harmony generally had the highest standard deviation in its population and would maintain this high amount of deviation throughout the experiment, occasionally increasing but mostly only slightly fluctuating. This makes sense considering the process of HS. Since all dimensions are chosen at random from an originally diverse population, and in comparison very few new vectors are developed so the population maintains its randomness and diversity and rarely converges on a single vector or a single dimensional value as long as there is even distribution in the random number generator being used.

3.2 Best and Worst Results Graphs

Analyzing Figures 2.19 - 2.36, it can be concluded that with the originally provided general parameters PSO gave the best results though in several cases, Figures 2.23, 2.24, and 2.35, FA provided the best results. Meanwhile, the best results of Harmonic rarely ever improved though the worst results of Firefly and Harmonic steadily improved over the iterations thereby narrowing the potential search space for the best results of the function. While this serves to help narrow the search space it cannot guarantee the best values being achieved and tuning was necessary to provide better results. In general however, PSO and FA have given me the best results I have seen from any of the current optimization algorithms implemented and I am surprised at their level of performance. These results are drastically better, especially for those functions which were normally fairly unresponsive to the other methodologies, such as Sine Envelope, Stretch V, Rastgrin, and Rosenbrock Saddle, and in some cases beat out the responsive functions such as Griewangk, DeJong's Sphere, Ackley One, Levy, and Alpine. With these general parameters, there were still many cases of long periods where the best result wouldn't improve for FA, and occasionally PSO.

3.3 Data Table Analysis

The first thing that should be noticed in tables 2.3 - 2.5 is a generally reduced standard deviation across all functions for these algorithms. This implies a more consistent set of results with these parameters and reproducible results with similar tuning values less dependent on the starting population. So not only are the best results generally better, they are consistent, which previous testing methodologies were unable to achieve. Some of the more interesting results were that in Particle Swarm Optimization, even though it got the best results, sometimes the results for functions like DeJong, Quartic, Griewangk, and Rosenbrock would hit these plateaus in a run and results would stop at certain thresholds. These thresholds were 10,000, 200,000,000, 63 or 125, and approximately 1,000,000 respectively. This was due to the speed of stagnation in the population, causing the particles to reach a trough and stagnate inside these common troughs.

Another important point is that the supremely large values which were prevalent at the end of other testing methodologies, really weren't that prevalent in these three algorithms. There were no outlier values, and the largest result of $3.403e^8$ in Firefly Algorithm was well below previous maxima $1.88e^{10}$ from Genetic Algorithm. Although Harmonic Search did not perform well with the generic parameters assigned, it certainly has potential with the proper tuning as discussed in the next subsection. Computationally, Harmonic Search is the least intensive and therefore was expected to perform the worst. PSO was the next most intensive and took some time to run all experiments for all equations but Firefly Algorithm was clearly the leader in this attribute as it took over a day to run and performs an astounding number of iterations at 125 million iterations per experiment.

3.4 Tuning

First, the disappointing tuning results. With Harmonic Search adjusting all of the various values in different ways didn't affect the results. The worst result of each run would get a lot better over the course of the iterations, but the best result would rarely if ever change. This was trying large adjustment and consideration rates with large and small bandwidths, small adjustment and consideration rates with different bandwidths, and one large rate, one small rat and different bandwidths. This algorithm would need more iterations as it isn't computationally intensive, as the larger tests with 500 iterations compared to these smaller 100 iteration tests produced better results with general parameters. Another update would be tweaking the way the Harmonic Search functions. Another student was getting good improving results by checking the HMCR first, otherwise checking the PAR next, or finally getting a random value rather than pitch adjusting a random member of the population. As a sidenote, his experiments used a higher PAR than HMCR and a bandwidth of 0.2.

Firefly algorithm depends more heavily on its parameters than Harmonic Search however. Our original gamma value of 1 was producing no changes in the results and the population was never changing. The reason for this is that the distances were so large that the light intensity of both fireflies was effectively 0 meaning no movement would occur. Thus the gamma was decreased to 0.00001 for the general tests. However, the smaller gamma the larger our steps in attractiveness will be, and for certain ranges, this would cause the population to behave in odd manners. For those equations with smaller overall value ranges a larger gamma produces better results whereas a smaller gamma for those with larger ranges. The beta or attractiveness value originally given was 0.2 and performed best from this to approximately 0.5 for all the equations. This is because it controlled how close to these better fireflies we would move and without traveling too far, as higher betas caused faster stagnation, the search spaces were well explored. Too small of a value and no progress was made in the small number of iterations tested. The final parameter was alpha which determined the amount of randomness added to the firefly's movement. Too large a value and it wouldn't move within the proper search space flying off into other spaces. The best values for this were actually small values of around 0.01 which drastically reduced the randomness in the firefly's movement and directed it closer to the better search spaces.

Finally, PSO performed best when the C2 was greater than C1 as it weighted the global best dimensional value higher but when C1 was too small the results would stagnate faster. The dampener of 0.9 was good but a smaller value contributed to more consistent result improvement though less improvement overall was achieved. When the value was too high, the population would hit the bounds of the range much faster. There has since been improvement to the Particle Swarm algorithm which will reset the velocity should it get too high rather than simply dampening the velocity. This implementation would provide some more diversity in the population.

3.5 Comparison to Genetic Algorithm

There were two main features of these results which stood out from Genetic Algorithm. First, the populations for these algorithms didn't stagnate anywhere near as fast as in Genetic Algorithm. While PSO was certainly comparable, the amount of deviation in the dimensions for FA and HS maintained a diverse population and the vectors being considered in PSO did not quickly become the same vector because certain dimensions spread throughout the population would still have some slight variance, rather than the large blocks of dimensions in Genetic Algorithm which remained the same and would quickly dominate the population. The second feature of these results was the wide range of results which were better than Genetic Algorithm for almost every objective function. Tournament Select did provide some good results for very specific functions but was not generally applicable to all of the objective functions, whereas for PSO and Firefly, the results achieved for all objective functions was better. Roulette Select relied on too much randomization and the distribution of the fitness in the population needed to be more random to provide better results as with the implemented sorted population, even distribution of the random number generator rarely hits the extrema, so the best and worst vectors were rarely being considered whereas the middle of the population was being considered most providing little improvement. In contrast to this, PSO and FA were constantly moving the dimensions of the vectors towards the best or better solutions and so a wider search space was being considered and the areas around the best results were being explored.