Central Washington University

CS471 Optimization

Winter 2020

Genetic Algorithm, Differential Evolution, Particle Swarm Optimization, Sine Cosine Algorithm

Student:
Chao Huang Lin
chao.huanglin@cwu.edu

Professor:
Dr. Donald Davendra
Donald.Davendra@cwu.edu

February 18, 2020

Contents

1 Introduction

For this project, four optimization algorithms will be tested. These are Genetic Algorithm (GA) and Differential Evolution Algorithm (DE), Particle Swarm Optimization (PSO) and Sine Cosine Algorithm (SCA).

The GA is a heuristic search and optimization technique that simulates the process of natural evolution. The main operations of GA are Selection, Crossover, Mutation, and Elitism.

The DE algorithm employs the difference of two randomly selected parameter vectors as the source fo random variations for a third parameter vector. Some advantages of DE are: few numbers of control parameters which make it easier to calibrate or tune, it is inherently parallel, and it has a faster convergence. There are many strategies of DE, in this project 10 different strategies will be tested.

The PSO is inspired by the flocking and schooling patterns of birds and fish, they key point of PSO is that it keeps tracking the personal best and global best variables, and make the particles move toward those best values.

The SCA created in 2015, actually is very similar to PSO. However, it only keeps track only the global optimal value, and in each iteration update it introduces sine and cosine formula which multiply with a decreasing range value.

2 Method

The GA, DE, PSO, SCA algorithms are coded using C++ object-oriented programming. The implemented classes are Population, PopulationBenchmark, Functions, Runner, GeneticAlgorithm, DifferentialEvolution, ParticleSwarm, SineCosine and mt19937ar. Additionally, python script with jupyter notebook is implemented read the configuration parameters and call the C++ executable, then it collects the result, displays it in table and figures, and print the output in latex format.

The obtained results are from 50 runs of each algorithm, comparing to the previous project the population size has increased to **500**, and the generations or iterations increased to **500**, number of dimensions = 30.

The computer used to run the project has the following specification: Intel Core i7-9750H 2.6GHZ with 16 GB of RAM

3 Important Notes

During the development of this project there were many problems that need to be fixed in order to get optimal results:

1) The mutation of GA and the trial of DE can make the data go outside of the range, to solve this problem a function is coded to truncate the values in the range.

2) The project has been run in single processor mode and parallel multiprocessor mode. The results presented in this document are only from a single processor mode since there was some problem with CPU clock time in parallel mode. According to [1] the clock() function in C++ measures the CPU time used by the entire program so other processes or threads that are not part of the algorithm that need measurement also get counted. (This problem might be caused by Python subprocess package)

3) Some DE algorithms did not converge, the cost was very high. This is caused by the parameters which are not calibrated, to solve this problem, the parameters of different algorithms of Differential Evolution had been calibrated manually one by one.

   The calibrated DE parameters are:

DE strategies	crossover rate	scaling factor F	scaling factor lambda
DE_best_1_exp	0.8	0.3	-
DE_rand_1_exp	0.9	0.1	-
DE_randbest_1_exp	0.9	0.4	0.4
DE_best_2_exp	0.9	0.25	-
DE_rand_2_exp	0.9	0.05	-
DE_best_1_bin	0.6	0.5	-
DE_rand_1_bin	0.8	0.2	-
DE_randbest_1_bin	0.7	0.5	0.5
DE_best_2_bin	0.8	0.4	-
DE_rand_2_bin	0.7	0.1	-

Table 1: Configuration parameters of DE obtained by manual tuning

   The calibrated GA parameters are:

crossover rate	mutation rate	mutation value range	mutation precision	elitism rate
0.9	0.05	0.1	1	0.3

Table 2: Configuration parameters of GA obtained by manual tuning

4 Results

The results of applying different optimization algorithms with 18 benchmarking functions are displayed in the following pages:

4.1 Function 1: Schwefel

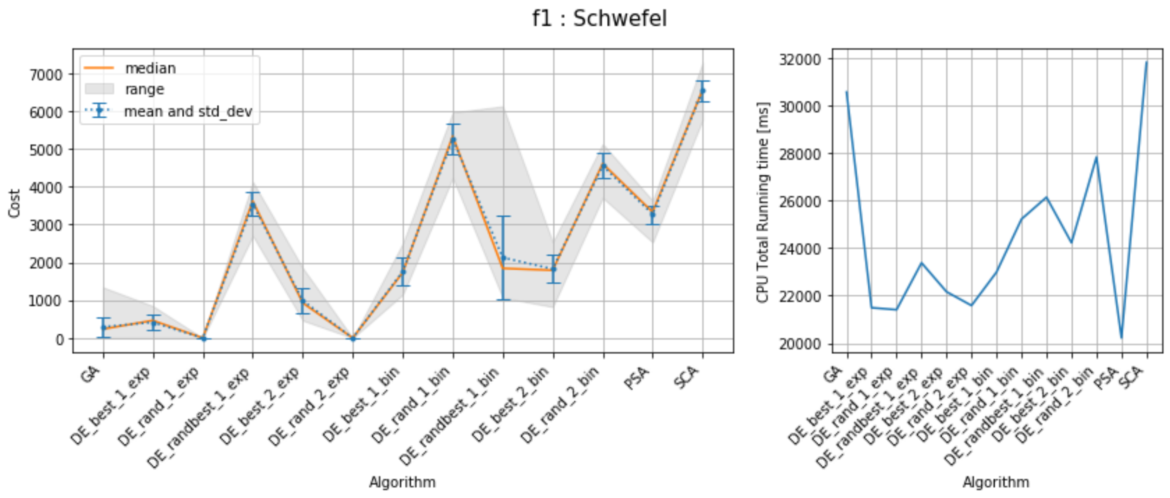

Figure 1: Cost and CPU total running time of Function 1: Schwefel

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	291.715	252.889	233.445	0.065	1342.860	30556.100
DE_best_1_exp	413.589	204.703	462.697	-0.001	846.203	21489.300
DE_rand_1_exp	0.001	0.001	0.001	-0.000	0.003	21405.800
DE_randbest_1_exp	3542.440	305.624	3618.260	2698.730	4127.590	23376.400
DE_best_2_exp	987.060	321.703	925.399	462.703	1849.480	22160.100
DE_rand_2_exp	-0.002	0.000	-0.002	-0.003	-0.001	21582.000
DE_best_1_bin	1757.670	359.702	1728.670	1147.530	2503.530	22986.300
DE_rand_1_bin	5261.550	396.785	5335.010	4231.200	5961.440	25214.300
DE_randbest_1_bin	2124.900	1093.450	1840.380	1066.570	6128.350	26134.600
DE_best_2_bin	1824.540	364.970	1789.090	821.697	2535.630	24220.600
DE_rand_2_bin	4551.430	324.357	4607.150	3705.000	5131.590	27823.400
PSA	3271.220	239.150	3327.850	2528.260	3633.530	20219.000
SCA	6546.260	276.539	6560.930	5767.570	7283.020	31801.900

Table 3: Function 1: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_2_exp, Cost (mean): -0.001977

4.2 Function 2: De Jong 1

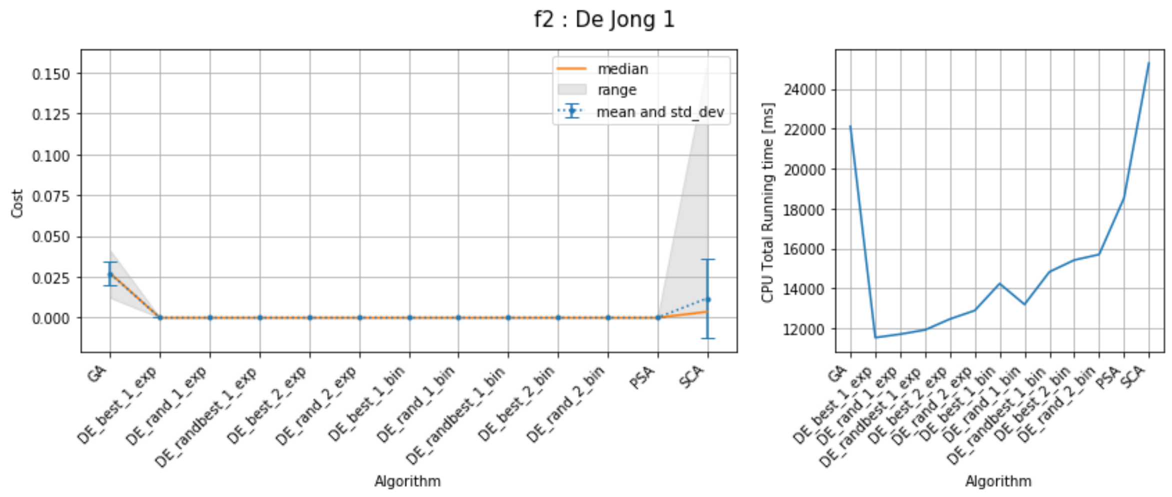

Figure 2: Cost and CPU total running time of Function 2: De Jong 1

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.027	0.007	0.027	0.012	0.041	22121.200
DE_best_1_exp	0.000	0.000	0.000	0.000	0.000	11523.900
DE_rand_1_exp	0.000	0.000	0.000	0.000	0.000	11695.500
DE_randbest_1_exp	0.000	0.000	0.000	0.000	0.000	11908.100
DE_best_2_exp	0.000	0.000	0.000	0.000	0.000	12454.800
DE_rand_2_exp	0.000	0.000	0.000	0.000	0.000	12882.300
DE_best_1_bin	0.000	0.000	0.000	0.000	0.000	14231.100
DE_rand_1_bin	0.000	0.000	0.000	0.000	0.000	13186.400
DE_randbest_1_bin	0.000	0.000	0.000	0.000	0.000	14820.700
DE_best_2_bin	0.000	0.000	0.000	0.000	0.000	15413.000
DE_rand_2_bin	0.000	0.000	0.000	0.000	0.000	15693.300
PSA	0.000	0.000	0.000	0.000	0.000	18522.800
SCA	0.012	0.024	0.004	0.000	0.156	25287.200

Table 4: Function 2: Statistical Analysis of the Cost

Best Algorithm:
PSA, Cost (mean): 0.000000
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_best_2_exp, Cost (mean): 0.000000 , P value: 0.312300

4.3 Function 3: Rosenbrock's Saddle

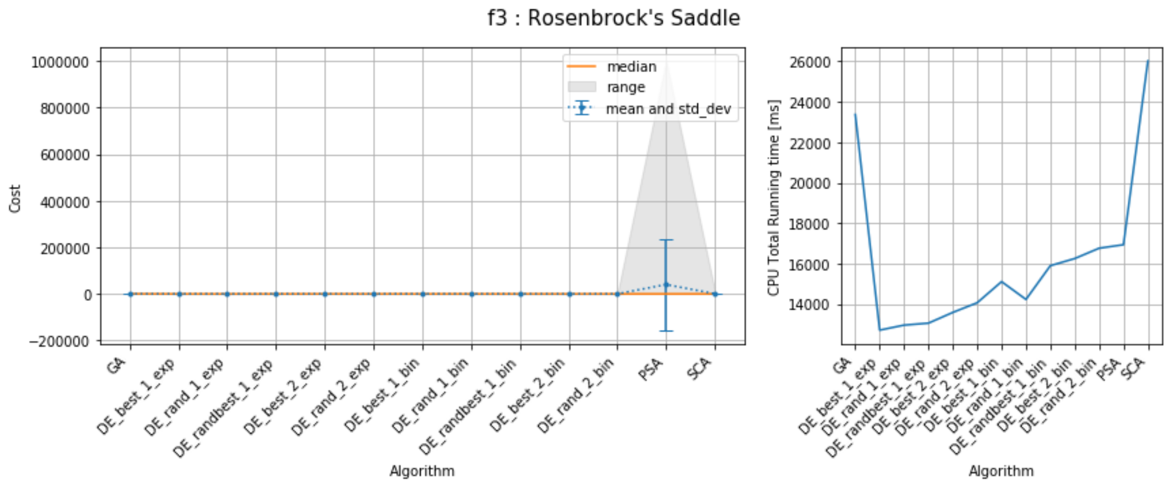

Figure 3: Cost and CPU total running time of Function 3: Rosenbrock's Saddle

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	257.352	884.297	137.786	22.558	6438.970	23371.600
DE_best_1_exp	27.969	27.609	19.983	6.134	181.193	12714.700
DE_rand_1_exp	27.116	14.129	24.682	0.254	82.130	12963.200
DE_randbest_1_exp	30.528	18.844	24.980	20.086	98.873	13060.400
DE_best_2_exp	46.667	149.105	21.051	6.904	1084.970	13590.900
DE_rand_2_exp	42.242	29.691	26.229	7.938	113.534	14069.100
DE_best_1_bin	30.677	33.944	17.140	9.213	204.623	15112.100
DE_rand_1_bin	31.228	16.637	26.366	24.792	90.204	14232.800
DE_randbest_1_bin	29.792	15.341	25.500	19.739	83.236	15900.600
DE_best_2_bin	6.766	10.657	4.102	0.013	67.366	16253.500
DE_rand_2_bin	50.293	29.508	27.080	24.573	93.474	16764.200
PSA	40247.300	195919.000	13.081	0.019	1000030.000	16938.100
SCA	665.449	1900.780	125.559	28.431	13094.800	26018.600

Table 5: Function 3: Statistical Analysis of the Cost

Best Algorithm:
DE_best_2_bin, Cost (mean): 6.765720
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_best_2_exp, Cost (mean): 46.666800 , P value: 0.059100
PSA, Cost (mean): 40247.300000 , P value: 0.146400

4.4 Function 4: Rastrigin

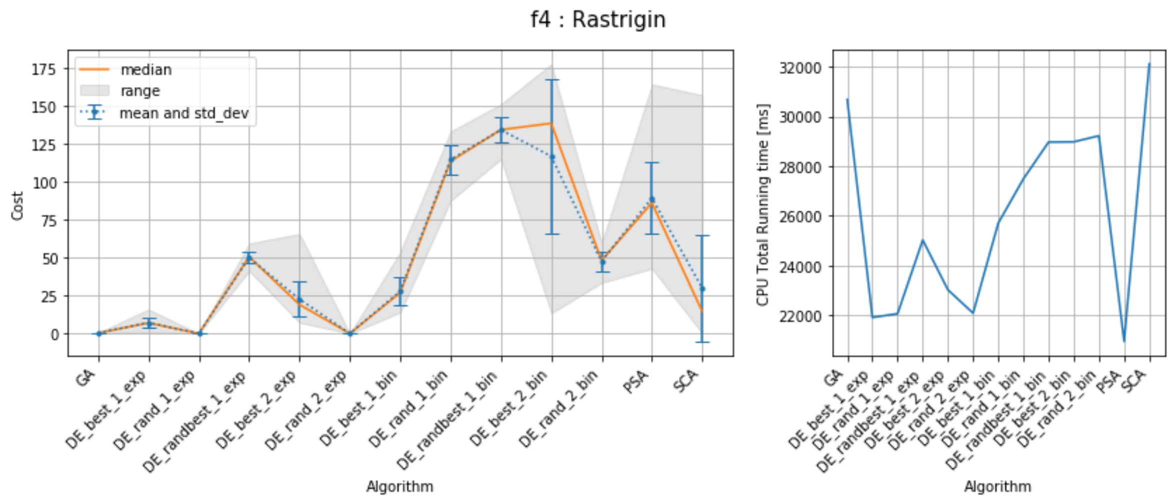

Figure 4: Cost and CPU total running time of Function 4: Rastrigin

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.326	0.093	0.329	0.160	0.602	30677.700
DE_best_1_exp	7.144	3.316	6.965	0.995	15.919	21918.600
DE_rand_1_exp	0.001	0.001	0.001	0.000	0.005	22068.400
DE_randbest_1_exp	49.952	3.926	50.451	41.164	59.406	25030.400
DE_best_2_exp	22.547	11.553	19.003	7.222	65.667	23028.400
DE_rand_2_exp	0.000	0.000	0.000	0.000	0.000	22090.700
DE_best_1_bin	27.855	9.024	26.864	13.930	53.728	25710.100
DE_rand_1_bin	114.483	9.400	113.420	87.475	133.161	27490.800
DE_randbest_1_bin	134.128	8.118	134.180	114.881	151.079	28967.100
DE_best_2_bin	116.669	51.123	138.424	13.387	177.584	28977.000
DE_rand_2_bin	47.581	6.281	48.055	33.423	59.602	29219.600
PSA	89.168	23.481	86.064	42.783	164.167	20958.900
SCA	29.896	35.064	14.473	0.010	157.073	32111.200

Table 6: Function 4: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_2_exp, Cost (mean): 0.000000

4.5 Function 5: Griewangk

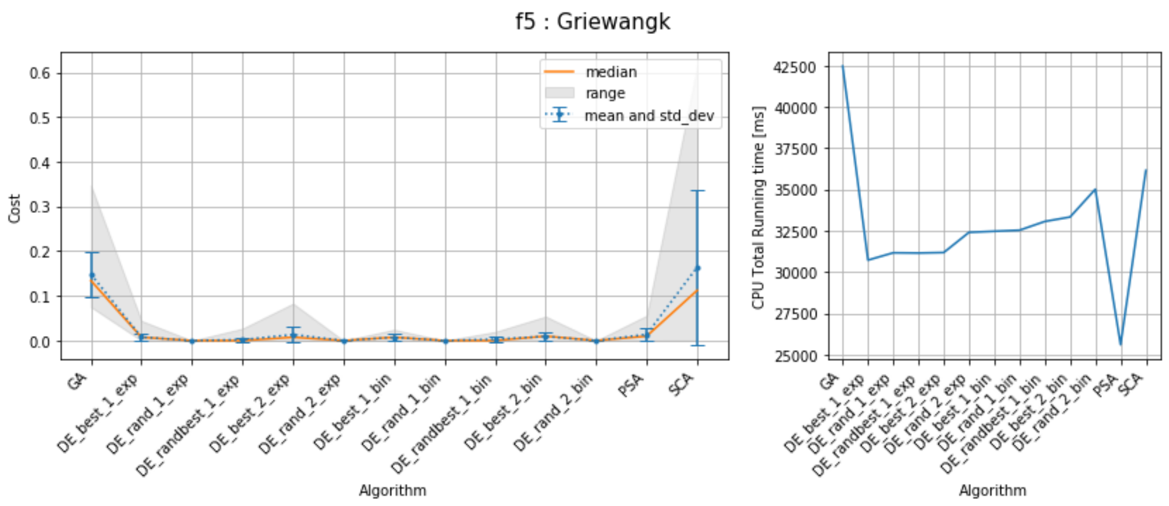

Figure 5: Cost and CPU total running time of Function 5: Griewangk

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.148	0.049	0.134	0.075	0.347	42468.000
DE_best_1_exp	0.008	0.009	0.007	0.000	0.044	30718.300
DE_rand_1_exp	0.000	0.000	0.000	0.000	0.000	31162.500
DE_randbest_1_exp	0.002	0.005	0.000	0.000	0.027	31148.500
DE_best_2_exp	0.013	0.018	0.007	0.000	0.083	31178.600
DE_rand_2_exp	0.000	0.000	0.000	0.000	0.000	32401.500
DE_best_1_bin	0.007	0.007	0.007	0.000	0.025	32465.700
DE_rand_1_bin	0.000	0.000	0.000	0.000	0.000	32529.500
DE_randbest_1_bin	0.003	0.005	0.000	0.000	0.020	33060.300
DE_best_2_bin	0.009	0.010	0.010	0.000	0.054	33329.300
DE_rand_2_bin	0.000	0.000	0.000	0.000	0.000	35011.700
PSA	0.014	0.015	0.010	0.000	0.056	25597.000
SCA	0.164	0.174	0.112	0.000	0.614	36150.800

Table 7: Function 5: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_2_bin, Cost (mean): 0.000000

4.6 Function 6: Sine Envelope Sine Wave

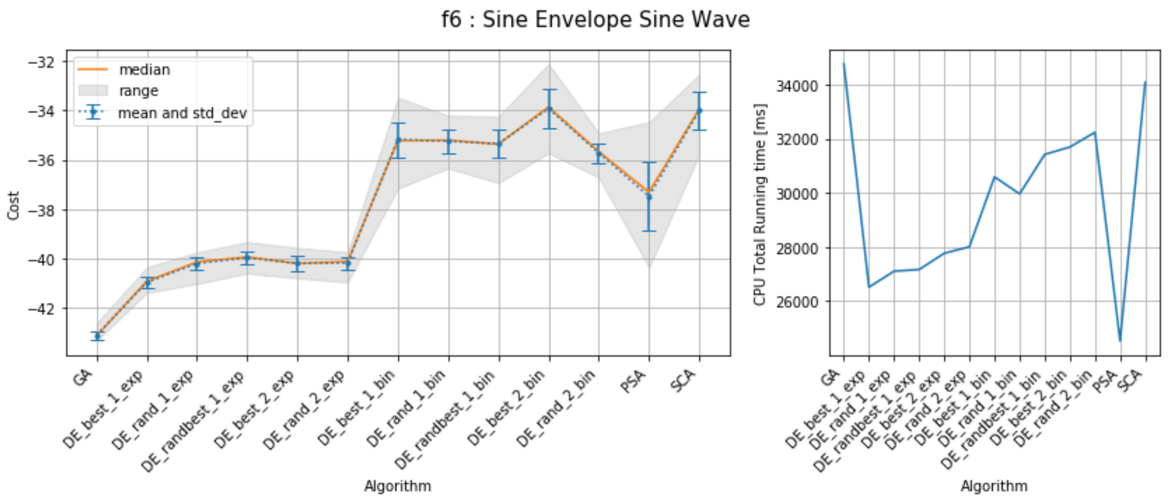

Figure 6: Cost and CPU total running time of Function 6: Sine Envelope Sine Wave

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	-43.097	0.166	-43.116	-43.326	-42.596	34769.500
DE_best_1_exp	-40.954	0.231	-40.913	-41.379	-40.340	26519.000
DE_rand_1_exp	-40.193	0.246	-40.136	-41.014	-39.743	27109.600
DE_randbest_1_exp	-39.959	0.260	-39.932	-40.593	-39.316	27170.500
DE_best_2_exp	-40.188	0.299	-40.194	-40.794	-39.559	27772.300
DE_rand_2_exp	-40.172	0.256	-40.115	-40.952	-39.727	28010.000
DE_best_1_bin	-35.180	0.707	-35.226	-37.162	-33.483	30593.200
DE_rand_1_bin	-35.246	0.469	-35.211	-36.346	-34.213	29964.500
DE_randbest_1_bin	-35.355	0.563	-35.371	-36.942	-34.248	31422.700
DE_best_2_bin	-33.936	0.787	-33.869	-35.732	-32.125	31695.000
DE_rand_2_bin	-35.725	0.398	-35.674	-36.723	-34.923	32239.700
PSA	-37.483	1.398	-37.318	-40.360	-34.475	24531.600
SCA	-34.012	0.783	-33.959	-35.791	-32.551	34082.600

Table 8: Function 6: Statistical Analysis of the Cost

Best Algorithm:
GA, Cost (mean): -43.097400

4.7 Function 7: Stretch V Sine Wave

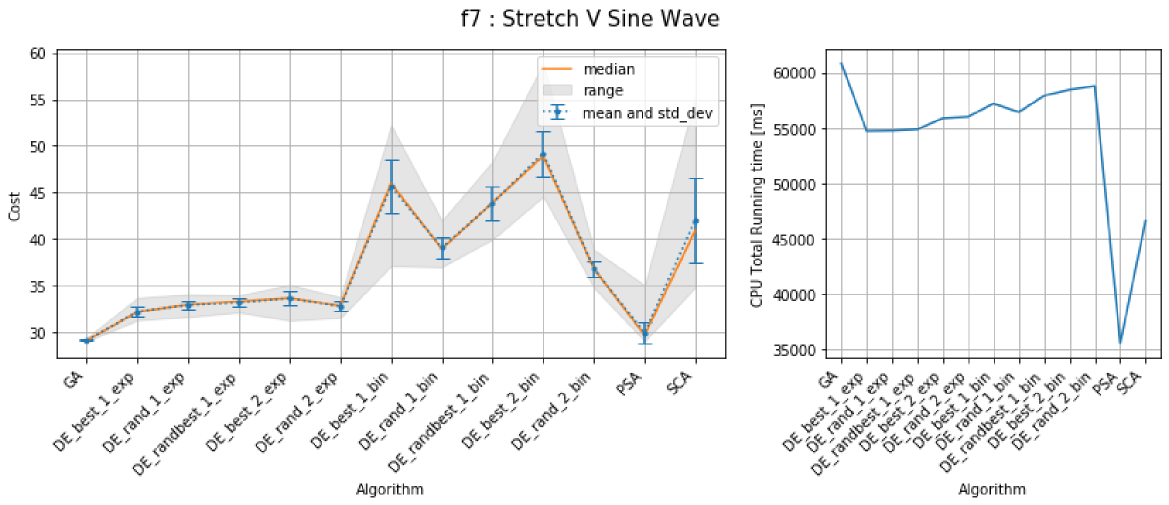

Figure 7: Cost and CPU total running time of Function 7: Stretch V Sine Wave

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	29.109	0.084	29.082	29.014	29.420	60843.000
DE_best_1_exp	32.202	0.579	32.163	31.270	33.704	54717.000
DE_rand_1_exp	32.902	0.504	32.941	31.626	34.050	54750.900
DE_randbest_1_exp	33.166	0.475	33.292	32.112	33.962	54867.100
DE_best_2_exp	33.636	0.769	33.649	31.232	35.076	55881.600
DE_rand_2_exp	32.756	0.521	32.815	31.547	33.759	56013.500
DE_best_1_bin	45.657	2.905	45.990	37.101	52.233	57204.400
DE_rand_1_bin	39.051	1.164	38.931	36.996	41.978	56431.900
DE_randbest_1_bin	43.853	1.795	43.885	39.957	48.346	57920.100
DE_best_2_bin	49.218	2.446	48.907	44.553	58.865	58456.700
DE_rand_2_bin	36.816	0.816	36.787	34.763	38.847	58792.800
PSA	29.904	1.089	29.672	29.012	35.035	35531.000
SCA	42.062	4.549	40.917	34.831	55.327	46592.500

Table 9: Function 7: Statistical Analysis of the Cost

Best Algorithm:
GA, Cost (mean): 29.108600

4.8 Function 8: Ackley One

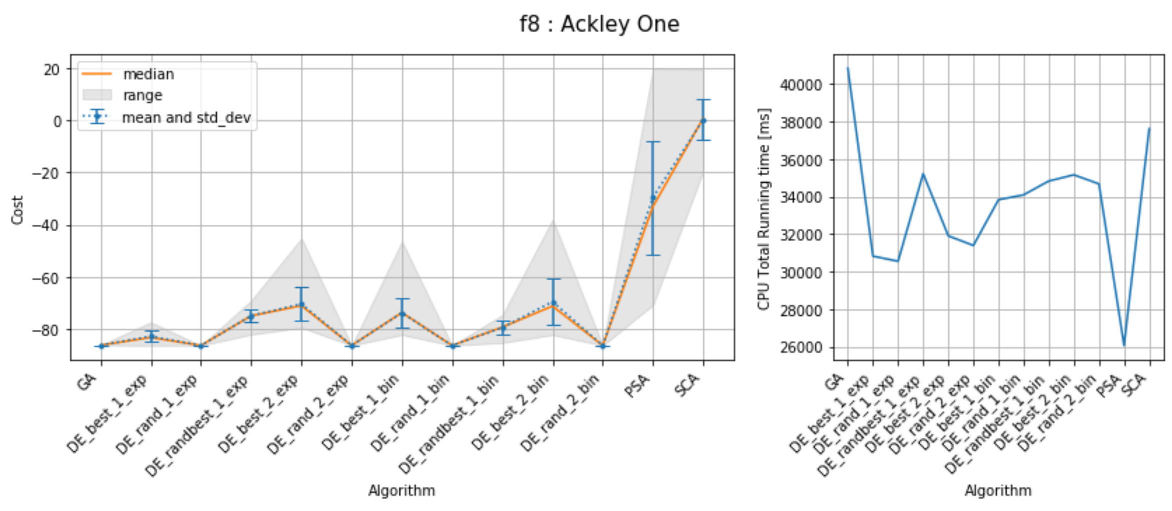

Figure 8: Cost and CPU total running time of Function 8: Ackley One

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	-86.227	0.034	-86.232	-86.288	-86.172	40844.100
DE_best_1_exp	-82.798	2.124	-83.265	-86.333	-77.374	30825.100
DE_rand_1_exp	-86.333	0.000	-86.333	-86.333	-86.333	30550.000
DE_randbest_1_exp	-74.903	2.393	-74.975	-82.121	-69.043	35223.000
DE_best_2_exp	-70.404	6.608	-71.073	-79.485	-45.173	31906.100
DE_rand_2_exp	-86.333	0.000	-86.333	-86.333	-86.333	31392.400
DE_best_1_bin	-73.816	5.568	-73.725	-82.242	-46.386	33833.200
DE_rand_1_bin	-86.333	0.000	-86.333	-86.333	-86.332	34094.200
DE_randbest_1_bin	-79.442	2.535	-79.319	-85.310	-74.594	34828.200
DE_best_2_bin	-69.564	8.844	-71.231	-82.242	-37.979	35161.900
DE_rand_2_bin	-86.333	0.000	-86.333	-86.333	-86.333	34678.400
PSA	-29.518	21.770	-33.060	-70.787	19.922	26057.900
SCA	0.245	7.895	0.761	-19.518	19.742	37620.600

Table 10: Function 8: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_1_exp, Cost (mean): -86.332800
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_rand_2_exp, Cost (mean): -86.332800 , P value: 1.000000
DE_rand_2_bin, Cost (mean): -86.332800 , P value: 1.000000

4.9 Function 9: Ackley Two

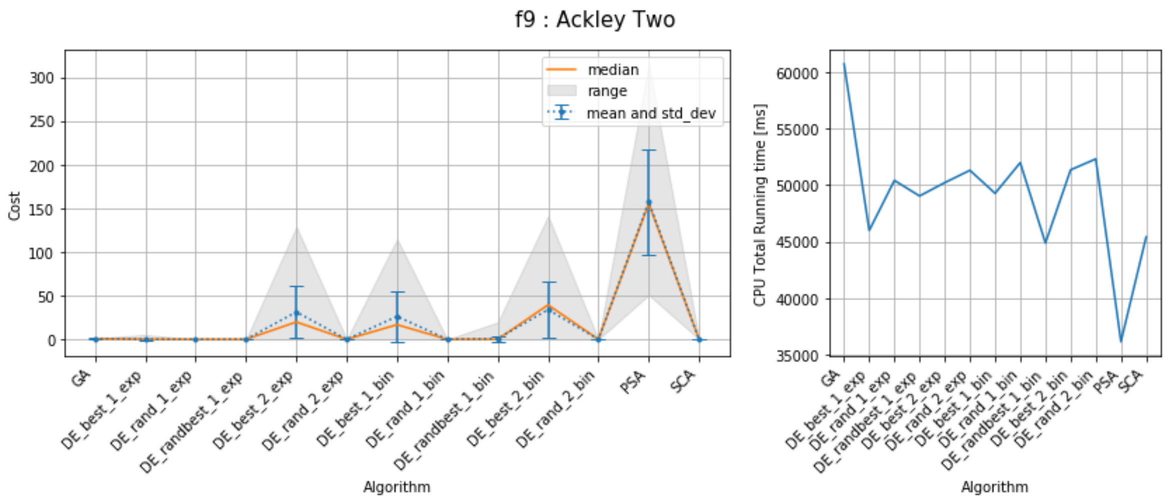

Figure 9: Cost and CPU total running time of Function 9: Ackley Two

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.912	0.188	0.853	0.624	1.349	60715.900
DE_best_1_exp	0.155	0.801	0.000	0.000	5.160	45983.300
DE_rand_1_exp	0.000	0.000	0.000	0.000	0.000	50407.800
DE_randbest_1_exp	0.000	0.000	0.000	0.000	0.000	49035.300
DE_best_2_exp	31.279	30.109	19.829	0.000	128.971	50212.800
DE_rand_2_exp	0.000	0.000	0.000	0.000	0.000	51302.700
DE_best_1_bin	26.216	28.915	16.697	-0.000	114.839	49268.600
DE_rand_1_bin	0.000	0.000	0.000	0.000	0.000	51982.400
DE_randbest_1_bin	0.754	2.977	-0.000	-0.000	19.619	44871.500
DE_best_2_bin	33.966	32.589	39.382	-0.000	141.211	51346.500
DE_rand_2_bin	0.000	0.000	0.000	0.000	0.000	52310.300
PSA	157.429	60.556	155.498	50.832	315.254	36144.900
SCA	0.004	0.005	0.002	0.000	0.023	45406.100

Table 11: Function 9: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_2_bin, Cost (mean): 0.000000
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_best_1_exp, Cost (mean): 0.154796 , P value: 0.171800
DE_randbest_1_bin, Cost (mean): 0.753563 , P value: 0.073400

4.10 Function 10: Egg Holder

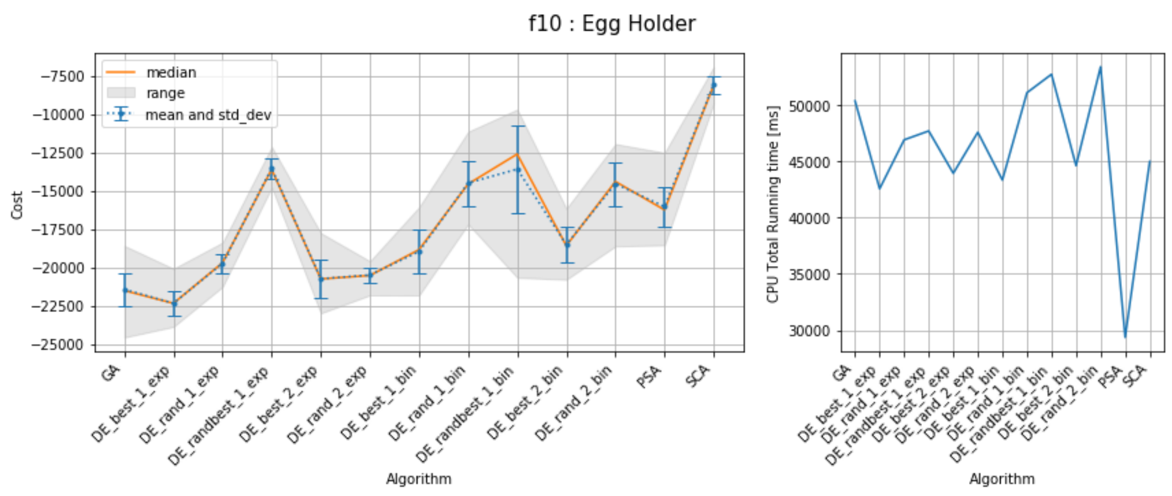

Figure 10: Cost and CPU total running time of Function 10: Egg Holder

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	-21442.100	1077.700	-21521.800	-24575.000	-18595.000	50363.000
DE_best_1_exp	-22349.800	829.100	-22353.400	-23881.700	-20088.400	42564.800
DE_rand_1_exp	-19745.600	628.453	-19704.700	-21273.100	-18377.500	46912.700
DE_randbest_1_exp	-13548.600	677.148	-13613.500	-14794.500	-12138.900	47699.000
DE_best_2_exp	-20752.800	1215.090	-20762.100	-22999.400	-17762.300	43939.800
DE_rand_2_exp	-20505.900	484.229	-20537.600	-21822.700	-19599.100	47585.600
DE_best_1_bin	-18974.300	1418.170	-18853.100	-21829.800	-16071.300	43342.700
DE_rand_1_bin	-14504.400	1476.430	-14568.700	-17193.800	-11148.500	51096.600
DE_randbest_1_bin	-13595.400	2855.480	-12616.600	-20674.600	-9698.160	52734.700
DE_best_2_bin	-18519.800	1170.740	-18593.700	-20805.000	-16163.400	44617.500
DE_rand_2_bin	-14551.900	1436.600	-14407.000	-18627.400	-11949.200	53400.700
PSA	-16045.600	1266.160	-16259.200	-18560.300	-12521.900	29364.100
SCA	-8083.420	590.408	-8123.710	-9308.710	-6919.080	44986.800

Table 12: Function 10: Statistical Analysis of the Cost

Best Algorithm:
DE_best_1_exp, Cost (mean): -22349.800000

4.11 Function 11: Rana

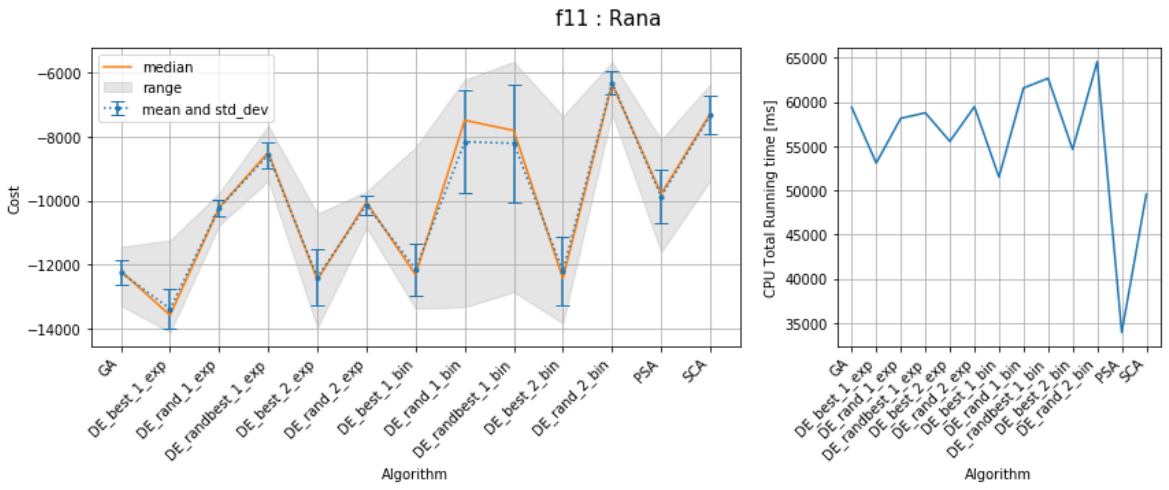

Figure 11: Cost and CPU total running time of Function 11: Rana

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	-12236.900	374.758	-12200.900	-13271.800	-11435.200	59400.100
DE_best_1_exp	-13371.700	639.893	-13574.200	-14111.400	-11234.000	53082.800
DE_rand_1_exp	-10222.100	271.100	-10219.400	-10770.600	-9757.920	58152.900
DE_randbest_1_exp	-8572.630	394.680	-8491.470	-9382.320	-7657.640	58772.300
DE_best_2_exp	-12390.600	889.330	-12467.100	-13963.500	-10403.000	55544.300
DE_rand_2_exp	-10131.400	294.876	-10049.800	-10844.600	-9713.110	59465.100
DE_best_1_bin	-12138.000	818.864	-12310.200	-13368.100	-8328.620	51507.700
DE_rand_1_bin	-8161.230	1616.130	-7489.140	-13320.300	-6226.960	61581.800
DE_randbest_1_bin	-8208.210	1846.680	-7811.030	-12849.300	-5659.630	62671.900
DE_best_2_bin	-12201.200	1082.510	-12431.400	-13825.600	-7361.840	54636.900
DE_rand_2_bin	-6322.750	375.721	-6358.950	-7282.700	-5673.070	64566.400
PSA	-9866.510	817.497	-9789.080	-11597.300	-8104.920	33922.900
SCA	-7329.940	609.371	-7289.290	-9373.180	-6363.650	49554.300

Table 13: Function 11: Statistical Analysis of the Cost

Best Algorithm:
DE_best_1_exp, Cost (mean): -13371.700000

4.12 Function 12: Pathological

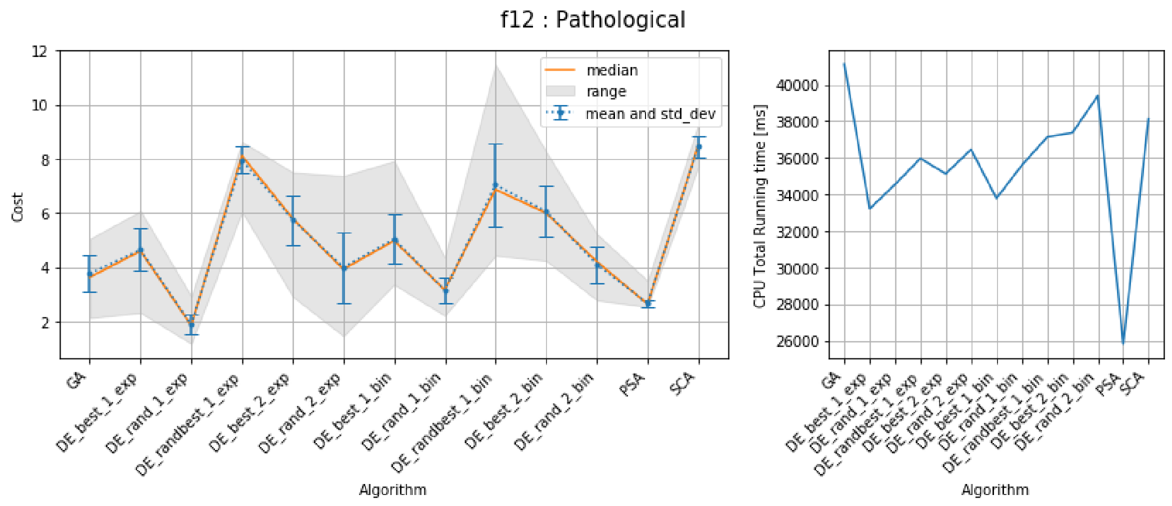

Figure 12: Cost and CPU total running time of Function 12: Pathological

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	3.771	0.687	3.631	2.130	5.055	41120.800
DE_best_1_exp	4.645	0.785	4.598	2.320	6.081	33212.900
DE_rand_1_exp	1.880	0.363	1.852	1.178	2.933	34544.000
DE_randbest_1_exp	7.959	0.489	8.117	6.049	8.634	35967.700
DE_best_2_exp	5.754	0.913	5.800	2.916	7.506	35129.100
DE_rand_2_exp	4.000	1.316	3.926	1.460	7.377	36458.400
DE_best_1_bin	5.045	0.925	4.978	3.360	7.922	33782.000
DE_rand_1_bin	3.153	0.457	3.161	2.214	4.366	35590.700
DE_randbest_1_bin	7.039	1.562	6.868	4.428	11.489	37140.300
DE_best_2_bin	6.062	0.942	5.999	4.238	8.265	37383.600
DE_rand_2_bin	4.100	0.676	4.207	2.780	5.231	39404.200
PSA	2.671	0.135	2.640	2.540	3.505	25825.900
SCA	8.454	0.389	8.504	7.703	9.290	38143.100

Table 14: Function 12: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_1_exp, Cost (mean): 1.879540

4.13 Function 13: Michalewicz

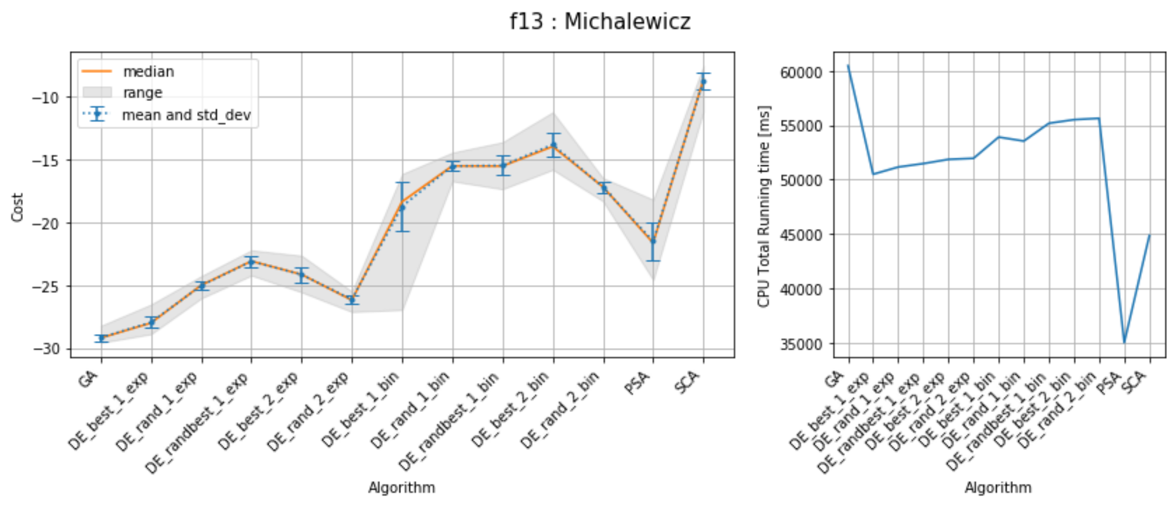

Figure 13: Cost and CPU total running time of Function 13: Michalewicz

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	-29.162	0.255	-29.195	-29.564	-28.219	60472.400
DE_best_1_exp	-27.954	0.429	-28.006	-28.884	-26.527	50495.800
DE_rand_1_exp	-25.040	0.352	-25.009	-26.044	-24.271	51158.600
DE_randbest_1_exp	-23.112	0.466	-23.093	-24.224	-22.219	51474.400
DE_best_2_exp	-24.164	0.614	-24.153	-25.556	-22.651	51863.100
DE_rand_2_exp	-26.136	0.353	-26.191	-27.116	-25.490	51958.700
DE_best_1_bin	-18.750	1.902	-18.374	-26.962	-16.173	53912.300
DE_rand_1_bin	-15.550	0.411	-15.537	-16.745	-14.469	53545.000
DE_randbest_1_bin	-15.499	0.775	-15.552	-17.374	-13.642	55181.200
DE_best_2_bin	-13.846	0.942	-13.992	-15.830	-11.258	55514.100
DE_rand_2_bin	-17.223	0.443	-17.198	-18.332	-16.489	55639.700
PSA	-21.507	1.535	-21.648	-24.590	-18.205	35007.700
SCA	-8.777	0.678	-8.701	-11.205	-7.584	44852.500

Table 15: Function 13: Statistical Analysis of the Cost

Best Algorithm:
GA, Cost (mean): -29.162300

4.14 Function 14: Masters' Cosine Wave

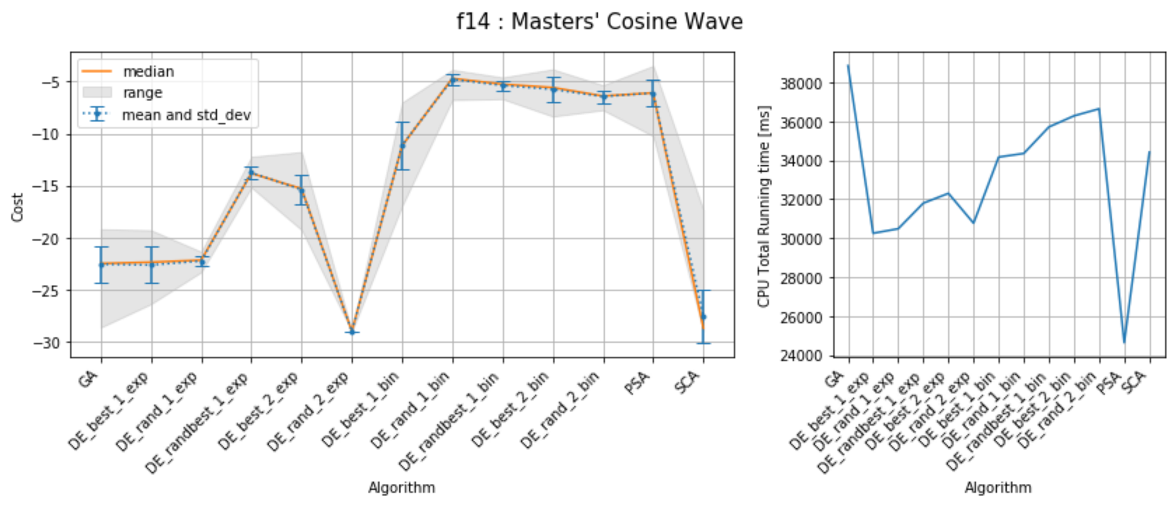

Figure 14: Cost and CPU total running time of Function 14: Masters' Cosine Wave

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	-22.551	1.774	-22.478	-28.620	-19.170	38860.600
DE_best_1_exp	-22.620	1.755	-22.355	-26.365	-19.271	30252.600
DE_rand_1_exp	-22.235	0.502	-22.158	-23.303	-21.337	30482.500
DE_randbest_1_exp	-13.754	0.580	-13.812	-15.174	-12.210	31799.300
DE_best_2_exp	-15.387	1.369	-15.296	-19.256	-11.753	32301.600
DE_rand_2_exp	-28.991	0.007	-28.992	-29.000	-28.961	30769.200
DE_best_1_bin	-11.126	2.351	-11.167	-16.951	-6.997	34167.800
DE_rand_1_bin	-4.776	0.533	-4.698	-6.764	-3.857	34352.300
DE_randbest_1_bin	-5.366	0.469	-5.251	-6.670	-4.589	35714.600
DE_best_2_bin	-5.745	1.145	-5.571	-8.340	-3.810	36286.100
DE_rand_2_bin	-6.434	0.592	-6.383	-7.735	-5.338	36647.900
PSA	-6.088	1.328	-6.084	-10.232	-3.476	24629.300
SCA	-27.556	2.529	-28.695	-29.000	-17.345	34416.300

Table 16: Function 14: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_2_exp, Cost (mean): -28.990800

4.15 Function 15: Quartic

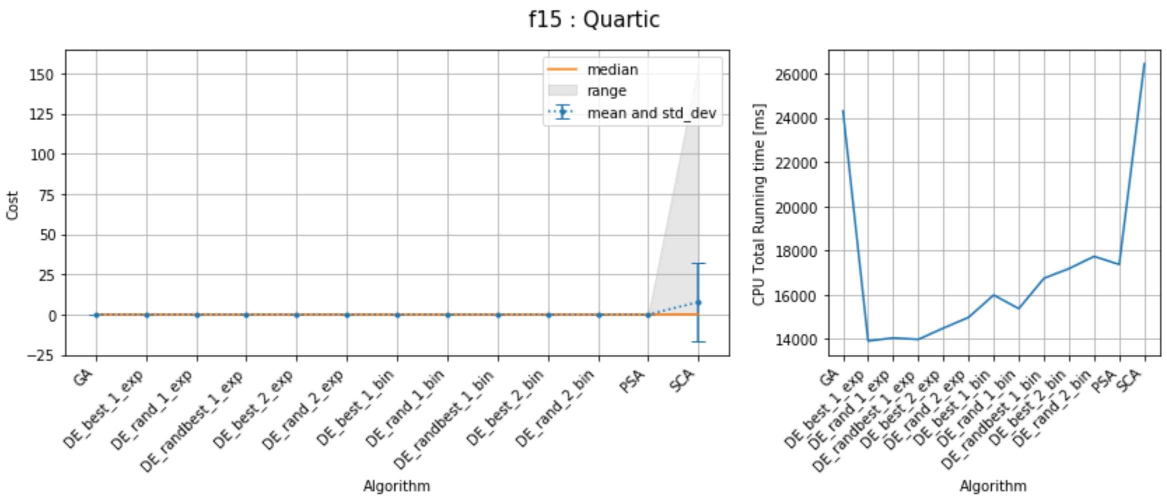

Figure 15: Cost and CPU total running time of Function 15: Quartic

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.004	0.003	0.004	0.001	0.013	24313.500
DE_best_1_exp	0.000	0.000	0.000	0.000	0.000	13904.000
DE_rand_1_exp	0.000	0.000	0.000	0.000	0.000	14042.600
DE_randbest_1_exp	0.000	0.000	0.000	0.000	0.000	13976.100
DE_best_2_exp	0.000	0.000	0.000	0.000	0.000	14488.100
DE_rand_2_exp	0.000	0.000	0.000	0.000	0.000	14973.200
DE_best_1_bin	0.000	0.000	0.000	0.000	0.000	15984.500
DE_rand_1_bin	0.000	0.000	0.000	0.000	0.000	15365.800
DE_randbest_1_bin	0.000	0.000	0.000	0.000	0.000	16738.400
DE_best_2_bin	0.000	0.000	0.000	0.000	0.000	17181.300
DE_rand_2_bin	0.000	0.000	0.000	0.000	0.000	17731.700
PSA	0.000	0.000	0.000	0.000	0.000	17367.900
SCA	7.731	24.212	0.136	0.000	156.060	26450.100

Table 17: Function 15: Statistical Analysis of the Cost

Best Algorithm:
DE_best_1_bin, Cost (mean): 0.000000
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_best_1_exp, Cost (mean): 0.000000 , P value: 0.311300
DE_randbest_1_exp, Cost (mean): 0.000000 , P value: 0.312400
DE_best_2_exp, Cost (mean): 0.000000 , P value: 0.267400
DE_rand_2_exp, Cost (mean): 0.000000 , P value: 0.298700

DE_randbest_1_bin, Cost (mean): 0.000000 , P value: 0.312000

4.16 Function 16: Levy

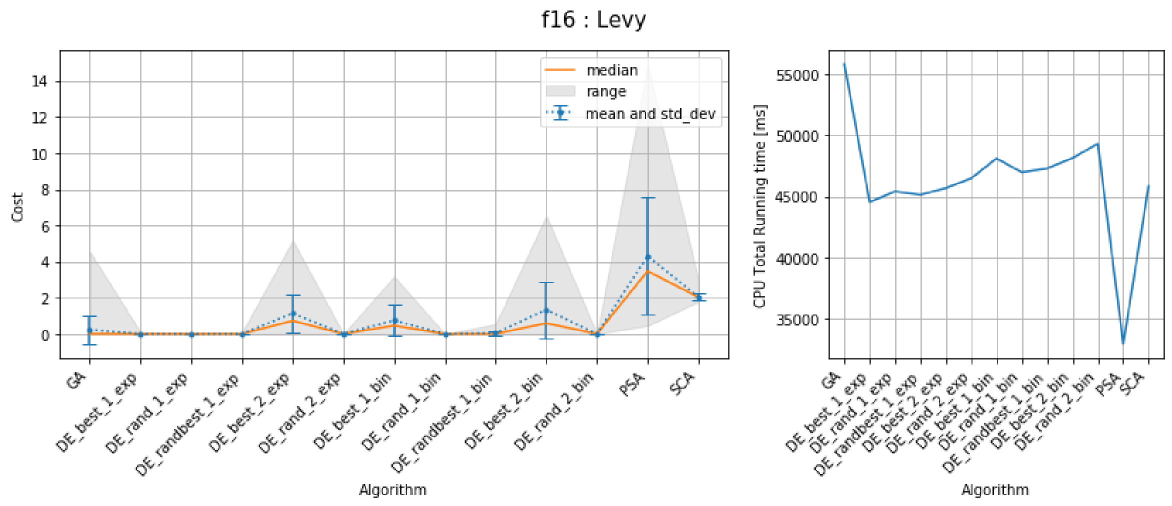

Figure 16: Cost and CPU total running time of Function 16: Levy

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.230	0.774	0.001	0.001	4.547	55783.300
DE_best_1_exp	0.004	0.018	0.000	0.000	0.090	44530.300
DE_rand_1_exp	0.000	0.000	0.000	0.000	0.000	45399.700
DE_randbest_1_exp	0.002	0.013	0.000	0.000	0.090	45126.400
DE_best_2_exp	1.134	1.065	0.723	0.000	5.177	45666.200
DE_rand_2_exp	0.000	0.000	0.000	0.000	0.000	46461.800
DE_best_1_bin	0.744	0.853	0.454	0.000	3.180	48093.800
DE_rand_1_bin	0.000	0.000	0.000	0.000	0.000	46963.000
DE_randbest_1_bin	0.041	0.103	0.000	0.000	0.544	47283.200
DE_best_2_bin	1.332	1.564	0.589	0.000	6.544	48122.700
DE_rand_2_bin	0.000	0.000	0.000	0.000	0.000	49301.700
PSA	4.303	3.234	3.472	0.454	14.906	32961.200
SCA	2.042	0.183	2.028	1.780	2.978	45821.000

Table 18: Function 16: Statistical Analysis of the Cost

Best Algorithm:
DE_rand_2_bin, Cost (mean): 0.000000
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_best_1_exp, Cost (mean): 0.003581 , P value: 0.148900
DE_randbest_1_exp, Cost (mean): 0.001791 , P value: 0.312400

4.17 Function 17: Step

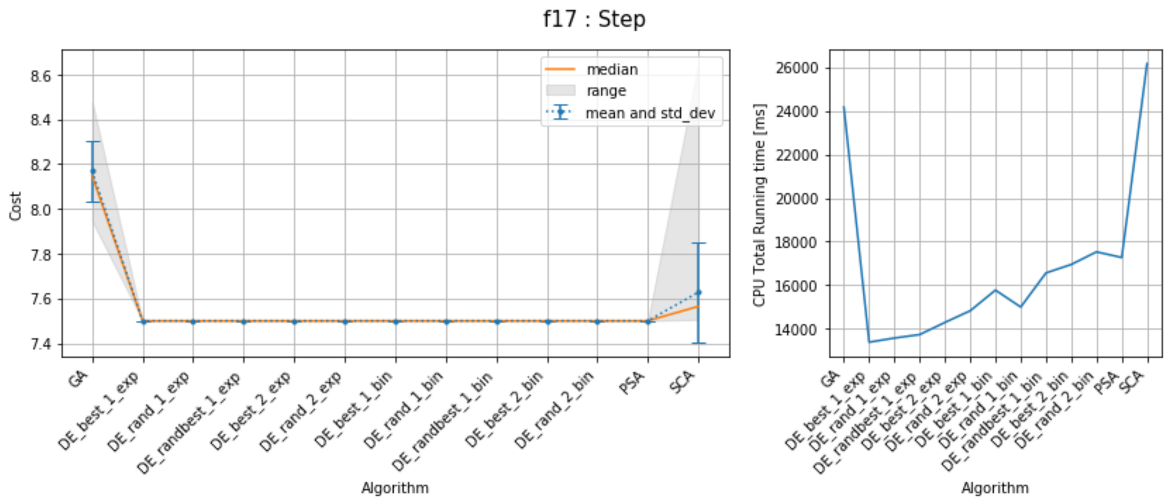

Figure 17: Cost and CPU total running time of Function 17: Step

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	8.169	0.136	8.149	7.946	8.481	24166.900
DE_best_1_exp	7.500	0.000	7.500	7.500	7.500	13384.400
DE_rand_1_exp	7.500	0.000	7.500	7.500	7.500	13573.500
DE_randbest_1_exp	7.500	0.000	7.500	7.500	7.500	13732.800
DE_best_2_exp	7.500	0.000	7.500	7.500	7.501	14291.200
DE_rand_2_exp	7.500	0.000	7.500	7.500	7.500	14825.900
DE_best_1_bin	7.500	0.000	7.500	7.500	7.500	15772.200
DE_rand_1_bin	7.500	0.000	7.500	7.500	7.500	14996.700
DE_randbest_1_bin	7.500	0.000	7.500	7.500	7.500	16562.200
DE_best_2_bin	7.500	0.000	7.500	7.500	7.500	16949.400
DE_rand_2_bin	7.500	0.000	7.500	7.500	7.500	17527.800
PSA	7.500	0.000	7.500	7.500	7.501	17268.900
SCA	7.627	0.223	7.564	7.505	8.649	26166.300

Table 19: Function 17: Statistical Analysis of the Cost

Best Algorithm:
DE_best_1_exp, Cost (mean): 7.500000
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_rand_1_exp, Cost (mean): 7.500000 , P value: 1.000000
DE_randbest_1_exp, Cost (mean): 7.500000 , P value: 1.000000
DE_rand_2_exp, Cost (mean): 7.500000 , P value: 1.000000
DE_best_1_bin, Cost (mean): 7.500000 , P value: 1.000000

DE_rand_1_bin, Cost (mean): 7.500000 , P value: 1.000000
DE_randbest_1_bin, Cost (mean): 7.500000 , P value: 1.000000
DE_best_2_bin, Cost (mean): 7.500000 , P value: 1.000000
DE_rand_2_bin, Cost (mean): 7.500000 , P value: 1.000000

4.18 Function 18: Alphine

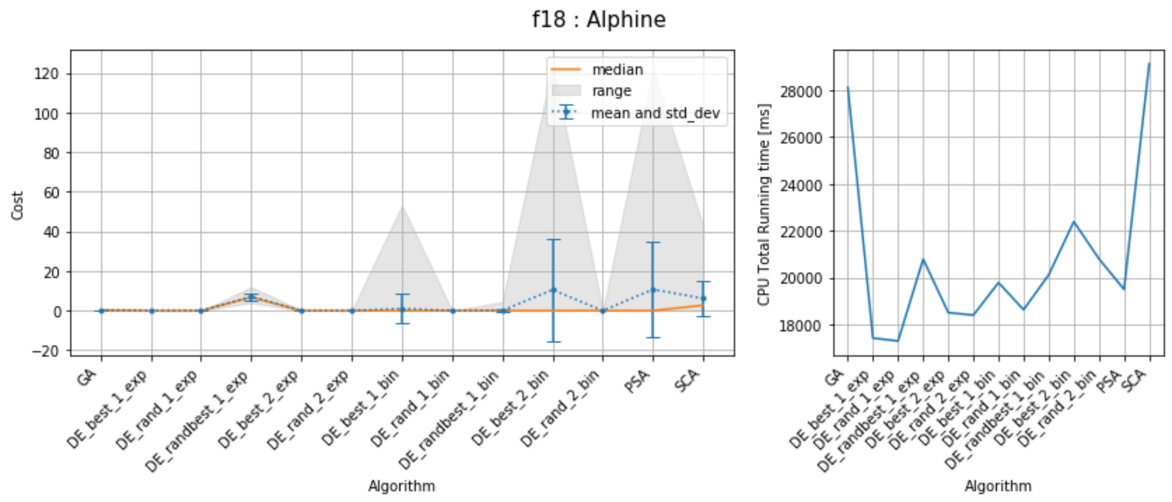

Figure 18: Cost and CPU total running time of Function 18: Alphine

algorithm	mean	std_dev	median	range_min	range_max	time_ms
GA	0.273	0.182	0.240	0.049	1.103	28117.800
DE_best_1_exp	0.000	0.000	0.000	0.000	0.000	17426.400
DE_rand_1_exp	0.003	0.000	0.003	0.002	0.003	17302.100
DE_randbest_1_exp	7.106	1.864	6.788	3.979	11.909	20792.700
DE_best_2_exp	0.014	0.053	0.000	0.000	0.342	18513.100
DE_rand_2_exp	0.002	0.000	0.002	0.002	0.003	18404.700
DE_best_1_bin	1.064	7.446	0.000	0.000	53.188	19791.200
DE_rand_1_bin	0.018	0.001	0.018	0.015	0.019	18636.300
DE_randbest_1_bin	0.089	0.626	0.000	0.000	4.472	20124.500
DE_best_2_bin	10.504	25.786	0.000	0.000	124.525	22391.200
DE_rand_2_bin	0.012	0.001	0.012	0.010	0.013	20807.300
PSA	10.733	24.130	0.001	0.000	121.910	19501.300
SCA	6.166	8.912	2.738	0.015	42.774	29122.300

Table 20: Function 18: Statistical Analysis of the Cost

Best Algorithm:
DE_best_1_exp, Cost (mean): 0.000005
Two-Sample Z-Test Hypothesis Testing: confidence interval = 95%

Null hypothesis: The best algorithm and the tested one are equal
DE_best_2_exp, Cost (mean): 0.013991 , P value: 0.064400
DE_best_1_bin, Cost (mean): 1.063840 , P value: 0.312400
DE_randbest_1_bin, Cost (mean): 0.089453 , P value: 0.312400

4.19 Summary

The table **??** shows a summary of the mean cost obtained by each algorithm, the lowest cost of each function is highlighted in yellow. It can be observed that the Genetic Algorithm found the minimum values in 10 functions, in some way Differential Algorithms are pretty good, they found the lowest cost in 9 functions. In function 8 both algorithms found the lowest cost.

function_id	best_algorithm	best_cost	similar_result obtained by hypothesis testing
1	DE_rand_2_exp	-0.002	
2	PSA	0.000	DE_best_2_exp
3	DE_best_2_bin	6.766	DE_best_2_exp, PSA
4	DE_rand_2_exp	0.000	
5	DE_rand_2_bin	0.000	
6	GA	-43.097	
7	GA	29.109	
8	DE_rand_1_exp	-86.333	DE_rand_2_exp, DE_rand_2_bin
9	DE_rand_2_bin	0.000	DE_best_1_exp, DE_randbest_1_bin
10	DE_best_1_exp	-22349.800	
11	DE_best_1_exp	-13371.700	
12	DE_rand_1_exp	1.880	
13	GA	-29.162	
14	DE_rand_2_exp	-28.991	
15	DE_best_1_bin	0.000	DE_best_1_exp, DE_randbest_1_exp, DE_best_2_exp, DE_rand_2_exp, DE_randbest_1_bin
16	DE_rand_2_bin	0.000	DE_best_1_exp, DE_randbest_1_exp
17	DE_best_1_exp	7.500	DE_rand_1_exp, DE_randbest_1_exp, DE_rand_2_exp, DE_best_1_bin, DE_rand_1_bin, DE_randbest_1_bin, DE_best_2_bin, DE_rand_2_bin
18	DE_best_1_exp	0.000	DE_best_2_exp, DE_best_1_bin, DE_randbest_1_bin

Table 21: Summary: mean cost of each optimization algorithm with different benchmark functions, the best cost of each function is highlighted.

5 Discussion

The previous results show that both GA and DE are good, almost half of the lowest cost is found by GA and another half by DE. Regarding the running time, it can be noticed that in general the GA are slower than DE since in GA it is necessary to do two sorting in each iteration that makes it more time-consuming. Additionally, the figures show that in general, the binomial crossover version of DE is slower than the exponential crossover.

6 Conclusion

In this project two optimization algorithms have been implemented and tested, they are Genetic Algorithm and Differential Evolution Algorithm. From the testing results, it can be observed there is not an absolute winner, GA is better with some functions, and DE with others. The CPU running time of both algorithms is similar. However, some versions of DE are faster than GA but those versions did not find the lowest cost.

To be able to compare the performance between different algorithms we assumed that the manual tuning of the configuration parameters of GA and DE achieved its best value. Additionally, we believe that the comparison is only valid for the mentioned configuration, running the algorithm with other parameters might produce a different result.

For future work, the comparison between different algorithms can be done in many other ways:

We can set a target cost and test which algorithm achieves the target cost first, in this way we can get a better comparison about which algorithm converges faster.

We can give each optimization algorithm a specific time constrain and stop the iterations when the algorithm reach the constrain, then we can compare the cost between different algorithms and find the best one.

References

[1] Measuring cpi time in c. `https://stackoverflow.com/questions/20167685/measuring-cpu-time-in-c`. Accessed: 2020-02-11.