SPRING SEMESTER 2020/21
COMP2024 Coursework

Group 2
Technical Report of Stochastic Optimizers

Marcus Wong J- Fatt (20115147) - GL
Wong Qing Joe (10268818)
Chuang Caleb (20204134)
Mohamad Arif bin M. Abu Baker (20116042)
Brendan Nicholas Sia Cheong Hui (20113100)

We declare that we have read and understood the University's Academic Integrity and Misconduct statements and policies.

LITERATURE REVIEW

Metaheuristics

Metaheuristic algorithms are computational intelligence paradigms especially used for sophisticated solving optimization problems according to Mohamed, A.B.(2018). Multimodal, complex, non linear and high dimensional problems can be solved using metaheuristics optimization algorithms. They are stochastic search algorithms that utilizes heuristics or rules applicable to a certain problem to accelerate or act as catalysts to achieve near-optimal solutions. Metaheuristics algorithms emulate behaviours and processors inspired by mechanisms around us present in our nature. In recent years, a large number of metaheuristic optimization algorithms have been proposed to tackle complex engineering problems according to Diego.O. (2019).

There are three general categories for **meta-heuristics: evolutionary, physics-based and swarm intelligence algorithms.**

Evolutionary (CMA-ES)

The Covariance Matrix Adaptation Evolution Strategy is an evolutionary algorithm targeted for higher level difficulty of non-linear non-convex black-box optimization issues in continuous domain. The CMA-ES is taken into account as progressive in evolutionary computation and has been adopted together of the quality tools for continuous optimization in several (probably a whole lot of) analysis labs and industrial environments round the world. The strategy ought to be applied, if by-product primarily based strategies, e.g. quasi-Newton BFGS or conjugate gradient, (supposedly) fail because of a rugged search landscape (e.g. discontinuities, sharp bends or ridges, noise, native optima, outliers).

The CMA-ES implements a stochastic variable-metric method. In the very particular case of a convex-quadratic objective function

$$f(x) = \frac{1}{2}(x - x^*)^T H(x - x^*)$$

The variance matrix $\boldsymbol{C_k}$ adapts to the inverse of the hessian matrix $\boldsymbol{H}$, up to a scalar issue and little random fluctuations. a lot of general, conjointly on the operation $\boldsymbol{g} \circ \boldsymbol{f}$, wherever $\boldsymbol{g}$ is strictly increasing and is order conserving and $\boldsymbol{f}$ is convex-quadratic, the variance matrix $\boldsymbol{C_k}$ adapts to $\boldsymbol{H}$, up to a scalar factor and little random fluctuations. Note that a generalized capability of evolution methods to adapt a covariance matrix reflective of the inverse-Hessian has been established for a static model counting on a quadratic approximation.

Gaussian distribution is the key source of random variations in CMAES. Simple random sampling is used to achieve this. However, it also suffers from sampling error which is caused by biased or unrepresentative samples when the sample size/ population is small. A strong sampling error occurs when the mean and covariance of a distribution are estimated from the vectors which are opposite the direction of the center of mass, the results will deviate largely from the optimum. CMAES tend to exploit small populations to speed up their convergence rate, hence the sampling error occurs where by a large portion of search place is not reached, none of the mutations lead to an improvement, hindering the progress of this generation. A derandomized sampling method is proposed to tackle this problem which is mirrored sampling. Sequential selection is also used together with mirrored sampling as both concepts complement each other well. Brockhoff. D. (2011).

Physics-Based

Several metaheuristic optimization algorithms have been developed by scientists in the recent years. These methods or algorithms have been emulated by biological or physical processes of nature. Scientists have been utilizing different chemical, biological and physical laws to improve and enhance new metaheuristic optimization algorithms. Physics-based algorithms are metaheuristic algorithms created by scientists based on the law of physics. There are a few physics-based metaheuristic algorithms available like Gravitational Search Algorithm (GSA), Central Force Optimization (CFO), Space Gravitational Algorithm and many more.

Swarm Intelligence

Swarm intelligence algorithms is one of the classes of nature-inspired algorithms which has become an area of interest for many researchers and scientists in the recent era. Though very similar to physics-based algorithms, SI algorithms instead rely on the simulated collective and social intelligence of creatures to navigate the search agents towards the objective. We may say that swarm intelligence is the collective conduct of scattered and self-organizing swarms. Swarm intelligence lies in the nexus of communications between swarm agents and the environment.

Particle Swarm Optimization Algorithm

PSO algorithm is a simple population-based stochastic optimization technique that simulates the collective conduct of a flock of birds and fish schools. It deals with the problems where the best solution can be presented as a surface in a space of n dimensions. A set of points or in other words, a swarm of particles, evolve their position in the search space with a velocity vector associated with each particle to find the global optimum at each iteration. A new population of particles is generated from the previous swarm using randomly generated weights and parameters specific to the algorithm (P. Szynkiewicz, 2018).

Following equations are used in the PSO training phase, where c1 and c2 are positive constants, R_1 and R_2 are random numbers from 0 to 1 and w represent the inertia weight.This equations define how the genotype values are changing along iterations (Blas & Mingo, 2010).

$$v_{\text{in}}(t+1) = wv_{\text{in}}(t) + c_1R_1(p_{\text{in}} - x_{\text{in}}(t)) + c_2R_2(p_{gn} - x_{in}(t))$$

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1)$$

The main disadvantage of the PSO algorithm is its sensitivity to a velocity. When the velocity is too low the convergence speed is low, but if it is too high, the algorithm falls into the local minimum (P. Szynkiewicz, 2018).

Grey Wolf Optimizer

An alternative optimiser algorithm that is not so well-known is the Grey Wolf Optimiser. To simulate the leadership hierarchy of grey wolves, four types of grey wolves—alpha, beta, delta and omega—are involved in ranking and organising the best search agents. The optimisation progression from exploration to exploitation is also modelled after the grey wolves' progression of hunting: searching for prey, encircling prey and finally attacking prey. S. Mirjalili, S. M. Mirjalili, A. Lewis, (2014).

Test results show that the GWO was able to provide highly competitive results compared to well-known heuristics in terms of exploitation, exploration and high local optima avoidance, besides having convergence. Random parameters A and C assist candidate solutions to have hyper-spheres with different random radii. The adaptive and decreasing value of a and A also guarantees a smooth transition from exploration in the first half to exploitation in the second. GWO also only has two main parameters for tuning. Lastly, it also has high performance in unknown and challenging search spaces, on both constrained and unconstrained problems.

The next position vector $X(t \vec{+} 1)$ of each search agent/grey wolf is a random position between the position of the prey and the current position of the wolf, using random variable $\vec{C}$. The variable $\vec{A}$ linearly decreases to transition from exploration to exploitation.

$$\vec{D} = |\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)|$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D}$$

The first three best solutions obtained so far are saved and used by the other search agents to update their positions.

Optimizer Configuration Setup

1+2ms CMA-ES

Mirrored sampling and sequential selection is applied on normal CMA-ES. In mirrored sampling, half of the mutation vectors (mirrored) completely depend on the other half (original) which ensures the difference between these two halves of mutation. The mirrored mutation is anti-parallel to the original hence no matter how the search space is partitioned, a mirrored pair would not miss one half of the search space. In sequential selection, the offspring are evaluated one by one sequentially, comparing them to their parents. The original selection scheme will be applied if the first λ-1 offspring are worse than the parent.

1,2 CMA-ES

In the optimiser algorithm utilised, the opts.TolHistFun value has been set according to a separate experiment carried out, the BI-Population CMA-ES on the BBOB-2009. The value has been set to ‘1e-12’. The sequential selection variant was selected for the intention of completing the iterations as soon as an offspring is evaluated to be better than the current λ solution, thereby saving additional functional evaluations.

PSO

The algorithm used is a simple PSO algorithm utilizing the global best model. The only design choice made was to select the absorbing boundaries to handle any particles leaving the search space, where the position is set to the boundary and the velocity is reset to zeros (El-Abd & Kamel, 2009).The swarm has 40 particles with the parameters set as c1 = c2 = 1.4944 and w = 0.792.

Grey Wolf

The default Grey Wolf Optimiser uses 25 search agents to ensure that the maximum function evaluations is divided cleanly without remainder, besides also maximising the number of evaluations that each search agent gets.

RESULTS

Table below shows results of all 4 optimizers that has been tested with ***20000 maximum function evaluations in 5 dimensions***

Summary Table

Functions	Optimizer							
	1+2ms CMA-ES		1,2 CMA-ES		PSO		Grey Wolf	
	Average	STD	Average	STD	Average	STD	Average	STD
Separable Functions								
f1	-4.45E-09	2.12E+00	-2.05E-09	3.54E-10	-2.30E-09	2.31E-09	3.46E-03	1.19E-02
f2	-1.85E-10	7.78E-11	-1.95E-09	1.20E-09	-2.41E-09	1.66E-09	7.35E+01	1.19E+02
f3	9.90E-01	0	4.00E+00	1.41E+00	3.96E-01	5.00E-01	3.21E+00	3.35E+00
f4	2.50E+00	2.12E+00	6.00E+00	0	1.53E+00	1.25E+00	4.80E+00	3.56E+00
f5	-1.00E-08	0	-1.00E-08	0	-1.00E-08	0	-1.00E-08	0
Low or Moderate Condition Functions								
f6	-2.52E-09	2.24E-09	-1.36E-09	1.05E-09	-2.11E-09	1.63E-09	1.30E-01	4.11E-01
f7	-4.20E-09	9.90E-10	9.10E-02	8.00E-02	1.24E-01	2.10E-01	1.09E+00	1.98E+00
f8	-1.90E-09	8.49E-10	-3.13E-09	3.64E-09	9.73E-03	2.00E-02	2.14E-01	2.97E-01
f9	-2.35E-09	1.06E-09	-8.25E-10	3.89E-10	1.02E+00	1.76E+00	1.19E+00	1.18E+00
High Condition Functions								
f10	-2.40E-09	2.69E-09	-9.05E-10	1.06E-10	2.87E+01	4.48E+01	3.64E+02	6.87E+02
f11	-2.55E-09	1.48E-09	-2.40E-09	1.27E-09	1.28E-01	1.90E-01	1.01E+01	9.57E+00
f12	-4.90E-10	3.35E-10	-7.20E-10	8.20E-10	1.86E+01	2.12E+01	9.49E+01	2.80E+02
f13	-1.19E-09	1.15E-09	-2.70E-10	2.26E-10	3.95E+00	3.37E+00	4.65E+00	8.45E+00
f14	-2.75E-10	9.19E-11	-2.85E-10	7.78E-11	8.81E-06	4.75E-06	1.74E-03	6.16E-03
Multi-Modal Functions								
f15	1.50E+00	7.10E-01	4.00E+00	2.83E+00	3.86E+00	2.31E+00	2.49E+00	1.38E+00
f16	5.65E-03	1.00E-02	2.10E+00	7.10E-01	2.47E-01	3.00E-01	3.82E-02	5.68E-02
f17	3.35E-02	3.00E-02	1.55E-01	1.20E-01	7.01E-02	2.10E-01	1.44E-01	3.68E-01
f18	5.75E-02	2.00E-02	1.29E+00	1.00E+00	3.32E-01	4.40E-01	2.98E-01	5.99E-01
f19	1.20E-01	6.00E-02	6.30E-01	3.00E-02	1.28E-01	1.00E-01	7.15E-02	3.24E-02
Multi-Modal with Weak Global Structure Functions								
f20	2.15E-01	3.00E-01	4.55-01	3.00E-01	2.70E-01	1.80E-01	8.27E-01	2.75E-01
f21	-7.00E-09	4.24E-09	-3.80E-09	3.68E-09	1.00E+00	6.80E-01	1.64E+00	1.38E+00
f22	-3.75E-09	3.18E-09	-4.15E-09	2.90E-09	6.97E-01	1.40E+00	1.94E+00	3.14E+00
f23	1.41E-01	1.10E-01	6.20E-01	1.70E-01	3.25E-01	1.70E-01	6.58E-01	1.85E-01
f24	3.15E+00	7.00E-02	8.35E+00	3.75E+00	6.83E+00	2.30E+00	8.63E+00	2.38E+00

Observation of Results

Overall, each optimizer produced the same results for f5 in separable functions.

The 1+2ms CMA-ES optimiser performs most consistently throughout low, moderate and high conditioned benchmark functions with a low standard deviation. The precision is higher on all multimodal functions as compared to other 3 optimizers, shown by the low average Δf_{target} values. Accuracy is higher on separable functions compared to other 3 optimizers, but standard deviation is large for f1 and f4. Though a remarkable zero standard deviation was obtained on a separable benchmark function (f3 and f4) by both of the CMA-ES optimisers, the average minimal score is still lower than PSO's.

The 1,2 CMA-ES optimiser follows similar patterns, accuracy and precision as the 1+2ms CMA-ES, except lagging slightly behind in certain functions, especially multimodal functions.

The PSO optimiser performs its best and most consistent in separable functions, especially f1 and f2. For Low or Moderate Condition functions, PSO scored the lowest Δf_{target} in f6. High Condition functions' results are not consistent, having a very high standard deviation, particularly in f10 and f12. The CMA optimisers still outperform PSO, except in the multi-modal functions.
The Grey Wolf Optimiser is the worst-performing of all optimisers, producing the least accurate average scores. Compared to other optimisers for separable functions, it loses out in f1 and f2. Neither does it produce good results in low or high condition functions. However, it performs competitively well in the set of multi-modal functions, having the least standard deviation in this category while placing second in accuracy. But it falls back to last place again in multi-modal functions with weak global structure.

Analysis of Results

Both CMA-ES optimisers perform best in separable functions f1-f4. This is because they are able to reduce the search process to D times one-dimensional search procedures, exploiting the separable property to get high accuracy in fewer function evaluations. In CMA-ES, whenever a solution with a lower objective function value is being met, the offspring will be selected to become the next parent, X_{k+1} and also as the best 2 offspring and the parent X_k. The optimiser demonstrates good ability at local minima avoidance and good small-scale exploration.

The 1+2ms CMA-ES optimiser performs best for low and moderate condition functions f6-f9, meaning that they are able to escape large plateaus and handle near-zero or zero gradients well. In functions f10-f14 with high conditioning, 1+2ms CMA-ES also performs the best, which shows that it can be precise with micro-movements towards the optimum when on a steep ridge or peak.

1+2ms CMA-ES uses mirrored sampling and sequential selection. At each iteration, half of the mutation vectors are mirrored with respect to parent $X_k$ensuring that both directions of the vector are evaluated. This is combined with sequential selection whereby it ensures that if the unmirrored solution is poor, it will intertwine newly sampled solutions and their respective mirrored versions hence, the new selection scheme will be applied. This will allow the optimizer to move in different vectors with small step sizes to detect small improvements in low to moderate condition functions. For high condition functions, the mirrored sampling technique uses a damping parameter to modify the step size, σ ensuring small steps made towards the optima. Besides, it's consideration of different vector directions also ensures that it does not exceed the optima.

In Multimodal functions, 1+2ms CMA-ES is the most accurate, but (slightly) less reliable. This optimiser has good exploitation but lesser exploration, since it gets stuck in certain local optima, but can minimize to a very low result if it lands in a good valley. 1+2ms CMA-ES is excellent in finding the local optima but whenever a local optima is found, it will only move back and forth in mirrored and unmirrored directions, hence the selection is limited after detecting no improvements, hence the lack in the ability to search for a global optimum.

On the other hand, GWO is the most reliable and precise in getting results of the same level of accuracy when there are many local minima, yet the average accuracy is (slightly) worse. This may mean better exploration and worse exploitation; it may jump from one local optima to the next without really minimizing the score. GWO uses a linear progression to transition exploration to exploitation. Even though the ratio would be balanced, there may need to be an imbalanced focus on exploitation to produce better results.
In functions f20-f24 that are multimodal with weak global structure, both CMA-ESs perform comparatively very well. They are able to ascertain the good scores, even in a search space that seems to have no pattern or structure to exploit. In multimodal functions f20-f24, population size of both the CMA-ES plays an important role in achieving the highest precision. CMA-ES generates a specific population size of offspring for each function evaluation and determines the closet offspring to the f_{target}, then generates another set of offspring around that particular offspring which is closest to f_{target}. This can ensure that the probability of finding a solution closest to f_{target} can be found and will increase based on the population size set by parameter tuning even though in a weak global structure, where improvements between points are hard to determine via exploring point by point.

Post-Processed Data

We ran the results of each optimizer and comparison between optimizers using the python script package available to produce tables and figures reporting the outcome of the benchmarking experiment. The complete post processing results generated from BB0B-2010 experiment data for each optimizer and its comparisons can be obtained from the following links:

1. 1,2 CMA-ES
2. 1+2ms CMA-ES
3. Grey Wolf
4. PSO
5. Comparison between 1,2 CMA-ES and 1+2ms CMA-ES
6. Comparison between 1+2ms CMA-ES and PSO

Expected running time (ERT) is used to calculate the overall performance of each optimizer as the expected number of function evaluations to reach a target function value for the first time. ERT computes to

$$ERT = \frac{\#FEs(f_{best} \geq f_{target})}{\#succ}$$

where $\#FEs(f_{best} \geq f_{target})$is the total number of function evaluations in which the best function value is not smaller than the function target value over all trials and $\#succ$ is the number of successful trials. The link to the post-processed document is included for each optimizer and the comparison between 2 of the optimizers.

In table 1, we display the comparison of all 4 optimizers by taking their ERT divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values in dimension 5 for all f1 - f24. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in italics, if ERT($\Delta f = 10^{-7}$) = ∞. #succ is the number of trials that reached the final target $f_{opt} + 10^{-8}$. Best results are printed in bold. (Nikolaus Hansen, 2010)

Table 1: Comparison between 4 Algorithms

Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f1	11	12	12	12	12	12	15/15
1,2 CMAES	5.5(4)	12(8)	22(9)	41(14)	58(14)	75(10)	15/15
1+2ms CMAES	**2.2**(2)	**5.3**(2)^{*2}	**8.8**(3)^{*3}	**15**(2)^{*4}	**21**(2)^{*4}	**28**(3)^{*4}	15/15
GWO	2.7(2)	12(7)	327(423)	7911(4579)	9394(4128)	4.1e4(4e4)	0/15
PSO	3.8(3)	28(10)	65(25)	180(26)	319(43)	463(68)	15/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f2	83	87	88	90	92	94	15/15
1,2 CMAES	57(27)	63(14)	66(13)	68(14)	69(12)	70(12)	15/15
1+2ms CMAES	**8.2**(3)^{*4}	**10**(3)^{*4}	**11**(1)^{*4}	**12**(1)^{*4}	**13**(1)^{*4}	**14**(1)^{*4}	15/15
GWO	2991(3251)	8366(9065)	1.7e4(2e4)	∞	∞	∞*1e5*	0/15
PSO	34(10)	42(8)	49(8)	68(6)	85(12)	102(12)	15/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f3	716	1622	1637	1646	1650	1654	15/15
1,2 CMAES	26(24)	∞	∞	∞	∞	∞*1e5*	0/15
1+2ms CMAES	4.7(7)	119(134)	∞	∞	∞	∞*1e5*	0/15
GWO	30(63)	123(97)	456(489)	911(911)	∞	∞*1e5*	0/15
PSO	**2.6**(1.0)	**6.9**(3)^{*3}	**52**(64)^{*2}	**53**(64)^{*2}	**54**(64)^{*2}	**55**(64)^{*2}	9/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f4	809	1633	1688	1817	1886	1903	15/15
1,2 CMAES	38(50)	909(1026)	∞	∞	∞	∞*1e5*	0/15
1+2ms CMAES	7.7(11)	910(980)	∞	∞	∞	∞*1e5*	0/15
GWO	38(66)	∞	∞	∞	∞	∞*1e5*	0/15
PSO	**3.2**(0.7)	**48**(63)*	**397**(499)	**370**(440)	**359**(421)	**357**(418)	2/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f5	10	10	10	10	10	10	15/15
1,2 CMAES	5.8(3)	7.6(8)	7.7(8)	7.7(8)	7.7(8)	7.7(8)	15/15
1+2ms CMAES	**2**(1)^{*2}	**2.8**(1)^{*2}	**2.8**(1)^{*2}	**2.8**(1)^{*2}	**2.8**(1)^{*2}	**2.8**(1)^{*2}	15/15
GWO	8.8(4)	31(20)	34(19)	34(19)	34(19)	34(19)	15/15
PSO	10(2)	15(5)	17(7)	17(9)	17(9)	17(9)	15/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f6	114	214	281	580	1038	1332	15/15
1,2 CMAES	4.0(3)	5.5(3)	7.1(8)	5.5(6)	4.5(4)	4.8(4)	15/15
1+2ms CMAES	**1.0**(0.3)^{*2}	**0.88**(0.2)^{*4}	**0.99**(0.2)^{*4}	**1.2**(0.4)^{*3}	**1.1**(1.0)^{*2}	**1.1**(0.7)^{*3}	15/15
GWO	2.4(1)	57(24)	267(224)	367(259)	∞	∞*1e5*	0/15
PSO	4.2(2)	9.4(3)	11(3)	12(2)	10(2)	11(2)	15/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f7	24	324	1171	1572	1572	1597	15/15
1,2 CMAES	14(9)	23(25)	89(107)	∞	∞	∞*1e5*	0/15
1+2ms CMAES	5.7(4)	**2.5**(1)	**2.7**(1)	**2.8**(2)^{*2}	**2.8**(2)^{*2}	**2.8**(2)^{*2}	15/15
GWO	**5.4**(3)	123(171)	68(85)	96(67)	96(67)	97(64)	9/15
PSO	10(6)	4.4(4)	54(85)	51(65)	51(65)	51(64)	9/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f8	73	273	336	391	410	422	15/15
1,2 CMAES	10(10)	16(18)	17(14)	18(12)	18(11)	18(11)	15/15
1+2ms CMAES	**2.5**(2)^{*2}	**3.7**(4)*	**3.7**(3)^{*3}	**3.7**(3)^{*4}	**3.8**(2)^{*4}	**3.9**(2)^{*4}	15/15
GWO	20(24)	205(137)	475(301)	∞	∞	∞*1e5*	0/15
PSO	15(6)	24(11)	94(80)	1812(1879)	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f9	35	127	214	300	335	369	15/15
1,2 CMAES	17(19)	23(24)	22(15)	20(12)	19(10)	18(9)	15/15
1+2ms CMAES	**3.3**(0.7)^{*3}	**8.4**(7)	**6.0**(4)^{*3}	**5.0**(3)^{*4}	**4.8**(3)^{*4}	**4.6**(3)^{*4}	15/15
GWO	7.4(3)	662(824)	∞	∞	∞	∞*1e5*	0/15
PSO	27(9)	327(412)	287(283)	4761(5160)	4367(4696)	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f10	349	500	574	626	829	880	15/15
1,2 CMAES	12(5)	10(2)	9.3(2)	9.3(2)	7.4(1)	7.2(1)	15/15
1+2ms CMAES	**2.0**(0.7)^{*4}	**1.7**(0.5)^{*4}	**1.7**(0.2)^{*4}	**1.7**(0.1)^{*4}	**1.4**(0.1)^{*4}	**1.4**(0.1)^{*4}	15/15
GWO	1347(1431)	∞	∞	∞	∞	∞*1e5*	0/15
PSO	514(473)	2811(3001)	2569(2874)	∞	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f11	143	202	763	1177	1467	1673	15/15
1,2 CMAES	33(16)	27(10)	7.7(3)	5.6(1)	4.7(1)	4.3(0.9)	15/15
1+2ms CMAES	**5.2**(3)^{*2}	**5.7**(2)^{*4}	**1.8**(0.3)^{*4}	**1.3**(0.1)^{*4}	**1.1**(0.1)^{*4}	**1.0**(0.1)^{*4}	15/15
GWO	480(701)	2101(2456)	∞	∞	∞	∞*1e5*	0/15
PSO	83(97)	222(168)	174(139)	∞	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f12	108	268	371	461	1303	1494	15/15
1,2 CMAES	42(67)	31(35)	28(30)	27(27)	11(11)	11(11)	15/15
1+2ms CMAES	**6.6**(8)*	**5.3**(5)^{*4}	**5.8**(5)^{*3}	**5.9**(4)^{*3}	**2.5**(2)^{*3}	**2.5**(2)^{*2}	15/15
GWO	1722(931)	1119(934)	4047(4317)	∞	∞	∞*1e5*	0/15
PSO	1092(1387)	5246(5601)	3793(4587)	3050(3359)	∞	∞*1e5*	0/15

Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f13	132	195	250	1310	1752	2255	15/15
1,2 CMAES	16(21)	51(50)	60(62)	21(15)	32(29)	57(67)	15/15
1+2ms CMAES	**3.8**(3)	**5.6**(4)^{*2}	**4.8**(3)^{*3}	**2.1**(1)^{*2}	**2.0**(1)^{*2}	**2.4**(2)^{*4}	15/15
GWO	552(492)	845(521)	3002(3002)	∞	∞	∞*1e5*	0/15
PSO	70(6)	1435(1804)	5627(6205)	∞	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f14	10	41	58	139	251	476	15/15
1,2 CMAES	6.0(5)	5.8(3)	6.8(2)	11(3)	15(3)	13(2)	15/15
1+2ms CMAES	**0.95**(1)	**1.5**(0.3)^{*3}	**2.0**(0.4)^{*4}	**2.6**(0.6)^{*4}	**3.3**(0.9)^{*4}	**2.8**(0.4)^{*4}	15/15
GWO	1.1(1)	3.2(1)	41(69)	712(41)	∞	∞*1e5*	0/15
PSO	1.9(3)	6.3(4)	16(4)	30(9)	488(513)	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f15	511	9310	19369	20073	20769	21359	14/15
1,2 CMAES	60(59)	158(156)	∞	∞	∞	∞*1e5*	0/15
1+2ms CMAES	**6.6**(7)	**17**(19)*	**74**(77)*	**71**(82)*	**69**(82)*	**67**(75)*	1/15
GWO	15(17)	80(86)	∞	∞	∞	∞*1e5*	0/15
PSO	15(14)	∞	∞	∞	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f16	120	612	2663	10449	11644	12095	15/15
1,2 CMAES	33(45)	410(462)	∞	∞	∞⋆3	∞*1e5*^{*3}	0/15
1+2ms CMAES	**1.2**(0.4)	**10**(11)	**28**(31)	143(148)	∞	∞*1e5*	0/15
GWO	2.4(3)	19(36)	33(21)	**24**(19)	∞	∞*1e5*	0/15
PSO	2.4(2)	16(5)	41(51)	42(48)	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f17	5.2	215	899	3669	6351	7934	15/15
1,2 CMAES	35(39)	66(99)	225(250)	∞	∞*	∞*1e5**	0/15
1+2ms CMAES	20(2)	12(12)	22(24)	∞	∞	∞*1e5*	0/15
GWO	**3.5**(3)	76(233)	95(67)	409(436)	∞	∞*1e5*	0/15
PSO	4.4(6)	**3.7**(1)	**20**(56)	**60**(68)	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f18	103	378	3968	9280	10905	12469	15/15
1,2 CMAES	10(9)	128(142)	∞	∞	∞	∞*1e5*	0/15
1+2ms CMAES	3.8(4)	34(54)	45(42)	∞	∞	∞*1e5*	0/15
GWO	**1.6**(1)	97(114)	**36**(25)	∞	∞	∞*1e5*	0/15
PSO	2.0(0.9)	**28**(32)	75(88)	∞	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f19	1	1	242	1.20E+05	1.20E+05	1.20E+05	15/15
1,2 CMAES	60(67)	1.3e4(1e4)	6119(6813)	∞	∞	∞*1e5*	0/15
1+2ms CMAES	**16**(15)	3389(2532)	585(662)	∞	∞	∞*1e5*	0/15
GWO	25(14)	**1228**(1377)	**288**(325)	∞	∞	∞*1e5*	0/15
PSO	26(24)	3819(3295)	742(863)	**5.8**(6)	**12**(14)	**12**(15)	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f20	16	851	38111	54470	54861	55313	14/15
1,2 CMAES	9.0(7)	19(20)	37(41)	26(30)	26(30)	26(29)	1/15
1+2ms CMAES	**2.9**(2)	7.3(7)	**7.8**(9)	**5.5**(7)	**5.4**(6)	**5.4**(6)	4/15
GWO	3.4(2)	89(95)	∞	∞	∞	∞*1e5*	0/15
PSO	4.5(6)	**2.8**(1)	17(21)	12(15)	12(14)	12(15)	2/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f24	1622	2.20E+05	6.40E+06	9.60E+06	1.30E+07	1.30E+07	3/15
1,2 CMAES	47(62)	∞	∞	∞	∞	∞*1e5*	0/15
1+2ms CMAES	**4.0**(5)	**6.8**(7)^{*2}	∞	∞	∞	∞*1e5*	0/15
GWO	54(63)	∞	∞	∞	∞	∞*1e5*	0/15
PSO	12(13)	∞	∞	∞	∞	∞*1e5*	0/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f21	41	1157	1674	1705	1729	1757	14/15
1,2 CMAES	2.8(3)	10(9)	33(40)	33(35)	32(35)	32(42)	12/15
1+2ms CMAES	3.2(1)	**4.3**(6)	**7.2**(10)	**7.1**(10)	**7.0**(10)	**6.9**(10)	15/15
GWO	**2.1**(2)	136(174)	168(211)	246(293)	265(293)	280(285)	3/15
PSO	2.4(2)	78(90)	241(299)	237(266)	235(289)	232(286)	3/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f22	71	386	938	1008	1040	1068	14/15
1,2 CMAES	23(36)	34(49)	39(56)	37(50)	36(48)	35(48)	13/15
1+2ms CMAES	5.6(12)	**4.5**(4)	**4.1**(5)	**3.9**(4)	**3.9**(4)⋆	**3.9**(4)⋆3	15/15
GWO	103(2)	133(259)	225(267)	430(494)	453(521)	463(422)	1/15
PSO	**1.8**(1)	41(130)	73(107)	70(100)	71(97)	74(96)	9/15
Δfopt	**1.00E+01**	**1.00E+00**	**1.00E-01**	**1.00E-03**	**1.00E-05**	**1.00E-07**	**#succ**
f23	3	518	14249	31654	33030	34256	15/15
1,2 CMAES	6.0(16)	95(81)	∞	∞	∞	∞*1e5*	0/15
1+2ms CMAES	3.0(3)	**1.5**(2)^{*3}	**8.8**(8)^{*3}	∞	∞	∞*1e5*	0/15
GWO	**2.4**(2)	50(58)	∞	∞	∞	∞*1e5*	0/15
PSO	2.5(3)	23(29)	102(107)	∞	∞	∞*1e5*	0/15

Figure 1: ALG0 = 1+2ms CMA-ES, ALG1 = PSO. Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to reach a target value $f_{opt} + \Delta f$ with $\Delta f = 10^k$,where k ∈ {1, −1, −4, −8} is given by the first value in the legend, for ALG1 (solid) and ALG0 (dashed). Light beige lines show the ECDF of FEvals for target value $\Delta f = 10^{-8}$ of algorithms benchmarked during BBOB-2009. Right sub-columns: ECDF of FEval ratios of ALG1 divided by ALG0, all trial pairs for each function. Pairs where both trials failed are disregarded, pairs where one trial failed are visible in the limits being > 0 or < 1. The legends indicate the number of functions that were solved in at least one trial (ALG1 first). The y-value at the transition between left and right sub-columns corresponds to the success probability of the trials.

The empirical (cumulative) distribution function F : R → [0, 1] is defined for a given set of real-valued data S, such that F(x) equals the fraction of elements in S which are smaller than x. The function F is monotonous and a lossless representation of the (unordered) set S (Nikolaus Hansen, 2010).

Figure 1 which shows ECDF for 1+2ms CMA-ES and PSO confirms that overall, the CMA-ES variant performs better than PSO optimizer. For dimension 5, the differences in the optimization of separable functions are not so significant and PSO still performs well relative to CMA-ES. However, in the case of ill-conditioned functions, CMA-ES hugely outranks PSO. A difference is also noticeable for the larger dimensions (20-D), especially in the case of ill-conditioned and moderate functions. For the more demanding, multi-modal problems, the results of both PSO and CMA-ES tested algorithms fall short especially in higher dimension 20-D.

From the post processed data, the difference in performance of the CMA-ES to PSO can be attributed to the various modifications of the CMA-ES algorithm which are considered to be among the top in the field of black-box optimization. PSO optimization techniques are also sensitive to hyperparameters of the algorithms and tuning of these parameters proved to be a challenging task. A better choice of the algorithm’s hyperparameters adapted to each function and dimension can seriously influence the final result. According to P. Szynkiewicz (2018), because CMA-ES does not require tedious parameter tuning, the choice of the strategy to be adopted while setting the internal parameters is not left to the user which makes it much more convenient than PSO.

Based on Table 1, a notable anomaly was observed for f3 and f4 mainly for both CMA-ES, whereas PSO was the best performing optimizer for both functions and managed to achieve the most successful trials. For Rastrigin function, f3 and Skewed Rastrigin function, f4 which are in a low dimension (5).n this case, the population size in both CMA-ES is small, hence the probability of reaching f_{target} is low: no successful trials were achieved within 20k evaluations. In contrast with the rest of the results for the other functions in 5-D, the CMA-ES variant optimizer generally performed better than all other optimizers tested.

CONCLUSION

This report documents the experimentation of selected stochastic optimisers for the single-objective continuous optimisation problems, namely the 1+2ms CMA-ES, 1,2 CMA-ES, PSO and GWO optimisers. In order to evaluate their performance, the BBOB-2010 benchmark test functions were utilised and tested against all selected optimisers. In this report, we organised the results of the optimisers after running the benchmark for dimension 5, then compared and analysed the difference in their results. Based on the performance reported, it is clear that the 1+2ms CMA-ES optimiser has the best performance among them, with GWO being the worst performing optimiser. 1+2ms CMA-ES has been shown to consistently solve the low and moderate and high functions (f6-f14) with minimal standard deviation. PSO performed excellently in separable functions (f1-f5), and GWO in Multi-modal(f15-f19). While the 1,2 CMA-ES performed well, 1+2ms CMA-ES performed slightly better overall .Based on the obtained results, we concluded that the best optimiser out of the 4, is the 1+2ms CMA-ES.

REFERENCES

Mohamed, A.B. (2018). Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications. Chapter 10: Metaheuristic Algorithms. Academic Press.

Diego.O. 2019. Metaheuristic Algorithms for Image Segmentation: Theory and Applications. Springer.

Rashedi, E., Nezamabadi, H., Saryazdi, S., (2009). Gravitational Search Algorithm. Information Sciences, 179, 2232-2248.

El-Abd, M., and Kamel, M. S. (2009). Black-Box Optimization Benchmarking for Noiseless Function Testbed using Particle Swarm Optimization. In Proceedings of the 2009 Genetic and Evolutionary Computation Conference. ACM, New York, NY, 2009, 2269–2273

S. Mirjalili, S. M. Mirjalili, A. Lewis, (2014). Grey Wolf Optimizer, Advances in Engineering Software, vol. 69, pp. 46-61

P. Szynkiewicz, A Comparative Study of PSO and CMA-ES Algorithms on Black-box Optimization Benchmarks, Journal of Telecommunications and Information Technology, vol. 8, pp. 1–13, 2018.

Blas, N.G. and de Mingo, L.F., 2010. BENCHMARK OF PSO-DE USING BBOB 2010. New Trends in Information Technologies, p.9

Brockhoff. D. 2011. Mirrored Sampling and Sequential Selection for Evolution Strategies. PPSN, Warsaw, Poland. Pp.11-21.

Steffen Finck and Raymond Ros, 2012. COCO (COmparing Continuous Optimizers) Software: User Documentation.

Nikolaus Hansen , Anne Auger , Steffen Finck and Raymond Ros, 2010.Real-Parameter Black-Box Optimization Benchmarking BBOB-2010: Experimental Setup

Source Code :

El-Abd, M., and Kamel, M. S. (2009). Black-Box Optimization Benchmarking for Noiseless Function Testbed using Particle Swarm Optimization. In Proceedings of the 2009 Genetic and Evolutionary Computation Conference. ACM, New York, NY, 2009, 2269–2273

Auger. A, Brockhoff. D, Hansen.N. (2010) Comparing the (1+1)-CMA-ES with a Mirrored (1+2)-CMA-ES with Sequential Selection on the Noiseless BBOB-2010 Testbed. GECCO workshop on Black-Box Optimization Benchmarking (BBOB'2010), pp.1543-1550.

Auger. A, Brockhoff. D, Hansen.N. (2010) Mirrored Variants of the (1,2)-CMA-ES Compared on the Noiseless BBOB-2010 Testbed. GECCO workshop on Black-Box Optimization Benchmarking (BBOB'2010), Jul 2010, Portland, OR, United States. pp.1551-1558

S. Mirjalili, S. M. Mirjalili, A. Lewis, (2014). Grey Wolf Optimizer, Advances in Engineering Software, vol. 69, pp. 46-61

Hansen, N. and S. Kern (2004). Evaluating the CMA Evolution Strategy on Multimodal Test Functions. Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings, pp. 282-291.