1. Introduction

This group report is an analysis of multiple stochastic optimizers submitted by the members of group 20 (excluding the leader). Each of the members will submit the best optimizer that they have tested based on the competition specification provided in the coursework. All the optimizers are to be compared between one another based on statistical analysis done in this report.

2. Abstract

The optimizer NIPOP-aCMA-ES is the best out of all the submitted optimizers (BIPOP-CMA-ES and DEAE) as it has the highest summation of dimensions scores compared to all other optimizers. The Δftarget output is used as the output for each iteration and benchmark function. The seed for the random number generator with the value of '1' is used. A maximum functional evaluation of 5000 for each iteration for 15 iterations across a total of 5 dimensions (dimension-2, dimension-3, dimension-5, dimension-10, dimension-20) is used for the benchmarking. The instances are starting from 1 to 15 (Hansen et al., 2010), as there will be multiple optimizers being submitted by multiple members of group 20. A way of determining the best optimizers will need to be found. Thus, all the optimizers will use the benchmark based on the Black-Box Optimization Benchmark 2010 (BBOB 2010) to compare which optimizer is the best, the benchmark will use the 24 noiseless functions. The findings in this report used are based on the leader's run on the optimizer, not the result given by the members as there is a difference between the result generated by the member and the result generated by the leader. The results are extracted inside the .info files generated by the optimizers.

3. Optimizer Algorithms

3.1. Covariance Matrix Adaptation - Evolutionary Strategy (CMA-ES)

CMA-ES is an evolutionary algorithm. Evolutionary algorithms are stochastic in nature which means that this algorithm will randomly create a set of candidate solutions and then apply a quality function to the set as an abstract fitness measure (Eiben and Smith, 2015). Furthermore, it is also used as a standard tool for a fairly large number of continuous optimisation environments throughout the globe (Hansen, 2006). The fitness values will then be evaluated and the candidates with the higher fitness ratings will be chosen as the seed for the next generation of candidates. The CMA-ES optimiser utilizes the Covariance Matrix for its evolutionary strategy. This optimizing framework possesses many appealing characteristics such as being covariant, derivative-free, scalable and many more. This optimiser is also extremely useful for obstacles that are non-convex, non-separable, ill-conditioned, multi-modal, and noisy evaluations (Dang, Vien and Chung, 2019). The pseudocode for the CMA-ES is shown in figure 3.1.1.

Step 1: Initialize the CMA-ES Parameters
- D ← no. of dimensions
- λ ← Offspring population size (4.0 + 3.0 Log(D))
- μ ← Parent population for next generation (floor(λ/2))
- σ_{start} ← Initial standard deviation.
- c_{cov} ← Covariance learning rate

Step 2: **While** stopping criterion is not met **do**

Step 3: Update the **Covariance Matrix** $C^{(g+1)}$ (see Covariance Matrix Adaption)

$$C^{(g+1)} \leftarrow (1 - c_{cov})C^{g} + \frac{c_{cov}}{\mu_{cov}} p_c^{(g+1)} p_c^{(g+1)^T} + c_{cov}\left(1 - \frac{1}{\mu_{cov}}\right) \times \sum_{i=1}^{\times} w_i \left(\frac{X_{1:\lambda}^{(g+1)} - M^{g}}{\sigma^{(g)}}\right)\left(\frac{X_{1:\lambda}^{(g+1)} - M^{g}}{\sigma^{(g)}}\right)^T$$

Step 4: Update the Step Size σ^{g} (see Step Size adaption)

$$\sigma_{g+1} \leftarrow \sigma_g \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{||P_\sigma||}{E\,N(0,I)} - 1\right)\right)$$

Step 5: Generate Sample Population for generation g+1

$$x_k^{(g+1)} \sim N(M^{(g)}, (\sigma^{(g)})^2 C^{(g)}) \quad \text{for } k = 1,\ldots,\lambda$$

Step 6: Update the mean for generation g+1

$$m^{(g+1)} \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)}$$

Step 7: Update best ever solution

Step 8: End **While**

Figure 3.1.1 pseudocode of the CMA-ES algorithm (Gagganapalli, 2015)

3.1.1. BI - Population Covariance Matrix Adaptation Evolution Strategy (BIPOP-CMA-ES)

BI-POP-CMA-ES is a variant of the CMA-ES algorithm that will run the algorithm with two possible regimes that will include the first regime with the population size doubled and the second regime with the population size reduced (Loshchilov, Schoenauer and Sèbag, 2013), A restart strategy has been proposed which could prevent premature convergence to local optima. This strategy is based on restarting and increasing the population size and initial mutation step-size each time once at least one stopping criterion of CMA-ES is met (Hansen, N., 2009). Figure 3.1.2 shows the pseudocode of the BIPOP-CMA-ES.

BiPop-(C)MA-ES for minimization

Initialize$(\mathbf{y}_{\text{lower}}, \mathbf{y}_{\text{upper}}, \sigma_{\text{init}}, \lambda_{\text{init}}, f_{\text{stop}}, b_{\text{stop}}, g_{\text{stop}} := \infty)$ (B1)
$\mathbf{y}_{\text{init}} := \mathbf{u}[\mathbf{y}_{\text{lower}}, \mathbf{y}_{\text{upper}}]$ (B2)
$(\mathbf{y}_{\text{min}}, f_{\text{min}}, b_1) :=$ (C)MA-ES$(\mathbf{y}_{\text{init}}, \sigma_{\text{init}}, \lambda_{\text{init}}, g_{\text{stop}}, TC)$ (B3)
If $f_{\text{min}} \leq f_{\text{stop}}$ **Then Return**$(\mathbf{y}_{\text{min}}, f_{\text{min}})$ (B4)
$n := 0;\ n_{\text{small}} := 0;$ (B5)
$b_{\text{large}} := 0;\ b_{\text{small}} := 0;$ (B6)
Repeat (B7)
  $n := n + 1$ (B8)
  $\lambda := 2^{n-n_{\text{small}}}\lambda_{\text{init}}$ (B9)
  $\mathbf{y}_{\text{init}} := \mathbf{u}[\mathbf{y}_{\text{lower}}, \mathbf{y}_{\text{upper}}]$ (B10)
  If $n > 2$ **AND** $b_{\text{small}} < b_{\text{large}}$ **Then** (B11)
    $\sigma_{\text{sInit}} := \sigma_{\text{init}}/100^{\mathbf{u}[0,1]}$ (B12)
    $\lambda_{\text{small}} := \left\lfloor \lambda_{\text{init}} \left(\frac{1}{2}\lambda/\lambda_{\text{init}}\right)^{(\mathbf{u}[0,1])^2} \right\rfloor$ (B13)
    $g_{\text{sStop}} := \left\lfloor \frac{1}{2} b_{\text{large}}/\lambda_{\text{small}} \right\rfloor$ (B14)
    $(\check{\mathbf{y}}, \check{f}, b_{\text{S}}) :=$ (C)MA-ES$(\mathbf{y}_{\text{init}}, \sigma_{\text{sInit}}, \lambda_{\text{small}}, g_{\text{sStop}}, TC)$ (B15)
    $b_{\text{small}} := b_{\text{small}} + b_{\text{S}}$ (B16)
    $n_{\text{small}} := n_{\text{small}} + 1$ (B17)
  Else (B18)
    $(\check{\mathbf{y}}, \check{f}, b_{\text{L}}) :=$ (C)MA-ES$(\mathbf{y}_{\text{init}}, \sigma_{\text{init}}, \lambda, g_{\text{stop}}, TC)$ (B19)
    $b_{\text{large}} := b_{\text{large}} + b_{\text{L}}$ (B20)
  EndIf (B21)
  If $\check{f} \leq f_{\text{min}}$ **Then** (B22)
    $\mathbf{y}_{\text{min}} := \check{\mathbf{y}};\ f_{\text{min}} := \check{f}$ (B23)
  EndIf (B24)
Until$(b_{\text{small}} + b_{\text{large}} + b_1 \geq b_{\text{stop}}$ OR $f_{\text{min}} \leq f_{\text{stop}})$ (B25)
Return$(\mathbf{y}_{\text{min}}, f_{\text{min}})$ (B26)

Figure 3.1.2 Pseudocode for BIPOP-CMA-ES (Beyer and Sendhoff, 2017)

BI-POP-CMA-ES also outperforms Cauchy Estimation-of-Distribution Algorithm (EDA) as it has a much higher reliability (success rate) and it outperforms in all dimensions, functions, and target levels, only a few exceptions in low dimensional functions. (Pošík, 2010).

In this report, two BIPOP-CMA-ES are used as two members of group 20 have submitted the same algorithm but different parameters for the competition. The difference between both members is that one of the members (20305249) set the stop fitness to 5000, while the other (20218675) has its stop fitness to be negative infinity, the stop fitness is one of the stopping criteria and to add on to the changes made, member 20218675 only has its setting of the random number generator only once, and it is done during the start of the optimizer, while member 20305249 sets the number generator seed every calling of the optimizer.

3.1.2. New Increasing Population size - active Covariance Matrix Adaptation - Evolution Strategy (NIPOP-aCMA-ES)

The NIPOP-aCMA-ES is another variant of CMA-ES. NIPOP-aCMA-ES stems from IPOP-CMA-ES which is introduced in the year 2005 with doubling the population size for each restart operation (Auger and Hansen, 2005). According to the results by Ros, (2010), the main differences between the BIPOP-CMA-ES and IPOP-CMA-ES are shown in their performances in the multi-modal functions and the weakly-structured functions. When the IPOP-CMA-ES is compared to the BIPOP-CMA-ES on the noiseless testbed, the IPOP was able to perform at best twice as fast compared to BIPOP on multi-modal functions f15 to f19, also f13 and f7 while it did not solve the weakly-structured functions f22 to f24 when BIPOP did give that the dimension of the search space was more than 10 but IPOP-CMA-ES performed less efficient on f19 if the dimension is lower than 5 (Ros, 2010). According to Loshchilov, Schoenauer and Sebag, (2012), the IPOP-CMA-ES showed relatively outstanding results in the BBOB-2009 and BBOB-2010.

NIPOP-aCMA-ES decreases the initial step size by a factor of 1.6 which is also used in NBIPOP-aCMA-ES but the budget for restart strategies is dependent on their performance in which the better performing restart strategy will be allocated a budget twice as large as the other (Loshchilov, Schoenauer and Sebag, 2012) where aCMA-ES means a weighted negative update of the covariance matrix of the CMA-ES algorithm, the NBIPOP also lets the budget to be adaptive for both restart schemes according to the performance of the respective scheme.

3.2. Differential Evolution with Adaptive Encoding (DEAE)

Differential Evolution with Adaptive Encoding is one of the variants in Differential Evolution (DE), which helps to improve the optimization process. DE was initially introduced by Storn, R and Price, K in 1997 and has been used in many scientific and engineering fields. DE is a stochastic population-based metaheuristic search algorithm that optimizes a problem by iteratively improving a candidate solution based on an evolutionary process (Georgioudakis and Plevris, 2020). DE is well known for its simplicity and efficiency on many practical problems. It uses a mutation operator followed by a crossover operator to generate offspring. The crossover operators in DE are rotation dependent, but mutation operators are rotationally invariant. On separable functions, the crossover helps to properly mix the good values of solution components in the population. On non-separable functions, however, it mostly only destroys the potentially good combinations of values generated by the mutation (Poaík and Klema, 2012). DEAE also performs better compared to other optimizers like Particle Swarm Optimizer (PSO) and Genetic Algorithms (GA) in terms of less clustering and re-initialization effect, with the addition of able to reach a good solution without local search (Kachitvichyanukul, 2012).

One of the solutions to overcome the crossover issues with non-separable functions is by using Adaptive Encoding (AE) to adjust the search space representation. Adaptive Encoding (AE) was introduced in 2008 to boost the performance of continuous domain search algorithms by finding a good representation of solutions. In optimization and search, the choice of the representation of the optimization problem is crucial (Hansen, 2008). If we were able to perform the crossover in a suitable coordinate system, we may enjoy the benefits of crossover even for the non-separable functions (Poaík and Klema, 2012). The pseudocode for DEAE is shown in figure 3.2

```
Initialize the population P ← {x_i}_{i=1}^{NP}.
Initialize the transformation matrix B ∈ ℝ^{D×D}
while stopping criteria not met do
    Transform P: P′ ← {x′_i | x′_i ← B^{-1} x_i}.
    for i ← 1 to NP do
        v′_i ← mutate(i, P′) (Eq. 1)
        u′_i ← crossover(x′_i, v′_i) (Eq. 2)
        Transform offspring back: u_i ← B u′_i.
        if f(u_i) < f(x_i) then
            x_i ← u_i
        end
    end
    B ← update(B, x_(1), ..., x_(μ))
end
```

Figure 3.2 Pseudocode for DEAE (Poaík, and Klema, 2012)

4. Results

After running all the optimizers, all the results are extracted and placed into the FSMap. The scores are made into a graphical format. A bar chart will show the overall performance of the optimizer in each dimension and overall (Figure 4.1) based on the scores. While the Δftarget is made into tables (Table 4.1, Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6) which are used to get the average and standard deviation of the 15 iterations for all noiseless functions in all 5 dimensions and overall dimensions.

The most noteworthy result that can be seen at a glance of the eye in Figure 4.1 is that the higher the dimensions the lower the performance of the optimizers. It also must be noted the performance of BIPOP-CMA-ES and DEAE dropped much faster between dimensions 5 and 10, while NIPOP-aCMA-ES have a considerable drop in performance between dimensions 10 and 20. It could also be seen that DEAE performs the best compared to other optimizers in lower dimensions (dimensions 2, 3, and 5). While NIPOP-aCMA-ES performs better compared to other optimizers in higher dimensions (dimensions 10 and 20). NIPOP-aCMA-ES performs similarly compared to BIPOP-CMA-ES in the lower dimensions. Lastly, BIPOP-CMA-ES with the parameter set by member 20218675 performs better compared to member 20305249 in all dimensions except dimension 10.

All optimizers perform great in functions 1 and function 5 in all dimensions except DEAE which does not perform well in dimension 20 compared to all other dimensions (Table 4.1, Table 4.2, Table 4.3, Table 4.4 and Table 4.5). In terms of consistency, there are only a few functions like f1, f14, and f20 which have (arguably) low standard deviation (Table 4.6).

Overall, BIPOP-CMA-ES and NIPOP-aCMA-ES show relatively similar results in terms of reliability, score, and Δftarget in dimensions 2, 3, and 5. The difference starts to be apparent at dimension 10 where NIPOP-aCMA-ES performs the best in f2 and f10 (Table 4.5).

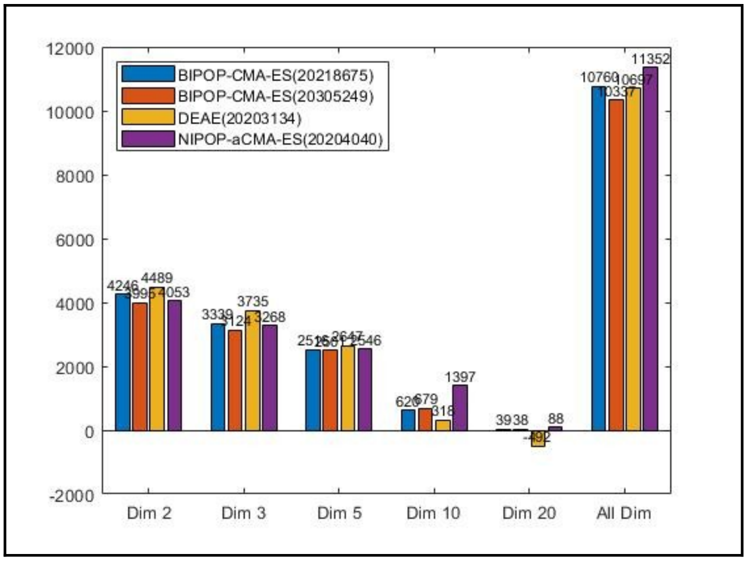

Figure 4.1 Bar Chart on the scores in each dimension and overall scores.

	Optimizers							
	BIPOP (20218675)		BIPOP (20305249)		NIPOP (20204040)		DEAE (20203134)	
Functions	Avg	std	Avg	std	Avg	std	Avg	std
Separable Function								
f1	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f2	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f3	2.20E-01	3.88E-01	4.23E-01	4.87E-01	1.77E-01	3.64E-01	1.00E-14	0.00E+00
f4	7.20E-01	5.60E-01	4.59E-01	5.08E-01	7.07E-01	6.98E-01	1.32E-01	3.48E-01
f5	1.03E-14	1.03E-15	1.03E-14	1.03E-15	1.03E-14	1.03E-15	1.03E-14	1.03E-15
Low or Moderate Condition Function								
f6	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f7	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f8	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f9	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
High Condition Function and Unimodal								
f10	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f11	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f12	5.53E-10	2.14E-09	5.73E-03	2.22E-02	2.33E-10	9.04E-10	1.80E-05	6.97E-05
f13	1.02E-12	1.62E-12	3.03E-13	4.51E-13	1.71E-12	2.10E-12	1.00E-14	0.00E+00
f14	7.27E-14	8.35E-14	2.23E-13	1.43E-13	1.18E-13	9.28E-14	1.00E-14	0.00E+00
Multi-Modal Functions								
f15	6.62E-02	2.56E-01	3.39E-01	4.34E-01	2.31E-01	3.60E-01	1.98E-01	4.10E-01
f16	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f17	4.04E-13	2.49E-13	1.07E-05	3.86E-05	2.53E-06	7.50E-06	1.93E-06	7.49E-06
f18	1.93E-06	7.49E-06	1.10E-01	3.85E-01	2.67E-06	7.82E-06	2.67E-01	1.03E+00
f19	3.21E-03	6.62E-03	3.73E-03	8.40E-03	6.93E-03	1.15E-02	3.27E-03	6.60E-03
Multi-Modal with Weak Global Structure Functions								
f20	3.63E-02	9.39E-02	1.40E-01	2.40E-01	1.53E-01	2.37E-01	2.13E-10	8.26E-10
f21	9.67E-03	3.35E-02	8.00E-03	3.10E-02	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f22	1.00E-14	0.00E+00	1.00E-14	0.00E+00	6.67E-04	2.58E-03	1.00E-14	0.00E+00
f23	1.26E-03	4.64E-03	1.61E-01	3.37E-01	2.51E-01	4.61E-01	2.79E-01	4.11E-01
f24	7.02E-01	8.26E-01	1.04E+00	7.62E-01	8.40E-01	8.17E-01	1.48E+00	8.87E-01

Table 4.1 average and standard deviation of Δftarget

for 15 iterations (Dim 2)

	Optimizers							
	BIPOP (20218675)		BIPOP (20305249)		NIPOP (20204040)		DEAE (20203134)	
Functions	Avg	std	Avg	std	Avg	std	Avg	std
Separable Function								
f1	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f2	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f3	7.95E-01	6.74E-01	1.55E+00	7.52E-01	8.28E-01	5.81E-01	2.65E-01	4.54E-01
f4	2.40E+00	1.64E+00	2.28E+00	1.54E+00	3.15E+00	1.33E+00	1.05E+00	9.22E-01
f5	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
Low or Moderate Condition Function								
f6	1.09E-14	2.47E-15	1.03E-14	0.00E+00	1.05E-14	1.41E-15	1.00E-14	0.00E+00
f7	1.00E-14	0.00E+00	7.33E-09	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f8	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f9	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
High Condition Function and Unimodal								
f10	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f11	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f12	1.73E-03	6.71E-03	3.94E-02	1.52E-01	2.13E-02	8.26E-02	1.00E-01	3.87E-01
f13	1.75E-12	2.18E-12	1.10E-11	0.00E+00	3.28E-12	4.44E-12	2.62E-14	2.11E-14
f14	1.78E-13	1.26E-13	2.07E-13	0.00E+00	2.01E-13	2.40E-13	1.00E-14	0.00E+00
Multi-Modal Functions								
f15	7.95E-01	6.74E-01	1.40E+00	8.16E-01	7.27E-01	5.91E-01	6.61E-01	6.14E-01
f16	1.81E-02	5.48E-02	1.81E-02	6.97E-02	1.13E-02	4.39E-02	3.03E-01	8.01E-01
f17	8.89E-06	2.44E-05	2.32E-05	0.00E+00	1.71E-06	4.39E-06	1.00E-14	0.00E+00
f18	5.95E-02	9.62E-02	1.84E-01	3.75E-01	1.13E-01	3.33E-01	1.93E-06	7.49E-06
f19	8.14E-02	9.41E-02	2.64E-01	3.94E-01	1.10E-01	1.11E-01	3.11E-02	3.00E-02
Multi-Modal with Weak Global Structure Functions								
f20	5.85E-01	4.48E-01	5.95E-01	3.41E-01	6.33E-01	3.53E-01	2.83E-01	3.07E-01
f21	2.59E-01	5.11E-01	2.00E-01	3.48E-01	9.20E-02	2.43E-01	9.20E-02	2.43E-01
f22	1.91E-01	3.12E-01	4.85E-01	9.03E-01	1.41E-01	2.47E-01	1.38E-01	2.86E-01
f23	5.23E-01	5.18E-01	3.55E-01	1.41E+00	4.01E-01	5.69E-01	8.51E-01	4.42E-01
f24	2.89E+00	1.92E+00	2.89E+00	7.62E-01	2.45E+00	1.36E+00	3.33E+00	1.11E+00

Table 4.2 average and standard deviation of Δftarget

for 15 iterations (Dim 3)

Functions	Optimizers							
	BIPOP (20218675)		BIPOP (20305249)		NIPOP (20204040)		DEAE (20203134)	
	Avg	std	Avg	std	Avg	std	Avg	std
Separable Function								
f1	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f2	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f3	3.00E+00	2.10E+00	3.00E+00	1.85E+00	3.30E+00	2.25E+00	3.53E+00	2.63E+00
f4	5.65E+00	1.91E+00	6.66E+00	3.77E+00	5.17E+00	1.19E+00	8.99E+00	9.09E+00
f5	1.03E-14	1.05E-15	1.03E-14	1.05E-15	1.03E-14	1.05E-15	1.03E-14	1.05E-15
Low or Moderate Condition Function								
f6	3.56E-14	3.37E-14	4.07E-14	2.74E-14	2.05E-14	1.35E-14	2.75E-12	3.76E-12
f7	2.20E-02	7.03E-02	3.50E-02	1.02E-01	1.49E-13	5.40E-13	4.87E-02	1.88E-01
f8	8.00E-09	3.10E-08	1.00E-14	0.00E+00	2.60E-01	1.01E+00	2.93E-01	7.75E-01
f9	1.00E-14	0.00E+00	1.12E-14	4.65E-15	1.59E-14	2.30E-14	3.00E-01	8.41E-01
High Condition Function and Unimodal								
f10	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f11	1.00E-14	0.00E+00	1.03E-14	1.03E-15	1.00E-14	0.00E+00	1.00E-14	0.00E+00
f12	3.17E-01	1.01E+00	1.24E+00	4.38E+00	9.30E-02	3.35E-01	6.03E-02	2.11E-01
f13	2.82E-11	4.00E-11	9.69E-08	2.49E-07	1.46E-10	3.56E-10	3.65E-13	2.59E-13
f14	4.68E-13	2.99E-13	4.25E-13	2.27E-13	5.25E-13	2.39E-13	1.00E-14	0.00E+00
Multi-Modal Functions								
f15	2.86E+00	1.64E+00	2.40E+00	1.69E+00	2.45E+00	1.65E+00	4.26E+00	3.75E+00
f16	2.08E-01	5.15E-01	4.29E-01	5.44E-01	3.35E-01	4.13E-01	2.81E-01	4.76E-01
f17	3.88E-04	1.03E-03	1.03E-03	2.57E-03	2.13E-03	4.70E-03	9.57E-03	2.45E-02
f18	5.98E-02	7.70E-02	1.19E-01	3.57E-01	5.63E-02	1.13E-01	9.16E-04	2.63E-03
f19	4.01E-01	2.45E-01	4.47E-01	3.30E-01	5.42E-01	3.53E-01	7.99E-01	2.92E-01
Multi-Modal with Weak Global Structure Functions								
f20	1.05E+00	3.12E-01	1.21E+00	2.47E-01	1.06E+00	3.46E-01	5.91E-01	3.35E-01
f21	1.20E+00	7.93E-01	7.43E-01	9.95E-01	9.77E-01	1.03E+00	5.53E-01	7.51E-01
f22	1.98E+00	2.30E+00	8.09E-01	8.03E-01	2.00E+00	2.82E+00	6.48E-01	1.04E+00
f23	1.29E+00	7.35E-01	5.87E-01	5.46E-01	8.21E-01	6.71E-01	1.42E+00	3.63E-01
f24	7.06E+00	2.03E+00	7.87E+00	2.68E+00	6.41E+00	3.48E+00	1.15E+01	3.42E+00

Table 4.3 average and standard deviation of Δftarget for 15 iterations (Dim 5)

Functions	Optimizers							
	BIPOP (20218675)		BIPOP (20305249)		NIPOP (20204040)		DEAE (20203134)	
	Avg	std	Avg	std	Avg	std	Avg	std
Separable Function								
f1	1.00E-14	0.00E+00	1.00E-14	0.00E+00	1.00E-14	0.00E+00	8.09E-13	2.12E-12
f2	1.30E+00	2.26E+00	2.24E-01	3.34E-01	1.19E-10	4.38E-10	7.12E+00	1.17E+01
f3	1.21E+01	4.98E+00	1.23E+01	3.45E+00	1.25E+01	6.94E+00	3.79E+01	2.06E+01
f4	1.61E+01	7.21E+00	2.19E+01	8.75E+00	1.53E+01	5.78E+00	4.97E+01	2.52E+01
f5	1.06E-14	1.40E-15	1.06E-14	1.40E-15	1.06E-14	1.40E-15	1.06E-14	1.40E-15
Low or Moderate Condition Function								
f6	2.21E-08	2.64E-08	2.76E-09	5.82E-09	5.66E-10	1.69E-09	4.95E-03	1.12E-02
f7	5.58E-01	3.14E-01	1.17E+00	8.71E-01	1.47E-01	2.13E-01	4.73E+00	6.43E+00
f8	9.49E-01	1.34E+00	1.68E-01	2.95E-01	8.62E-01	1.64E+00	4.12E+00	1.49E+00
f9	5.85E-01	8.04E-01	5.71E-01	1.42E+00	3.48E-01	1.06E+00	3.74E+00	2.06E+00
High Condition Function and Unimodal								
f10	1.72E+00	4.62E+00	6.27E-01	1.49E+00	2.20E-10	7.45E-10	3.18E+01	7.87E+01
f11	2.39E-02	6.97E-02	5.78E-03	2.19E-02	1.72E-14	2.57E-14	4.91E+00	8.17E+00
f12	2.30E+00	5.49E+00	3.81E+00	5.73E+00	2.31E-01	4.13E-01	9.11E+00	1.21E+01
f13	2.96E-01	5.25E-01	1.99E-01	3.89E-01	2.21E-02	4.55E-02	1.15E+00	3.44E+00
f14	1.14E-06	9.75E-07	9.09E-07	7.67E-07	1.20E-09	8.98E-10	6.35E-05	8.55E-05
Multi-Modal Functions								
f15	6.93E+00	4.54E+00	9.79E+00	4.55E+00	9.77E+00	7.03E+00	4.25E+01	1.03E+01
f16	1.42E+00	1.48E+00	3.25E+00	2.72E+00	3.86E+00	4.90E+00	1.02E+01	4.48E+00
f17	5.33E-03	8.86E-03	5.11E-02	1.34E-01	4.62E-01	7.36E-01	3.14E-01	2.94E-01
f18	2.72E-01	5.72E-01	2.77E-01	3.26E-01	1.31E+00	2.26E+00	1.30E+00	1.21E+00
f19	1.96E+00	9.37E-01	2.39E+00	9.59E-01	1.76E+00	9.65E-01	2.87E+00	5.56E-01
Multi-Modal with Weak Global Structure Functions								
f20	1.87E+00	3.06E-01	1.67E+00	3.20E-01	1.50E+00	3.23E-01	1.21E+00	3.06E-01
f21	1.75E+00	1.16E+00	1.82E+00	1.90E+00	1.65E+00	1.97E+00	4.04E+00	4.73E+00
f22	5.75E+00	7.92E+00	6.69E+00	6.24E+00	6.66E+00	8.80E+00	5.79E+00	1.08E+01
f23	2.13E+00	4.71E-01	1.78E+00	5.22E-01	2.05E+00	6.31E-01	1.95E+00	3.98E-01
f24	2.90E+01	1.41E+01	3.47E+01	1.45E+01	3.31E+01	7.39E+00	5.23E+01	8.63E+00

Table 4.4 average and standard deviation of Δftarget for 15 iterations (Dim 10)

Functions	Optimizers							
	BIPOP (20218675)		BIPOP (20305249)		NIPOP (20204040)		DEAE (20203134)	
	Avg	std	Avg	std	Avg	std	Avg	std
Separable Function								
f1	3.77E-14	2.69E-14	2.27E-14	2.73E-14	2.51E-14	2.69E-14	1.15E+00	1.31E+00
f2	1.00E+03	0.00E+00	1.00E+03	0.00E+00	3.74E+02	3.04E+02	9.63E+02	1.05E+02
f3	2.93E+01	7.39E+00	3.62E+01	7.71E+00	3.85E+01	1.65E+01	1.81E+02	7.03E+01
f4	5.03E+01	1.37E+01	5.56E+01	1.45E+01	5.53E+01	1.93E+01	2.59E+02	6.11E+01
f5	2.82E-14	3.39E-14	2.82E-14	3.39E-14	2.82E-14	3.39E-14	5.00E+00	1.66E+01
Low or Moderate Condition Function								
f6	4.74E-02	5.19E-02	1.85E-02	2.75E-02	4.53E-03	4.78E-03	1.08E+02	7.06E+01
f7	4.87E+00	2.37E+00	4.69E+00	2.93E+00	2.87E+00	1.34E+00	6.87E+01	5.68E+01
f8	1.49E+01	3.88E+00	1.33E+01	1.44E+00	1.63E+01	1.61E+01	1.59E+02	1.94E+02
f9	1.84E+01	1.60E+01	1.65E+01	1.57E+01	1.58E+01	1.65E+01	6.17E+01	3.22E+01
High Condition Function and Unimodal								
f10	9.85E+02	5.68E+01	9.33E+02	1.81E+02	4.15E+02	2.67E+02	9.78E+02	7.03E+01
f11	3.77E+02	2.07E+02	4.73E+02	2.17E+02	1.82E+00	1.71E+00	4.82E+01	2.84E+01
f12	6.45E+00	9.32E+00	2.88E+00	3.82E+00	6.71E+00	1.08E+01	1.00E+03	0.00E+00
f13	2.15E+00	3.64E+00	4.80E+00	6.02E+00	4.08E+00	4.81E+00	1.38E+02	1.43E+02
f14	4.89E-04	1.84E-04	3.47E-04	9.84E-05	4.29E-05	1.28E-05	1.41E+00	1.38E+00
Multi-Modal Functions								
f15	2.83E+01	1.42E+01	2.82E+01	7.80E+00	3.56E+01	1.30E+01	2.16E+02	7.06E+01
f16	4.33E+00	7.32E+00	3.27E+00	1.90E+00	6.07E+00	3.25E+00	2.71E+01	2.71E+00
f17	1.59E-02	2.76E-02	1.27E-01	1.52E-01	2.33E-01	2.73E-01	3.36E+00	1.29E+00
f18	2.96E-01	3.35E-01	5.47E-01	7.04E-01	1.58E+00	2.66E+00	1.39E+01	4.26E+00
f19	4.93E+00	6.69E-01	4.78E+00	3.84E-01	4.97E+00	4.73E-01	5.13E+00	5.55E-01
Multi-Modal with Weak Global Structure Functions								
f20	1.98E+00	3.47E-01	1.87E+00	2.72E-01	1.77E+00	3.31E-01	1.77E+00	3.22E-01
f21	6.80E+00	7.01E+00	6.41E+00	7.32E+00	6.48E+00	6.84E+00	9.84E+00	1.31E+01
f22	1.23E+01	1.08E+01	1.54E+01	2.10E+01	1.05E+01	9.97E+00	1.07E+01	1.47E+01
f23	3.12E+00	5.91E-01	3.01E+00	3.83E-01	2.91E+00	8.85E-01	3.35E+00	4.81E-01
f24	1.20E+02	1.81E+01	1.26E+02	2.72E+01	1.14E+02	4.53E+01	1.67E+02	1.39E+01

Table 4.5 average and standard deviation of Δftarget for 15 iterations (Dim 20)

Functions	Optimizers							
	BIPOP (20218675)		BIPOP (20305249)		NIPOP (20204040)		DEAE (20203134)	
	Avg	std	Avg	std	Avg	std	Avg	std
Separable Function								
f1	1.55E-14	1.61E-14	1.25E-14	1.29E-14	1.30E-14	1.32E-14	2.30E-01	7.35E-01
f2	2.00E+02	4.03E+02	2.00E+02	4.03E+02	7.48E+01	2.01E+02	1.94E+02	3.90E+02
f3	9.08E+00	1.18E+01	1.07E+01	1.40E+01	1.11E+01	1.65E+01	4.46E+01	7.71E+01
f4	1.50E+01	1.98E+01	1.74E+01	2.20E+01	1.59E+01	2.23E+01	6.38E+01	1.04E+02
f5	1.39E-14	1.65E-14	1.39E-14	1.65E-14	1.39E-14	1.65E-14	1.00E+00	7.48E+00
Low or Moderate Condition Function								
f6	9.48E-03	2.96E-02	3.69E-03	1.41E-02	9.05E-04	2.77E-03	2.17E+01	5.33E+01
f7	1.09E+00	2.18E+00	1.18E+00	2.26E+00	6.04E-01	1.29E+00	1.47E+01	3.69E+01
f8	3.18E+00	6.19E+00	2.69E+00	5.36E+00	3.48E+00	9.54E+00	3.28E+01	1.06E+02
f9	3.80E+00	1.01E+01	3.42E+00	9.51E+00	3.22E+00	9.56E+00	1.32E+01	2.82E+01
High Condition Function and Unimodal								
f10	1.97E+02	3.97E+02	1.87E+02	3.84E+02	8.30E+01	2.03E+02	2.02E+02	3.94E+02
f11	7.55E+01	1.77E+02	9.45E+01	2.12E+02	3.64E-01	1.04E+00	1.06E+01	2.30E+01
f12	1.81E+00	5.34E+00	1.60E+00	3.87E+00	1.41E+00	5.41E+00	2.02E+02	4.02E+02
f13	4.88E-01	1.81E+00	1.00E+00	3.25E+00	8.21E-01	2.66E+00	2.78E+01	8.33E+01
f14	9.80E-05	2.12E-04	6.95E-05	1.46E-04	8.59E-06	1.82E-05	2.83E-01	8.28E-01
Multi-Modal Functions								
f15	7.80E+00	1.25E+01	8.42E+00	1.12E+01	9.76E+00	1.49E+01	5.27E+01	8.92E+01
f16	1.20E+00	3.66E+00	1.39E+00	2.12E+00	2.06E+00	3.58E+00	7.58E+00	1.08E+01
f17	4.32E-03	1.40E-02	3.57E-02	1.01E-01	1.39E-01	3.89E-01	7.37E-01	1.45E+00
f18	1.37E-01	3.18E-01	2.47E-01	4.68E-01	6.10E-01	1.67E+00	3.09E+00	5.81E+00
f19	1.47E+00	1.95E+00	1.58E+00	1.89E+00	1.48E+00	1.93E+00	1.77E+00	2.03E+00
Multi-Modal with Weak Global Structure Functions								
f20	1.10E+00	8.11E-01	1.10E+00	7.10E-01	1.02E+00	6.63E-01	7.71E-01	7.01E-01
f21	2.00E+00	3.99E+00	1.84E+00	4.09E+00	1.84E+00	3.95E+00	2.90E+00	7.17E+00
f22	4.05E+00	7.52E+00	4.68E+00	1.12E+01	3.85E+00	7.20E+00	3.45E+00	9.01E+00
f23	1.41E+00	1.23E+00	1.18E+00	1.17E+00	1.29E+00	1.22E+00	1.57E+00	1.13E+00
f24	3.19E+01	4.66E+01	3.45E+01	4.95E+01	3.14E+01	4.78E+01	4.72E+01	6.37E+01

Table 4.6 average and standard deviation of Δftarget for 15 iterations (Overall)

5. Discussion

During the generation of the result, the algorithm NIPOP-aCMA-ES produces different outputs after every full run. The reason why it generates different outputs is that the seed is being set into a global state which is being reused every run. For this issue, all the results in this report generated from the optimizers will be based on the first run (running the experiment after a fresh restart of MATLAB).

This report also uses the leader generated results as some of the member's results do not produce the same result as the leader generated result. The reason why it uses the leader generated result is to standardize the performance of the device, and also prevent tampering with the results given by the members if there are any.

There are some cases where the functional evaluation has gone over 5000 slightly, the reason for the occurrence is that the calling of the optimizer will undergo multiple evaluations inside the optimizer and the population size does play a factor in this issue as the population size is used to generate the amount of offspring which is being used to undergo evaluation. Which in turn will make the functional evaluation a little over the max limit.

In lower dimensions (Table 4.1, Table 4.2 and Table 4.3), DEAE performs well, this could be contributed to the fact that there aren't many options (elements) for the DEAE to undergo mutation on the parents to generate offspring and limited options for crossing-over choices to allow improvement in the offsprings. In a higher dimension, where the options are much greater, it makes the mutation and crossing over to be much more troublesome as there are many permutations available. The opposite could also hold true, as there are higher dimensions, more crossing-over and mutation combinations are available. Thus, the option of changing the elements increases, increasing the permutations. This in turn makes searching for the optimum permutation harder, this would also cause the standard deviation to increase as the dimension increases from the lower dimensions (Table 4.1, Table 4.2, Table 4.3) to the higher dimensions (Table 4.4, Table 4.5) as the optimizer is trying to find the optimum permutation (global optimum) but failed as shown in the average in each table. According to Klemš, (2011) his findings show that the use of the exponential crossover may allow a better performance compared to binomial crossover in higher dimensions.

In higher dimensions (Table 4.4 and Table 4.5) and overall (Table 4.6) for all unimodal functions (f1, f2, f6, f10-f14) NIPOP-aCMA-ES perform better compared to BIPOP-CMA-ES and DEAE. This is because NIPOP-aCMA-ES increases the variance in the direction of the successful steps or decreases the variance when it is getting unsuccessful steps based on the aCMA property (Jastrebsk and Arnold, 2006) this would help on the unimodal functions as NIPOP-aCMA-ES could decrease the variance if it has generated offsprings which are over the global optimum (unsuccessful steps). Lower step size in NIPOP-aCMA-ES also helps as it will cause the selection of the range of numbers used for the offspring to begin to be smaller much faster compared to BIPOP-CMA-ES, this would also mean it will be beneficial for separable functions as the search space is limited this will allow the searching of the optimum (global or local) to be much faster (for every subsequent dimension which has reached optimal).

6. Conclusion

NIPOP-aCMA-ES outperforms the other optimizers in higher dimensions while DEAE performs best in lower dimensions compared to all other optimizers. But overall, NIPOP-aCMA-ES performs well as it has the highest score out of all optimizers. The setting of the random number generator also played a factor in affecting the performance of BIPOP-CMA-ES as setting the random number generator once yielded better performance in dimensions 2, 3, 5, 20 compared to setting the number generator every calling of the optimizer.

7. References

Auger, A and Hansen, N (2005), A Restart CMA Evolution Strategy with Increasing Population Size, *IEEE Congress on Evolutionary Computation*, pp. 1769-1776, doi: [10.1109/CEC.2005.1554902](https://doi.org/10.1109/CEC.2005.1554902).

Beyer. H and Sendhoff. B (2017), Simplify Your Covariance Matrix Adaptation Evolution Strategy, *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 746-759, doi: 10.1109/TEVC.2017.2680320.

Georgioudakis. M and Plevris V. (2020), A Comparative Study of Differential Evolution Variants in Constrained Structural Optimization. *Frontiers in Built Environment*. vol. 6, no.102, doi: 10.3389/fbuil.2020.00102.

Hansen, N. (2008). Adaptive encoding: How to render search coordinate system invariant. *Parallel Problem Solving from Nature – PPSN X*, Dortmund, Germany, pp 205–214. (inria-00287351v3)

Hansen, N. (2009), Benchmarking a Bi-population CMA-ES on the BBOB-2009 function testbed. GECCO (Companion), *GECCO '09: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers,* Montréal, Canada, pp 2389–2396, doi: 10.1145/1570256.1570333.

Hansen, N., Auger, A., Finck, S., and Ros, R. (2010), *Real-Parameter Black-Box Optimization Benchmarking 2010: Experimental Setup*. (inria-00462481)

Hansen, N. (2006), *The CMA Evolution Strategy*, viewed 26th April, <https://cma-es.github.io/>.

Hansen, N., (2016), The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*.

Jastrebski. G. A and Arnold. D. V. (2006), Improving Evolution Strategies through Active Covariance Matrix Adaptation, *2006 IEEE International Conference on Evolutionary Computation*, pp. 2814-2821, doi: 10.1109/CEC.2006.1688662.

Klemš. V, (2011), Differential evolution with adaptive encoding. Master's thesis, Czech Technical University in Prague. Available at: https://cyber.felk.cvut.cz/thesis/detail.phtml?id=177

Kachitvichyanukul, V. (2012), Comparison of Three Evolutionary Algorithms: GA, PSO, and DE, *Industrial Engineering and Management Systems*. Korean Institute of Industrial Engineers. doi: 10.7232/iems.2012.11.3.215

Loshchilov, I, Schoenauer. M and Sebag. M (2012), Black-box optimization benchmarking of NIPOP-aCMA-ES and NBIPOP-aCMA-ES on the BBOB-2012 noiseless testbed, *Genetic and Evolutionary Computation Conference*, New York, pp. 269-276, doi: 10.1145/2330784.2330823

Loshchilov, I., Schoenauer, M., and Sèbag, M. (2013). Bi-population CMA-ES algorithms with surrogate models and line searches. *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*., Amsterdam, The Netherlands, pp. 1177-1184, doi: 10.1145/2464576.2482696

Pošík, P. (2010). Comparison of Cauchy EDA and BIPOP-CMA-ES algorithms on the BBOB noiseless testbed. *GECCO '10: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, New York, pp. 1697-1702. Doi: 10.1145/1830761.1830791.

Poaík, P., and Klema, V. (2012). Benchmarking the differential evolution with adaptive encoding on noiseless functions. *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion - GECCO Companion '12*. doi: 10.1145/2330784.2330813

Ros, R, (2010), Black-box optimization benchmarking the IPOP-CMA-ES on the noiseless testbed: comparison to the BIPOP-CMA-ES, *Genetic and Evolutionary Computation Conference*, New York, pp, 1511-1578, doi: 10.1145/1830761.1830766

Storn, R., and Price, K. (1995). Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Technical Report, *Journal of Global Optimization 11*, Berkeley, CA. Available at: https://www.researchgate.net/publication/246648860_Differential_Evolution_A_Simple_and_Efficient_Adaptive_Scheme_for_Global_Optimization_Over_Continuous_Space