**SPRING SEMESTER 2021/22**

**COMP2024 Coursework**

|  |  |
| --- | --- |
| Project title | Technical Report of Stochastic Optimizers |
| Date | ??/04/2022 |
| Group | 6 |
| Leader | Tan Zhun Xian (20313854) |
| Other Group Members | * Heng Yuen Qi * Azzam Eusuf Chaai Bin Amier * Wong Khai Hin * Sabrina Felicia Kusumawati |
| Supervisor | * Tissa Chandesa * Siang Yew Chong |

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

**Table of Contents**

Cover Page ---------------------------------------------------------------------------------------------------------------------------------- ***1***

Literature Review --------------------------------------------------------------------------------------------------------------------- ***2 - 3***

* + Metaheuristics --------------------------------------------------------------------------------------------------------- 2
  + Evolutionary ------------------------------------------------------------------------------------------------------------ 2
  + Physics-based ---------------------------------------------------------------------------------------------------------- 2
  + Swarm intelligence ---------------------------------------------------------------------------------------------------- 2
  + CMAES ------------------------------------------------------------------------------------------------------------------- 2
  + Differential Evolution --------------------------------------------------------------------------------------------- 2 - 3
  + Genetic Algorithm ----------------------------------------------------------------------------------------------------- 3
  + Particle Swarm Optimization --------------------------------------------------------------------------------------- 3

Optimizer Configurations --------------------------------------------------------------------------------------------------------------- ***3***

* + BIPOP-CMA-ES --------------------------------------------------------------------------------------------------------- 3
  + DEAE ---------------------------------------------------------------------------------------------------------------------- 3
  + GA ------------------------------------------------------------------------------------------------------------------------- 3
  + PSO\_Bounds ------------------------------------------------------------------------------------------------------------ 3

Results ----------------------------------------------------------------------------------------------------------------------------------- ***4 - 5***

* + Average ------------------------------------------------------------------------------------------------------------------ 4
  + STD ------------------------------------------------------------------------------------------------------------------------ 4
  + Results Table ------------------------------------------------------------------------------------------------------------ 5

Observation of Results ------------------------------------------------------------------------------------------------------------------- ***6***

Analysis of Results ------------------------------------------------------------------------------------------------------------------------ ***6***

Post-Processed Data ----------------------------------------------------------------------------------------------------------------- ***6 - 7***

Conclusion ---------------------------------------------------------------------------------------------------------------------------------- ***7***

References ---------------------------------------------------------------------------------------------------------------------------------- ***8***

**Literature Review**

**Metaheuristics**

A metaheuristic is a problem-independent, high-level algorithmic framework that provides a set of recommendations or techniques for developing heuristic optimization algorithms [1]. Genetic/evolutionary algorithms, tabu search, simulated annealing, and ant colony optimization are all examples of metaheuristics, but there are many more. A metaheuristic is a problem-specific implementation of a heuristic optimization algorithm that follows the rules provided in a metaheuristic framework. Glover (1986) invented the phrase, which combines the Greek prefix meta- (metá, high-level beyond) with heuristic (from the Greek heuriskein or euriskein, to search).

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

**Evolutionary**

Although these models are crude simplifications of biological reality, nearly three decades of research and applications have clearly proved that modelling the search process of natural evolution can provide very strong, direct computer algorithms. The resulting evolutionary algorithms [2] are based on a population of individuals' collective learning process, with each individual representing a search point in the space of possible solutions to a particular problem. The population is randomly started, and it progresses toward better and better portions of the search space by randomised selection (which can be deterministic in some algorithms), mutation, and recombination processes (which is completely omitted in some algorithmic realizations). The environment provides high-quality information (fitness value) on the search points, and the selection process favours those who are fitter to reproduce more frequently than those who are less fit. The recombination mechanism permits parental information to be mixed while being passed down to progeny, and mutation promotes diversity into the population.

**Physics-based**

Several optimization approaches, particularly metaheuristic optimization methods, have been created by scientists in recent years. People have used nature's power to solve issues. As a result, those metaheuristic methods have mimicked natural physical and biological processes. Big Bang Big Crunch [3] optimization method based on universe evolution and Gravitational Search Algorithm [4] based on gravity law were created and implemented to solve complicated problems in 2007.

**Swarm intelligence**

Swarm intelligence (SI) [5] is a subset of artificial intelligence (or bio-inspired computation in general) (AI). It was first coined by Gerardo Beni and Jing Wang in 1989 in the context of building cellular robotic systems, and it has been recognised as an emerging field. The growing popularity of such SI-based algorithms can be attributed to a number of factors, the most important of which being the flexibility and variety they provide. The algorithms' primary features are their ability to self-learn and adapt to environmental variations, which has sparked a lot of attention and led to the identification of various application areas. Swarm intelligence has gained appeal in recent years as NP-hard issues have become more prevalent, making finding a global optimum in a real-time setting nearly impossible.

**CMAES**

Covariance matrix adaptation evolution strategy (CMA-ES) is a special approach for numerical optimization [6]. It is an evolution strategy that adapts the full covariance matrix of a normal search (mutation) distribution. In comparison to other evolutionary algorithms, the CMA-ES has an important property which is its invariance against linear transformations of the search space. It will exhibit the same performance on a given objective function and on the same function where a linear transformation is applied. Only with a corresponding strategy (distribution) parameter will this be true. In theory, this transformation is learned by the CMA algorithm.

To continue analysing the CMA-ES algorithm, a comparative study was done by Szynkiewicz on the CMA-ES and Particle Swarm Optimization (PSO) with the help of the black-box optimization benchmarks [7]. In his research, the efficiency and availability of the two biologically-inspired algorithms which were designed to tackle non-convex and ill-conditioned black-box optimizations problem was reported. The worst performing algorithm was the local version of the PSO algorithm followed by its global version [7]. The CMA-ES was better and outranked both PSO methods in all of the cases [7]. This is because CMA-ES is considered to be the state-of-the-art in the field of black-box optimization problems. To continue, even though the classic variant of the CMA-ES was outclassed by a reference algorithm in the research [7], the performance of the CMA-ES algorithm could be improved further by tuning its parameters.

To improve the CMA-ES algorithm, multiple restart strategies have been proposed. By applying the restart strategies, the probability of the algorithm to find the global optima is increased [17]. This produces variants such as the increasing population covariance matrix adaptation evolution strategy (IPOP-CMA-ES) and the bi-population covariance matrix adaptation evolution strategy (BIPOP-CMA-ES). A comparative study was done by Loshchilov using these two algorithms to solve the CEC 2013 benchmark problem. Although in some cases that the two algorithms produce minor differences, overall, the BIPOP based CMA-ES algorithm performs better on composition functions [17].

In 2010, a study was done by Ros where the IPOP-CMA-ES is used in benchmarking the black box optimization problem on the noiseless testbed. The result of the benchmark was then used to compare with the BIPOP-CMA-ES algorithms’ result. The result shows that the IPOP-CMA-ES algorithm solves multi-modal functions faster than the BIPOP-CMA-ES at best two times faster but it does not solve the weakly structured functions [18]. This makes the BIPOP-CMA-ES more favourable as although it produces some output slower, it makes up for it by having the capability to solve all of the functions that were benchmarked. In a competition where the time taken to solve the functions does not matter, the obvious choice is the BIPOP-CMA-ES.

**Differential Evolution**

Evolutionary algorithms (EAs) are inspired by the natural evolution of species. It involves mutation and crossover. A lot of optimisation problems have been solved by EAs in many areas. However, we need to carefully choose the appropriate parameters and encoding schemes in order to optimise the problem that we have. Otherwise, computational costs will be expensive [8].

Differential evolution (DE) is one of the EAs. It is a population-based optimisation algorithm that can be used to solve many practical problems and the results are quite satisfactory. However, it is not so efficient to solve non-separable functions. This is in light of the fact that crossover helps in separable function and on the other hand, it destroys possible combination of good offspring in non-separable function. To solve this problem, several ways are introduced. One of the ways is to use adaptive encoding (AE). This variant of DE is called Differential Evolution with Adaptive Encoding (DEAE). The algorithm will be made rotationally invariant by this AE framework [12]. This type of algorithm is needed as modern benchmark normally uses rotation matrices. This matrix is used to increase the difficulty to solve the problem as real-world problems are normally very difficult to solve. The notion of rotation invariant is that we can get the same output after we rotate the input [13].

In general, AE framework consists of three steps, that are encode, decode and update. Encoding means transforming the population into a space which gives benefits to the modification operators. A key to take note is that the offspring are evaluated in the original space. So, it needs to be decoded first. After that, the transformation matrix is adjusted in the update step [14].

AE is actually the recovery of Covariance Matrix Adaptation (CMA) evolution strategy, but with cumulative step-size control. Several experiments show that non-separable and ill-conditioned problems will normally be sped up by this type of adaptive encoding. In AE, search performance can be improved because search problems can be made decoupled [16].

**Genetic Algorithm**

Binary Genetic Algorithm or also known as Genetic Algorithm (GA) was introduced by John Henry Holland and his associates in 1975 in a book published by the MIT press [9]. This algorithm is a computational model inspired by evolution. In relation to its name, the way component vectors are configured within this algorithm follows the genetic structure of a chromosome [9]. Its main idea takes from natures natural selection/survival of the fittest and works by simulating evolution, starting with an initial set of solutions or hypotheses and generating future “generations” of solutions [10].

For more clarification, GA starts by generating a “population” containing many (possibly even thousands) of solutions depending on the nature of the problem [19]. It then processes each individual solution through a fitness function where each individual is given a fitness score [19]. This fitness score will affect the chances an individual has on being selected as a “parent” for the next generation [10]. The information contained in selected individuals are then combined from 2 or more parents to generate “offspring” via a Crossover operator [20]. A mutation operator then maintains the diversity of the solutions present in the population by simulating real life evolutions. This is done by modifying the information of a given individual slightly using operators like displacement, simple inversion, and scramble mutations [20]. A population with the mutated information is then created and used to generate any subsequent populations [10]. The values of the objective functions in the new generation are then determined by decoding the string and used to determine the fitness of their solutions, thus completing the cycle of a generation [10]. If the solution improves then the best solution in a generation is stored and this is repeated till the solutions reach a convergence [10].

**Particle Swarm Optimization**

Particle Swamp Optimizer, aka PSO, is a population-based stochastic optimization method commonly used to solve continuous nonlinear functions [15] The technique was developed by Dr Eberhart and Dr Kennedy, inspired by the social behaviour of a flock of birds or a school of fish [21]. PSO starts with a set of random particles and then iteratively seeks optimal solutions by updating generations. Each particle is updated in each cycle by alternating between the two best values. The first is the most successful option, aka fitness, thus far. In addition, the fitness value is saved. This is referred to as pbest. Another best value that the particle swarm optimizer monitors are the best value attained so far by any particle in the population. This best value is known as gbest and is a global best. When a particle has topological neighbours from the population, the best value is a local best and is referred to as lbest. With the two best values, the particle updates its position and its velocity [21].

PSO\_Bounds was introduced by EL-Abd and Kamel, where the concept of population-based incremental learning (PBIL) is amalgamated into Particle Swamp Optimizer. Essentially, PSO\_Bounds is a Hybrid of PSO and PBIL. PBIL is a stochastic guided search algorithm in which the learning rate and search rate can be adjusted and manipulated [22].

**Optimizer Configurations**

**BIPOP-CMA-ES**

After a first single run with default population size, we use two interlaced multi-start regimes, each equipped with a function evaluation budget accounting for the so far conducted function evaluations, after a single run with default population size. A complete run of either one or the other strategy is initiated, depending on which budget figure is lower. Under the first regime, the first and last restarts are carried out.

In the first regime, we resume with rising population size, increasing the population size by a factor of two before each restart. There are a total of nine restarts which comes out to a maximum factor of 512. In the second regime, multi-start regime with small population size is applied. The second multi-start regime begins if and only if its recent budget is less than that of the first multi-start regime with growing populations.

**DEAE**

This Adaptive Encoding (AE) framework will make the DE algorithm rotationally invariant [12]. Because rotation matrices are commonly used in recent benchmarks, this type of algorithm is required. This matrix is used to make the problem more difficult to solve, as real-world problems are typically tough to solve. The term "rotation invariant" refers to the fact that we can achieve the same result when rotating the input [13].

Encode, decode, and update are the three phases that make up the AE framework. Encoding is the process of converting a population into a place that helps the moderators. It's important to notice that the progeny is assessed in the original space. As a result, it must first be decoded. In the update step [14], the transformation matrix is then changed.

**Genetic Algorithm**

The GA optimizer was initialised with floor (sqrt (5000) \* 4) = 282 population size. The individuals in the population have DIM number of chromosomes with 32 bit each. The selection operator was set to tournament selection with a 2-point crossover operator. Mutation probability was set to even.

**PSO\_Bounds**

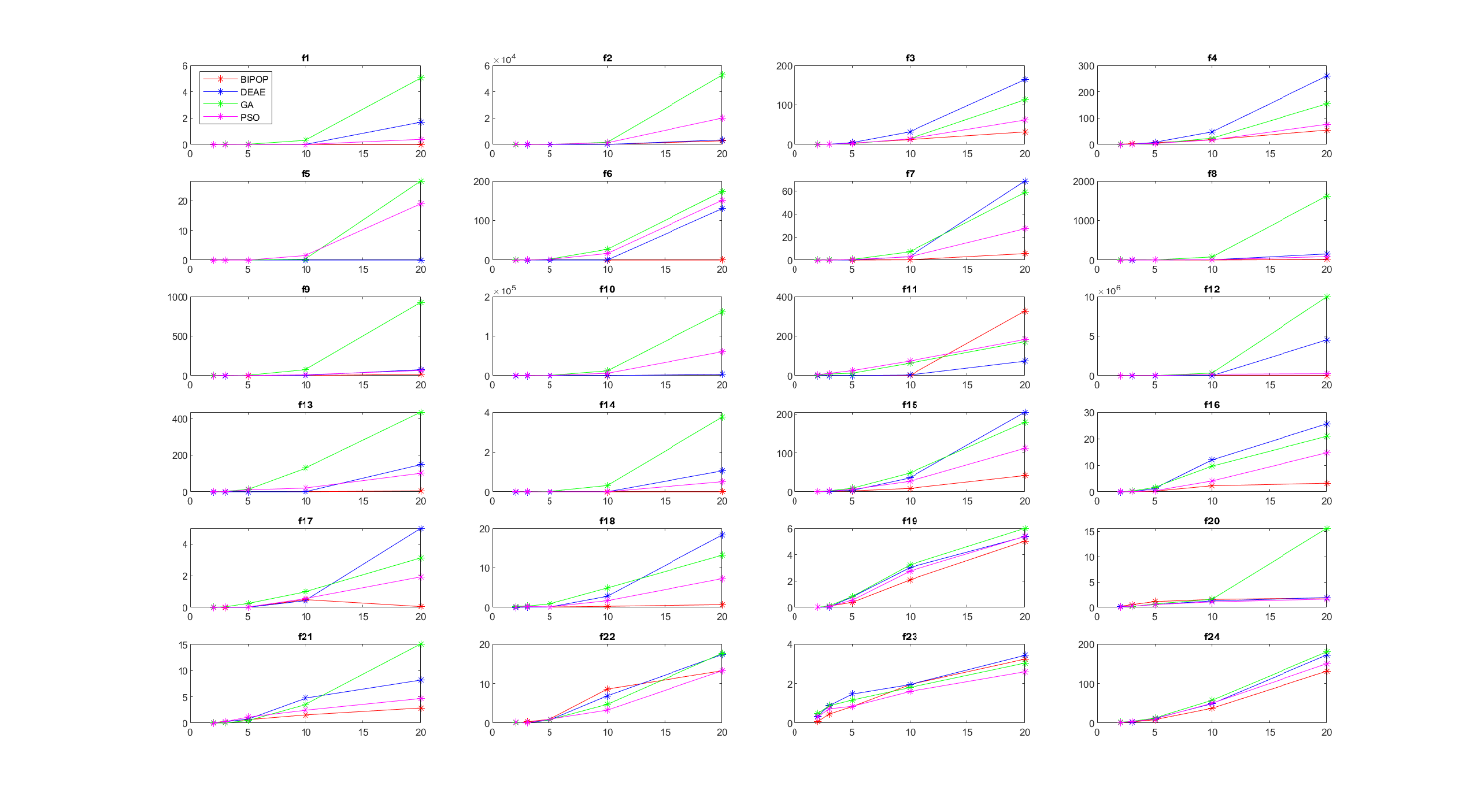
The algorithm that was employed was a simple PSO algorithm that used the global best model. The only design decision was to use absorbing borders to handle any particles that left the search space, with the position set to the boundary and the velocity set to zeros [15]. The swarm consists of 40 particles with the following parameters: c1 = c2 = 2.

**Results**

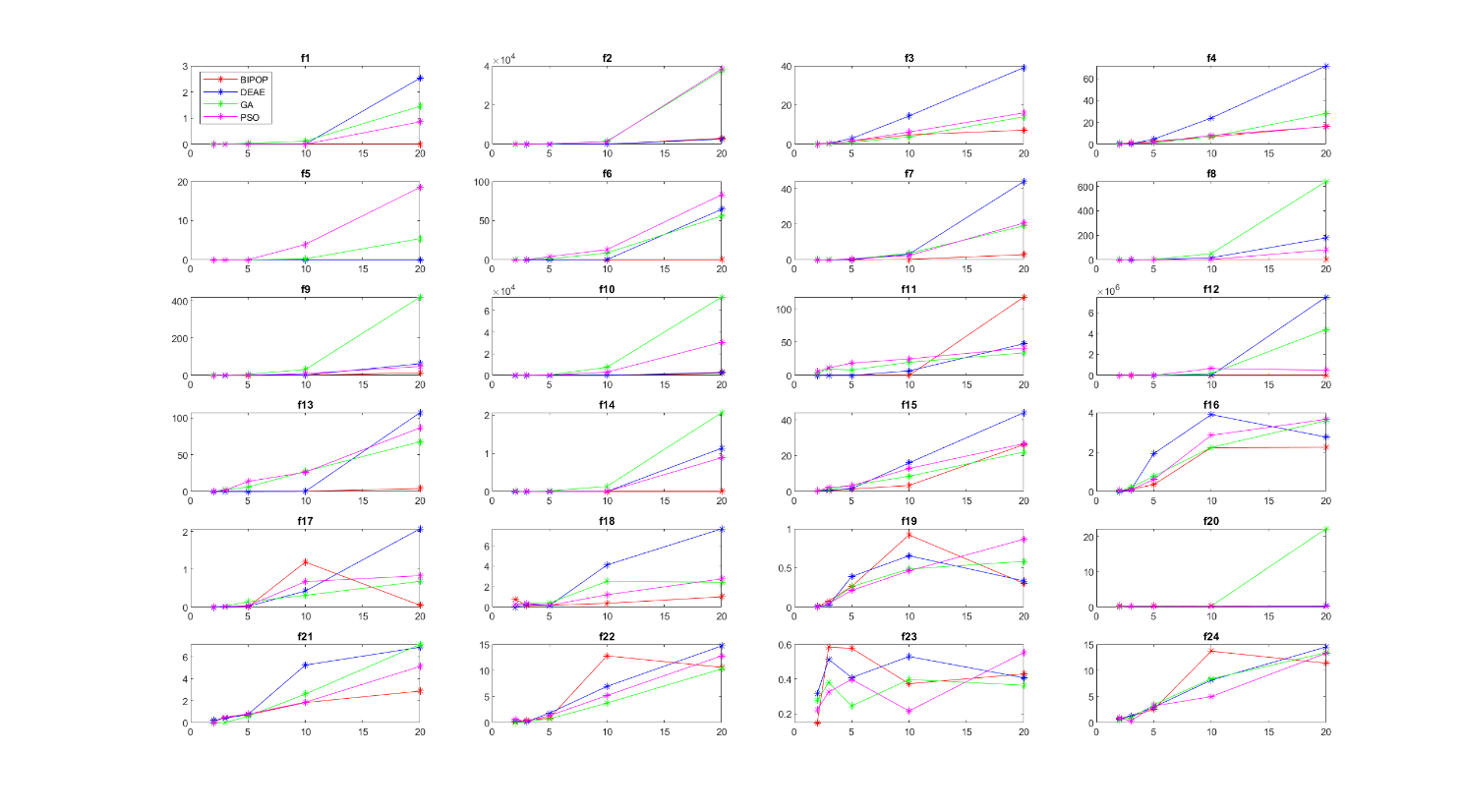
Results below show the results of the 4 optimizers tested with 5000 maximum function evaluation 15 times with initial seed 20313854 at 5 dimensions [2,3,5,10,20] plotted on a graph. Average and STD are calculated and shown below. Full results table are also shown below.

Boxplots are also provided at these links: SortByDimensions | SortByOptimizers

**Average of Δftarget**

****

**STD of Δftarget**

****

**Full Results Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Functions** | **2-D** | | | | | | | | **3-D** | | | | | | | |
| **Average** | | | | **STD** | | | | **Average** | | | | **STD** | | | |
| **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** |
| **Separable Functions** | | | | | | | | | | | | | | | | |
| **f1** | **0** | **0** | 1.28E-06 | **0** | **0** | **0** | 2.99E-06 | **0** | **0** | **0** | 2.04E-04 | **0** | **0** | **0** | 2.84E-04 | **0** |
| **f2** | **0** | **0** | 7.73E-04 | 6.97E-01 | **0** | **0** | 1.32E-03 | 2.48E+00 | **0** | **0** | 9.53E-01 | 5.53E+00 | **0** | **0** | 2.47E+00 | 1.47E+01 |
| **f3** | 6.63E-02 | **0** | 1.17E-03 | **0** | 2.48E-01 | **0** | 2.25E-03 | **0** | 7.09E-01 | 4.85E-01 | **1.15E-01** | 2.65E-01 | 5.74E-01 | 5.23E-01 | **2.40E-01** | 4.40E-01 |
| **f4** | 1.08E+00 | 1.99E-01 | 7.53E-02 | **2.87E-14** | 6.36E-01 | 5.39E-01 | 2.57E-01 | **7.26E-14** | 2.57E+00 | 1.07E+00 | **4.83E-01** | 1.21E+00 | 1.14E+00 | **4.39E-01** | 7.58E-01 | 9.86E-01 |
| **f5** | **6.51E-16** | **6.51E-16** | **6.51E-16** | **6.51E-16** | **3.73E-15** | **3.73E-15** | **3.73E-15** | **3.73E-15** | **-2.96E-16** | **-2.96E-16** | **-2.96E-16** | **-2.96E-16** | **8.98E-16** | **8.98E-16** | **8.98E-16** | **8.98E-16** |
| **Low or Moderate Condition Functions** | | | | | | | | | | | | | | | | |
| **f6** | 1.18E-16 | **0** | 3.25E-02 | 4.13E-11 | 4.43E-16 | **0** | 1.07E-01 | 1.09E-10 | 2.96E-15 | **0** | 4.52E-01 | 2.99E-03 | 4.75E-15 | **0** | 6.23E-01 | 1.12E-02 |
| **f7** | **0** | **0** | 2.08E-02 | 1.66E-15 | **0** | **0** | 7.06E-02 | 3.86E-15 | **7.99E-16** | 5.20E-04 | 9.94E-03 | 2.05E-13 | **1.89E-15** | 1.95E-03 | 2.69E-02 | 2.51E-13 |
| **f8** | **0** | **0** | 1.93E-02 | 5.50E-06 | **0** | **0** | 2.46E-02 | 1.26E-05 | **0** | **0** | 3.55E-01 | 4.17E-01 | **0** | **0** | 3.52E-01 | 1.10E+00 |
| **f9** | **0** | **0** | 1.55E-02 | 5.35E-07 | **0** | **0** | 1.55E-02 | 1.12E-06 | **0** | **0** | 3.22E-01 | 9.26E-02 | **0** | **0** | 2.92E-01 | 2.38E-01 |
| **High Condition Functions** | | | | | | | | | | | | | | | | |
| **f10** | **0** | **0** | 3.32E+00 | 6.91E+00 | **0** | **0** | 3.57E+00 | 1.81E+01 | **0** | **0** | 1.08E+02 | 5.41E+00 | **0** | **0** | 1.78E+02 | 4.78E+00 |
| **f11** | **0** | **0** | 2.87E+00 | 4.49E+00 | **0** | **0** | 3.64E+00 | 5.65E+00 | **0** | **0** | 8.11E+00 | 1.15E+01 | **0** | **0** | 9.50E+00 | 1.14E+01 |
| **f12** | **2.56E-03** | 2.46E-01 | 7.96E-01 | 7.09E-01 | **9.58E-03** | 9.13E-01 | 1.24E+00 | 2.08E+00 | **3.52E-02** | 4.06E+00 | 1.37E+01 | 9.94E+00 | **1.32E-01** | 1.49E+01 | 1.75E+01 | 1.64E+01 |
| **f13** | 1.31E-12 | **2.96E-16** | 3.69E-01 | 3.38E-02 | 2.41E-12 | **8.98E-16** | 3.71E-01 | 4.69E-02 | 3.95E-12 | **2.65E-14** | 2.36E+00 | 1.00E+00 | 5.56E-12 | **4.98E-14** | 2.50E+00 | 1.76E+00 |
| **f14** | 8.85E-14 | **-7.40E-18** | 5.98E-03 | 5.39E-06 | 8.92E-14 | **2.77E-17** | 1.93E-02 | 9.51E-06 | 3.27E-13 | **-7.40E-18** | 1.58E-03 | 5.18E-05 | 2.76E-13 | **2.77E-17** | 1.65E-03 | 4.85E-05 |
| **Multi-Modal Functions** | | | | | | | | | | | | | | | | |
| **f15** | 2.32E-01 | 1.99E-01 | 7.99E-01 | **1.33E-01** | 4.00E-01 | 3.98E-01 | 4.71E-01 | **3.38E-01** | **7.74E-01** | 7.96E-01 | 2.48E+00 | 1.67E+00 | **5.83E-01** | 7.45E-01 | 1.25E+00 | 2.26E+00 |
| **f16** | **0** | **0** | 2.22E-02 | 2.28E-02 | **0** | **0** | 3.34E-02 | 7.90E-02 | **3.36E-02** | 3.41E-02 | 2.68E-01 | 6.28E-02 | 1.17E-01 | 8.79E-02 | 2.06E-01 | **8.57E-02** |
| **f17** | 3.82E-13 | **-6.48E-18** | 1.32E-02 | 1.98E-06 | 1.45E-13 | **2.42E-17** | 1.17E-02 | 7.17E-06 | **2.85E-06** | 1.13E-05 | 3.41E-02 | 6.52E-03 | **7.32E-06** | 2.89E-05 | 2.93E-02 | 2.27E-02 |
| **f18** | 2.09E-01 | **2.23E-08** | 2.49E-01 | 6.99E-02 | 7.81E-01 | **8.33E-08** | 2.62E-01 | 2.55E-01 | 7.85E-02 | **7.08E-02** | 3.35E-01 | 2.37E-01 | **1.55E-01** | 2.49E-01 | 3.74E-01 | 3.70E-01 |
| **f19** | 1.63E-03 | 5.24E-03 | 4.69E-03 | **1.05E-03** | 4.34E-03 | 7.40E-03 | 6.42E-03 | **3.92E-03** | 9.82E-02 | **1.30E-02** | 1.08E-01 | 5.46E-02 | 7.93E-02 | **2.62E-02** | 4.57E-02 | 4.61E-02 |
| **Multi-Modal with Weak Global Structure Functions** | | | | | | | | | | | | | | | | |
| **f20** | 2.07E-01 | 1.97E-01 | **4.23E-02** | 7.90E-02 | 3.49E-01 | 2.79E-01 | **1.24E-01** | 2.01E-01 | 5.44E-01 | 2.11E-01 | **1.82E-01** | 2.31E-01 | 2.92E-01 | **2.44E-01** | 2.84E-01 | 2.70E-01 |
| **f21** | 4.61E-02 | 6.20E-02 | 1.59E-04 | **0** | 1.73E-01 | 2.32E-01 | 5.73E-04 | **0** | 3.16E-01 | 2.37E-01 | **1.44E-03** | 2.49E-01 | 4.64E-01 | 4.11E-01 | **5.01E-03** | 5.12E-01 |
| **f22** | 7.18E-02 | 9.22E-02 | **1.26E-02** | 2.19E-01 | 1.92E-01 | 2.35E-01 | **2.39E-02** | 6.57E-01 | 2.97E-01 | **4.61E-02** | 1.46E-01 | 9.22E-02 | 5.23E-01 | **1.73E-01** | 2.29E-01 | 2.35E-01 |
| **f23** | **6.17E-02** | 3.41E-01 | 4.69E-01 | 2.96E-01 | **1.48E-01** | 3.16E-01 | 2.78E-01 | 2.23E-01 | **4.49E-01** | 8.78E-01 | 8.69E-01 | 7.11E-01 | 5.84E-01 | 5.13E-01 | 3.79E-01 | **3.25E-01** |
| **f24** | **1.27E+00** | 1.87E+00 | 1.88E+00 | 1.41E+00 | 8.58E-01 | 6.65E-01 | **5.95E-01** | 9.00E-01 | **2.67E+00** | 2.92E+00 | 4.23E+00 | 3.53E+00 | 1.22E+00 | 1.28E+00 | 9.42E-01 | **3.32E-01** |
| **Results** | | | | | | | | | | | | | | | | |
| **Best** | 12/24 | 15/24 | 3/24 | 7/24 | 11/24 | 15/24 | 4/24 | 7/24 | 14/24 | 13/24 | 5/24 | 2/24 | 12/24 | 14/24 | 3/24 | 5/24 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Functions** | **5-D** | | | | | | | | **10-D** | | | | | | | |
| **Average** | | | | **STD** | | | | **Average** | | | | **STD** | | | |
| **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** |
| **Separable Functions** | | | | | | | | | | | | | | | | |
| **f1** | **0** | **0** | 1.61E-02 | 6.89E-14 | **0** | **0** | 3.78E-02 | 1.53E-13 | **2.66E-15** | 1.17E-12 | 3.18E-01 | 2.97E-07 | **4.91E-15** | 3.95E-12 | 1.19E-01 | 5.86E-07 |
| **f2** | **0** | **0** | 5.28E+01 | 8.99E+01 | **0** | **0** | 9.19E+01 | 2.64E+02 | **1.96E-01** | 4.34E+01 | 1.63E+03 | 1.26E+03 | **3.68E-01** | 8.64E+01 | 1.51E+03 | 1.28E+03 |
| **f3** | 3.32E+00 | 4.75E+00 | **2.01E+00** | 2.71E+00 | 1.61E+00 | 2.90E+00 | **9.68E-01** | 1.66E+00 | **1.23E+01** | 3.19E+01 | 1.43E+01 | 1.42E+01 | 4.79E+00 | 1.44E+01 | **3.73E+00** | 6.23E+00 |
| **f4** | 6.34E+00 | 7.83E+00 | **3.05E+00** | 3.88E+00 | 2.76E+00 | 4.62E+00 | **1.18E+00** | 1.61E+00 | 1.83E+01 | 4.77E+01 | 2.38E+01 | **1.72E+01** | **6.72E+00** | 2.39E+01 | 6.91E+00 | 8.15E+00 |
| **f5** | **-7.40E-15** | **-7.40E-15** | **-7.40E-15** | **-7.40E-15** | **2.91E-14** | **2.91E-14** | **2.91E-14** | **2.91E-14** | **-7.40E-15** | **-7.40E-15** | 3.03E-01 | 1.53E+00 | **2.95E-14** | **2.95E-14** | 2.53E-01 | 3.93E+00 |
| **Low or Moderate Condition Functions** | | | | | | | | | | | | | | | | |
| **f6** | **3.53E-14** | 4.04E-12 | 2.28E+00 | 1.67E+00 | **3.72E-14** | 6.84E-12 | 1.13E+00 | 4.04E+00 | **3.26E-09** | 5.60E-02 | 2.71E+01 | 1.65E+01 | **7.84E-09** | 1.52E-01 | 8.75E+00 | 1.30E+01 |
| **f7** | **5.00E-15** | 1.14E-02 | 5.32E-01 | 4.04E-01 | **9.71E-15** | 4.25E-02 | 3.62E-01 | 5.88E-01 | **2.34E-01** | 3.00E+00 | 7.17E+00 | 2.81E+00 | **2.56E-01** | 3.20E+00 | 3.68E+00 | 2.42E+00 |
| **f8** | **1.55E-02** | 5.24E-01 | 4.23E+00 | 9.61E-01 | **5.81E-02** | 1.34E+00 | 4.19E+00 | 5.18E-01 | **9.69E-01** | 7.03E+00 | 7.75E+01 | 7.62E+00 | **1.51E+00** | 1.67E+01 | 5.01E+01 | 2.90E+00 |
| **f9** | **0** | 4.72E-01 | 5.29E+00 | 1.50E+00 | **0** | 1.21E+00 | 5.79E+00 | 1.34E+00 | **7.03E-01** | 3.68E+00 | 7.55E+01 | 1.03E+01 | **1.32E+00** | 2.17E+00 | 3.17E+01 | 1.01E+01 |
| **High Condition Functions** | | | | | | | | | | | | | | | | |
| **f10** | **0** | **0** | 6.41E+02 | 3.16E+02 | **0** | **0** | 4.86E+02 | 4.61E+02 | **1.01E-01** | 1.82E+01 | 1.20E+04 | 5.89E+03 | **1.46E-01** | 2.00E+01 | 7.44E+03 | 2.98E+03 |
| **f11** | 4.74E-16 | **0** | 1.23E+01 | 2.53E+01 | 1.77E-15 | **0** | 8.13E+00 | 1.85E+01 | **1.36E-03** | 3.07E+00 | 6.29E+01 | 7.39E+01 | **3.63E-03** | 6.84E+00 | 1.98E+01 | 2.49E+01 |
| **f12** | 1.64E-01 | **9.29E-02** | 3.04E+03 | 1.51E+01 | 4.08E-01 | **1.74E-01** | 2.75E+03 | 1.65E+01 | **5.00E-01** | 8.54E+00 | 3.24E+05 | 1.76E+05 | **9.55E-01** | 1.44E+01 | 1.81E+05 | 6.59E+05 |
| **f13** | 1.45E-09 | **3.45E-13** | 1.31E+01 | 1.06E+01 | 2.51E-09 | **2.48E-13** | 6.15E+00 | 1.38E+01 | 2.91E-01 | **5.40E-02** | 1.29E+02 | 2.04E+01 | 4.61E-01 | **1.72E-01** | 2.74E+01 | 2.62E+01 |
| **f14** | 7.75E-13 | **8.53E-15** | 1.90E-02 | 2.84E-04 | 1.49E-12 | **2.82E-14** | 1.88E-02 | 1.71E-04 | **6.06E-07** | 2.68E-05 | 3.15E-01 | 2.03E-03 | **6.24E-07** | 4.11E-05 | 1.34E-01 | 6.65E-04 |
| **Multi-Modal Functions** | | | | | | | | | | | | | | | | |
| **f15** | **2.32E+00** | 3.53E+00 | 9.20E+00 | 6.20E+00 | **1.39E+00** | 1.62E+00 | 3.42E+00 | 3.26E+00 | **8.69E+00** | 3.57E+01 | 4.82E+01 | 2.70E+01 | **3.40E+00** | 1.60E+01 | 8.61E+00 | 1.29E+01 |
| **f16** | **2.38E-01** | 1.18E+00 | 1.67E+00 | 3.74E-01 | **3.51E-01** | 1.92E+00 | 7.81E-01 | 6.14E-01 | **2.27E+00** | 1.20E+01 | 9.64E+00 | 4.07E+00 | **2.22E+00** | 3.90E+00 | 2.23E+00 | 2.85E+00 |
| **f17** | **1.17E-03** | 7.57E-03 | 2.51E-01 | 2.26E-02 | **3.16E-03** | 2.29E-02 | 1.40E-01 | 3.44E-02 | 4.93E-01 | **4.30E-01** | 9.89E-01 | 5.54E-01 | 1.19E+00 | 4.24E-01 | **3.13E-01** | 6.75E-01 |
| **f18** | 9.08E-02 | **7.49E-02** | 8.96E-01 | 1.35E-01 | 1.78E-01 | **1.49E-01** | 4.10E-01 | 1.73E-01 | **2.72E-01** | 2.80E+00 | 4.91E+00 | 1.69E+00 | **3.71E-01** | 4.11E+00 | 2.52E+00 | 1.21E+00 |
| **f19** | **3.83E-01** | 7.93E-01 | 8.27E-01 | 5.50E-01 | 2.63E-01 | 3.90E-01 | 2.62E-01 | **2.14E-01** | **2.08E+00** | 3.02E+00 | 3.22E+00 | 2.75E+00 | 9.22E-01 | 6.55E-01 | 4.86E-01 | **4.63E-01** |
| **Multi-Modal with Weak Global Structure Functions** | | | | | | | | | | | | | | | | |
| **f20** | 1.14E+00 | **5.42E-01** | 6.74E-01 | 5.47E-01 | 3.46E-01 | **2.32E-01** | 2.44E-01 | 3.19E-01 | 1.57E+00 | 1.28E+00 | 1.60E+00 | **1.08E+00** | 3.34E-01 | 2.85E-01 | 3.10E-01 | **2.58E-01** |
| **f21** | 6.36E-01 | 7.27E-01 | **4.65E-01** | 1.13E+00 | 7.08E-01 | 7.54E-01 | **5.98E-01** | 7.89E-01 | **1.54E+00** | 4.74E+00 | 3.52E+00 | 2.43E+00 | **1.84E+00** | 5.25E+00 | 2.61E+00 | 1.87E+00 |
| **f22** | 9.59E-01 | **7.50E-01** | 8.15E-01 | 9.57E-01 | 8.75E-01 | 1.86E+00 | **7.48E-01** | 1.37E+00 | 8.57E+00 | 6.86E+00 | 4.70E+00 | **3.25E+00** | 1.28E+01 | 6.94E+00 | **3.77E+00** | 5.20E+00 |
| **f23** | **8.27E-01** | 1.46E+00 | 1.16E+00 | 8.43E-01 | 5.77E-01 | 4.08E-01 | **2.45E-01** | 3.98E-01 | 1.94E+00 | 1.94E+00 | 1.79E+00 | **1.59E+00** | 3.73E-01 | 5.30E-01 | 3.97E-01 | **2.16E-01** |
| **f24** | **8.10E+00** | 1.10E+01 | 1.22E+01 | 1.01E+01 | **2.56E+00** | 3.00E+00 | 3.44E+00 | 3.20E+00 | **3.72E+01** | 4.87E+01 | 5.73E+01 | 4.98E+01 | 1.37E+01 | 8.14E+00 | 8.38E+00 | **5.00E+00** |
| **Results** | | | | | | | | | | | | | | | | |
| **Best** | 14/24 | 11/24 | 4/24 | 1/24 | 12/24 | 10/24 | 6/24 | 1/24 | 18/24 | 3/24 | 0/24 | 4/24 | 16/24 | 2/24 | 3/24 | 4/24 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Functions** | **20-D** | | | | | | | | **Overall** | | | | | | | |
| **Average** | | | | **STD** | | | | **Average** | | | | **STD** | | | |
| **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** | **BIPOP** | **DEAE** | **GA** | **PSO** |
| **Separable Functions** | | | | | | | | | | | | | | | | |
| **f1** | **2.59E-14** | 1.69E+00 | 5.04E+00 | 3.88E-01 | **2.97E-14** | 2.53E+00 | 1.45E+00 | 8.60E-01 | **5.71E-15** | 3.37E-01 | 1.08E+00 | 7.76E-02 | **1.68E-14** | 1.32E+00 | 2.09E+00 | 4.15E-01 |
| **f2** | **2.74E+03** | 3.45E+03 | 5.26E+04 | 2.00E+04 | 3.04E+03 | **2.59E+03** | 3.75E+04 | 3.84E+04 | **5.49E+02** | 6.99E+02 | 1.08E+04 | 4.28E+03 | **1.75E+03** | 1.80E+03 | 2.68E+04 | 1.89E+04 |
| **f3** | **3.17E+01** | 1.64E+02 | 1.13E+02 | 6.18E+01 | **7.19E+00** | 3.91E+01 | 1.40E+01 | 1.60E+01 | **9.61E+00** | 4.03E+01 | 2.59E+01 | 1.58E+01 | **1.25E+01** | 6.58E+01 | 4.44E+01 | 2.48E+01 |
| **f4** | **5.43E+01** | 2.60E+02 | 1.54E+02 | 7.63E+01 | 1.63E+01 | 7.15E+01 | 2.80E+01 | **1.61E+01** | **1.65E+01** | 6.34E+01 | 3.62E+01 | 1.97E+01 | **2.14E+01** | 1.05E+02 | 6.08E+01 | 3.01E+01 |
| **f5** | **2.29E-14** | **2.29E-14** | 2.67E+01 | 1.90E+01 | **3.81E-14** | **3.81E-14** | 5.40E+00 | 1.85E+01 | **1.69E-15** | **1.69E-15** | 5.40E+00 | 4.11E+00 | **2.76E-14** | **2.76E-14** | 1.09E+01 | 1.13E+01 |
| **Low or Moderate Condition Functions** | | | | | | | | | | | | | | | | |
| **f6** | **1.93E-02** | 1.30E+02 | 1.73E+02 | 1.51E+02 | **1.32E-02** | 6.44E+01 | 5.58E+01 | 8.30E+01 | **3.87E-03** | 2.59E+01 | 4.06E+01 | 3.38E+01 | **9.72E-03** | 5.93E+01 | 7.17E+01 | 6.98E+01 |
| **f7** | **5.58E+00** | 6.87E+01 | 5.87E+01 | 2.72E+01 | **2.91E+00** | 4.41E+01 | 1.91E+01 | 2.07E+01 | **1.16E+00** | 1.43E+01 | 1.33E+01 | 6.08E+00 | **2.57E+00** | 3.36E+01 | 2.45E+01 | 1.41E+01 |
| **f8** | **1.48E+01** | 1.54E+02 | 1.62E+03 | 8.72E+01 | **2.52E+00** | 1.79E+02 | 6.40E+02 | 8.16E+01 | **3.16E+00** | 3.24E+01 | 3.41E+02 | 1.92E+01 | **5.98E+00** | 1.01E+02 | 7.03E+02 | 5.00E+01 |
| **f9** | **1.76E+01** | 7.52E+01 | 9.27E+02 | 6.51E+01 | **1.43E+01** | 6.25E+01 | 4.20E+02 | 4.80E+01 | **3.65E+00** | 1.59E+01 | 2.02E+02 | 1.54E+01 | **9.46E+00** | 4.08E+01 | 4.10E+02 | 3.34E+01 |
| **High Condition Functions** | | | | | | | | | | | | | | | | |
| **f10** | 3.37E+03 | **3.10E+03** | 1.61E+05 | 6.04E+04 | **2.13E+03** | 2.77E+03 | 7.27E+04 | 3.07E+04 | 6.73E+02 | **6.23E+02** | 3.48E+04 | 1.33E+04 | **1.65E+03** | 1.75E+03 | 7.13E+04 | 2.74E+04 |
| **f11** | 3.27E+02 | **7.30E+01** | 1.72E+02 | 1.83E+02 | 1.18E+02 | 4.76E+01 | **3.39E+01** | 4.13E+01 | 6.53E+01 | **1.52E+01** | 5.17E+01 | 5.97E+01 | 1.41E+02 | **3.60E+01** | 6.66E+01 | 7.05E+01 |
| **f12** | **3.64E+00** | 4.52E+06 | 9.96E+06 | 2.42E+05 | **4.06E+00** | 7.48E+06 | 4.36E+06 | 5.00E+05 | **8.68E-01** | 9.05E+05 | 2.06E+06 | 8.36E+04 | **2.34E+00** | 3.80E+06 | 4.41E+06 | 3.84E+05 |
| **f13** | **5.40E+00** | 1.48E+02 | 4.34E+02 | 1.01E+02 | **4.05E+00** | 1.07E+02 | 6.74E+01 | 8.64E+01 | **1.14E+00** | 2.97E+01 | 1.16E+02 | 2.66E+01 | **2.81E+00** | 7.63E+01 | 1.69E+02 | 5.57E+01 |
| **f14** | **3.89E-04** | 1.05E+00 | 3.75E+00 | 5.05E-01 | **1.12E-04** | 1.13E+00 | 2.06E+00 | 8.83E-01 | **7.80E-05** | 2.10E-01 | 8.18E-01 | 1.02E-01 | **1.64E-04** | 6.57E-01 | 1.74E+00 | 4.43E-01 |
| **Multi-Modal Functions** | | | | | | | | | | | | | | | | |
| **f15** | **4.18E+01** | 2.05E+02 | 1.79E+02 | 1.12E+02 | 2.62E+01 | 4.41E+01 | **2.20E+01** | 2.67E+01 | **1.08E+01** | 4.90E+01 | 4.80E+01 | 2.95E+01 | **1.98E+01** | 8.17E+01 | 6.88E+01 | 4.46E+01 |
| **f16** | **3.23E+00** | 2.56E+01 | 2.09E+01 | 1.47E+01 | **2.24E+00** | 2.76E+00 | 3.59E+00 | 3.66E+00 | **1.15E+00** | 7.77E+00 | 6.51E+00 | 3.85E+00 | **1.95E+00** | 1.03E+01 | 8.25E+00 | 6.01E+00 |
| **f17** | **5.12E-02** | 5.00E+00 | 3.13E+00 | 1.93E+00 | **5.11E-02** | 2.07E+00 | 6.82E-01 | 8.31E-01 | **1.09E-01** | 1.09E+00 | 8.84E-01 | 5.03E-01 | **5.68E-01** | 2.18E+00 | 1.23E+00 | 8.85E-01 |
| **f18** | **6.96E-01** | 1.83E+01 | 1.32E+01 | 7.27E+00 | **1.01E+00** | 7.65E+00 | 2.43E+00 | 2.76E+00 | **2.69E-01** | 4.24E+00 | 3.91E+00 | 1.88E+00 | **6.44E-01** | 8.09E+00 | 5.19E+00 | 3.08E+00 |
| **f19** | **5.03E+00** | 5.37E+00 | 6.00E+00 | 5.39E+00 | **3.04E-01** | 3.35E-01 | 5.83E-01 | 8.67E-01 | **1.52E+00** | 1.84E+00 | 2.03E+00 | 1.75E+00 | **1.96E+00** | 2.12E+00 | 2.33E+00 | 2.13E+00 |
| **Multi-Modal with Weak Global Structure Functions** | | | | | | | | | | | | | | | | |
| **f20** | 1.86E+00 | 1.93E+00 | 1.56E+01 | **1.64E+00** | 2.76E-01 | 3.41E-01 | 2.22E+01 | **2.10E-01** | 1.06E+00 | 8.32E-01 | 3.61E+00 | **7.15E-01** | 6.95E-01 | 7.30E-01 | 1.16E+01 | **6.30E-01** |
| **f21** | **2.86E+00** | 8.19E+00 | 1.51E+01 | 4.67E+00 | **2.88E+00** | 6.88E+00 | 7.14E+00 | 5.13E+00 | **1.08E+00** | 2.79E+00 | 3.81E+00 | 1.69E+00 | **1.88E+00** | 5.04E+00 | 6.71E+00 | 3.01E+00 |
| **f22** | **1.32E+01** | 1.73E+01 | 1.76E+01 | **1.32E+01** | 1.06E+01 | 1.47E+01 | **1.03E+01** | 1.28E+01 | 4.63E+00 | 5.02E+00 | 4.66E+00 | **3.55E+00** | 9.15E+00 | 9.91E+00 | 8.32E+00 | **7.95E+00** |
| **f23** | 3.23E+00 | 3.42E+00 | 3.03E+00 | **2.59E+00** | 4.29E-01 | 4.07E-01 | **3.64E-01** | 5.53E-01 | 1.30E+00 | 1.61E+00 | 1.46E+00 | **1.20E+00** | 1.24E+00 | 1.14E+00 | 9.56E-01 | **8.85E-01** |
| **f24** | **1.31E+02** | 1.72E+02 | 1.80E+02 | 1.50E+02 | **1.14E+01** | 1.45E+01 | 1.34E+01 | 1.33E+01 | **3.61E+01** | 4.72E+01 | 5.12E+01 | 4.30E+01 | **4.99E+01** | 6.49E+01 | 6.81E+01 | 5.68E+01 |
| **Results** | | | | | | | | | | | | | | | | |
| **Best** | 20/24 | 3/24 | 0/24 | 3/24 | 17/24 | 2/24 | 4/24 | 2/24 | 19/24 | 3/24 | 0/24 | 3/24 | 20/24 | 2/24 | 0/24 | 3/24 |

**\*Best Results of the 4 optimizers are bolded for easy viewing.**

**BIPOP – BIPOP-CMA-ES, DEAE – DEAE, GA – Genetic Algorithm, PSO – PSO\_Bounds**

**Observation of Results**

At lower dimensions, all optimizers perform similarly. However, once we reach higher dimensions, we find that the results quickly diverge. Generally, the BIPOP performs the best out of the 4 optimizers, having the lowest average score more often than not. However, an exception does occur for f11 as BIPOP was found to perform the worst at 20-D. We find that GA has the most occurrences of highest average scores at 20-D followed by DEAE which has the highest average score half as often as GA.

The BIPOP optimizer was found to have the lowest average scores across the board. After losing out to DEAE at 2-D, BIPOP manages to surpass the other optimizers beginning from 3-D before getting a supermajority of the lowest average scores starting from 10-D. BIPOP performs quite well at low, moderate and high condition functions especially at lower dimensions, achieving a substantial amount of 0 average scores. BIPOP also performs quite consistently at these conditions, achieving low standard deviations. Of course, when the average score is 0 then standard deviation is also 0. BIPOP also generally performs better than the other optimizers at multi-modal functions, achieving low average scores and standard deviations.

The DEAE optimizer manages to beat out BIPOP at 2-D, getting the lowest average scores. However, this does not last as the share of lowest average scores decreases at the number of dimensions increases as the scores are slowly dominated by BIPOP. Remarkably, at lower dimensions, DEAE was found to achieve more 0 average scores than BIPOP, dominating the low, moderate and high condition functions at lower dimensions. However, as mentioned above, this does not last as we reach higher dimensions. Unfortunately, DEAE was shown to get its fair share of highest average score as shown above especially on higher dimensions like 20-D.

The GA optimizer was only shown to perform well in separable functions and multi-modal functions with weak global structure at lower dimensions, dominating f3-f5 at 3-D and 5-D while performing well at f20-f22 at lower dimensions. After 10-D, GA has not been able to get even a single lowest average score. GA was also shown to get the highest average scores especially at 20-D.

The PSO optimizer was shown to perform extremely well at separable functions at 2-D, even outperforming the likes of BIPOP and DEAE. However, as the number of dimensions grows, separable functions was then dominated by GA and then by BIPOP. However, PSO was shown to dominate in multi-modal functions with weak global structure especially in higher dimensions. This is quite a surprise as usually BIPOP dominates at higher dimensions.

Overall, we can see that as the number of dimensions gets larger, the average results for all functions also increases. The overall average shows that BIPOP dominates the results overwhelmingly, with DEAE and PSO fighting over the remainder. GA was completely wiped out.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **fsmap** | **BIPOP** | **DEAE** | **GA** | **PSO** |
| **2-D** | 4160 | 4345 | 969 | 2919 |
| **3-D** | 3305 | 3572 | 469 | 1573 |
| **5-D** | 2553 | 2567 | -44 | 433 |
| **10-D** | 729 | 361 | -511 | -95 |
| **20-D** | 60 | -484 | -802 | -564 |
| **Total** | 10807 | 10361 | 81 | 4266 |

Looking at fsmap scores, we see that BIPOP beats out DEAE with only a slim margin. PSO received only half the score as the top 2 optimizers while GA received an embarrassingly low score. Although DEAE was seen to perform much worse at higher dimensions than PSO, its score was much higher than PSO.

**Analysis of Results**

The reason for the similar results of optimizers at lower dimension is the low complexity of the functions which allow the optimal result to be quickly and easily reached. As the number of dimensions increases, the average score also increases as the complexity of the function increases with increasing dimensions.

In separable functions f1 - f4, BIPOP performs the best. This is due to its ability to simplify the search process to D times one-dimensional search procedures, allowing them to achieve high accuracy with fewer function evaluations by utilising the separable property. When a solution with a lower objective function value is found in CMA-ES, the children are chosen as the next parent, Xk+1 as well as the best two offspring and the parent Xk. The optimizer performs well in terms of avoiding local minima and small-scale exploration.

Low and moderate condition functions f6 - f9 performs the best with the BIPOP optimizer, which means they can avoid huge plateaus and handle near-zero or zero gradients well. BIPOP also works best in functions f10 - f14 with high conditioning, demonstrating that it can be precise with micro-movements towards the optimum when on a steep ridge or peak.

BIPOP is also the most accurate and most consistent at Multimodal functions f15 – f19. This optimizer has good exploitation but limited exploration since it gets caught in local optima. However, if it lands in a favourable valley, it can minimise to a very low result.

Finally, in multimodal functions with weak global structure f20 – f24, BIPOP performs quite well. It can find strong scores even when there appears to be no pattern or structure to exploit in the search area. The population size of BIPOP plays a significant role in achieving the best precision in multimodal functions f20-f24. For each function evaluation, BIPOP generates a given population size of offspring, identifies the closest offspring to the target, then generates another set of offspring based on that closest offspring. Even in a weak global structure, where improvements between locations are difficult to discern via point-by-point exploration, this can ensure that the likelihood of finding a solution closest to the target can be identified and will rise depending on the population size set by parameter tuning.

For the other optimizers, DEAE, GA and PSO are able to perform relatively well at lower dimensions but their results get worst as the number of dimensions increases. This is due to the increase in function complexity as the number of dimensions increases. This might be also due to the fact that the optimizers fall into a local optimum with a large convergence basin and are unable to escape if exploitation is prioritised over exploration. If exploration is prioritised over exploitation, then the optimizer may jump from one local optima to the next without minimizing the score.

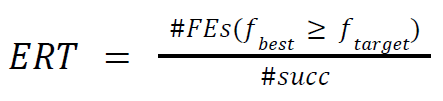
Looking at the average plots and the fsmap results, we find the even though DEAE was shown to get its fair share of highest average scores, it manages to vastly outscore PSO and GA in the fsmap results. This is due to the fact that fsmap uses the function result = floor (-log10 (score)) which varies logarithmically. So, even if DEAE performs poorly at higher dimensions, it will not affect its results as much as performing poorly at lower dimensions such as in the case of GA.

**Post-Processed Data**

We used the python script package provided to construct tables and figures reflecting the results of the benchmarking experiment, and we compared the results of each optimizer. The complete post processing findings for each optimizer derived from BB0B-2010 experiment data, as well as comparisons, can be found at the following links:

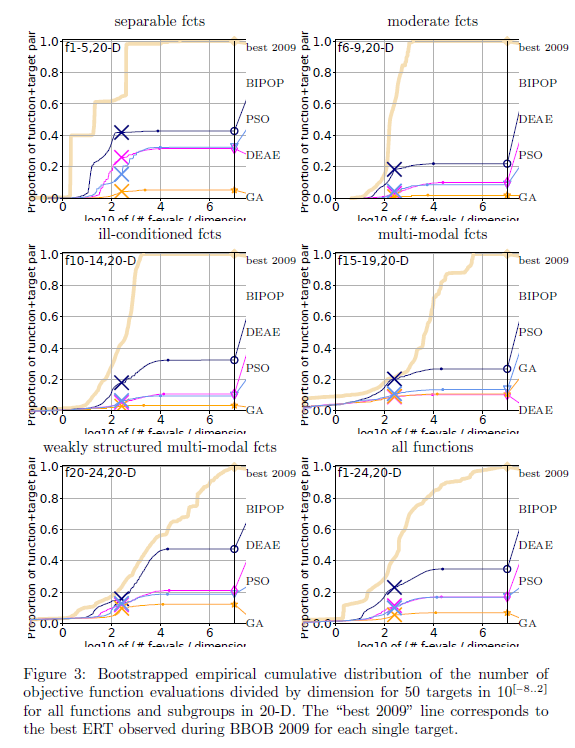
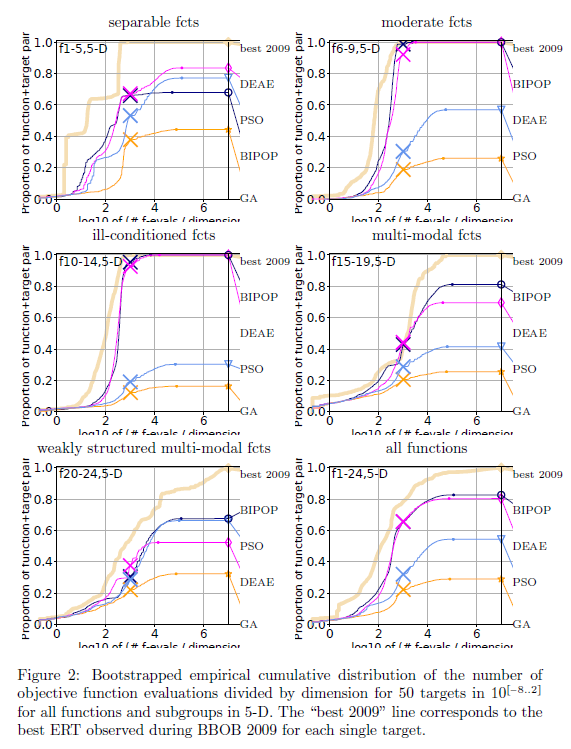
1. BIPOP
2. DEAE
3. GA
4. PSO
5. Comparison between all 4 optimizers

The overall performance of each optimizer is calculated using the Expected running time (ERT) and denotes the expected number of function evaluations to reach a target function value for the first time. ERT is defined as



where #𝐹𝐸𝑠 (𝑓𝑏𝑒𝑠𝑡 ≥ 𝑓𝑡𝑎𝑟𝑔𝑒𝑡) 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 post-processed data of each document has been linked as well as the comparison between the 4 optimizers.

In table 1, we show the bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension for 50 targets in 10[-8..2] for all functions and subgroups in 5-D and 20-D. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each single target.



The empirical cumulative distribution (ECDF) above shows that for 5-dimensions, aside from losing to DEAE and PSO at separable functions, BIPOP clearly performs better than the other optimizers by consistently getting the best results. However, DEAE was able to consistently get second place, performing almost identically to BIPOP. The only exception to this is the weakly structured multimodal functions where PSO was able to beat out DEAE to become to get the second-best results. The gap between second and third place is quite significant in some cases like moderate functions and ill conditioned functions followed by a slightly smaller gap between third and last place. GA was consistently the worst optimizer of the 4.

For the 20-dimensions ECDF, BIPOP was able to get the best results out of all the optimizers in all types of functions. This time, BIPOP was able to beat out the other optimizers decisively, creating a noticeable gap between first and second place. However, the second and third place optimizers perform nearly identically. With the exception of multimodal functions, the second and third place optimizers alternate between DEAE and PSO. Finally, GA was also shown to almost exclusively occupy last place.

The extraordinary performance of BIPOP can be ascribed to the various modifications of the CMA-ES algorithm, which are considered to be among the greatest in the field of black-box optimization, as seen in the post-processed data. PSO's low performance can be attributed to the optimizer's sensitivity to hyperparameter adjustment, which has proven difficult. PSO may have performed significantly better in the tests if the hyperparameters had been chosen more carefully, as they can have a significant impact on the end results. However, in the case of BIPOP, no time-consuming parameter adjusting is required, making it far more convenient to use than PSO, which is a significant advantage.

**Conclusion**

The experiment of selected stochastic optimizers for single-objective continuous optimisation problems, particularly the BIPOP, DEAE, GA, and PSO optimizers, is documented in this study. The BBOB-2010 benchmark test functions were used to evaluate their performance, and they were tested against all of the selected optimizers. We organised the findings of the optimizers after they ran the benchmark for dimensions 2,3,5,10, and 20, then compared and analysed the differences in their results in this report. Based on the results, it is evident that the BIPOP optimizer has the best performance, while the GA optimizer has the worst performance. BIPOP has been proven to reliably solve most functions (f1-f19), yielding the greatest results, whereas PSO excelled in multi-modal functions with weak global optimums (f20-f24). DEAE was a close contender with BIPOP for best optimizer at lower dimensions but lost out to it at higher dimensions. Based on the obtained results, we can safely conclude that the best optimiser out of the 4, is the BIPOP.

**References**

[1] K. Sörensen and F. Glover. Metaheuristics. In S.I. Gass and M. Fu, editors, Encyclopedia of Operations Research and Management Science, New York, To appear. Springer.

[2] Back, T., Hoffmeister, F., & Schwefel, H.-P. (1992). Applications of evolutionary algorithms. Report of the Systems Analysis Research Group SYS-2/92, University of Dortmund, Department of Computer Science.

[3] Alatas, B., (2011). Uniform Big Bang–Chaotic Big Crunch Optimization, Communication Nonlinear Science and Numerical Simulation, 16(9), 3696-3703.

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

[5] Kar AK (2016) Bio-inspired computing—a review of algorithms and scope of applications. Expert Syst Appl 59:20–32

[6] Auger, A., & Hansen, N. (2005, September). A restart CMA evolution strategy with increasing population size. In 2005 IEEE congress on evolutionary computation (Vol. 2, pp. 1769-1776). IEEE.

[7] Szynkiewicz, P. (2018). Comparative study of PSO and CMA-ES algorithms on black-box optimization benchmarks. Journal of Telecommunications and Information Technology.

[8] V. L. Huang, S. Z. Zhao, R. Mallipeddi, P. N. Suganthan, "Multi-objective optimization using self-adaptive differential evolution algorithm", Evolutionary Computation 2009. CEC '09. IEEE Congress on, pp. 190-194, 2009.

[9] Fouskakis, D., & Draper, D. (2002). Stochastic optimization: a review. International Statistical Review, 70(3), pp. 315-349.

[10] Mathew, T. V. (2012). Genetic algorithm. Report submitted at IIT Bombay.

[11] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In Proc. of IEEE International Conference on Neural Networks, volume 4, pages 1942-1948, 1995.

[12] Petr Pošík and Václav Klemš. 2012. Benchmarking the differential evolution with adaptive encoding on noiseless functions. In Proceedings of the 14th annual conference companion on Genetic and evolutionary computation. ACM, 189--196.

[13] Caraffini, F., & Neri, F. (2018). Rotation invariance and rotated problems: An experimental study on differential evolution. In International Conference on the Applications of Evolutionary Computation (pp. 597-614). Springer, Cham.

[14] Klemš, V. Differential evolution with adaptive encoding (Doctoral dissertation, Master’s thesis, Czech Technical University in Prague, 2011. Available online, https://cyber.felk.cvut.cz/theses/papers/177.pdf) [Accessed 15 April 2022].

[15] 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

[16] Hansen, N. (2008). Adaptive encoding: How to render search coordinate system invariant. In International Conference on Parallel Problem Solving from Nature (pp. 205-214). Springer, Berlin, Heidelberg.

[17] Loshchilov, I. (2013, June). CMA-ES with restarts for solving CEC 2013 benchmark problems. In 2013 IEEE Congress on Evolutionary Computation (pp. 369-376). Ieee.

[18] Ros, R. (2010, July). Black-box optimization benchmarking the IPOP-CMA-ES on the noiseless testbed: comparison to the BIPOP-CMA-ES. In Proceedings of the 12th annual conference companion on Genetic and evolutionary computation (pp. 1503-1510).

[19] Kumar, M., Husain, M., Upreti, N., & Gupta, D. (2010). Genetic algorithm: Review and application. Available at SSRN 3529843.

[20] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80(5), pp. 8091-8126.

[21] Hu, X. (2006). PSOTutorial. Particle Swarm Optimization: Tutorial. Retrieved April 20, 2022, from <http://www.swarmintelligence.org/tutorials.php>

[22] Hughes, E. J. (1998). Optimisation using population based Incremental Learning (PBIL). IEEE Xplore. Retrieved April 25, 2022, from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=744258>

**Source Code**

[BIPOP] <http://www.cmap.polytechnique.fr/~nikolaus.hansen/publications.html#hansenTEC2009>

[DEAE] <https://numbbo.github.io/data-archive/bbob/>

[GA] <https://www.mathworks.com/matlabcentral/fileexchange/52856-binary-and-real-coded-genetic-algorithms>

[PSO] <https://numbbo.github.io/data-archive/bbob/>