Interdisciplinary Journal of Research and Development
ISSN 2410-3411 (online) / ISSN 2313-058X (print)
Vol 11 No 2 / July 2024

Performance Analysis of Metaheuristic Algorithms on Benchmark Functions

Edjola Naka*

Eris Zeqo

Alsa Kaziu

Faculty of Information Technology,
University "Aleksandër Moisiu",
Durres, Albania
**Corresponding Author*

Received: 8 June 2024 / Accepted: 25 July 2024 / Published: 29 July 2024

Doi: 10.56345/ijrdv11n202

Abstract

The discipline of optimization can be used to maximize or minimize several problems. The use of metaheuristic algorithms is a strategy that often works well for global optimization. They are a type of stochastic algorithm that, via trial and error, finds workable solutions to difficult optimization problems in a reasonable amount of time, but they do not provide assurance that the answers are optimal. This paper aims to offer a comparative analysis of several metaheuristics in searching for the optimal solution. The selected metaheuristics are Artificial Bee Colony, Ant Lion Optimizer, Bat, Black Hole, Cuckoo Search, Cat Swarm Optimization, Dragonfly, Differential Evolution, Firefly, Genetic, Gravitational-Based Search, Grasshopper Optimization, Grey Wolf Optimizer, Harmony Search, Krill-Herd, Moth-Flame Optimizer, Particle Swarm Optimization, Sine Cosine, Shuffled Frog-Leaping, and Whale Optimization algorithms. For this evaluation, 18 benchmark test functions, categorized as unimodal, multimodal, and fixed-dimension multimodal are used to examine various properties, such as accuracy, escape from the local optimum, and convergence. As an indicator of how effectively these metaheuristics work, metrics like minimum, maximum, average, and standard deviation of fitness are provided. There are no optimization algorithms that are adequate for all problems, as the No Free Lunch theorem suggests, but the metaheuristics that are more effective than the others will be demonstrated. This study could be helpful for young researchers to identify the most prominent metaheuristics for achieving a better global optimum.

Keywords: *Optimization, Metaheuristic algorithm, Benchmark function, Performance metric*

1. Introduction

The objective of global optimization is to identify the optimal solution x^* from a set X on a set of objective functions $F = \{f_1, f_2, \dots, f_n\}$. The determination of the global optimum is a crucial objective in the search process, necessitating the avoidance of regions proximate to local minima to the greatest extent feasible. Optimization algorithms are employed to assess the optimal solution. Optimization algorithms can be classified into several categories, including non-heuristic, heuristic, metaheuristic (MH), surrogate-based, hyper-heuristic, and hybrids (Stork, Eiben, & Bartz-Beielstei, 2022).

MHs refer to optimization algorithms that are not specific to resolve a particular problem and aim to provide a solution that is seen as satisfactory (Abdel-Basset , Abdel-Fatah, & Sangaiah , 2018). MHs accomplish two primary directions, namely exploration and exploitation. The utilization of exploration in MHs results in a global search approach, whereas exploitation leads to a local search tendency. The effectiveness of the search process is conditioned upon achieving an appropriate balance between these two functions. The aim is to enhance the quality of available solutions via iterative processes, commencing with a set of feasible solutions generated randomly and subsequently exploring and exploiting the solution space. Despite the absence of a guarantee of optimality, these algorithms have been subjected to testing and have demonstrated the ability to provide a satisfactory and acceptable solution (Bandaru & Deb, 2017).

In recent years, various classifications of MHs have been proposed. MHs can be classified into two categories, namely metaphor-based and non-metaphor-based, as stated in (Abdel-Basset , Abdel-Fatah, & Sangaiah , 2018). Metaphor-based MHs refer to algorithms that replicate natural phenomena, human behavior in contemporary society, mathematical concepts, and other related phenomena. Conversely, non-metaphor-based MHs do not rely on simulation to establish their search strategy. Another classification is by their source of inspiration, which includes evolutionary, swarm intelligence, physical law-based, and miscellaneous approaches (Rajwar, Deep, & Das, 2023). The number of MHs has been observed to keep increasing over time. The authors conducted an investigation into the proposal of MHs, resulting in a list of approximately 540 MHs that were examined up until the year 2022 (Rajwar, Deep, & Das, 2023). As per the data obtained from Google Scholar until December 31, 2022, the MHs that have received the highest number of citations are PSO with 75000 citations, followed by GA with 70000 citations, Simulated Annealing with 50000 citations, DE with 30000 citations, and Ant Colony Optimization (ACO) with 15000 citations

The availability of a diverse range of algorithms provides the opportunity to select an appropriate one for addressing a problem based on its characteristics. The utilization of benchmark test functions is a viable approach to assessing the balance between exploration and exploitation in MHs. Numerous benchmark functions have been suggested in prominent CEC competition sessions pertaining to extensive global optimization across various years, CEC'05, CEC'13, CEC'14, CEC'17, CEC'20, CEC'21 (Suganthan, et al., 2005), (Liang, Qu, Suganthan, & Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, 2013), (Liang, Qu, & Suganthan, 2013), (Wu, Mallipeddi, & Suganthan, 2016), (Liang, Suganthan, Qu, Gong, & Yue, 2020), (Mohamed, Sallam, Agrawal, Hadi, & Mohamed, 2023). Also, a number of engineering problems have been suggested for the purpose of testing their efficacy in real-world scenarios. Some examples include the design of a three-bar truss, cantilever beam, 52-bar truss, tension/compression spring, welded beam, and pressure vessel, among others (Saremi, Mirjalili, & Lewis , 2017), (Mirjalili, Mirjalili, & Andrew, 2014).

The objective of this study is to conduct a comparative analysis of 20 MHs across 18 test functions in order to determine which of these algorithms yields superior fitness values across the majority of functions. The present paper's organization is structured in the following manner: Section II provides a concise overview of the employed methodologies, including the algorithms, benchmark functions, and similar related work. The subsequent section outlines the parameters, the experiment analysis, and the outcomes. Section IV presents the conclusions, and discussion about the results. The paper also includes the mathematical formulas for the test functions in the Appendix.

2. Materials and Methods

In this section, general information is provided about the MH algorithms, and the list of benchmark functions. Additionally, a brief review of relevant literature from comparable studies is presented.

2.1 Metaheuristic optimization algorithms

2.1.1 Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm is developed based on the behavior of bees in terms of foraging and abandoning food sources (Karaboga & Akay, 2009) and is in the category of swarm-based MHs. In the initial stage, the ABC algorithm generates an initial population of food source positions, which are solutions randomly distributed among a group of employed bees or onlooker bees. Following the initialization phase, the population of positions, or solutions, undergoes a series of iterative cycles, during which the employed bees, onlooker bees, and scout bees engage in search processes. During all the phases each of the bees alters its memory of the location and evaluates the fitness (quantity of nectar) present at a potential source. In the event that the fitness surpasses that of its prior discovery, the bee will commit the updated location to memory while disregarding the previous one. The aforementioned process is iterated for a specified number of cycles or until a termination criterion is met.

2.1.2 Ant Lion Optimizer algorithm

The Ant Lion Optimizer (ALO) is a MH that is based on swarm intelligence and aims to replicate the hunting behavior of antlions in their natural habitat (Mirjalili, The Ant Lion Optimizer, 2015). Antlions are known to prey on ants. The method comprises five primary steps, namely: (1) conducting random walks of ants; (2) constructing traps; (3) entrapping ants within the constructed traps; (4) capturing prey; and (5) reconstructing the traps. Ants employ random walks to navigate through a search area that is subject to the presence of antlion traps. Antlion traps have an impact on the random behavior of ants. The process of selecting antlions for optimization is accomplished through the utilization of a roulette wheel selection approach, which is based on the antlions' respective fitness values. The antlion that exhibits the highest level of fitness in each iteration is preserved as the elite. In addition, the updates to the positions of ants are conditioned upon the random walks of both the selected antlion and the elite. If an ant exhibits superior fitness compared to an antlion, the antlion is substituted for the corresponding ant. The antlion might replace the elite if it exhibits better fitness.

2.1.3 Bat Algorithm

Yang was the first to introduce the Bat Algorithm (BA), which takes its inspirations from the echolocation practices of bats that involve varying emission pulse rates and loudness (Yang, A New Metaheuristic Bat-Inspired Algorithm, 2010). This falls under the classification of swarm-based MHs. Each simulated bat exhibits stochastic flight patterns, characterized by a velocity vector at a given position and modulated by frequency, wavelength, and loudness parameters. The bat regulates its frequency, loudness, and pulse emission rate during the process of locating and capturing its prey. The local random walk technique enhances the intensity of the search process. The process of selecting the optimal solution continues until specific termination conditions are satisfied. The proposed approach employs a frequency-tuning methodology to regulate the dynamic behavior of a bat swarm.

2.1.4 Black Hole Optimization algorithm

Hatamlou introduced the Black Hole algorithm (BH), which belongs to the class of physics-based MHs (Hatamlou, 2013). The algorithm in question draws inspiration from the black hole phenomenon and attempts to replicate its characteristic of gravitational attraction towards other celestial bodies in the universe. The algorithm initiates by generating a population of candidate solutions, commonly referred to as "stars," in a random manner. Following initialization, the population's fitness values are assessed and the individual with the highest fitness value is designated as the black hole, while the remaining individuals constitute the normal stars. As a star approaches a black hole, it is possible for it to encounter a region with a lower gravitational potential energy than that of the black hole. In the event of such an occurrence, the black hole goes through a change to the position of the aforementioned star, and conversely. The optimal solution is often referred to as a "black hole" and all alternative solutions are expected to converge towards it. The algorithms may terminate upon reaching a maximum number of iterations or achieving a satisfactory level of fitness.

2.1.5 Cuckoo Search Algorithm

Yang and Deb drew inspiration from the breeding practices of specific cuckoo species, particularly their brood parasitism, to develop the Cuckoo Search (CS) swarm-based algorithm (Yang & Deb, Cuckoo Search via Lévy flights, 2009). Several assumptions are made in relation to this algorithm. 1) the reproductive behavior of cuckoos involves the deposition of a single egg into a randomly selected host nest. 2) the best nests with high-quality eggs will carry over to the next generation. 3) the probability of a host bird discovering a cuckoo egg depends on a range of factors, with values ranging from 0 to 1, and 4) the number of available host nests remains constant. A cuckoo is stochastically generated through a Levy flight mechanism and subsequently subjected to a fitness assessment. In a nest, each individual egg can be viewed as a representation of a solution, while the presence of a cuckoo egg can be interpreted as a new solution. The objective is to employ new and potentially superior alternatives (cuckoos) for the purpose of substituting suboptimal solutions within the nests. Once the optimal nest has been identified, the solutions are ranked, and the most favorable one is preserved.

2.1.6 Cat Swarm Optimization algorithm

Shu-An Chu et al. introduced the Cat Swarm Optimization (CSO) swarm-based MH (Chu, Tsai, & Pan , 2006). This approach employs cats and their behavioral patterns to address optimization problems and represent solution sets. The CSO algorithm is comprised of two distinct modes, specifically tracing and seeking modes. Each feline entity embodies a distinct set of solutions, characterized by its unique position, fitness value, and flag. The position consists of M dimensions within the search space, with individual velocities assigned to each dimension. The fitness value represents the efficacy of the solution set (cat), while the flag serves to categorize the cats as either being in seeking or tracing mode. Therefore, it is necessary to establish the number of felines to be involved in the iteration and subject them to the algorithmic process. The optimal cat from each cycle is stored in the memory, and the cat found at the ultimate cycle will serve as the final solution.

2.1.7 Dragonfly optimization algorithm

Mirjalili proposed the Dragonfly Algorithm (DA), which draws inspiration from the hunting and migration strategies of dragonflies (Mirjalili, 2016). The hunting strategy commonly referred to as static swarming" (feeding) involves the coordinated flight of all members of a swarm in small clusters within a restricted region to locate sources of food. Dragonflies employ a migratory technique known as a dynamic swarm. During this phase, dragonflies exhibit a tendency to fly in larger groups, which can facilitate their migration as a swarm. The exploration and exploitation phases of dragonflies are guided by five distinct factors: separation, alignment, cohesion, food factor, and enemy factor. The objective is to guarantee the survival of the swarm by drawing it towards sources of nourishment and diverting its attention from potential threats. The optimal outcome identified during an iterative process is designated as the food source, while the worst outcome identified is designated as the enemy. The artificial dragonflies' neighbors are determined by defining a radius around each individual specimen. In the final phase of optimization, the dragonflies will integrate into a dynamic swarm that will ultimately converge toward the optimal global solution. The iterative process involves updating the step vector and position vectors of each dragonfly until the specified termination condition is satisfied.

2.1.8 Differential Evolution Algorithm

Storn and Price introduced the Differential Evolution (DE) algorithm (Storn & Price , 1997). DE is a MH search algorithm that employs an evolutionary approach to optimize a given problem. The algorithm iteratively enhances a candidate solution to achieve the optimal solution. The algorithm explores the design space through the utilization of a population of potential solutions, also known as individuals. These individuals are subjected to a particular process that involves the combination of existing solutions to generate novel solutions. The utilization of mutation and crossover operators is a common practice in generating novel solutions. In each iteration of the algorithm, individuals with superior objective values are selected to be retained in the population, while those with inferior objective values are eliminated. This process ensures that only individuals with improved objective values are included in subsequent iterations. The aforementioned procedure iterates until a specified termination condition is met.

2.1.9 Firefly Algorithm

Yang initially proposed the Firefly algorithm (FFA), a MH algorithm based on swarm intelligence (Yang, Firefly Algorithms for Multimodal Optimization, 2009). The algorithm emulates the behavioral patterns of fireflies through the utilization of their flashing lights. The level of attraction exhibited by a firefly is directly correlated to its brightness, which is dependent upon its objective function. The algorithm involves the assignment of light intensity to fireflies, which are randomly generated feasible solutions, based on their performance in the objective function. The aforementioned intensity shall be utilized for the purpose of calculating the luminosity of the firefly, which exhibits a direct correlation with its light intensity. In the context of minimization problems, the solution that yields the smallest functional value will be attributed with the greatest intensity of light. After assigning the intensity or brightness of the solutions, each firefly will subsequently pursue other fireflies that exhibit superior light intensity. The most luminous firefly will execute a localized exploration by undertaking stochastic movements within its immediate vicinity.

2.1.10 Genetic Algorithm

Darwin's theories of biological evolution serve as the foundation for the genetic algorithm (GA), which Holland modified for optimization purposes (Holland, 1992). Subsequently, Mitchell provided a more comprehensive exposition of GA (Mitchell, 1998) [23]. The GA algorithm is designed to imitate the natural phenomena of chromosomes and genes. In the context of optimization problems, it can be observed that every solution can be represented as a chromosome, with each gene within the chromosome corresponding to a variable of the problem. The GA employs a combination of primary operators, namely selection, crossover, and mutation, to enhance the chromosomes in every succeeding generation. The process of evolution typically commences with a population of arbitrarily generated individuals and proceeds iteratively, with each iteration's population referred to as a generation. The evaluation of the fitness of each member of a given population is a recurring process across generations. The process of selection is employed to choose the fittest individuals from the existing population, and subsequently, their genomes undergo modification through crossover and random mutation to generate a new generation. The algorithm terminates when either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population.

2.1.11 Gravitational Based Search Algorithm

Rashedi et al. introduced a physics-based optimization algorithm that draws inspiration from the laws of gravity and mass interactions, named Gravitational Based Search algorithm (GSA) (Rashedi, Nezamabadi-pour, & Saryazdi, 2009). The algorithm under consideration regards agents as objects and evaluates their efficacy based on their respective masses. The phenomenon of gravitational attraction results in the mutual attraction of objects, leading to a collective motion of all objects towards those with greater mass. The algorithm ensures the exploitation step by causing the heavier masses, which represent optimal solutions, to move at a slower rate than the lighter ones. GSA assigns four distinct specifications to each agent, namely: position, inertial mass, active gravitational mass, and passive gravitational mass. Stated differently, every mass offers a solution, and the procedure is guided by appropriately modifying the gravitational and inertia masses. Over time, it is anticipated that objects will be drawn towards the most massive object due to gravitational attraction. This mass is expected to offer an optimal solution within the given search space. It can be inferred that agents exhibiting superior performance possess a larger gravitational mass. Consequently, the agents exhibit a tendency to gravitate towards the most optimal agent.

2.1.12 Grasshopper Optimization Algorithm

Saremi et al. have developed the Grasshopper Optimization Algorithm (GOA) which emulates the collective behavior of grasshopper swarms through mathematical modeling (Saremi, Mirjalili, & Lewis , 2017). Each individual grasshopper within the population is representative of a solution, and its calculation is determined by three distinct forces: social interaction with other grasshoppers, wind advection, and the force of gravity acting upon the solution. The process starts with a given population of grasshoppers, followed by an assessment of each solution through the computation of its value utilizing the fitness function. Upon assessing each solution within the population, we allocate the optimal solution based on its respective value. Subsequently, the coefficient parameter c is updated to reduce the sizes of the attraction, repulsion, and comfort zones. Afterward, the solutions within the population are revised by taking into account the distance between the grasshopper, which represents the solution, and the other grasshopper, which represents another solution. The normalization process ensures that the distance is scaled to a range between 1 and 4. Following that, the optimal location of the superior grasshopper is determined by applying the diminishing coefficient factor c which facilitates the alignment of swarm convergence towards the designated objective.

2.1.13 Grey Wolf Optimizer

Mirjalili et al. proposed the Grey Wolf Optimization (GWO) algorithm, which emulates the hierarchical leadership structure and hunting behavior of gray wolves found in the natural world (Mirjalili, Mirjalili, & Andrew, Grey Wolf Optimizer, 2014). It falls into the category of swarm-based MHs. The leadership hierarchy is simulated through the utilization of four distinct categories of gray wolves, namely alpha, beta, delta, and omega. The top three performers are denoted as alpha, beta, and delta, while the remaining individuals are classified as omega. The GWO employs a hunting strategy that is directed by the three key individuals. Wolves often exhibit leadership behavior by guiding their pack members towards the most optimal areas when searching for resources. The process of iterative searching involves the evaluation of potential prey locations by the group of alpha, beta, and delta. The omega undergoes a relocation process based on the positioning of three other wolves. Furthermore, the three primary stages of hunting, namely locating prey, surrounding prey, and engaging in an attack on prey, are executed. The outcome generated by the first search agent will be retrieved.

Harmony Search algorithm

Geem et al. first introduced the Harmony Search (HS) algorithm, which drew its inspiration from the fundamentals of musical harmony improvisation (Geem, Kim, & Loganathan, 2001). It is a physics-based algorithm. During an improvisation, a musician is presented with three potential options: firstly, to perform a renowned musical piece, consisting of a sequence of harmonious pitches, from memory with precision; secondly, to play identically to a familiar composition while making slight adjustments to the pitch; or thirdly, to create original or arbitrary notes. The names of this phases are harmony memory, pitch adjustment, and randomization. The algorithm commences by initializing harmonics and subsequently generates new solutions by employing one of the aforementioned procedures. The first approach involves the selection of a limited number of optimal harmonies, while the subsequent approach introduces alterations to the solutions comparable to a genetic mutation. Additionally, the incorporation of randomization serves to enhance the variety of available solutions. Ultimately, the optimal solution has been chosen.

2.1.14 Krill-Herd Algorithm

The Krill-Herd (KH) algorithm is based on a simulation of the collective herding behavior that krill organisms exhibit when searching for food. This swarm-based MH was proposed by Gandomi and Alavi in their work (Gandomi & Alavi , 2012). The minimum distances of each individual krill from food and from the highest density of the herd are considered the objective functions of the krill's movement. The position of individual krill is affected by three aspects: krill movement due to other individuals; foraging behavior; and random diffusion. In order to enhance the efficacy of the algorithm, genetic reproduction mechanisms, namely crossover and mutation, have been integrated into the algorithm, drawing inspiration from the classical DE algorithm. Then the krill positions are updated until the end criterion is reached.

2.1.15 Moth-Flame Optimizer algorithm

Mirjalili developed the Moth-Flame Optimizer (MFO), a widely used swarm-based algorithm that emulates the transverse navigation mechanism of moths during night flights (Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, 2015). The MFO algorithm under consideration operates under the assumption that the candidate solutions can be represented as moths, while the variables of the problem are indicative of the moths' respective positions within the space. The fitness value corresponds to the output of the fitness (objective) function for every individual in a given population. The algorithm incorporates the presence of flames. Both moths and flames can be considered potential solutions. Flames serve as the moths' most advantageous position thus far as they move through the search space as search agents. Thus, every moth conducts a search in close proximity to a flame and modifies its position upon discovering an improved solution. Through this mechanism, the moth is able to retain its optimal solution without any loss. During the optimization process, it is observed that every moth in the present iteration navigates around its respective flame by means of a logarithmic spiral. Each moth is assigned a specific flame.

2.1.16 Particle Swarm Optimization

Kennedy and Eberhart introduced the initial swarm-based MH of particle swarm optimization (PSO) (Kennedy & Eberhart, 1995). A group of particles navigates through a search space of D dimensions in pursuit of an optimal solution. Every individual particle is characterized by its current position and velocity vector. The conventional PSO procedure commences with the random initialization of the position and velocity vectors. During each iteration, the optimal position discovered by an individual particle as well as the optimal position discovered by the entire swarm, are utilized to update the velocity and position of the particles.

2.1.17 Sine Cosine Algorithm

The sine cosine algorithm (SCA) is a meta-heuristic algorithm that was developed by Mirjalili inspired by a mathematical approach (Mirjalili, 2016). The utilization of trigonometric sine and cosine functions for updating the positions of individuals towards the optimal solution characterizes a population-based MH approach to address optimization problems. The utilization of four parameters, namely r1, r2, r3, and r4, serves the purpose of preventing the occurrence of local optima and maintaining a balance between exploratory and exploitative search patterns. The parameter r1 plays a crucial role in determining the direction of a solution's position update. The parameter r2 governs the magnitude of a solution's movement towards or away from its destination. A further stochastic variable, denoted as r3, allocates a probabilistic weight to the target location. The final random parameter, denoted as r4 functions as a binary switch, determining whether the trigonometric function utilized will be the sine or cosine.

2.1.18 Shuffled frog-leaping algorithm

In 2006, Eusuff et al. introduced a memetic MH algorithm known as the shuffled frog-leaping algorithm (SFLA) (Eusuff, Lansey, & Pasha, 2006). It falls in the category of swarm-based algorithms. According to research, frogs are regarded as carriers of memes that facilitate the exchange of information with other frogs, thereby influencing the relocation of their habitats to areas within the swamp that are abundant in food resources. Initially, a cohort of frogs is randomly produced and subsequently partitioned into various memeplexes within the swamp. Subsequently, the selection of frogs within each memeplex is carried out through triangular probability distributions, thereby facilitating the formation of novel sub-memeplexes. In the context of memeplex optimization, the host exhibiting the lowest performance within each sub-memeplex engages in meme optimization and repositions its stone by means of information exchange with either the most high-performing host in the entire population or the most high-performing host within the sub-memeplex. Upon completion of memetic evolution, the sub-memeplexes are ultimately reorganized and aggregated into a novel population, which then undergoes a continuous iterative process of searching for the most nutritionally abundant regions within the swamp.

2.1.19 Whale Optimization Algorithm

Mirjalili and Lewis developed the Whale Optimization Algorithm (WOA) to simulate the social behavior of humpback whales (Mirjalili & Lewis, 2016) [31]. It falls in the category of swarm-based MHs. The method employed by the subject for hunting is commonly referred to as the bubble-net feeding method. The process comprises three primary stages, namely, prey encirclement, bubble-net assault, and prey exploration. Whales possess the ability to locate their prey and subsequently surround it in a circular manner. The bubble net attack is a technique that involves identifying the target's location and subsequently attempting to encircle the area surrounding the optimal solution through a process referred to as "shrinking encircling". The calculation of the distance between the present location and the newly encircled points is utilized to update the location. The phenomenon under discussion is commonly referred to as the "spiral updating position". In the interim, individual whales engage in foraging behavior by utilizing the indications provided by other species. The whale with the highest fitness value is identified, and this process is repeated until the desired outcomes are attained or the iteration limit is surpassed.

2.2 Benchmark test functions

Test functions are crucial for evaluating and comparing the effectiveness of optimization techniques. Numerous functions with a variety of properties have been proposed over the years for evaluating the global optimum. In the work of Jamil and Yang, 175 unconstrained optimization test problems are summarized and can be used to assess how well optimization methods perform (Jamil & Yang, 2013). Each function's description and classification are given in depth. Modality, basins, valleys, separability, and dimensionality are a few of these characteristics (Jamil & Yang, 2013), (Hussain, Mohd Salleh, Cheng, & Naseem, 2017). We considered a list of 18 test functions divided into benchmark unimodal, multimodal, and fixed-dimension multimodal functions. The unimodal functions (F1–F7), which have just one global optimum, are appropriate for measuring the exploitation and convergence of the algorithms. Each of the multimodal functions (F8–F12) in the second category has a huge number of local optimum points and a global optimum. They are quite useful in examining how algorithms explore and avoid local optima. Meanwhile, the capability to explore in low dimensions is demonstrated using the fixed-dimension multimodal functions (F13–F18). The mathematical formula, the size, the range of variables, and the minimal ideal value are all provided for each function in accordance with the recommendations made in (Moghdani & Salimifard, 2018)'s work and are shown in the Appendix.

2.2.1 Related work

A comprehensive review of relevant literature has been carried out to examine comparable publications that assess the efficacy of distinct MHs across various benchmark functions. On May 10th, 2024, a search was conducted on Google Scholar using the keywords "review," "metaheuristic algorithms," and "benchmark functions." The selection criteria for papers require the inclusion of at least one of the twenty specified MHs. The selection process for papers is limited to those that exclusively compare MHs through the use of benchmark functions. Furthermore, research papers that aimed to contrast a novel or enhanced MH with other pre-existing MHs were excluded from consideration as their primary objective was not to compare these methodologies but rather to evaluate their relative performance against other MHs. Given the widespread evaluation of the field of metaheuristics and their performance comparisons, the following paragraph highlights a few of the selected articles.

Demirhan et al. conducted a comparative analysis of PSO and FFA on five distinct benchmark functions. The utilization of the PSO was observed to produce comparatively superior outcomes in contrast to the primary firefly algorithm (Demirhan, Özkaraca, & Güvenç, 2021). Another study has successfully addressed the FFA in a total of 11 benchmark functions. The obtained results have been subjected to a comparative analysis with those obtained through the implementation of other MH algorithms such as BA, Bacteria Foraging Algorithm (BFA), and CS. The comparative analysis indicates that FFA exhibited superior performance compared to BA and BFA while demonstrating comparable outcomes to CS (Dhawan, 2022). (Ab Wahab, Nefti-Meziani, & Atyabi, 2015) conducted a comparative analysis of various swarm-intelligence algorithms across 30 benchmark functions. The algorithms encompassed in this set are GA, ACO, PSO, DE, ABC, Glowworm Swarm Optimization, and CS algorithms. DE seems to be the best overall performing approach, outperforming other methods in 24 out of 30 functions followed by PSO with the best performance in 19 out of 30. The third best is GA with 14 out 30 best performances and closely followed by ACO with 13 out of 30 best performances. On 10 benchmark functions, including unimodal and multimodal ones, Joshi et al. examined the performance of GA, Simulated Annealing, and PSO (Josh, Gyanchandani, & Wadhvan, 2021). According to the results, PSO outperformed GA and SA for most of the benchmark functions that were used. An experimental examination of the most recent 10 swarm-based MH algorithms was conducted on 12 well-known benchmark test functions with various characteristics for additional performance analysis. The FA, GWO, and Animal Migration Optimization (AMO) are very promising MH algorithms that can be utilized to resolve challenging unimodal and multimodal optimization issues, according to the simulation results (Hussain, Mohd Salleh, Cheng, & Shi, 2017). In the work of (Karaboga & Akay, 2009) it was compared the performance of ABC algorithm with those of GA, DE, PSO algorithms on a large set of unconstrained test functions, such as unimodal, multimodal, regular, irregular, separable, non-separable and multidimensional. One major conclusion was that ABC is more successful and most robust on multimodal functions included in the set respect to the other MHs. (Mirjalili, The Ant Lion Optimizer, 2015) compared ALO with PSO, GA, States of Matter Search (SMS), BA, Flower Pollination Algorithm (FPA), CS, and FFA where ALO outperformed most of them regarding unimodal functions emphasizing its capability in high exploitation. Moreover, ALO shows the fastest convergence on multimodal test functions. (Mirjalili, Mirjalili, & Andrew, Grey Wolf Optimizer, 2014) has tested GWO in the included 18 functions here, and it is highlighted that GWO is superior in unimodal (exploiting), and multimodal functions (exploration), where in the last ones, it outperforms PSO, and GSA, and DE.

3. Experiment Analysis and Results

This section will present the results of applying the 20 MHs to each selected benchmark function.

3.1 Experiment parameters

Forty search agents are taken into consideration for each method throughout 100 iterations in order to solve the unimodal and multimodal benchmark functions. The dimensions (Dim) and the range of each metaheuristic (numVar) when applied to each benchmark function is defined as in the Appendix section. Each MH is applied 30 times in order to conduct a fair comparison. For each MH, the fitness value is displayed. The final fitness corresponds to the optimal value generated from the objective function of each benchmark function. The average, standard deviation, and best and worst fitness values from the 30 runs are finally presented. Lower average and standard deviation values demonstrate an algorithm's greater ability to avoid local solutions and determine the global optimum. Finally, we calculate the execution time required from each MH. All the selected MH are provided from the metaheuristicOpt R package (Riza, et al., 2019). Table 1 provides a summary of the default parameters for each MH.

Table 1. Parameters of each metaheuristic

Metaheuristic	Parameters
PSO	Vmax (maximal velocity) =2, ci (individual cognitive) =1.49445, cg (group cognitive) =1.49445, w (inertia weight) =0.729
ABC	-
ALO	-
BA	maxFrequency = 0.1, minFrequency = -0.1, gamma (increase pulse rate) = 1, alphaBA (decrease loudness) = 0.1
BHO	-
CS	abandonedFraction = 0.5
CSO	mixtureRatio = 0.5, tracingConstant = 0.1, maximumVelocity = 1, smp = as.integer(20), srd = 20, cdc = as.integer(numVar)
DA	-
DE	scalingVector = 0.8, crossOverRate = 0.5, strategy of mutation = "best 1"
FFA	B0 (attractiveness firefly) = 1, gamma (light absorption coefficient.) = 1, alphaFFA (randomization parameter) = 0.2
GA	Pm (mutation probability) = 0.1, Pc (crossover probability) = 0.8
GBS	gravitationalConst = max(rangeVar), kbest = 0.1
GOA	-
GWO	-
HS	PAR (Pinch Adjusting Ratio) = 0.3, HMCR (Harmony Memory Consideration Rate) = 0.95, bandwith = 0.05
KH	maxMotionInduced = 0.01, inertiaWeightOfMotionInduced = 0.01, epsilon = 1e-05, foragingSpeed = 0.02, inertiaWeightOfForagingSpeed = 0.01, maxDifussionSpeed = 0.01, constantSpace = 1
MFO	-
SCA	-
SFL	numMemeplex = as.integer(40/3), frogLeapingIteration = as.integer(10)
WOA	-

3.2 *Results of average and standard deviation of fitness on each function*

Tables 2 and 3 display the mean (Avg) and standard deviation (SD) of the 20 MH fitness values. The selected functions are unimodal functions that are used to evaluate the exploitation phase and the rate of algorithm convergence. The minimum average and SD are highlighted for each table. Based on the results of the simulations, BA has effectively provided the best solution for F1, F2, F3, and F4 out of the seven unimodal functions that were studied in this study. The design of BA is more focused on exploitation than exploration (Shehab, et al., 2023), explaining its good results in unimodal functions. Subsequently, it can be observed that GWO yields superior outcomes in terms of the mean fitness for F5 and F6, and GWO has also been considered superior in unimodal functions (Mirjalili, Mirjalili, & Andrew, 2014). However, the standard deviation of BA indicates that the fitness values are more closely clustered around the mean compared to GWO. SFL produces the optimal mean value for F7, while GOA yields the most favorable standard deviation.

Table 2. Results on the unimodal test functions (10 first MHs)

	Statistic	PSO	ABC	ALO	BA	BHO	CS	CSO	DA	DE	FFA
F1	Avg	4.58	17933.07	2764.80	**8.67E-185**	1496.72	48445.87	8.33E-08	2999.99	6446.02	62519.56
	SD	2.15	3083.76	1125.17	**0**	692.25	4132.50	2.37E-07	1732.65	1399.97	7692.35
F2	Avg	1.39	50.90	80.55	**3.89E-85**	10.2	60700530145773882	0.0011	22.02	40.89	1.33E+13
	SD	0.29	6.22	34.10	**2.13E-84**	2.73		0.0022	8.93	7.30	4.27E+13
F3	Avg	109.73	250866	40927.44	**4.84E-191**	35847.86	663735.1	1.1E-06	37578.74	72654.8	875225
	SD	50.97	63242.06	19034.59	**0**	14783.29	70391.67	4.95E-06	22529.17	14097.11	123419.5
F4	Avg	0.88	75.38	24.57	**2.17E-103**	14.58	76.91	2.28E-06	40.91	64.26	80.05
	SD	0.19	2.78	5.79	**9.09E-103**	2.62	2.65	2.94E-06	11.99	4.80	3.96
F5	Avg	117.99	30628529	797176.6	28.99	1548025	136934963	28.99	1039747	7587554	137857996
	SD	61.56	9121657	734992.4	**0.01**	1067404	22428131	0.03	1257752	3286001	22031016
F6	Avg	4.78	18083.21	2889.01	7.4	1832.22	47046.72	2.04	3465.51	6584.32	62055.01
	SD	1.74	3806.66	1217.92	**0.18**	909.87	4130.3	0.77	2699.13	1655.01	7997.47
F7	Avg	6.16	15.82	1.99	0.59	0.78	59.99	0.52	1.81	4.28	3.84

	SD	6.31	5.64	0.73	0.263	0.29	12.19	0.262	1.06	1.75	1.46

Table 3. Results on the unimodal test functions (10 second MHs)

	Statistic	GA	GBS	GOA	GWO	HS	KH	MFO	SCA	SFL	WOA
F1	Avg	5480.63	44.64	351.53	6.9E-12	616.97	18583.08	6060.14	3468.54	159.05	1.5E-23
	SD	1255.78	42.53	208.30	6.6E-12	118.14	14866.78	5085.11	2518.66	59.64	4.9E-23
F2	Avg	20.70	7.59	20.6	2.75E-07	5.45	1.30E+12	50.82	6.33	4.22	3.03E-14
	SD	3.34	2.56	17.91	1.9E-07	0.88	5.82E+12	16.49	3.55	1.26	4.17E-14
F3	Avg	66840.55	1625.64	4719.41	9.66E-11	7964.71	219272.7	87717.43	41139.43	1889.79	4.09E-22
	SD	20043.37	1909.94	4251.99	1.39E-10	2702.32	197110	53200.13	34082.72	933.04	1.14E-21
F4	Avg	51.28	9.71	13.83	0.009	31.7	60.08	69.10	67	20.91	66.58
	SD	6.08	2.66	4.07	0.004	3.42	16.68	8.10	9.3	4.27	19.75
F5	Avg	2894991	2666.84	24355.74	**27.37**	93781.63	23474242	6805662	11727249	6238.6	28.47
	SD	1298344	3318.17	21006.95	0.6	40761.76	26196728	15260678	14790659	4537.96	0.34
F6	Avg	4894.70	67.7	272.74	**0.68**	632.41	18929.72	5603.51	3011.13	149.00	0.78
	SD	1226.10	78.27	158.42	0.35	173.47	15071.85	4089.70	1977.88	124.25	0.28
F7	Avg	2.15	1.17	0.77	0.61	0.71	7.34	4.7	4.92	**0.48**	0.46
	SD	0.66	0.43	**0.261**	0.27	0.27	3.11	3.42	3.5	0.31	0.30

Tables 4 and 5 display the mean and standard deviation of fitness in relation to the multimodal functions. Multimodal functions have several optimal solutions and are used to test the exploration phase of the algorithm. BA has attained the optimal global value for F9 and F11, as well as the most optimal metrics for F10. The available data suggests that GWO produces the best results for F12 and has demonstrated effectiveness in multimodal functions (Grey Wolf Optimizer, 2014). HS provided the minimum average for F8, while ALO exhibited the lowest standard deviation for the same function. HS has been very suitable to solve multi-modal optimization problems, as have its improved variants (Tuo , et al., 2015) which can explain its results for the F8 function.

Table 4. Results on the multimodal test functions (10 first MHs)

	Statistic	PSO	ABC	ALO	BA	BHO	CS	CSO	DA	DE	FFA
F8	Avg	-1009.98	-1198.97	-1632.06	-790.29	-844.36	-646.06	-787.49	-1153.47	-1589.26	-554.87
	SD	135.52	64.14	**0**	104.90	87.87	47.72	191.81	107.19	109.54	98.10
F9	Avg	75.61	376.76	159.39	**0**	174.70	764.78	1.55E-06	253.1	336.32	824.18
	Sd	14.73	38.94	30.03	**0**	33.91	40.02	3.05E-06	65.03	26.51	91.66
F10	Avg	20.56	20.34	8.62	**4.44E-16**	20.26	21.16	2.97E-06	20.31	20.32	21.01
	Sd	0.50	0.06	5.94	**0**	0.07	0.06	4.92E-06	0.16	0.10	0.19
F11	Avg	0.17	5.56	1.78	**0**	0.86	13.24	1.61E-09	1.88	2.653	16.35
	Sd	0.05	0.81	0.23	**0**	0.15	1.21	3.7E-09	0.58	0.32	1.66
F12	Avg	0.17	38.07	11.56	1.32	11.05	129.67	0.19	11.5	50.02	59.21
	Sd	0.35	5.39	5.92	0.29	4.46	11.65	0.10	5.58	9.65	11.07

Table 5. Results on the multimodal test functions (10 second MHs)

	Statistic	GA	GBS	GOA	GWO	HS	KH	MFO	SCA	SFL	WOA
F8	Avg	-1613.64	-674.10	-1355.28	-1146.45	**-1868.55**	54.75	-1662.94	-819.61	-961.70	-1721.9
	Sd	48.18	140.04	118.40	122.55	9.31	646.15	69.89	76.62	74.99	127.89
F9	Avg	214.23	45.03	174.13	18.29	54.4	398.92	302.68	210.75	61.91	8.34E-14
	Sd	28.67	11.44	26.70	10.93	8.76	73.55	61.87	59.94	17.71	8.92E-14
F10	Avg	20.18	13.406	20.48	20.92	20.03	20.44	20.004	20.39	7.12	1.49E-12
	Sd	0.04	2.64	0.22	0.06	0.005	0.13	0.008	0.08	4.01	3.60E-12
F11	Avg	2.32	0.63	1.08	0.01	1.17	4.26	2.55	1.74	1.01	0.02
	Sd	0.28	0.31	0.06	0.02	0.04	1.89	1.28	0.49	0.10	0.09
F12	Avg	11.87	0.87	8.82	**0.048**	1.89	36.32	39.98	20.31	1.20	0.052
	Sd	2.46	0.58	5.87	**0.02**	0.6	23.44	9.3	11.27	0.61	0.04

Both Table 6 and Table 7 display the mean and standard deviation of fitness findings for the fixed-dimension multimodal.

Regarding the benchmark functions, only for the F17 function, PSO, ALO, DA, DE, and MFO yielded the optimal value (3). For the other functions, the optimums are not the same, as shown in Table 10 of the Appendix. Concerning the remaining functions, it has been observed that in certain instances, the minimum average is attained by a multitude of metaheuristic algorithms (12 in total), such as in the case of F13. The algorithms were PSO, ABC, ALO, BA, BHO, CSO, DE, GWO, MFO, SCA, SFL, and WOA.

Table 6. Results on the fixed-dimension multimodal test functions (10 first MHs)

	Statistic	PSO	ABC	ALO	BA	BHO	CS	CSO	DA	DE	FFA
F13	Avg	**2.1810381125**	**21810381125**	**21810381125**	**21810381125**	**21810381125**	22306364392	**21810381125**	21810381126	**21810381125**	27747592099
	Sd	0.03	0.0003	0.003	2.0E-06	2.37E-05	4.61E+08	0.15	5.67	**4.37E-07**	1.344E+10
F14	Avg	0.002	0.002	0.006	0.001	0.001	0.01	**0.0006**	0.004	0.005	2.10
	Sd	0.005	0.0008	0.007	0.0009	0.003	0.008	**0.0001**	0.007	0.008	9.48
F15	Avg	4.6525E-08	4.95E-08	4.65107E-08	0.0004	4.65105E-08	0.04	0.004	5.13E-08	4.651013E-08	0.49
	Sd	3.22E-11	3.93E-09	7.00E-13	0.0004	4.33E-13	0.03	0.004	2.55E-08	1.87E-14	0.60
F16	Avg	**16.960**	**16.960**	**16.960**	16.961	17.38	17.36	16.97	**16.960**	**16.960**	18.43
	Sd	1.7E-13	5.97E-15	8.05E-15	0.001	0.26	0.33	0.02	1.47E-13	**0**	1.48
F17	Avg	**3**	3.007	**3**	3.017	3.000001	3.66	5.87	**3**	**3**	46.37
	Sd	5.03E-10	0.01	3.53E-12	0.02	3.56E-06	0.7	15.4	1.72E-09	2.86E-12	42.76
F18	Avg	**-0.30**	**-0.30**	**-0.3**	-0.28	-0.19	-0.15	**-0.3**	**-0.3**	**-0.3**	**-0.3**
	Sd	**0**	**0**	**0**	0.03	0.03	0.05	**0**	**0**	**0**	**0**

Nine meta-heuristic algorithms, namely PSO, ABC, ALO, DA, DE, GWO, MFO, SCA, and WOA, achieve the optimal mean in a further case, F16. Out of the 20 metaheuristic algorithms considered (namely PSO, ABC, ALO, CSO, DA, DE, FFA, GBS, GOA, GWO, KH, MFO, SCA, and WOA), 14 of them demonstrate an average of -0.3 and a standard deviation of 0 for F18. This suggests that the majority of these algorithms are well-suited for exploring low-dimension optimization problems. The CSO yields the most optimal fitness for F14, while the MFO generates the best optimum for F15.

Table 7. Results on the fixed-dimension multimodal test functions (10 second MHs)

	Statistic	GA	GBS	GOA	GWO	HS	KH	MFO	SCA	SFL	WOA
F13	Avg	21837174687	21810411144	21810381126	**21810381125**	21815673278	21815289372	**21810381125**	**21810381125**	**21810381125**	**21810381125**
	Sd	29011825	35157.05	0.86	1.00E-06	12139470	6254273	1.87E-06	2.92E-06	0.032	3.88E-06
F14	Avg	0.009	0.009	0.01	0.009	0.005	0.28	0.001	0.002	0.002	0.002
	Sd	0.01	0.02	0.01	0.01	0.007	0.80	0.0004	0.004	0.007	0.004
F15	Avg	0.012	5.91E-06	4.687E-08	1.56E-07	1.27E-05	0.001	**4.651012E-08**	0.00027	4.65396E-08	4.32E-07
	Sd	0.015	6.95E-06	2.9E-10	1.18E-07	2.93E-05	0.001	**4.05E-17**	0.0005	1.61E-10	1.67E-06
F16	Avg	17.08	279.16	16.97	**16.960**	16.97	137.62	**16.960**	**16.960**	19.06	**16.960**
	Sd	0.11	25.91	2.76E-11	7.82E-10	0.006	16.96	**0**	2.92E-06	2.14	5.99E-14
F17	Avg	11.74	4.54	5.7	3.0004	3.93	3.008	**3**	3.0012	3.90	3.0002
	Sd	12.87	5.56	14.79	0.0007	4.93	0.007	**3.02E-15**	0.002	4.93	0.0008
F18	Avg	-0.27	**-0.3**	**-0.3**	**-0.3**	-0.3	**-0.3**	**-0.3**	**-0.3**	-0.06	**-0.3**
	Sd	0.02	**0**	**0**	**0**	0.0003	**0**	**0**	**0**	0.04	**0**

3.3 The results of the best and worst fitness solution

Another approach to this analysis will be to evaluate how each MH has behaved in finding the minimum (best) and maximum (worst) fitness for each benchmark function. Figure 1 displays the lowest possible fitness for each function across all MHs. In the same way, Figures 2 and 3 display the best fitness for the remaining 12 functions. According to the findings of Figs. 1, 2, and 3, BA algorithms find the best fitness in 9 out of 18 functions (F1, F2, F3, F4, F9, F10, F11, F13, and F18), WOA 6 times (F7, F8, F9, F11, F13, and F18), PSO 5 times (F12, F13, F14, F17, and F18), and GWO 4 times (F5, F6, F13, and F18). For F13, 14 MH determine optimal fitness; for F18, 15 MH achieve this; for F17, 9 MH; for F15, 4 MH; and for F9 and F11, just 3 MH.

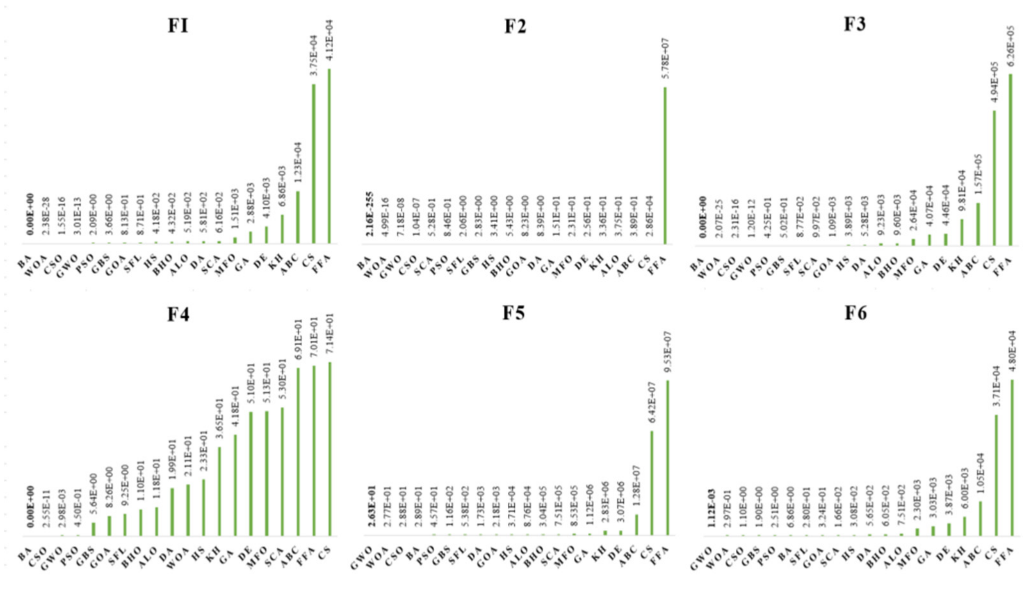

Figure 1. Best fitness for the F1-F6 test functions

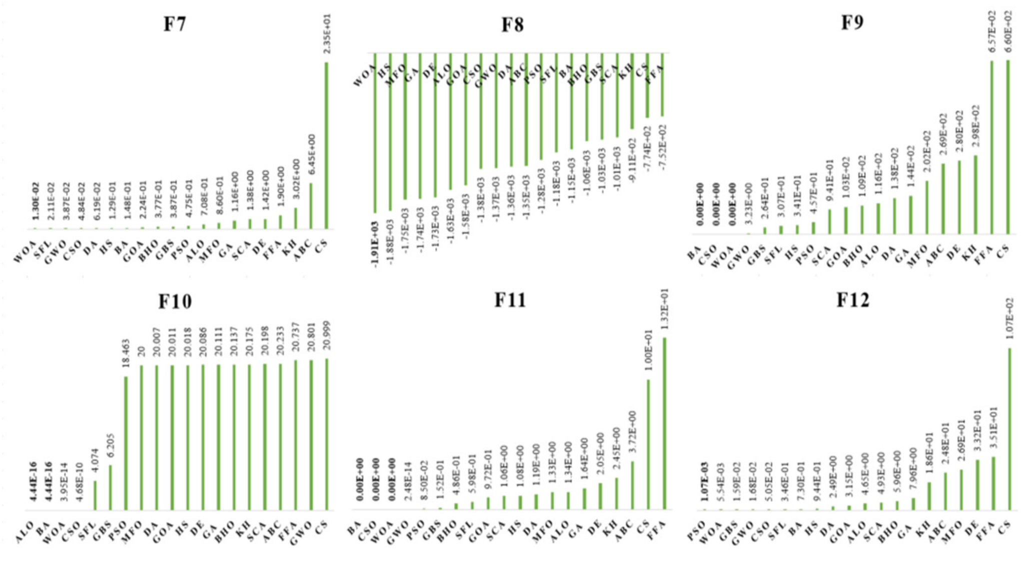

Figure 2. Best fitness for the F7-F12 test functions

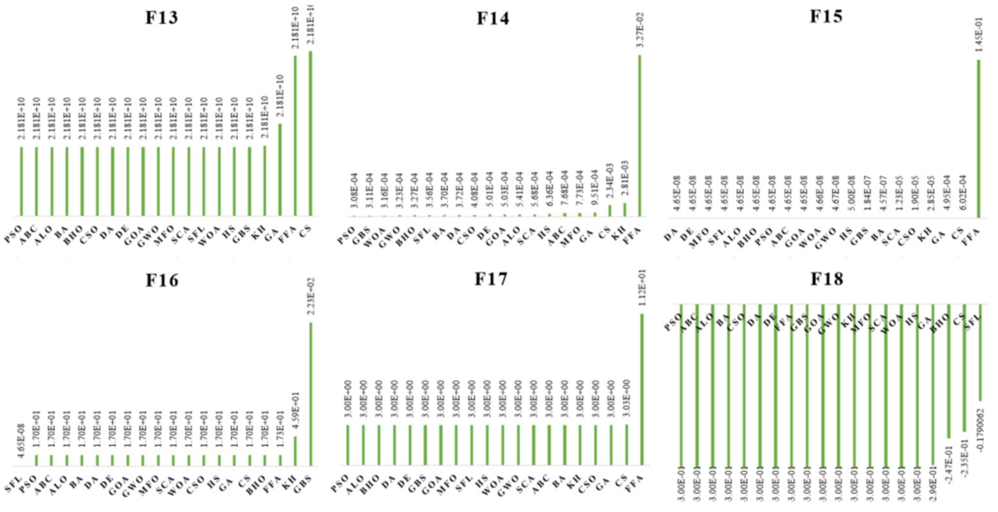

Figure 3. Best fitness for the F13-F18 test functions

Figures 4 through 6 illustrate the worst fitness offered by each benchmark function in connection with the 20 MH. The findings indicate that out of the 20 functions examined, the FFA metaheuristic approach yielded the highest level of fitness in 12 of them, which are F1, F2, F3, F5, F6, F9, F10, F11, F13, F14, F15, and F17. The next MH is KH, which yields the worst fitness for the functions F4, F8, and F16. CS yields the highest fitness result with respect to the F7 and F12 functions, while SFL shows the highest fitness for F18.

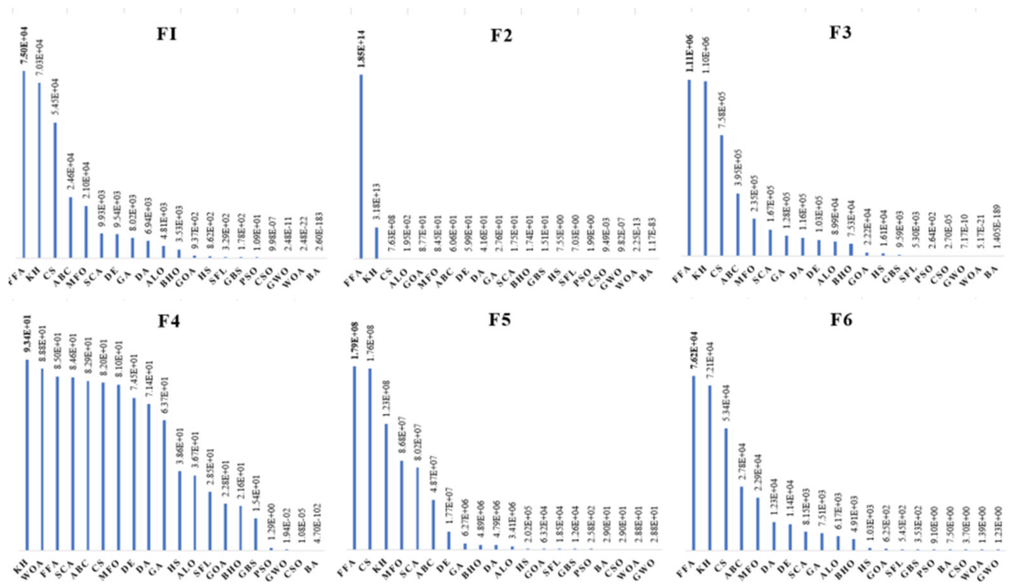

Figure 4. Worst fitness for the F1-F6 test functions

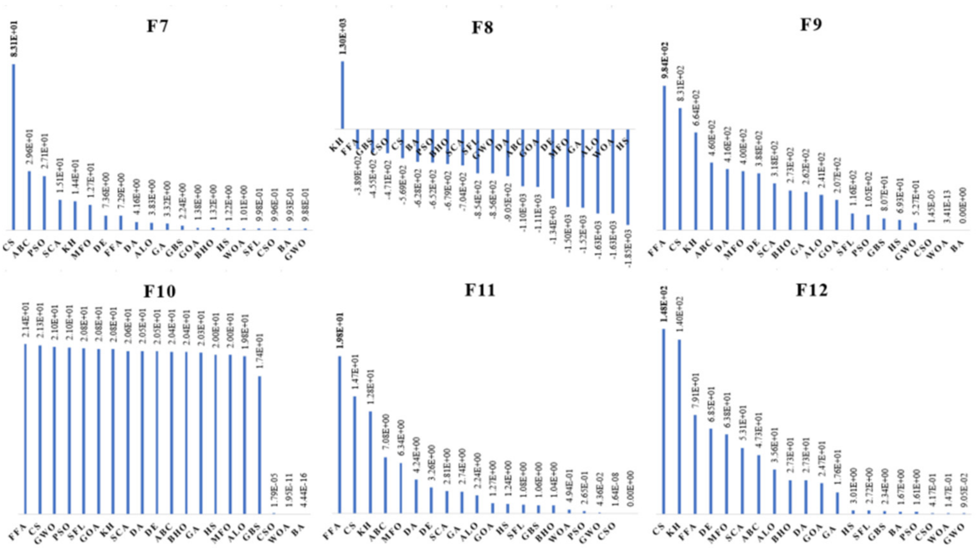

Figure 5. Worst fitness for the F7-F12 test functions

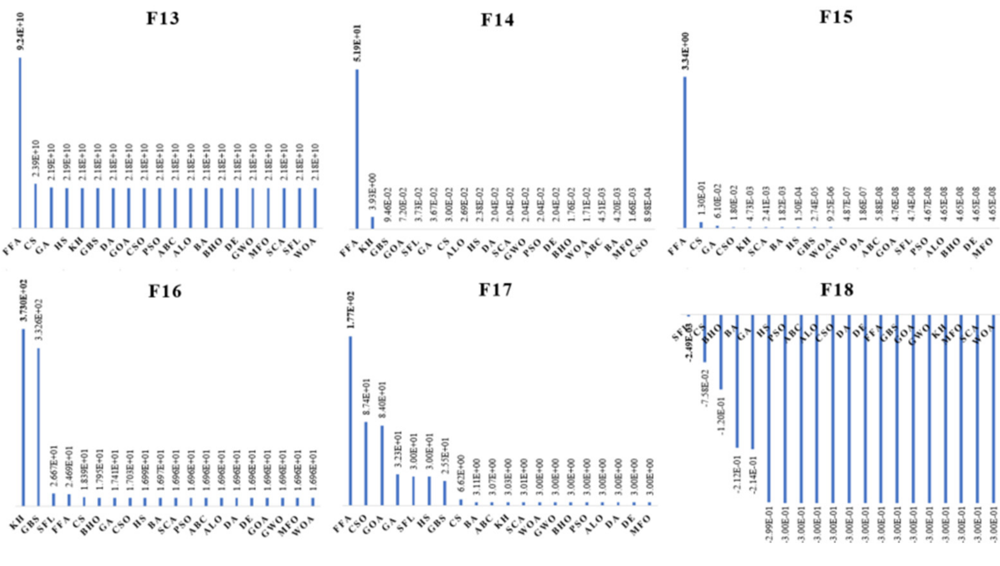

Figure 6. Worst fitness for the F13-F18 test functions

3.3.1 Execution time

Another significant metric refers to the amount of time of execution required by each MH. The experiments were conducted utilizing a personal computer equipped with an Intel(R) Core(TM) i5-3337U CPU 1.80 GHz and 8GB RAM, running on the Windows 10 operating system. The RStudio environment is utilized for all executions. The data presented in Figure 7 displays the average execution time of each MH algorithm, computed across all benchmark functions. The figure indicates that the CSO algorithm exhibits a comparatively longer execution time, averaging 26.132 minutes across the 18 test functions. Subsequently, GOA shows a duration of 23.481 minutes, followed by KH with a duration of 13.04

minutes. The graphical comparison of MHs reveals that CS demonstrates the quickest execution time of 0.428 minutes, followed by BHO with 0.551 minutes and WOA with 0.563 minutes. DE, GA, ABC, BA, and MFO illustrate execution times of 0.602, 0.652, 0.699, 0.771, and 0.839 minutes, respectively. On average, the 18 functions can be processed in less than one minute by all of the algorithms mentioned above.

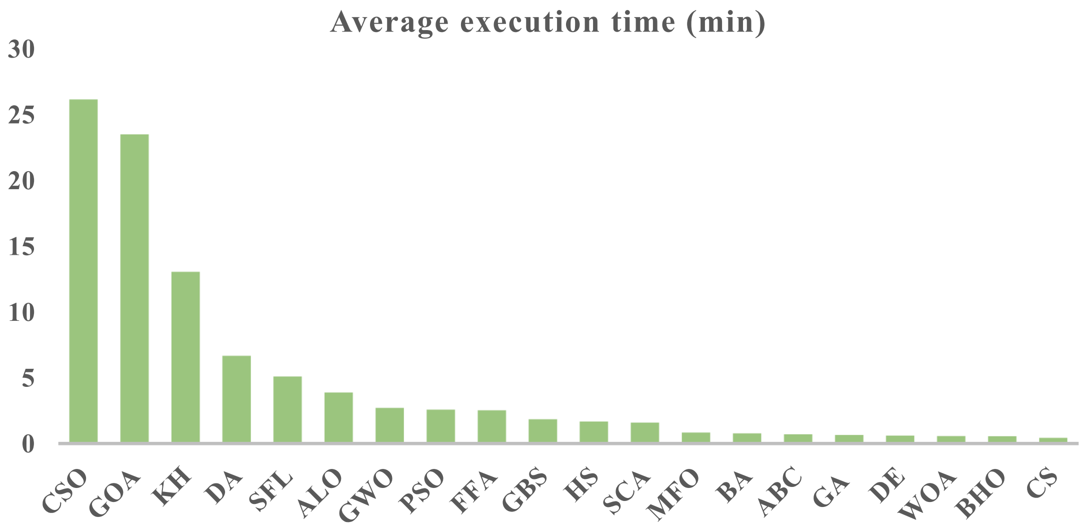

Figure 7. The execution time of each MH

4. Conclusions

This paper presents a comparative analysis of metaheuristic optimization algorithms and their capacity to attain the global optimum. In the present comparison, a total of 18 benchmark test functions have been employed, classified based on their respective characteristics as either unimodal, multimodal, or fixed-dimension multimodal functions. In the context of metaheuristics, a total of 20 algorithms have been utilized, with 14 of them being swarm-based (namely, CSO, GOA, KH, DA, SFL, ALO, GWO, PSO, FFA, MFO, BA, ABC, WOA, and CS), 2 being evolutionary-based (GA and DE), 3 appearing physical-based (GBS, HS, and BHO), and 1 remaining mathematical-based (SCA). The fitness value generated for each metaheuristic, representing the objective function's value from each benchmark function, is considered for evaluating the quality of the solutions. The evaluation of metaheuristics is conducted based on various metrics such as average, standard deviation, minimum, and maximum fitness. Furthermore, an analysis of the velocity of each metaheuristic was conducted by calculating the average execution time across the total of 18 functions.

The BA algorithm has demonstrated superior performance in optimizing 7 benchmark functions (4 unimodal, and 3 multimodal), as demonstrated by its evaluation of the average fitness, which has provided its effectivity in exploration, and exploitation for the selected benchmark functions. Next is GWO, which has yielded a high average fitness for two unimodal and one multimodal function, making it highly valuable on finding the global optimum. In contrast to the other two categories of benchmark functions, a significant number of metaheuristics generate the majority of fixed-dimension average fitness values, which distinguishes them from the aforementioned categories. Most metaheuristics are well-suited for discovering the optimal solution for the final category.

With respect to the worst fitness achieved for each benchmark function, the results indicate that the FFA metaheuristic optimization algorithm yields notably higher fitness values in twelve benchmark functions, followed by KH for three functions, CS for two functions, and SFL for one function. Regarding providing the best fitness, the optimization algorithm BA has demonstrated superior performance in approximately 50% of fitness evaluations, followed by WOA, PSO, and GWO.

In terms of computational efficiency, it has been observed that CSO, GOA, and KH algorithms require a relatively longer time for their execution, whereas CS, BHO, and WOA algorithms show faster computational performance.

In future studies, alternative benchmark functions may be employed to attain the global maximum of the MH algorithms, and to explore more carefully their ability in exploration and exploitation. Alternative values of MH parameters may have an impact on or enhance the overall optimal performance of the MH. Different dimensions and ranges of benchmark functions can also have an impact on the optimal solution. Given the ongoing advancements, alternatives, and recent MH will be considered to be applied in the future for the similar functions, and not only.

References

Ab Wahab, M. N., Nefti-Meziani, S., & Atyabi, A. (2015). A Comprehensive Review of Swarm Optimization Algorithms. *PLOS ONE, 10*(5). Retrieved from https://doi.org/10.1371/journal.pone.0122827

Abdel-Basset , M., Abdel-Fatah, L., & Sangaiah , A. K. (2018). Metaheuristic Algorithms: A Comprehensive Review. In A. K. Sangaiah, M. Sheng, & Z. Zhang, *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications* (pp. 185-231). Elsevier. Retrieved from https://doi.org/10.1016/B978-0-12-813314-9.00010-4

Bandaru, S., & Deb, K. (2017). Metaheuristic Techniques. In R. N. Sengupta, J. Dutta, & A. Gupta, *Decision Sciences - Theory and Practice* (pp. 693-750). CRC Press, Taylor & Francis Group, LLC.

Chu, S.-C., Tsai, P.-w., & Pan , J.-S. (2006). Cat Swarm Optimization. In Q. Yang, & G. Webb, *PRICAI 2006: Trends in Artificial Intelligence , Lecture Notes in Computer Science ((LNAI,volume 4099))* (pp. 854–858). Berlin, Heidelberg: Springer. Retrieved from https://doi.org/10.1007/978-3-540-36668-3_94

Demirhan, M., Özkaraca, O., & Güvenç, E. (2021). Performance Analysis of Particle Swarm Optimization and Firefly Algorithms with Benchmark Functions. In J. Hemanth, T. Yigit, B. Patrut, & A. Angelopoulou, *Trends in Data Engineering Methods for Intelligent Systems. Lecture Notes on Data Engineering and Communications Technologies ((LNDECT,volume 76))* (pp. 643–653). Springer, Cham. Retrieved from https://doi.org/10.1007/978-3-030-79357-9_60

Dhawan, S. (2022). Optimization Performance Analysis of Firefly Algorithm using Standard Benchmark Functions. *International Journal of Engineering Research & Technology (IJERT), 11*(2). doi:10.17577/IJERTV11IS020165

Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization, 38*(2), 129-154. Retrieved from https://doi.org/10.1080/03052150500384759

Gandomi , A. H., & Alavi , A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation, 17*, 4831-4845. Retrieved from https://doi.org/10.1016/j.cnsns.2012.05.010

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation, 76*(2), 60-68. Retrieved from https://doi.org/10.1177/003754970107600201

Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information Sciences, 222*, 175-184. Retrieved from https://doi.org/10.1016/j.ins.2012.08.023

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems - An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* MIT Press.

Hussain, K., Mohd Salleh, M. N., Cheng, S., & Naseem, R. (2017). Common Benchmark Functions for Metaheuristic Evaluation: A Review. *International Journal on Informatics Visualization, 1*(4-2), 218-223. Retrieved from http://dx.doi.org/10.30630/joiv.1.4-2.65

Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2017). Comparative Analysis of Swarm-Based Metaheuristic Algorithms on Benchmark Functions. In Y. Tan, H. Takagi, & Y. Shi, *Advances in Swarm Intelligence. Lecture Notes in Computer Science ((LNTCS,volume 10385))* (pp. 3-11). Springer, Cham. Retrieved from https://doi.org/10.1007/978-3-319-61824-1_1

Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150-194. Retrieved from https://doi.org/10.1504/IJMMNO.2013.055204

Josh, M., Gyanchandani, M., & Wadhvan, D. R. (2021). Analysis Of Genetic Algorithm, Particle Swarm Optimization and Simulated Annealing On Benchmark Functions. *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1152-1157). Erode, India: IEEE. doi:10.1109/ICCMC51019.2021.9418458

Karaboga, D., & Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation, 214*(1), 108-132. Retrieved from https://doi.org/10.1016/j.amc.2009.03.090

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks.* Perth, WA, Australia: IEEE. doi:10.1109/ICNN.1995.488968

Liang, J., Qu, B. Y., & Suganthan, P. N. (2013). *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization.* Technical Report 201311.

Liang, J., Qu, B. Y., Suganthan, P. N., & Hernández-Díaz, A. G. (2013). *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization.* Technical Report 201212.

Liang, J., Suganthan, P. N., Qu, B. Y., Gong, D. W., & Yue, C. T. (2020). *Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session on Multimodal Multiobjective Optimization.* Technical Report 201912. doi:10.13140/RG.2.2.31746.02247

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228-249. Retrieved from https://doi.org/10.1016/j.knosys.2015.07.006

Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software, 383*, 80-98. Retrieved from

https://doi.org/10.1016/j.advengsoft.2015.01.010
Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications, 27*, 1053–1073. Retrieved from https://doi.org/10.1007/s00521-015-1920-1
Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120-133. Retrieved from https://doi.org/10.1016/j.knosys.2015.12.022
Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software, 95*, 51-67. Retrieved from https://doi.org/10.1016/j.advengsoft.2016.01.008
Mirjalili, S., Mirjalili, S. M., & Andrew, L. (2014). Grey Wolf Optimizer. *Advances in Engineering Software, 69*, 46-61. Retrieved from https://doi.org/10.1016/j.advengsoft.2013.12.007
Mitchell, M. (1998). *An Introduction to Genetic Algorithms.* MIT Press.
Moghdani, R., & Salimifard, K. (2018). Volleyball Premier League Algorithm. *Applied Soft Computing, 64*, 161-185. Retrieved from https://doi.org/10.1016/j.asoc.2017.11.043
Mohamed, A. W., Sallam, K. M., Agrawal, P., Hadi, A. A., & Mohamed, A. K. (2023). Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems. *Neural Computing and Applications, 35*, 1493–1517. Retrieved from https://doi.org/10.1007/s00521-022-07788-z
Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review, 56*, 13187-13257. Retrieved from https://doi.org/10.1007/s10462-023-10470-y
Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences, 179*(13), 2232-2248. Retrieved from https://doi.org/10.1016/j.ins.2009.03.004
Riza, L. S., Iip, Nugroho, E. P., Adi Prabowo, M. B., Junaeti, E., & Abdullah, A. G. (2019, June 19). metaheuristicOpt: Metaheuristic for Optimization, R Package, v.2.0.0. Retrieved from https://cran.r-project.org/web/packages/metaheuristicOpt/metaheuristicOpt.pdf
Saremi, S., Mirjalili, S., & Lewis , A. (2017). Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software, 105*, 30-47. Retrieved from https://doi.org/10.1016/j.advengsoft.2017.01.004
Shehab, M., Abu-Hashem, M. A., Shambour, M. K., Alsalibi, A. I., Alomari, O. A., Gupta, J. N., . . . Abualigah, L. (2023). A Comprehensive Review of Bat Inspired Algorithm: Variants, Applications, and Hybridization. *Archives of Computational Methods in Engineering, 30*, 765-797. Retrieved from https://doi.org/10.1007/s11831-022-09817-5
Stork, J., Eiben, A. E., & Bartz-Beielstei, T. (2022). A new taxonomy of global optimization algorithms. *Natural Computing, 21*, 219-242. Retrieved from https://doi.org/10.1007/s11047-020-09820-4
Storn, R., & Price , K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization, 11*, 341-359. Retrieved from https://doi.org/10.1023/A:1008202821328
Suganthan, P. N., Hansen, N., Liang, J., Deb, K., Chen, Y.-p., Auger, A., & Tiwari, S. (2005). *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization.* Natural Computing.
Tuo , S., Zhang, J., Yong, L., Yuan, X., Liu, B., Xu, X., & Deng, F. (2015). A harmony search algorithm for high-dimensional multimodal optimization problems. *Digital Signal Processing, 46*, 151-163. Retrieved from https://doi.org/10.1016/j.dsp.2015.08.008
Wu, G., Mallipeddi, R., & Suganthan, P. N. (2016). *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization.* Technical Report.
Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. In O. Watanabe, & T. Zeugmann, *Stochastic Algorithms: Foundations and Applications. Lecture Notes in Computer Science ((LNTCS,volume 5792))* (pp. 169–178). Berlin, Heidelberg: Springer. Retrieved from https://doi.org/10.1007/978-3-642-04944-6_14
Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. In J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor, *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) - Studies in Computational Intelligence ((SCI,volume 284))* (pp. 65-74). Berlin Heidelberg: Springer. Retrieved from https://doi.org/10.1007/978-3-642-12538-6_6
Yang, X.-S., & Deb, S. (2009). Cuckoo Search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC).* Coimbatore, India: IEEE. doi:10.1109/NABIC.2009.5393690

APPENDIX

Table 8. Unimodal benchmark functions

Function	Dim	Range	fmin	Name
$f_1(x) = \sum_{1=1}^{n} x_i^2$	30	[-100, 100]	0	Sphere
$f_2(x) = \sum_{1=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0	Schwefel 2.22
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	[-100,100]	0	Schwefel 1.2
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100,100]	0	Schwefel 2.21
$f_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	[-30,30]	0	Rosenbrock 1
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100,100]	0	Step 2
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[-1.28,1.28]	0	Quartic

Table 9. Multimodal benchmark functions

Function	Dim	Range	f_{min}	Name
$f_8(x) = \sum_{1=1}^{n} -x_i \sin\sqrt{\lvert x_i \rvert}$	30	[-100, 100]	0	Schwefel 2.26
$f_9(x) = \sum_{1=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-10,10]	0	Rastrigin
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{1=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i-1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-100,100]	0	Ackley's path
$f_{11}(x) = \frac{1}{4000}\sum_{1=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-100,100]	0	Griewank
$f_{12}(x) = \frac{\pi}{n}\{\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4) \quad y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-30,30]	0	Generalized Penalized function no.1

Table 10. Fixed-dimension multimodal benchmark functions

Function	Dim	Range	f_{min}	Name
$f_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}^{-1}\right)$	2	[-65, 65]	1	Shekel's Foxholes
$f_{14}(x) = \sum_{1=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5,5]	0.00030	Kowalik
$f_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316	Six-Hump Camel
$f_{16}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[-5,5]	0.398	Brain
$f_{17}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] . [30 + (2x_1 - 3x_2)^2 . (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3	Goldstein Price
$f_{18}(x) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[1,3]	-3.86	Hartman 3