2022-1241 IJOI
https://www.ijoi-online.org/

SELECTING PROPER POPULATION-BASED METAHEURISTIC ALGORITHMS FOR SOLVING THE DIFFERENT MODALS OF UNCONSTRAINED OPTIMIZATION PROBLEMS

Shou-Cheng Hsiung*, Department of Industrial Management, I-Shou University, Taiwan
*Corresponding Author: melvinisstrong@gmail.com

I-Ming Chao, I-Shou University, Taiwan (ROC), iming@isu.edu.tw

Hsiang-Chen Hsu, I-Shou University, Taiwan (ROC), hchsu@isu.edu.tw

Abstract

Population-based metaheuristic (P-metaheuristic) algorithms are trendy for solving optimization problems. However, according to the NFL theorem, no metaheuristic is suited for solving all kinds of optimization problems. This study aims to determine the proper algorithms for the different modals of unconstrained optimization problems. We conducted a post-analysis of one-way analysis of variance (ANOVA) to test 14 P-metaheuristics, including evolutionary algorithms and swarm intelligence algorithms, on 23 unconstrained optimization benchmark functions. Experimental results show that the Harris hawks optimization (HHO) algorithm and gray wolf optimizer (GWO) are robust and more suitable for unimodal functions. In addition to the HHO being the best, the whale optimization algorithm (WOA) and GWO are also good choices for multi-modal functions. The cuckoo search (CS) algorithm dominated over fixed-dimension multi-modal functions. The study found that HHO, GWO, and WOA have similar mechanisms, such as searching (exploration), encircling, and attacking (exploitation) prey. The HHO and CS adopt the Lévy-flight-style random walk strategy to enhance the exploitation and exploration capabilities. Consequently, we acquired the proper P-metaheuristic to solve different modals and found superior mechanism to develop better P-metaheuristics in the future.

Key words: metaheuristic algorithms, population-based, swarm intelligent, exploration and exploitation capabilities, post-analysis, one-way ANOVA.

Introduction

Many real-world applications of unconstrained optimization aim to obtain the optimum (minimize or maximize) of D dimensional objective function (Tuba et al., 2011), as formulated below:

$$Min\ (or\ Max)\ f(x), x = (x_1, x_2, \cdots, x_D)$$

where D is the number of decision variables (dimensions) to be optimized.

Finding the unconstrained minimizer is a critical problem for many practical applications. Furthermore, techniques for solving unconstrained optimization problems form the foundation for most methods for solving constrained optimization problems. The size of real-world unconstrained optimization problems varies from small problems (e. g. D between 2 and 10) to huge problems (e. g. D in the hundreds or thousands). In many cases, the objective function f(x) is a computer routine that is expensive to evaluate, so even small problems tend to be time-consuming and difficult to solve (Dennis & Schnabel, 1989). That motivates us to utilize the unconstrained optimization problems to find the proper algorithm.

There are two main categories of techniques for solving optimization problems: The "Exact" methods are suitable for tackling smaller problems, while the "Approximate" methods can solve the no-differentiability and large-scale problems in a reasonable time (Ezugwu et al., 2021).

Approximate algorithms can be further divided into specific heuristics and metaheuristics. Specific heuristics are problem-dependent. Metaheuristics are high-level problem-independent techniques; in other words, metaheuristics are more general, faster, and suitable for solving large problems (Ezugwu et al., 2021; Talbi, 2009). Metaheuristics can find reasonable quality solutions to many complexes and NP-hard (non-deterministic polynomial time-hard) problems (Ezugwu et al., 2021). The characteristics of metaheuristics assist other scientists in quick learning, improving (Chao et al., 2020), and applying them to their problems (Pan et al., 2019).

Generally speaking, metaheuristic algorithms can be divided into two main categories (Talbi, 2009): Single-solution-based metaheuristics, such as simulated annealing (SA) (Kirkpatrick et al., 1983), and population-based metaheuristics, such as genetic algorithm (GA) (Goldberg, 1989).

As the name indicates, the Single-solution-based metaheuristics (S-metaheuristics) search process starts with one candidate solution and improves it through iterations. The population-based metaheuristics (P-metaheuristics) uses a set of agents (i.e., population) to perform the optimization process in each iteration. Compared to S-metaheuristics, P-metaheuristics have the following strengths: multiple agents sharing the information of the search space, which results in sudden jumps toward the promising area, assisting each other to avoid local optima, and having better exploration capability (Mirjalili et al., 2014). The no free lunch (NFL) theorem has proved that no metaheuristic

could be best suited for solving all kinds of optimization problems (Wolpert & Macready, 1997).

Taken together, the advantages of P-metaheuristics and the NFL theorem make us want to realize which P-metaheuristic algorithm is more suitable for solving a certain kind of problem. So, we try to conduct the one-way analysis of variance (ANOVA) to find better algorithms for the different modals of unconstrained optimization problems. One-way ANOVA could determine whether there are any statistically significant differences between the means of two or more independent (unrelated) groups. And we try to find common characteristics of superior algorithms to develop an advanced algorithm in the future.

The rest of this paper is organized as follows. Section 2 reviews the works of literature on metaheuristic algorithms. Section 3 presents the one-way ANOVA, 23 benchmark functions, and parameters of the 14 methods. Section 4 introduces the experimental results and analyzes them. Finally, in Section 5, conclusions are given.

Literature Review

All metaheuristic algorithms must be equipped with a randomization mechanism and a particular tradeoff between local search and global exploration (Yang, 2011). Therefore, they have higher local optima avoidance ability than conventional or classical optimization algorithms (Mirjalili, 2016). Metaheuristic algorithms with stochastic mechanisms regard optimization problems as black-boxes (Droste et al., 2006). In other words, the derivation of the mathematical models is not necessary because such optimization mechanisms only change the inputs and monitor the system's outputs to minimize (maximize) the objection function (outputs). Therefore metaheuristic algorithms can readily apply to different fields (Mirjalili, 2016).

As mentioned earlier, the general classification of the metaheuristics are S-metaheuristics and P-metaheuristics (Talbi, 2009). P-metaheuristics can be categorized into three main groups: evolution-based, physics-based, and swarm-based methods (Ezugwu et al., 2021; Heidari et al., 2019; Mirjalili & Lewis, 2016).

Evolution-based algorithms are also called evolutionary algorithms (EA), inspired by evolution behaviors in nature (Heidari et al., 2019). The most popular EA is GA (Goldberg, 1989), which evolves a set of initial random solutions. Each new population is a recombination and mutation of the individuals in the previous generation. Since the best individuals assessed by the objective function have a higher probability of being selected and generating a new population, the new population may be better than the previous generation(s). Another well-known EA is differential evolution (DE) (Storn & Price, 1997). Its main evolutionary behaviors are mutation, crossover, and selection. The mutation of DE is to generate a mutant vector by adding the weighted difference between two vectors to a third vector. These three random mutually different vectors are sampled from the previous population except the target vector. Jaya (a Sanskrit word meaning victory) is based on the

idea that the candidate solution should be close to the best solution and avoid the worst solution (Venkata Rao, 2016).

Physics-based algorithms are inspired by the physical laws or imitate the physical rules in the universe, such as multi-verse optimizer (MVO) (Mirjalili et al., 2015) and sine cosine algorithm (SCA) (Mirjalili, 2016). The main inspirations of MVO are based on three concepts in cosmological theory: white hole, black hole, and wormhole. These three concepts implement exploration, exploitation, and local search, respectively. MVO employs the wormhole existence probability (WEP), which increases linearly over the iterations, to emphasize exploitation as the evolution of the optimization process. On the contrary, the traveling distance rate (TDR), which decreases over the iterations to enhance the local search around the best-obtained universe (solution). According to the value of sine and cosine functions, the SCA explores or exploits the search space and constantly updates the position of candidate solutions by adopting the best solution obtained so far.

The swarm-based methods are also called swarm intelligence (SI) algorithms (Beni & Wang, 1993). The inspiration of SI techniques mostly mimic the social behaviors (e.g., distributed, self-organized systems, decentralized) of creatures in nature, such as flocks of birds, schools of fish, groups of wolves (Ezugwu et al., 2021). SI algorithms preserve search space information throughout the iteration process, and utilize memory to save the best solution obtained so far. These algorithms have fewer parameters, fewer operators compared, and are easy to implement (Mirjalili et al., 2014).

The most popular SI technique is particle swarm optimization (PSO) (Eberhart & Kennedy, 1995), which was inspired by the social behavior of birds flocks. The PSO algorithm employs multiple particles, and each particle in the swarm represents a candidate solution to the optimization problem. Each particle is updated according to the position of the global best particle and its own (local) best position.

Firefly algorithm (FFA) was inspired by the flashing characteristics of fireflies and their social behaviors of attracting each other by brightness (Yang, 2009). The brightness is proportional to the attractiveness and determined by the objective function. Then cuckoo search (CS) algorithm was inspired by the combination of special interesting brood parasitic behavior of cuckoos and the Lévy-flight-style random walk process, modeled a power-law step-length distribution (Yang & Deb, 2009). Lévy-Flights is a series of straight flight paths punctuated by a sudden 90◦turn and successfully applied to many optimization algorithms (Pavlyukevich, 2007). Lévy-Flights and random walks enhance the exploitation and exploration capabilities, respectively. This particular operator will make sure the optimization process will not be trapped in a local optimum (Yang & Deb, 2009). Bat algorithm (BAT) was inspired by the echolocation behavior of bats when they are navigating and hunting (Yang, 2010). BAT reduces the loudness and increases pulse

emission rate when they approach their prey.

Grey Wolf optimizer (GWO) was inspired by the leadership hierarchy of grey wolves (Canis lupus) and their hunting mechanism, divided into three leading operators (i.e. searching, encircling, and attacking prey) (Mirjalili et al., 2014). The social order assists GWO in saving the best three solutions obtained so far. The searching operator implements the exploration. Encircling and attacking mechanisms conducts the exploitation. Moth-flame optimization (MFO) algorithm was inspired by the navigation method of moths in nature (Mirjalili, 2015). MFO modeled the deadly spiral movement of each moth around a relative flame. The moth's position update can take place around different flames, and the unique mechanism makes the abrupt movement of moths in the search space and enhance the exploration. MFO gradually decreases in the number of flames to balance exploration and exploitation capabilities. Whale optimization algorithm (WOA) was inspired by humpback whales' unique bubble-net hunting behavior (Mirjalili & Lewis, 2016). The exploitation phase is implemented by the bubble-net attacking method, which can be modeled by the shrinking encircling mechanism or the spiral updating position. Salp swarm algorithm (SSA) was inspired by the efficient swarming behavior of the salp chain when navigating and foraging (Mirjalili et al., 2017). The population is divided into two groups: leader and followers. SSA assumed that the best solution obtained so far is the food source pursued by the leader of the salp chain. Each follower updated its position by the average of the previous salp and itself. SSA utilizes the adaptive gradually decrease parameter to balance the leader salp's exploration and exploitation capabilities. Harris hawks optimizer (HHO) was inspired by the cooperative and surprise pounce chasing style behavior of Harris hawks (Heidari et al., 2019). HHO provides four operators of searching strategy based on stochastic parameters to enhance the exploitation phase. HHO integrates the Lévy-Flights-based patterns with short-length jumps into the rapid dive mechanism to mimic the irregular, abrupt, and rapid dive behaviors of Harris hawks to enhance the exploitative capability.

Because of the excellent exploration capability, local optima avoidance, search space information sharing, and other advantages, the research community prefer to develop population-based metaheuristics than single solution-based recently. (Ezugwu et al., 2021; Mirjalili et al., 2014) Yang (2011) indicated that the searching process has two phases: exploration (diversification) and exploitation (intensification) regardless of the various classes of P-metaheuristics. In the exploration phase, metaheuristics should use and enhance its randomized operators to thoroughly explore diverse regions and borders of the search space. A well-designed metaheuristic should have a sufficiently rich random mechanism to efficiently allocate more randomized solutions to different regions of the feature space in the early steps of the optimization process. The exploitation phase is normally implemented after the exploration phase. In the exploitation phase, the metaheuristic tries to visit the local regions around the best solution

intensively instead of all-inclusive regions of the feature space (Heidari et al., 2019; Yang, 2011). Finding a reasonable, delicate balance between exploration and exploitation is the most critical to the overall efficiency and performance of any metaheuristic algorithm. Otherwise, the probability of getting stuck in local optima (LO) and premature convergence drawbacks increases (Heidari et al., 2019; Yang, 2011).

Although the NFL theorem has proved that no metaheuristic could be best suited for solving all kinds of optimization problems, the research community still attempted to find better algorithms for optimization, especially for complex NP-hard optimization problems (Wolpert & Macready, 1997; Yang, 2011). This work focuses on finding the proper, efficient P-metaheuristic for different kinds of unconstrained benchmark functions, not for all types of problems.

Research Method

One-way ANOVA

Analysis-of-Variance (ANOVA) is a statistical model used to analyze the differences among or between the means of their groups (Fisher, 1925). ANOVA is based on the law of total variance (Sum-of-Squares, SS), where the variances of observations in particular factors are partitioned into components attributable to different sources of variation. The principal purpose of ANOVA is to research the relationship between the continuous data type of the dependent variable and the categorical data type of independent variables. ANOVA can provide the statistical t-test, and F-test applies two or more categories of means of the independent variables respectively to test whether they are equal. If there are significant differences, we perform post-analysis or multi-comparison for selecting proper metaheuristic algorithms for the different modals of unconstrained optimization problems.

In this paper, we adopt one-way ANOVA to compare the effects of the dependent variable caused by different treatments of the particular independent variable. We want to know whether different P-metaheuristic algorithms have a significant impact on solving benchmark functions to help us find the proper P-metaheuristic for different types of unconstrained problems.

Benchmark Functions

To find proper and efficient P-metaheuristics for different kinds of unconstrained problems, we utilize typical 23 benchmark functions used in the recent pieces of literature (Heidari et al., 2019; Mirjalili, 2015; Mirjalili & Lewis, 2016; Mirjalili et al., 2014; Yao & Liu, 1997; Yao et al., 1999).

F1- F7 are unimodal functions with only one global optimum but no local optima (Mirjalili, 2016). F6 is a discontinuous step function. F7 is a noisy quartic function (Yao et al., 1999). F8- F13 are multi-modal functions with multiple local optima and increase exponentially with the function's dimension (Mirjalili, 2016). F1- F13 are high-dimensional problems. They also could be scalable on dimension to expand the desired number of the design variables. F14 - F23 are fixed-dimensional functions

with only a few local minima. They cannot adjust the number of design variables but provide different types of search space compared to multi-modal (F8-F13) functions (Mirjalili & Lewis, 2016). Over all, the unimodal functions (F1-F7) with unique global optima and no other local optima can test the exploitation (intensification) capability of optimization algorithms. On the contrary, multi-modal functions (F8-F23) can reveal the exploration (diversification) and the ability to avoid the stagnation of poor local optima (Heidari et al., 2019). Figure 1. shows the typical 2D plots of the benchmark functions considered in this paper.

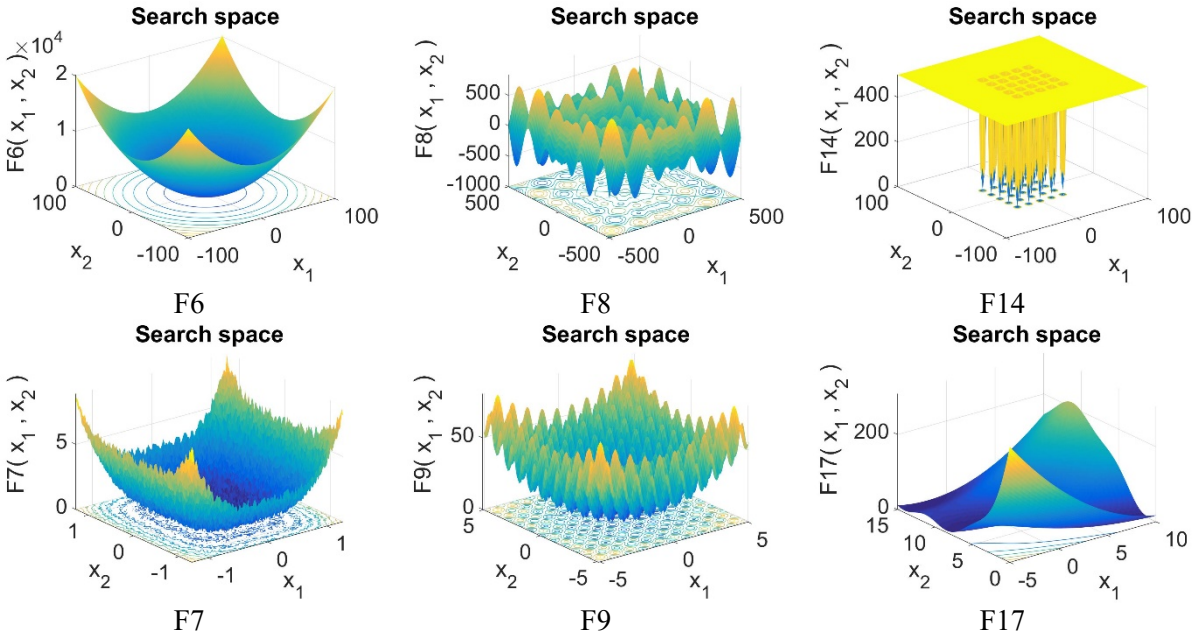

Figure 1. Typical 2D plots of benchmark functions

Compared Algorithms and Parameter Setup

To find the proper algorithm, we compared with 14 well-known and state-of-the-art P-metaheuristic algorithms such as PSO, MVO, GWO, MFO, CS, BAT, WOA, FFA, SSA, GA, HHO, SCA, JAYA, and DE.

All algorithms were implemented under PyCharm 2018.3.7 (Community Edition) on a laptop with a Windows 10 64-bit Home edition, i5-8250U CPU, and 12 GB RAM. The population size and maximum iterations of all optimizers are set to 30 and 500, respectively. Other parameters of those Population-based metaheuristic algorithms we choose in this study are shown in Table 1.

Experimental Results and Analysis

Evaluation of Exploitation Capability (Functions F1–F7)

For each benchmark function, those chosen P-metaheuristic algorithms run 30 times starting from different populations randomly generated. The average (AVG) and the corresponding standard deviation (STD) of statistical results are shown in Table 2.

In terms of AVG from Table 2, the HHO and WOA algorithms are more competitive. The HHO algorithm obtained the best average (AVG) on F3, F4, F5, and F7 (4 out of 7), and the WOA algorithm acquired the best average on F1 and F2 (2 out of 7). The post-analysis of one-way ANOVA in Table 3 shows that the HHO and GWO always belong to the first subset, but the WOA algorithm obtained a worse subset on F3 and F4. In other words, HHO and GWO algorithms are robust and more suitable for unimodal functions. It's worth mentioning that there is no significant difference on F2 for all selected algorithms, and the BAT algorithm got the worst subset for all unimodal functions.

Evaluation of Exploration Capability (Functions F8–F23)

F8- F13 are multi-modal functions with multiple local optima. In terms of AVG from Table 4, the HHO algorithm is the best. The post-analysis of one-way ANOVA from Table 5 shows that only the HHO stays in the first subset for each multi-modal function (F8- F13). We can then determine that the HHO is the most suitable for multi-modal functions. On the other hand, besides HHO, the WOA and GWO are also good choices for multi-modal functions.

F14 - F23 are fixed-dimensional functions with only a few local minima. In terms of AVG from Table 6, the CS algorithm got 5 in 10 of the best average (AVG) on F14 and F19 through F22. The DE algorithm got 3 in 10 on F18, F19, and F23. The post-analysis of one-way ANOVA from Table 7 shows that only the CS stays in the first subset for each multi-modal function (F14- F23). We can determine that the CS is the most suitable meta-heuristic algorithm for fixed-dimensional functions. The DE is competitive except on F21 and F22. In addition to CS, the DE is also a good option for fixed-dimensional functions.

Discussion of Results

According to previous sections, we can recognize that the HHO and GWO algorithms are robust and more suitable for unimodal functions. In addition to the HHO, the WOA and GWO are also good choices for multi-modal functions. Except for the CS, the DE is also a good option for fixed-dimensional multi-modal functions.

Therefore, the HHO, GWO, WOA, and CS population-based meta-heuristic algorithms are worth investigating. The HHO, GWO, and WOA have similar mechanisms, such as searching (exploration), encircling, and attacking (exploitation) prey. The HHO and CS utilize the Lévy-flight-style random walk strategy to enhance the exploitation and

Table 1. The parameters setting of algorithms

No.	Algorithm	Parameter	Value
1	Particle swarm optimization, PSO (Eberhart & Kennedy, 1995)	Inertia weight minimum, W_{min} Inertia weight maximum, W_{max} Local best parameter, C1 Global best parameter, C2	0.2 0.9 2 2
2	Multi-verse optimizer, MVO (Mirjalili et al., 2015)	Wormhole existence probability minimum, WEP_{min} Wormhole existence probability maximum, WEP_{max} Exploitation accuracy, p	0.2 1 6
3	Grey wolf optimizer, GWO (Mirjalili et al., 2014)	Convergence constant, a	[2, 0]
4	Moth-flame optimization, MFO (Mirjalili, 2015)	Convergence constant, r Spiral factor, b	[-1, -2] 1
5	Cuckoo search, CS (Yang & Deb, 2009)	Discovery rate of alien eggs/ solutions, P_a	0.25
6	Bat algorithm, BAT(Yang, 2010)	Loudness, A Pulse rate, r Frequency minimum, Q_{min} Frequency maximum, Q_{max}	0.5 0.5 0 2
7	Whale optimization algorithm, WOA (Mirjalili & Lewis, 2016)	Convergence constant, a Convergence constant, a2 Spiral factor, b	[2,0] [-1,-2] 1
8	Firefly algorithms, FFA (Yang, 2009)	Randomization parameter, α Attractiveness parameter, β_{min} Absorption coefficient, γ	0.5 0.20 1
9	Salp swarm algorithm, SSA (Mirjalili et al., 2017)	Convergence constant, c_1	$[2,2e^{-1}]$
10	Genetic algorithm, GA (Goldberg, 1989)	Crossover probability Mutation Probability Elitism parameter	1 0.1 2
11	Harris hawks optimization, HHO (Heidari et al., 2019)	Factor of escaping energy, E1 Levy flight constant, β	[2,0] 1.5
12	Sine cosine algorithm, SCA (Mirjalili, 2016)	Convergence constant, a	[2,0]
13	JAYA(Venkata Rao, 2016)	No algorithm-specific parameter	-
14	Differential evolution, DE (Storn & Price, 1997)	Mutation factor, F Crossover ratio, CR	0.5 0.7

Table 2. Results of unimodal benchmark functions

Benchmark		PSO	MVO	GWO	MFO	CS	BAT	WOA	FFA	SSA	GA	HHO	SCA	JAYA	DE
F1	AVG	1.43E-04	1.18E+00	2.50E-27	1.00E+03	1.24E+01	1.90E+04	2.26E-72	6.65E-03	2.17E-08	3.04E+03	6.74E-62	4.34E+01	9.61E-05	1.61E-04
	STD	1.05E-04	3.27E-01	4.23E-27	3.00E+03	4.86E+00	7.17E+03	1.21E-71	2.30E-03	5.09E-09	7.66E+02	3.63E-61	8.49E+01	2.28E-04	9.85E-05
F2	AVG	5.03E+00	8.92E-01	8.19E-17	4.23E+01	6.16E+00	2.15E+03	2.04E-50	6.83E-01	9.32E-01	1.59E+01	6.51E-35	1.98E-02	2.76E-04	4.73E-03
	STD	6.20E+00	6.71E-01	5.61E-17	1.99E+01	1.57E+00	1.03E+04	7.09E-50	8.94E-01	9.42E-01	2.61E+00	2.73E-34	2.70E-02	2.26E-04	2.09E-03
F3	AVG	7.71E+01	2.30E+02	8.40E-06	1.86E+04	2.16E+03	5.20E+04	4.31E+04	9.42E+02	5.99E+02	1.51E+04	3.52E-48	8.40E+03	2.20E+04	2.00E+04
	STD	3.18E+01	1.11E+02	1.68E-05	1.09E+04	5.38E+02	2.44E+04	1.23E+04	5.42E+02	5.31E+02	3.05E+03	1.90E-47	4.73E+03	1.00E+04	3.73E+03
F4	AVG	1.12E+00	2.19E+00	6.57E-07	5.82E+01	1.12E+01	5.77E+01	6.02E+01	4.47E-01	7.25E+00	4.94E+01	4.96E-33	3.53E+01	1.90E+01	1.19E+01
	STD	2.26E-01	9.48E-01	4.36E-07	1.03E+01	2.20E+00	1.04E+01	2.25E+01	3.04E-01	2.62E+00	5.81E+00	2.02E-32	1.20E+01	6.66E+00	4.73E+00
F5	AVG	2.29E+02	5.29E+02	2.69E+01	1.26E+04	6.88E+02	1.86E+07	2.81E+01	1.91E+02	9.95E+01	1.82E+06	1.10E-02	7.65E+04	3.12E+02	4.00E+01
	STD	5.48E+02	6.68E+02	6.45E-01	3.04E+04	3.31E+02	1.37E+07	5.45E-01	3.96E+02	1.52E+02	1.48E+06	1.52E-02	2.37E+05	7.57E+02	2.96E+01
F6	AVG	1.83E-04	1.12E+00	7.01E-01	1.33E+03	1.26E+01	2.15E+04	4.26E-01	6.09E-03	2.34E-08	3.29E+03	1.27E-04	1.43E+01	4.04E+00	1.77E-04
	STD	3.66E-04	3.08E-01	4.05E-01	3.38E+03	4.41E+00	6.64E+03	2.25E-01	2.24E-03	7.92E-09	1.04E+03	1.38E-04	1.39E+01	6.96E-01	1.11E-04
F7	AVG	2.81E+00	3.87E-02	2.28E-03	3.89E+00	1.01E-01	9.22E+00	3.88E-03	3.80E-01	9.41E-02	1.36E+00	1.82E-04	1.19E-01	6.45E-02	4.27E-02
	STD	4.44E+00	1.39E-02	1.09E-03	6.17E+00	3.51E-02	4.13E+00	3.17E-03	1.42E-01	3.45E-02	6.58E-01	2.38E-04	1.03E-01	6.63E-02	1.31E-02

Table 3. Homogeneous subsets of different algorithms by post-analysis of one-way ANOVA on unimodal functions

Benchmark	PSO	MVO	GWO	MFO	CS	BAT	WOA	FFA	SSA	GA	HHO	SCA	JAYA	DE
F1	1	1	1	2	1	3	1	1	1	2	1	1	1	1
F2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F3	1	1	1	3	1	5	4	1	1	3	1	2	3	3
F4	2	2	1	8	3	7	6	2	3	7	1	5	4	3
F5	1	1	1	1	1	2	1	1	1	1	1	1	1	1
F6	1	1	1	2	1	3	1	1	1	2	1	1	1	1
F7	3	1	1	3	1	4	1	1	1	2	1	1	1	1

Table 4. Results of multi-modal benchmark functions

Benchmark		PSO	MVO	GWO	MFO	CS	BAT	WOA	FFA	SSA	GA	HHO	SCA	JAYA	DE
F8	AVG	-5.01E+03	-7.74E+03	-6.12E+03	-8.38E+03	-1.00E+04	-3.33E+03	-1.06E+04	-7.41E+03	-7.37E+03	-1.04E+04	-1.25E+04	-3.74E+03	-5.09E+03	-6.09E+03
	STD	1.20E+03	7.04E+02	6.61E+02	7.27E+02	1.47E+03	1.18E+03	1.58E+03	7.18E+02	6.18E+02	2.95E+02	2.58E+02	2.31E+02	6.90E+02	4.72E+02
F9	AVG	1.13E+02	1.20E+02	1.39E+00	1.77E+02	1.01E+02	9.55E+01	1.89E-15	8.04E+01	4.65E+01	8.14E+01	0.00E+00	4.86E+01	1.08E+02	1.86E+02
	STD	3.09E+01	2.40E+01	2.35E+00	4.14E+01	1.30E+01	4.66E+01	1.02E-14	1.80E+01	1.68E+01	1.25E+01	0.00E+00	4.44E+01	4.45E+01	9.61E+00
F10	AVG	1.04E-01	1.92E+00	1.01E-13	1.53E+01	4.38E+00	1.63E+01	4.35E-15	3.79E-01	1.89E+00	1.11E+01	4.44E-16	1.33E+01	2.65E-02	6.21E-01
	STD	3.31E-01	5.42E-01	1.73E-14	7.72E+00	4.88E-01	1.10E+00	2.31E-15	5.04E-01	8.19E-01	7.79E-01	0.00E+00	8.79E+00	1.13E-01	3.31E+00
F11	AVG	8.64E-03	8.55E-01	5.51E-03	9.20E+00	1.11E+00	1.83E+02	8.53E-03	2.01E-02	8.76E-03	2.62E+01	0.00E+00	9.21E-01	1.21E-01	5.86E-03
	STD	7.93E-03	6.97E-02	8.28E-03	2.71E+01	3.90E-02	8.11E+01	4.59E-02	5.69E-03	9.90E-03	6.20E+00	0.00E+00	3.93E-01	1.62E-01	1.46E-02
F12	AVG	1.25E-01	2.41E+00	1.49E-01	9.74E+00	3.87E+00	2.52E+07	7.56E-02	5.89E-01	5.20E+00	2.64E+05	7.67E-06	3.41E+04	1.16E+01	3.66E-01
	STD	1.43E-01	1.22E+00	9.62E-02	4.68E+00	1.03E+00	2.27E+07	8.68E-02	4.49E-01	2.51E+00	1.02E+06	1.29E-05	1.16E+05	5.47E+01	7.06E-01
F13	AVG	2.64E-03	1.69E-01	6.61E-01	4.49E+00	8.61E+00	7.73E+07	5.77E-01	5.04E-03	2.17E+00	1.02E+06	8.83E-05	2.92E+04	2.02E+04	1.23E-03
	STD	4.62E-03	1.20E-01	2.72E-01	1.02E+01	3.51E+00	5.42E+07	3.64E-01	6.90E-03	6.19E+00	1.02E+06	1.08E-04	6.08E+04	9.40E+04	1.47E-03

Table 5. Homogeneous subsets of different algorithms by post-analysis of one-way ANOVA on multi-modal functions

Benchmark	PSO	MVO	GWO	MFO	CS	BAT	WOA	FFA	SSA	GA	HHO	SCA	JAYA	DE
F8	8	5	6	4	3	9	3	5	5	2	1	9	8	7
F9	5	5	1	6	5	5	1	3	2	4	1	2	5	7
F10	1	2	1	4	2	4	1	1	2	3	1	3	1	1
F11	1	1	1	2	1	3	1	1	1	2	1	1	1	1
F12	1	1	1	1	1	2	1	1	1	1	1	1	1	1
F13	1	1	1	1	1	2	1	1	1	1	1	1	1	1

Table 6. Results of fixed-dimension multi-modal benchmark functions

Benchmark		PSO	MVO	GWO	MFO	CS	BAT	WOA	FFA	SSA	GA	HHO	SCA	JAYA	DE
F14	AVG	2.61E+00	9.98E-01	4.27E+00	1.53E+00	9.98E-01	1.05E+01	2.41E+00	1.20E+00	1.03E+00	9.98E-01	1.43E+00	1.99E+00	1.11E+00	1.23E+00
	STD	2.52E+00	4.60E-11	3.56E+00	1.29E+00	1.04E-15	5.54E+00	2.60E+00	4.73E-01	1.78E-01	9.17E-05	9.43E-01	1.87E+00	5.32E-01	9.43E-01
F15	AVG	3.02E-03	6.03E-03	4.39E-03	1.68E-03	5.67E-04	9.51E-03	5.86E-04	7.57E-04	8.54E-04	1.23E-03	3.12E-04	1.02E-03	1.33E-03	2.06E-03
	STD	5.79E-03	8.53E-03	7.99E-03	3.49E-03	1.72E-04	9.78E-03	3.30E-04	2.49E-04	2.85E-04	4.18E-04	3.76E-06	3.55E-04	6.61E-04	5.10E-03
F16	AVG	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.00E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	STD	4.15E-16	2.62E-07	1.58E-08	4.44E-16	2.35E-13	1.47E-01	3.31E-10	2.21E-09	1.94E-14	2.15E-04	2.83E-09	5.81E-05	9.84E-05	4.44E-16
F17	AVG	3.98E-01	4.75E-01	3.98E-01	3.98E-01	3.98E-01	5.52E-01	3.98E-01	3.98E-01	3.98E-01	3.99E-01	3.98E-01	4.00E-01	3.98E-01	3.98E-01
	STD	0.00E+00	4.14E-01	3.32E-06	0.00E+00	1.95E-09	5.76E-01	1.52E-05	6.07E-09	4.18E-14	8.90E-04	1.34E-06	2.15E-03	1.58E-03	0.00E+00
F18	AVG	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	7.50E+00	3.00E+00	3.00E+00	3.00E+00	3.01E+00	5.70E+00	3.00E+00	3.21E+00	3.00E+00
	STD	1.64E-15	2.50E-06	4.42E-05	1.97E-15	2.16E-15	1.57E+01	1.35E-04	4.42E-08	2.27E-13	1.32E-02	8.10E+00	8.78E-05	4.97E-01	1.30E-15
F19	AVG	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.84E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.85E+00	-3.61E+00	-3.86E+00
	STD	2.94E-03	2.79E-06	2.75E-03	8.88E-16	6.73E-16	1.39E-01	7.63E-03	9.80E-10	3.42E-13	2.36E-04	2.00E-05	2.72E-03	1.99E-01	8.88E-16
F20	AVG	-3.19E+00	-3.25E+00	-3.26E+00	-3.23E+00	-3.32E+00	-3.27E+00	-3.19E+00	-3.28E+00	-3.21E+00	-3.30E+00	-3.27E+00	-2.91E+00	-1.82E+00	-3.24E+00
	STD	2.98E-01	6.03E-02	7.27E-02	6.21E-02	2.70E-06	5.93E-02	1.75E-01	5.90E-02	3.81E-02	4.44E-02	6.04E-02	2.80E-01	4.56E-01	5.71E-02
F21	AVG	-8.76E+00	-6.75E+00	-8.34E+00	-6.75E+00	-1.01E+01	-5.44E+00	-8.91E+00	-7.69E+00	-8.44E+00	-6.62E+00	-5.55E+00	-1.77E+00	-1.46E+00	-9.36E+00
	STD	2.22E+00	2.37E+00	2.53E+00	2.85E+00	9.64E-07	3.00E+00	2.13E+00	3.25E+00	2.84E+00	3.07E+00	1.52E+00	1.61E+00	9.53E-01	2.24E+00
F22	AVG	-6.82E+00	-7.13E+00	-9.36E+00	-5.54E+00	-1.02E+01	-4.77E+00	-7.30E+00	-7.60E+00	-6.84E+00	-5.87E+00	-5.06E+00	-3.02E+00	-1.55E+00	-8.76E+00
	STD	3.30E+00	3.17E+00	2.09E+00	3.40E+00	8.38E-06	3.33E+00	2.87E+00	3.43E+00	3.43E+00	3.49E+00	6.92E-05	1.93E+00	8.67E-01	2.85E+00
F23	AVG	-8.67E+00	-8.82E+00	-1.05E+01	-8.90E+00	-1.05E+01	-4.33E+00	-7.63E+00	-9.87E+00	-9.18E+00	-8.37E+00	-5.47E+00	-3.18E+00	-1.88E+00	-1.03E+01
	STD	3.07E+00	2.82E+00	1.01E-03	2.89E+00	1.72E-05	2.90E+00	3.05E+00	1.87E+00	2.66E+00	2.85E+00	1.34E+00	1.60E+00	8.13E-01	1.20E+00

Table 7. Homogeneous subsets of different algorithms by post-analysis of one-way ANOVA on multi-modal functions

Benchmark	PSO	MVO	GWO	MFO	CS	BAT	WOA	FFA	SSA	GA	HHO	SCA	JAYA	DE
F14	3	1	3	1	1	4	3	1	1	1	1	2	1	1
F15	3	2	3	1	1	4	1	2	1	1	1	1	1	1
F16	1	1	1	1	1	2	1	1	1	1	1	1	2	1
F17	1	1	1	1	1	2	1	1	1	1	1	1	1	1
F18	1	1	1	1	1	2	1	1	1	1	1	1	1	1
F19	1	1	1	1	1	2	1	1	1	1	1	1	3	1
F20	1	1	1	1	1	1	1	1	1	1	1	2	3	1
F21	3	3	2	2	1	4	2	2	2	4	4	5	5	2
F22	5	4	2	6	1	7	4	3	3	6	6	8	8	3
F23	3	3	2	3	1	5	3	2	3	3	4	6	6	1

exploration capabilities (Heidari et al., 2019; Yang & Deb, 2009). As mentioned above, Lévy-Flights is a series of straight flight paths punctuated by a sudden 90◦ turn and successfully applied to many optimization algorithms (Pavlyukevich, 2007; Shlesinger, 2006). This particular operator will ensure the optimization process will not be trapped in a local optimum (Yang & Deb, 2009). The special interesting brood parasitic behavior of the CS algorithm enhances the exploration capability (Yang & Deb, 2009). It might explain why the CS obtained better experimental performance than HHO and other algorithms on fixed-dimensional multi-modal functions. Some researchers employ the chaotic sequences to specify the fraction probability on the CS to improve the exploration capability (Pan et al., 2019).

Conclusion and Future Directions

We selected 14 algorithms from state-of-the-art Population-based meta-heuristic algorithms for solving 23 unconstrained optimization benchmark functions in this work. According to the NFL theorem, no meta-heuristic is suited for solving all kinds of optimization problems. Therefore, we executed the post-analysis of the one-way ANOVA on the experimental results. The analysis findings show that the HHO and GWO algorithms are robust and more suitable for unimodal functions. In addition to the HHO being the best, the WOA and GWO are also good choices for multi-modal functions. The CS algorithm performs more competitively on fixed-dimensional multi-modal functions than other algorithms. If we don't know which function the problem belongs to, the HHO and CS algorithms are excellent initial ways to deal with them.

We also identify the common characteristics of HHO, GWO, WOA, and CS, which are superior algorithms on different modals functions. The HHO, GWO, and WOA have similar mechanisms, such as searching (exploration), encircling, and attacking (exploitation) prey. The HHO and CS adopt the Lévy-flight-style random walk strategy to enhance the exploitation and exploration capabilities. These potential and competitive conclusions could help us to achieve efficient, advanced, and stable equilibria among the exploratory and exploitative propensity algorithms to escape the local optima stagnation and find the most satisfactory solution in the future.

References

Beni, G., & Wang, J. (1993). Swarm Intelligence in Cellular Robotic Systems. In *Robots and Biological Systems: Towards A New Bionics?* (pp. 703-712). Springer.

Chao, I.-M., Hsiung, S.-C., & Liu, J.-L. (2020). Improved Whale Optimization Algorithm Based on Inertia Weights for Solving Global Optimization Problems. *Advances in Technology Innovation, 5*(3), 147-155.

Dennis, J. E., & Schnabel, R. B. (1989). A View of Unconstrained Otimization. In *Optimization* (pp. 1–72). Elsevier.

Droste, S., Jansen, T., & Wegener, I. (2006). Upper and Lower Bounds for Randomized Search Heuristics in Black-Box Optimization. *Theory of computing systems, 39*(4), 525-544.

Eberhart, R., & Kennedy, J. (1995, 27 November-1 December). Particle Swarm Optimization. *Proceedings of The IEEE International Conference on Neural Networks*, Perth, Western Australia.

Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., & Muhuri, P. K. (2021). Metaheuristics: A Comprehensive Overview and Classification along with Bibliometric Analysis. *Artificial Intelligence Review, 54*(6), 4237-4316. https://doi.org/10.1007/s10462-020-09952-0

Fisher, R. A. (1925). Statistical Methods for Research Workers. In. Springer.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems, 97*, 849-872. https://doi.org/10.1016/j.future.2019.02.028

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *science, 220*(4598), 671-680.

Mirjalili, S. (2015). Moth-Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm. *Knowledge-Based Systems, 89*, 228-249. https://doi.org/10.1016/j.knosys.2015.07.006

Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowledge-Based Systems, 96*, 120-133. https://doi.org/10.1016/j.knosys.2015.12.022

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Advances in Engineering Software, 114*, 163-191. https://doi.org/10.1016/j.advengsoft.2017.07.002

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software, 95*, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2015). Multi-Verse Optimizer: A Nature-Inspired Algorithm for Global Optimization. *Neural Computing and Applications, 27*(2), 495-513. https://doi.org/10.1007/s00521-015-1870-7

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software, 69*, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Pan, J. S., Liu, J. L., & Hsiung, S. C. (2019, February 22 - 24, 2019). Chaotic Cuckoo Search Algorithm for Solving Unmanned Combat Aerial Vehicle Path Planning Problems. *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, Zhuhai, China.

Pavlyukevich, I. (2007). Lévy Flights, Non-Local Search and Simulated Annealing. *Journal of Computational Physics, 226*(2), 1830-1844.

Shlesinger, M. F. (2006). Search Research. *Nature, 443*(7109), 281-282.

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.

Talbi, E. G. (2009). *Metaheuristics: from Design to Implementation* (Vol. 74). John Wiley & Sons, Inc.

Tuba, M., Subotic, M., & Stanarevic, N. (2011, April 28-30, 2011). Modified Cuckoo Search Algorithm for Unconstrained Optimization Problems. *Proceedings of fhe European Computing Conference*, Paris, France.

Venkata Rao, R. (2016). Jaya: A simple and New Optimization Algorithm for Solving Constrained and Unconstrained Optimization Problems. *International Journal of Industrial Engineering Computations, 7*(1), 19-34. https://doi.org/10.5267/j.ijiec.2015.8.004

Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82. https://doi.org/10.1109/4235.585893

Yang, X. S. (2009). Firefly Algorithms for Multimodal Optimization. In W. O. & Z. T. (Eds.), *Stochastic Algorithms: Foundations and Applications. SAGA 2009. Lecture Notes in Computer Science* (Vol. 5792, pp. 169-178). Springer, Berlin, Heidelberg.

Yang, X. S. (2010). A New Metaheuristic Bat-Inspired

Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (pp. 65-74). Springer.

Yang, X. S. (2011). Review of Metaheuristics and Generalised Evolutionary Walk Algorithm. *International Journal of Bio-Inspired Computation, 3*(2), 77-84.

Yang, X. S., & Deb, S. (2009, December). Cuckoo Search via Lévy Flights. *Proceeding of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, India.

Yao, X., & Liu, Y. (1997). Fast Evolution Strategies. *Control And Cybernetics, 26*(3), 467-496.

Yao, X., Liu, Y., & Lin, G. (1999, July). Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82-102.