PROGRESS IN APPLIED SCIENCE AND TECHNOLOGY
Faculty of Science and Technology
Rajamangala University of Technology Thanyaburi
Vol.11 No.1 (2021): 10–24
ISSN (Print): 2730-3012
ISSN (Online): 2730-3020
https://ph02.tci-thaijo.org/index.php/past

Received 25th January 2021,
Revised 17th March 2021,
Accepted 22nd March 2021

DOI: 10.14456/past.2021.7

Global Performance Test of Metaheuristics Optimization and Engineering Applications

Somboon Sukpancharoen[1*]
[1] *Department of Mechatronics and Robotics Engineering, Rajamangala University of Technology Thanyaburi, Pathum Thani, 12110, Thailand*
*E-mail: somboon_s@rmutt.ac.th

Abstract

In this research study, 10 of the latest metaheuristics' performance characteristics are compared in the context of 19 unconstrained benchmark functions, where the large dimension of the test challenge is 100. Optimization problems may often encompass a large number of design variables, exerting complex effects upon the specific objective function. The performance is evaluated as the algorithm seeks a global optimum and avoids becoming trapped in a local optimum through the use of best values, mean and standard deviation (Stdev.). This study also uses Friedman Aligned Ranks and applies the Quade Ranks test to examine the differences in performance as the algorithms seek their solutions. Analysis of exploitation is conducted using Friedman Aligned Rank tests, while exploration is addressed using the Quade Ranks test. The study revealed that the different algorithms use different approaches to look for their solutions, with a significance level of 0.05. Finally, the comparison of the ten algorithms' performance is presented in this paper in the context of solving constrained mechanical and chemical engineering problems.

Keywords: Metaheuristics, Optimization techniques, Unconstrained benchmark functions, Constrained engineering optimization, Friedman Aligned Ranks, Quade Ranks

1. Introduction

Among the various real-world problems addressed by artificial intelligence or machine learning involve a number of different types, including discrete, continuous, constrained, or unconstrained problems. However, studies have shown that the suitable approaches for these various problem types are not always sufficient to handle large-scale multimodal real-world problems that may be non-continuous or non-differentiable (1-3). Therefore, metaheuristic algorithms are suitable to address a wide range of problems and have thus been designed and employed as competitive alternative solvers based on their ease of implementation and inherent simplicity. Furthermore, these techniques' fundamental operating processes are not reliant upon the gradient details for the objective landscape, nor are the mathematical traits necessary.

All optimization algorithms act by performing the exploration and exploitation processes within a particular search space (4), while to obtain the best solutions, the two processes must be suitably balanced. During the exploration stage, the algorithm will search the most promising areas within the search space, while the exploitation stage finds the best solutions within those most promising areas (5). Optimal solutions or solutions close to optimal can only be found when the exploration and exploitation processes are suitably tuned. Given that there are so many optimization algorithms available, it is always questionable whether there is a need to develop different optimization algorithms.

Metaheuristics have increasingly taken precedence in recent studies (6–7) and are now more widely applied. There is also a comparison metaheuristic with benchmark functions (8). They can be categorized into four broad types, as shown in Figure 1: genetic evolution-based algorithms, animal and plant-based algorithms, human activity-based algorithms, and physics-based algorithms (9). The algorithms can be described as problem-independent since they can be used to seek approximate optimal solutions in the context of highly complex nonlinear problems that cannot be addressed by traditional deterministic approaches due to the time that conventional techniques would require (10). Accounting for their increased popularity in the science and engineering fields today, metaheuristic optimization algorithms offer many advantages over the conventional approaches (11): they are flexible, simple, have a derivation-free mechanism, and avoid local optima. The drawback is that an exact solution cannot be guaranteed, but the solutions generated are generally acceptable and will be achieved much more quickly.

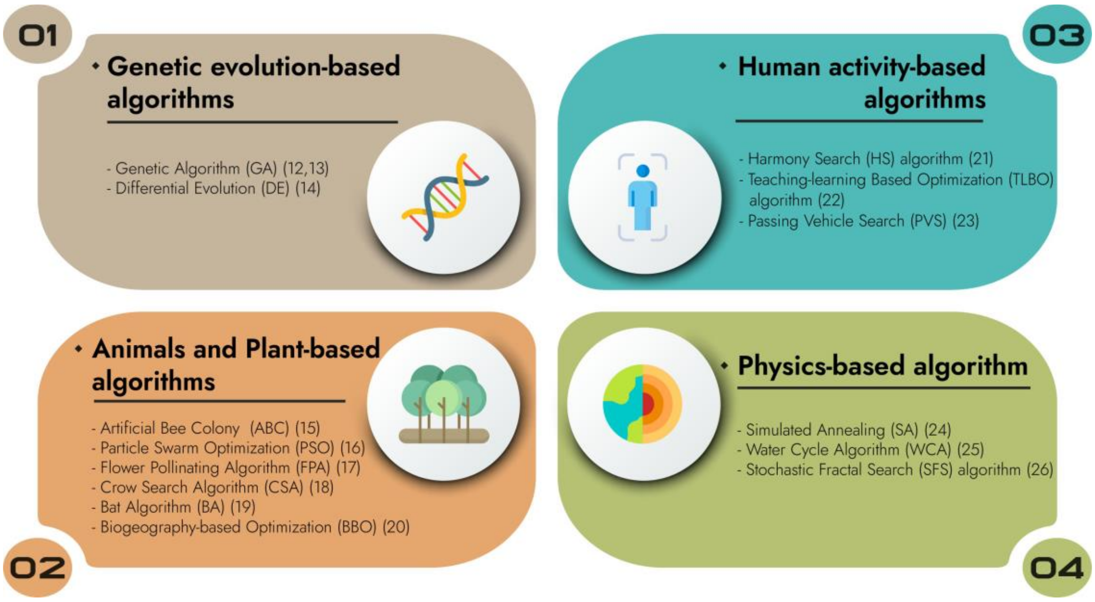


Figure 1 The four main categories for metaheuristic algorithm inspiration

When new optimization algorithms are created, it is necessary to verify their operation and draw comparisons with alternative algorithmic approaches. To do this, benchmark evaluation functions may be employed to measure the efficiency of the algorithm. To assess the output from the optimization process, the various algorithms are tested, and comparisons are drawn by using the benchmark functions. In this study, ten novel algorithms from 2018 to 2020 are assessed for optimization. They are shown in Table 1 and include Beetle Antennae Search (BAS), Atom Search Optimization (ASO), Henry Gas Solubility Optimization (HGSO), Sunflower Optimization (SFO), Seagull Optimization Algorithm (SOA), Mayfly Algorithm (MA), Marine Predators Algorithm (MPA), Parasitism Predation Algorithm (PPA), Slime Mould Algorithm (SMA), and Tunicate Swarm Algorithm (TSA). A total of 19 benchmark test functions were also selected to verify and analyze the algorithms' performance.

The fields of chemical and mechanical engineering or similar industries have used the problem-solving capacity of metaheuristics, including Gray Wolf Optimizer (GWO) or PSO using an ABC algorithm (27). Simultaneously, researchers have also sought to employ such algorithms as GWO with multi-objective optimization (MOO) to address problems such as producing biodiesel from waste cooking oil to lower the capital costs and procedural costs advantage of the primary functions of MOO (28). The field of energy conservation applied an HS algorithm to perform the synthesis of heat-integrated distillation sequences. This method requires that the distillation sequence be first considered, followed by the heat integration, to hold the overall annual cost to a minimum. The HS algorithm can be improved to address this particular problem and proved itself to be both robust and effective in this context (29). In electrical engineering, Electrical Load Dispatch (ELD) offers a means of combining many power generating units in an optimal formation, minimizing the overall fuel cost while simultaneously generating sufficient power to meet the load demand. The technique uses a modified directional Bat Algorithm (dBA) to ensure that operational constraints are respected. It can be seen that the use of dBA allows operational costs to be reduced with improved convergence attributes and lower computational time (30).

This research paper begins in Section 2 with an overview of metaheuristic optimization algorithms. Section 3 describes the benchmark functions tests and the procedures for performance testing. The ranking system to assess each algorithm's performance is also discussed. Section 4 presents the results, including the best value, mean, Stdev., Friedman Aligned Ranks test and Quade Ranks test outcomes. Section 5 contains an analysis of constrained mechanical-chemical engineering problems, and the principal findings and conclusions are presented in Section 6.

Table 1 Various algorithms and their creators

Algorithm	Proposed by	Proposed year
BAS	Jiang and Li (31)	2018
ASO	Zhao et al. (32)	2019
HGSO	Hashim et al. (33)	2019
SFO	Gomes et al. (34)	2019
SOA	Dhiman and Kumar (35)	2019
MA	Zervoudakis and Tsafarakis (36)	2020
MPA	Faramarzi et al. (37)	2020
PPA	Mohamed et al. (38)	2020
SMA	Li et al. (39)	2020
TSA	Kaur et al. (40)	2020

2. A Review of metaheuristic algorithms used for the optimization

2.1 Beetle Antennae Search (BAS)

BAS has its basis in the way the longhorn beetle behaves as it searches for food. The algorithm mirrors the activity of the beetle's antennae and resulting walking, which is naturally performed by the beetle. The principal steps are therefore detection and scanning (31).

2.2 Atom Search Optimization (ASO)

The basis of ASO algorithms is the atomic motion model, which is structured mathematically to represent the way atoms interact, taking into account the Lennard-Jones (41) interaction and restriction forces that would result from the bond length potential. This approach is both simple and offers ease of practical use.

The first five atoms which offer the ideal fitness value are designated as $KBest$, and Figure 2 presents the atom population forces. The cases identified as A_1 , A_2 , A_3 and A_4 makeup $KBest$, as Figure 2 shows, while A_5 , A_6 and A_7 serve to entice or provide resistance to each of the $KBest$ atoms while A_1 , A_2 , A_3 and A_4 either resist or entice each other. Within the population, each atom has a limitation force derived from the best atom A_1 , with the exception of A_1 (x_{best}) (32).

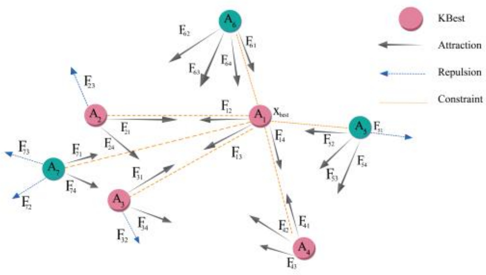


Figure 2 Forces within the system of atoms with $KBest$ for $K = 5$

2.3 Henry Gas Solubility Optimization (HGSO)

Henry's law provides the foundation for the Henry Gas Solubility Optimization (HGSO) algorithm to find solutions to complex problems involving optimization. This integral gas law explains the amount of gas that can be dissolved within a particular form and volume of a liquid at a given temperature. HGSO copies the buoyancy of gas for the purposes of extraction and discovery within the search space while avoiding the problem of local optimization (33).

2.4 Sunflower Optimization (SFO)

The SFO algorithm is an iterative and population-based global heuristic algorithm for optimization. It differs from conventional algorithms in making use of terminology, including root velocity and robust pollination. In the natural world, sunflowers follow the same daily routine, following the sun throughout the day as it moves across the sky. During the night, the movement direction is reversed as they await the sunrise the following morning.

In testing the SFO algorithm, Gomes et al. discovered that good positions could be located in regular feature testing, thus confirming the right level of efficiency. However, certain adjustments were necessary to optimize various dimensions to ensure maximized computational (34).

2.5 Seagull Optimization Algorithm (SOA)

The SOA is based on the way seagulls naturally move and make their attacks on migratory birds as their victims cross the sea (42). Their attacks typically involve a spiral movement as they approach their targets. This can be represented mathematically, as shown in Figure 3. It is essential to consider this action to correlate with the target function to be further refined if necessary. It is thus, feasible to develop a new optimization algorithm. In this context, SOA is based on seagulls' two biological activities: Moving, and attacking, respectively to exploration and exploitation (35).

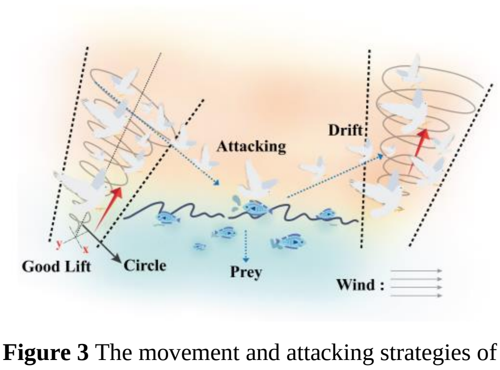


Figure 3 The movement and attacking strategies of seagulls

2.6 Mayfly Algorithm (MA)

In 2020, Zervoudakis and Tsafarakis developed the mayfly algorithm (MA) inspired by mayflies' mating behavior. Upon hatching, mayflies become adults, and while the lifespan may be short, it is the healthiest mayflies that can survive. Solutions to an optimization problem are given by the location of the mayfly within a search space. The algorithm works by first creating two mayfly sets to serve as males and females (36).

2.7 Marine Predators Algorithm (MPA)

The marine predator's algorithm (MPA) uses the foraging activity of sea creatures as the model, in the form of Lévy and Brownian behaviors, along with the optimization policy, which considers the rate at which encounters between marine predators and their prey take place. MPA employs the natural rules that govern the optimal foraging strategy in line with prey and predators' meeting rate within the natural ocean setting. There are three critical phases of MPA optimization, Phase 1: The high-velocity ratio when the prey speed exceeds that of the predator, Phase 2: The unit velocity ratio when the predator and prey have equal speeds, giving the impression that both are seeking prey, Phase 3: The low-velocity ratio when the predator speed exceeds that of the prey. This situation will arise during the last stage of the optimization process and is associated with more significant exploitation potential. Figure 4 presents a visual representation of the three phases involved in MPA optimization (37).

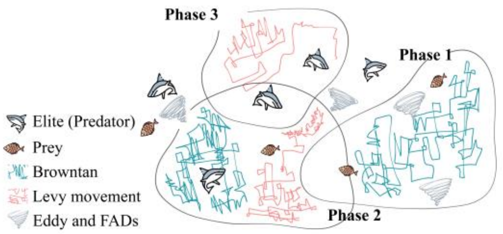


Figure 4 Three optimization phases using MPA

2.8 Parasitism Predation Algorithm (PPA)

The PPA is based on the complicated relationship between predators, parasites, and hosts, represented respectively by cats, cuckoos, and crows, which are sued to produce the crow-cuckoo-cat device model, which is able to address the issues arising in problems with a lack of cohesion and large data dimensions (38).

2.9 Slime Mould Algorithm (SMA)

The SMA is based on the slime mould known as Physarum polycephalum, which was initially understood to be a fungus when Howard (43) described its life cycle in 1931. It is eukaryotic which thrives in locations that have high humidity and low temperatures. Its principal nutritional phase is plasmodium, representing the slime mould's complex activity stage, replicated by the algorithm for its primary testing process. In this phase, the slime mold's organic components seek food before occupying that food and consuming it by the emission of enzymes. As the slime mould migration occurs, the front end takes on a fan-shaped appearance, as seen in Figure 5, while an intersected venous network is developed behind this to facilitate cytoplasm movement (39,44).

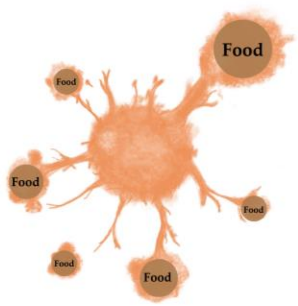


Figure 5 The slime mould foraging morphology

2.10 Tunicate Swarm Algorithm (TSA)

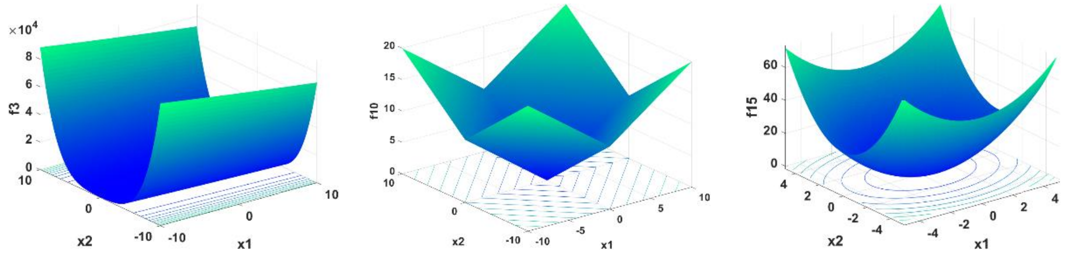
The TSA uses swarm intelligence to address real-world optimization problems using community or multi-solution swarm intelligence to analyze the search space better to find the global optima. However, such an approach cannot solve all optimization problems, as explained in the No Free Lunch Theorem (45). These shortcomings inspired to seek a novel population-based metaheuristic algorithm that might address those problem types that cannot currently be solved using current techniques (40).

3. Methodology

3.1 Benchmark functions test

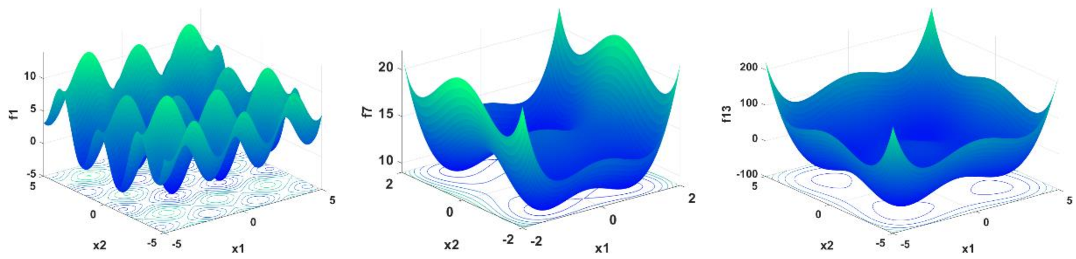
When assessing an algorithm's optimal performance, its achievements must be compared with those of other algorithms for validation by testing with a set of benchmark test functions (46). These benchmarks, which are artificial problems, allow assessments to be carried out for the algorithm's efficiency, reliability, and validity under examination. Accordingly, the performance can be objectively rated.

This research study examined 19 benchmark test functions, which are presented in Table 2. These were drawn from the work of Jamil and Yang (46). For the ten optimization algorithms examined in this study, the benchmark functions have at least one optimum. Table 2 shows that ten are shown to be unimodal across the 19 benchmark functions as a whole, while the remainder is multimodal. It is possible to assess the algorithms' exploitation capabilities using the unimodal test functions, while algorithms can be better explored in greater depth by using the multimodal functions. The unimodal function plots can be seen in Figure 6, while the multimodal plots are presented in Figure 7.



(a) (b) (c)

Figure 6 3D surface plot of unimodal function: (a) Dixon Price, (b) Schwefel 2.20, (c) Sum Squares



(a) (b) (c)

Figure 7 3D surface plot multimodal function:(a) Alpinen1, (b) Qing, (c) Styblinski-Tank

3.2 Performance testing procedure

A variety of benchmark functions were selected to encompass a range of features. There were 19 in total. Ten optimization algorithms were then examined in the context of the 19 unconstrained test problems. The algorithms' experimentation process is carried out with MATLAB Version R2020a, while the simulation uses Microsoft Windows 10 (64 bit) with a Core i7 processor with 2 GHz and memory of 8 GB. The experiments for all algorithms employed the same conditions to ensure fairness, with the maximum number of evaluations set to 50,000. The dimensions for all algorithms were set to be 100 when problem-solving. The effects of any other random factors which might influence the algorithm results were mitigated by running the comparative algorithms separately for 30 iterations in each function before taking the final result. The measurements included best values, mean and Stdev. These values were used for further analysis.

Table 2 The selected 19 benchmark testing functions

Function Name **Function Formulation** **Dim (D), [lower bound, upper bound]**
Alpinen1 $f_1(x)=\sum_{i=1}^{d}\left\|x_i\sin(x_i)+0.1x_i\right\|$ D=100, [-10,10]
Brown $f_2(x)=\sum_{i=1}^{d-1}\left(x_i^2\right)^{\left(x_{i+1}^2+1\right)}+\left(x_{i+1}^2\right)^{\left(x_i^2+1\right)}$ D=100, [-5,5]

Table 2 (Countinued)

Function Name **Function Formulation** **Dim (D), [lower bound, upper bound]**
Dixon& Price $f_3(x)=(x_1-1)^2+\sum_{i=2}^{d}i\left(2x_i^2-x_{i-1}\right)^2$ D=100, [-10,10]
Happy Cat $f_4(x)=\left[\left(\left(\|x\|^2-d\right)^2\right]^{\frac{1}{8}}+\frac{1}{d}\left(\frac{1}{2}\|x\|^2+\sum_{i=1}^{d}x_i\right)+\frac{1}{2}$ D=100, [-2,2]
Periodic $f_5(x)=1+\sum_{i=1}^{d}\sin^2(x_i)-0.1e^{\left(\sum_{i=1}^{d}x_i^2\right)}$ D=100, [-15,15]
Powellsum $f_6(x)=\sum_{i=1}^{d}\|x_i\|^{i+1}$ D=100, [-1,1]
Qing $f_7(x)=\sum_{i=1}^{d}(x^2-i)^2$ D=100, [-525,525]
Quartic $f_8(x)=\sum_{i=1}^{d}ix_i^4+random[0,1)$ D=100, [-2,2]

Table 2 (Countinued)

Function Name	**Function Formulation**	**Dim (D), [lower bound, upper bound]**
Rastrigin	$f_9(x)=10d+\sum_{i=1}^{d}\left[x_i^2-10\cos(2\pi x_i)\right]$	D=100, [-10,10]
Schwefel 2.20	$f_{10}(x)=\sum_{i=1}^{d}\lvert x_i\rvert$	D=100, [-110,110]
Schwefel 2.22	$f_{11}(x)=\sum_{i=1}^{d}\lvert x_i\rvert+\prod_{i=1}^{d}\lvert x_i\rvert$	D=100, [-110,110]
Schwefel 2.23	$f_{12}(x)=\sum_{i=1}^{d}x_i^{10}$	D=100, [-15,15]
Styblinski-Tank	$f_{13}(x)=\frac{1}{2}\sum_{i=1}^{d}(x_i^4+16x_i^2+5x_i)$	D=100, [-5,5]
Sum of Different Powers	$f_{14}(x)=\sum_{i=1}^{d}\lvert x_i\rvert^{i+1}$	D=100, [-10,10]
Sum Squares	$f_{15}(x)=\sum_{i=1}^{d}ix_i^2$	D=100, [-10,10]
Trid	$f_{16}(x)=\sum_{i=1}^{d}(x_i-1)^2-\sum_{i=2}^{d}x_ix_{i-1}$	D= 50, [-6000,6000]
Xin-She Yang N.1	$f_{17}(x)=\sum_{i=1}^{d}\phi\lvert x_i\rvert^{i}$; ϕ is a random form [0,1]	D=100, [-10,10]
Xin-She Yang N.3	$f_{18}(x)=\exp\left(-\sum_{i=1}^{d}\left(\frac{x_i}{15}\right)^{10}\right)-2\exp\left(-\sum_{i=1}^{d}x_i^2\right)\prod_{i=1}^{d}\cos^2(x_i)$	D=100, [-10,10]
Xin-She Yang N.4	$f_{19}(x)=\left(\sum_{i=1}^{d}\sin^2(x_i)-e^{-\sum_{i=1}^{d}x_i^2}\right)e^{-\sum_{i=1}^{d}\sin^2\sqrt{\lvert x_i\rvert}}$	D=100, [-20,20]

In this study, rankings are to be applied as a performance metric, based on assigning a rank to each algorithm for each benchmark test function's performance. A ranking of 1 is given to the best performer, while ten is assigned to the worst. Exploitation and Exploration performance are measured through the Friedman Aligned Ranks test and Quade Ranks tests, respectively and weightings are used for the row ranks, whereby the factor used is a function of the row range (47).

4. Results and discussion

4.1 Comparing each algorithm in terms of the best value

Statistical outcomes for the optimal results for each of the algorithms on benchmark functions are presented in Table 3, with numbers in the bold font representing the best values in each of the benchmark function tests, which serve as the global optimum. Those figures indicated with an asterisk are solutions that are a near-global optimum. It was found that the PPA and SMA algorithms were able to produce the best values for each of the functions when compared to the alternative algorithms when considering 13 of the 19 functions for PPA and 12 functions for SMA. However, SMA could determine a more significant number of responses for the global optimal point than PPA, whereas the remaining algorithms, including BAS, SFO, ASO, MA, and TSA, could still not beat other algorithms whose best values were superior. SMA can avid the local optimum in functions 6 and 17, thus exhibiting the capacity to explore around the optimum. In this regard, SMA is the sole algorithm capable of reaching an optimal solution in the case of those two functions. When considering the best values shown in Table 3, it is apparent that SMA has a more remarkable ability in the context of multimodal problems than it does when faced with a unimodal problem.

Furthermore, it demonstrates better solutions with scalable problems than is the case for non-scalable problems. Its ability to obtain solutions for particular problems is better than the alternatives, but separable problems are not addressed effectively as non-scalable problems. It achieves better performance with continuous forms than with non-continuous forms, and the data for functions 3, 4, 7, and 10 suggest that PPA offers the best solutions to these problems and can work around the optimum point, demonstrating a similar mathematical basis to SMA. The difference is that PPA is better able than SMA to solve unimodal problems rather than multimodal problems and can handle both separable and non-separable problems.

Meanwhile, HGSO achieves the best value for six functions and a near-global optimal solution for six functions. It demonstrates good exploitation capability close to the optimum point, and while the mathematical basis has similarities to that of PPA, HGSO is better able to handle separable problems than non-separable problems. Table 3 reveals that BAS can find a near-optimal solution for function 12, although this was the only function for BAS to achieve any search success.

Table 3 Comparison of the best value for each algorithm

#	BAS	ASO	HGSO	SFO	SOA	MA	MPA	PPA	SMA	TSA
f_1	254.94766	0.01466	3.898E-226*	1.138E+02	0.01466	99.09982	1.885E-27*	0.13813	8.741E-294*	105.46802
f_2	6.356E+29	4.299E-09*	**0.00000**	4.634E-04*	2.453E-18*	3.513E+21	3.196E-47*	**0.00000**	**0.00000**	1.656E-31*
f_3	1.676E+08	1.08275	0.66667	0.82381	0.66667	0.05156	0.66667	**0.00000**	0.25779	0.66667
f_4	9.0014	0.7794	2.0588	0.8963	0.7075	0.6739	0.6773	**0.0000**	0.0003*	0.6560
f_5	42.532	1.005*	**0.900**	1.001*	1.804	39.272	**0.900**	**0.900**	**0.900**	25.617
f_6	2.44260	1.022E-23*	1.693E-13*	2.302E-05*	3.115E-59*	1.350E-09*	4.702E-124*	3.133E-08*	**0.00000**	7.850E-159*
f_7	1.445E+12	2.61073	2.252E+05	8.879E+04	2.096E+05	9.787E+11	5.771E+03	**0.00000**	1.795E+03	4.178E+04
f_8	1.147E+04	0.32513	4.389E-06*	5.497E-04*	4.803E-04*	5.626E+02	3.529E-04*	2.094E-06*	3.575E-05*	7.810E-03*
f_9	3.476E+03	1.622E+02	**0.00000**	1.32411	**0.00000**	0.57518	**0.00000**	**0.00000**	**0.00000**	7.586E+02
f_{10}	3.219E+12	7.87779	5.055E-223*	5.93744	6.729E-11*	4.460E+03	5.242E-26*	**0.00000**	3.073E-289*	1.993E-17*
f_{11}	2.457E+146	8.469E+02	5.326E-235*	7.993E+00	2.156E-11*	1.748E+93	1.179E-26*	**0.00000**	**0.00000**	6.931E-18*
f_{12}	6.423E-09*	1.052E-11*	**0.00000**	4.258E-16*	2.229E-58*	2.093E+12	1.472E-190*	**0.00000**	**0.00000**	2.836E-39*
f_{13}	-1024.511	-3562.905*	-1602.034	-2069.497	-1999.315	-1337.578	-3499.138*	-39.166	**-3916.616**	-2643.394
f_{14}	4.115E+89	2.127E+07	**0.00000**	1.106E-06*	1.066E-84*	3.195E-35*	4.105E-116*	**0.00000**	**0.00000**	1.677E-106*
f_{15}	1.514E+05	8.725E-08*	**0.00000**	0.48165	1.855E-18*	3.038E-07*	8.998E-46*	**0.00000**	**0.00000**	1.779E-30*
f_{16}	8.149E+08	1.444E+07	3.510E+07	-6.980E+03	5.299E+00	-5.447E+03	-1.893E+03	0.000E+00	-1.099E+04*	-2.284E+02
f_{17}	8.844E+82	1.241E+12	1.055E-150*	2.331E-04*	1.155E-22*	1.628E+13	1.016E-33*	2.382E-277*	**0.00000**	1.02971
f_{18}	4.227E-272	0.71510	0.17655	-0.97137*	0.17655	0.48233	**-1.00000**	**-1.00000**	**-1.00000**	0.50908
f_{19}	6.011E-27	2.845E-39	8.921E-40	-5.427E-01*	8.138E-40	2.602E-27	1.252E-42	**-1.00000**	**-1.00000**	5.557E-39

*Near-global optimal value

4.2 Comparing the mean and standard deviation for each of the algorithm

Table 4 presents a comparison of the ten algorithms' outcomes in terms of mean and Stdev. Excellent performance or robustness is indicated by a Stdev. approaching zero, while the numbers shown in bold font represent the best values for the mean of each benchmark function along with a low value for the standard deviation. It can be seen from Table 4 that the best optimizer is PPA, ahead of SMA and HGSO, respectively. PPA owes its success to its ability to avoid the local optima, as observed in its performance on functions 3, 4, 7, 11, and 19. For function 11, the Stdev. for PPA, SMA, and HGSO were all equal to zero, indicating excellent robustness. However, the data do not show whether the HGSO and SMA algorithm can avoid the local optima. This also applies to function 19 for the same reasons. Table 4 indicates that the MPA, SOA, TSA, SFO, ASO, and MA algorithms can find the mean of the optimal point, which is close to a global optimum for 11 functions (MPA), nine functions (SOA), seven functions (TSA), five functions (SFO), four functions (ASO), and two functions (MA). These algorithms show that they help exploit analysis and address unimodal problems much more quickly than multimodal problems.

Furthermore, these algorithms show better performance with continuous problems than discontinuous problems and better solve scalable problems than the non-differential problem or separable problems. However, one exception is the ASO algorithm, which is better in non-separable problems than separable problems.

Figure 8 presents the column bar of average mean, best, worst, and Stdev. ranking and presents the average overall ranking line. It can be concluded that algorithms can be classified for global performance into four groups according to the average overall ranking by considering percentiles. The 1st quartile optimizer is the SMA method, the 2nd quartile optimizers are the PPA, MPA, and HGSO methods, the 3rd quartile optimizers are the TSA, SOA, and SFO methods, and the 4th quartile optimizers are the ASO, MA, and BAS methods.

Table 4 Comparative mean and Stdev. of each algorithm

#	Statistical	BAS	ASO	HGSO	SFO	SOA	MA	MPA	PPA	SMA	TSA
f_1	Mean	283.42905	0.62405	**3.986E-196**	1.806E+02	0.62405	107.41074	6.575E-26	0.55345	1.593E-183	136.95194
	Stdev.	16.85964	0.86697	**0.00000**	4.137E+01	0.86697	7.15790	5.900E-26	0.66192	**0.00000**	24.48173
f_2	Mean	2.881E+31	8.271E-05	**0.00000**	0.06486	8.625E-12	3.993E+23	6.324E-46	**0.00000**	**0.00000**	1.647E-28
	Stdev.	4.017E+31	2.612E-04	**0.00000**	0.07136	2.704E-11	6.931E+23	9.363E-46	**0.00000**	**0.00000**	2.604E-28
f_3	Mean	2.058E+08	10.95289	0.66667	9.02345	0.66667	1.712E+03	0.66667	**0.00000**	0.52392	0.66667
	Stdev.	2.866E+07	9.23969	3.447E-07	8.20522	3.545E-06	5.385E+03	4.260E-08	**0.00000**	0.20693	3.545E-06
f_4	Mean	33.9179	0.8477	9.7681	1.0055	0.8296	0.8166	0.7560	**0.0000**	0.1101	0.7800
	Stdev.	11.0751	0.0473	4.3569	0.1033	0.0841	0.0883	0.0504	**0.0000**	0.1284	0.0973
f_5	Mean	46.116	4.034	**0.900**	1.018	18.983	42.397	2.340	**0.900**	**0.900**	28.013
	Stdev.	2.39423	3.75850	**1.170E-16**	0.01693	12.70658	1.88719	2.19037	**1.170E-16**	**1.170E-16**	2.03404
f_6	Mean	3.79708	2.059E-21	1.766E-07	1.912E-04	8.825E-07	3.114E-07	2.071E-117	3.133E-08	**0.00000**	2.540E-134
	Stdev.	0.99072	4.790E-21	3.719E-07	2.370E-04	2.791E-06	2.933E-07	5.985E-117	6.975E-24	**0.00000**	8.032E-134
f_7	Mean	1.533E+12	4.150E+09	2.567E+05	1.737E+05	2.320E+05	1.062E+12	1.246E+04	**0.00000**	2.605E+03	6.571E+04
	Stdev.	6.509E+10	6.395E+09	1.880E+04	7.746E+04	1.458E+04	4.988E+10	9.041E+03	**0.00000**	730.79736	1.400E+04
f_8	Mean	1.509E+04	0.48365	4.336E-05	0.02773	1.236E-03	8.940E+02	9.930E-04	**1.771E-05**	1.291E-04	0.01250
	Stdev.	2.615E+03	0.12207	2.193E-05	0.05648	7.306E-04	2.245E+02	5.231E-04	**1.190E-05**	6.072E-05	3.889E-03
f_9	Mean	4.123E+03	2.092E+02	**0.00000**	1.905E+01	0.09981	7.154E+01	**0.00000**	**0.00000**	**0.00000**	8.985E+02
	Stdev.	3.042E+02	3.870E+01	**0.00000**	1.210E+01	0.31564	4.907E+01	**0.00000**	**0.00000**	**0.00000**	1.331E+02
f_{10}	Mean	5.078E+12	18.09511	1.889E-198	20.77224	4.691E-10	4.676E+03	4.377E-24	**2.043E-304**	5.015E-202	8.344E-17
	Stdev.	1.009E+12	7.10930	**0.00000**	11.98781	3.460E-10	1.278E+02	5.509E-24	**0.00000**	**0.00000**	7.071E-17
f_{11}	Mean	2.474E+163	1.146E+03	1.103E-191	2.606E+01	1.656E-10	1.111E+112	1.285E-24	**3.192E-284**	1.516E-168	2.405E-17
	Stdev.	2.000E+164	2.138E+02	**0.000E+00**	1.555E+01	1.359E-11	3.071E+112	1.643E-24	**0.000E+00**	**0.000E+00**	2.331E-17
f_{12}	Mean	4.712E+12	1.304E-05	**0.00000**	6.821E-03	3.062E-30	2.779E+12	2.411E-179	**0.00000**	**0.00000**	1.099E-31
	Stdev.	2.000E+12	2.903E-05	**0.00000**	0.02122	9.684E-30	3.824E+11	**0.00000**	**0.00000**	**0.00000**	2.930E-31
f_{13}	Mean	-535.4620	-3354.8381	-1417.3697	-1891.1113	-1871.0441	-1249.3802	-3371.5149	-39.1662	**-3916.5952**	-2532.3162
	Stdev.	326.038	88.898	113.657	92.265	122.023	73.385	83.714	**0.000**	0.042	101.998
f_{14}	Mean	5.11E+95	1.85E+12	**0.00000**	9.21E-03	1.35E-72	1.25E-21	3.83E-110	**0.00000**	**0.00000**	1.29E-43
	Stdev.	1.124E+96	3.169E+12	**0.00000**	1.821E-02	3.985E-72	3.962E-21	1.026E-109	**0.00000**	**0.00000**	4.066E-43
f_{15}	Mean	1.696E+05	0.03840	**0.00000**	2.71401	1.554E-16	7.91966	1.893E-44	**0.00000**	**0.00000**	2.400E-28
	Stdev.	1.333E+04	0.10981	**0.00000**	2.77089	2.786E-16	2.424E+01	1.804E-44	**0.00000**	**0.00000**	2.396E-28
f_{16}	Mean	9.087E+08	5.131E+07	3.510E+07	7.837E+03	1.998E+01	1.792E+04	-1.523E+03	3.000E-01	**-1.099E+04**	-6.210E+01
	Stdev.	7.078E+07	1.473E+07	**0.000E+00**	1.805E+04	9.245E+00	2.926E+04	2.287E+02	4.830E-01	1.060E+03	8.103E+01
f_{17}	Mean	1.441E+94	5.078E+29	1.953E-91	6.008E-03	7.115E-06	1.143E+18	7.104E-22	1.295E-255	**0.00000**	275.50450
	Stdev.	4.558E+94	1.602E+30	6.174E-91	5.763E-03	2.249E-05	2.393E+80	2.091E-21	**0.000E+00**	**0.00000**	531.44520
f_{18}	Mean	6.664E-111	0.72947	0.17655	-0.20473	0.17655	0.54667	0.05889	**-1.00000**	**-1.00000**	0.54291
	Stdev.	2.107E-110	9.743E-03	2.926E-17	0.39070	2.926E-17	0.06506	0.37206	**0.00000**	**0.00000**	0.01844
f_{19}	Mean	5.246E-22	3.306E-25	8.921E-40	-5.596E-02	8.790E-40	2.019E-23	5.182E-42	**-1.00000**	-0.99999	3.287E-38
	Stdev.	1.388E-21	5.343E-25	**0.000E+00**	1.711E-01	2.819E-41	4.149E-23	5.570E-42	**0.00000**	4.302E-05	3.115E-38

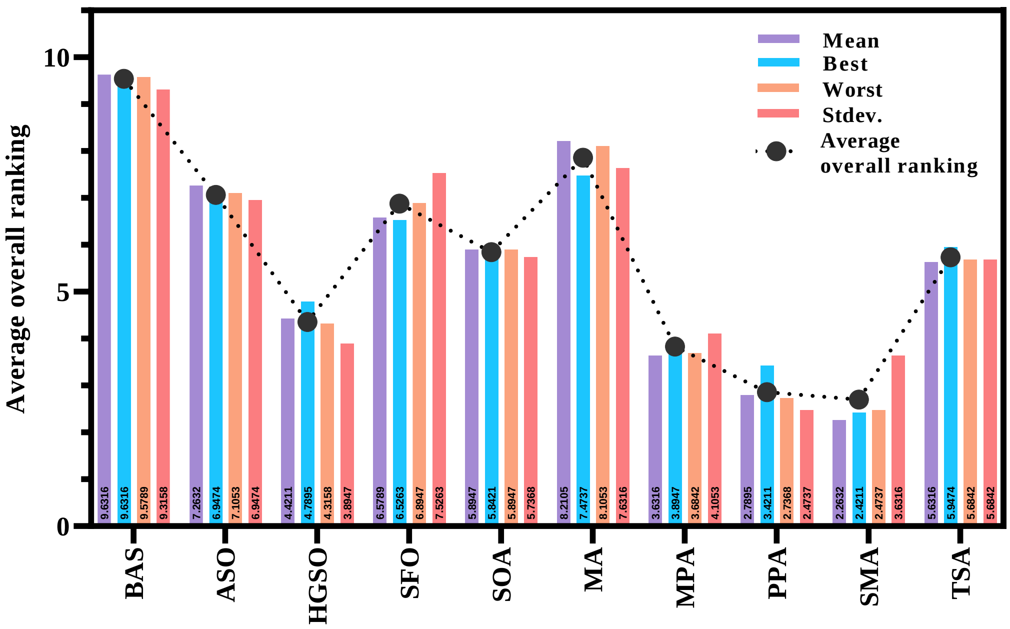


Figure 8 Average overall ranking for each algorithm

4.3 Authentication

4.3.1 Friedman Aligned Ranks test for exploitation

The Friedman Aligned Ranks test uses data rankings rather than looking directly at the data values themselves. For this reason, it is possible to avoid the classical assumptions concerning the parametric analysis of variance (ANOVA), which include a normal distribution and constant variance. Aligned observations have assigned ranks known as aligned ranks. The T value is compared to the Chi square-distribution (X^2) while the degrees of freedom set to be 9. Table 5 presents the aligned ranks in this study using the Friedman tests for exploitation with the ten algorithms for unimodal functions. The rankings alone can provide an indication allowing comparison of the algorithms. On average, SMA was ranked first with a score of 22.1, followed by PPA (26.2) and MPA (36.4). In the last place was BAS (85.6).

For the Friedman testing the statistics included T = 54.849, while the P-value calculated from the X_9^2 = 16.91898, leading to the rejection of the null hypothesis. Thus, the findings indicate that the ten algorithms employ different approaches to determine solutions in exploitation problems, with a significance level of 0.05.

4.3.2 Quade Ranks test for exploration

In this scenario, the values are compared to the F-distribution percentiles when the degrees of freedom are equal $F_{8.72}$. S_j represents the sum for each of the classifiers. In this study, exploration was evaluated by the Quade Ranks test. The aligned observations and rankings (in brackets) are shown in Table 6 in the context of multimodal functions for each of the ten algorithms. SMA was ranked first overall (2.0) ahead of PPA (2.7) and MPA (3.2), while BAS was once again last (8.9).

V_8 = 41.2866 has a distribution that follows the F distribution with 9-1=8 and 72 degrees of freedom. Comparisons of the P-value were made using the $F_{8.72}$ =2.06983164, leading to the rejection of the null hypothesis. Therefore, it was concluded that the ten algorithms find their exploration problem solutions using significantly different approaches, at a significance level of 0.05.

Table 5 Results of ranks test for unimodal function: Friedman Aligned Ranks test

#	BAS	ASO	HGSO	SFO	SOA	MA	MPA	PPA	SMA	TSA
f_2	97	55	20	61	47	96	33	20	20	38
f_3	90	77	69	76	71	83	68	20	65	71
f_6	74	43	51	56	53	52	29	50	20	28
f_{10}	95	78	24	80	49	84	41	21	23	45
f_{11}	100	82	25	81	48	99	40	22	27	44
f_{12}	94	54	20	57	37	93	26	20	20	36
f_{14}	98	92	20	58	32	42	31	20	20	35
f_{15}	87	59	20	73	46	75	34	20	20	39
f_{16}	91	89	88	85	79	86	2	64	1	3
f_{18}	30	72	63	6	63	67	60	5	5	66
Average Friedman Aligned Ranks test	**85.6**	**70.1**	**40.0**	**63.3**	**52.5**	**77.7**	**36.4**	**26.2**	**22.1**	**40.5**

Table 6 Results of ranks test for multimodal function: Quade Ranks test

#	BAS	ASO	HGSO	SFO	SOA	MA	MPA	PPA	SMA	TSA
f_1	18	2	-18	14	2	6	-10	-6	-14	10
f_4	9	3	7	5	1	-1	-5	-9	-7	-3
f_5	13.5	1.5	-7.5	-4.5	4.5	10.5	-1.5	-7.5	-7.5	7.5
f_7	36	20	12	-4	4	28	-20	-36	-28	-12
f_8	22.5	12.5	-17.5	7.5	-2.5	17.5	-7.5	-22.5	-12.5	2.5
f_9	31.5	17.5	-10.5	3.5	-3.5	10.5	-10.5	-10.5	-10.5	24.5
f_{13}	21	-15	9	-3	3	15	-21	27	-27	-9
f_{17}	31.5	40.5	-22.5	4.5	-4.5	22.5	-13.5	-31.5	-40.5	13.5
f_{19}	4.5	2.5	0.5	-2.5	-0.5	3.5	-1.5	-4.5	-3.5	1.5
S_j	187.5	84.5	-47.5	20.5	3.5	112.5	-90.5	-100.5	-150.5	35.5
Average Quade Ranks test	**8.80**	**6.40**	**4.20**	**5.30**	**5.10**	**7.10**	**3.20**	**2.70**	**2.00**	**5.70**

5. Mechanical and Chemical Engineering Constrained Problem

5.1 Boiler-turbo generator system optimization

This problem involves a power system using steam generated from burning wood pulp to supply a small power house. A pair of turbo generators are used to generate electricity. The first turbine is of a double extraction type which has a pair of intermediate streams that leave at 195 psi and 62 psi. The resulting maximum capacity for generation and minimum load are respectively 6250 kW and 2500 kW. In the last phase, a condensate is produced which then serves as feed water for the boiler. The second turbine is of a single extraction type and has one intermediate stream at 195 psi along with an exit stream that leaves at 62 psi. This turbine does not produce any condensate, while the maximum capacity for generation and the minimum load are respectively 9000 kW and 3000 kW. Greater efficiency is offered by the first turbine as a consequence of the release of energy via steam condensation, although its power output is lower than that of the second turbine. Any excess steam produced can bypass the two turbines and pass directly to the two levels of steam via the pressure-reducing valves which can be seen in Figure 9.

Figure 9 presents a system that is suitable for modeling as constraints in combination with an objective function. The aim is to lower the system operating cost to a minimum by the optimal selection of power generation or purchase and steam flow rates while considering the various requirements and limitations imposed by the system (48).

I_i indicates the inlet flow rate i [lb$_m$/h] of the turbine, while HE_i and LE_i are the exit flow rates from the turbine i to the 195 psi header and the 62 psi header [lb$_m$/h] respectively. The condensate flow rate for Turbine 1 is given by C [lb$_m$/h], while P_i represents the power output i [kW]. BF_1 and BF_2 show the respective bypass flow rates from 635 psi to the 195 psi header and 195 psi to the 62 psi header [lb$_m$/h]. HPS, MPS, and LPS are the flow rates through the 635, 195, and 62 psi headers [lb$_m$/h] respectively. PP represents purchased power [kW], while EP indicates excess power [kW] (calculated as the difference between PP and base power). Finally, PRV represents the pressure-reducing valve.

The hourly operational cost for the system is indicated as (48):

Minimize: $f = 0.00261HPS + 0.0239PP + 0.00983EP$

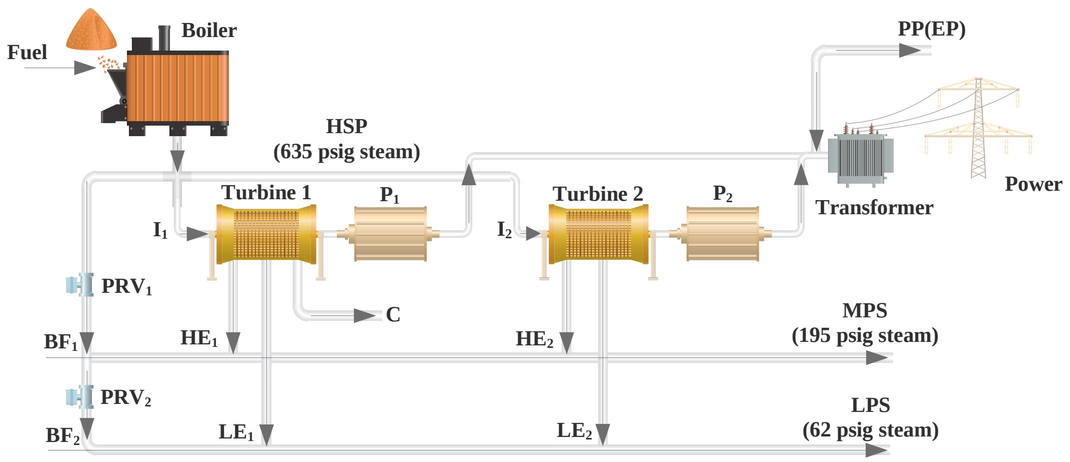


Figure 9 Boiler/turbo-generator process

In this problem, there are a total of 16 continuous independent variables. The constraints are shown in Table 7, while the decision variables can be seen in Table 8, which also summarizes their upper and lower bounds.

Table 7 Constraints applied in the problem of optimizing the boiler-turbo generator system (48)

Turbine 1	Turbine 2
$P_1 \le 6250$	$P_2 \le 9000$
$P_1 \ge 2500$	$P_2 \ge 3000$
$HE_1 \le 192,000$	$I_2 \le 244,000$
$C \le 62,000$	$LE_2 \le 142,000$
$I_1 - HE_1 \le 132,000$	

Table 7 (Countinued)

Material balances	Power purchased
$HPS - I_1 - I_2 - BF_1 = 0$	$EP + PP \ge 12,000$
$I_1 + I_2 + BF_1 - C - MPS - LPS = 0$	**Demand**
$I_1 - HE_1 - LE_1 - C = 0$	$MPS \ge 271,536$
$I_2 - HE_2 - LE_2 = 0$	$LPS \ge 100,623$
$HE_1 + HE_2 + BF_1 - BF_2 - MPS = 0$	
$LE_1 + LE_2 + BF_2 - LPS = 0$	
Energy balances	
$1359.8I_1 - 1267.8HE_1 - 1251.4LE_1 - 192C - 3413P_1 = 0$	
$1359.8I_2 - 1267.8HE_1 - 1251.4LE_2 - 3413P_2 = 0$	

The solutions for this optimization problem can be seen in Table 9. They were obtained via the previously described optimization methods. The set population is set to be 50 for 1,000 iterations for every algorithm. The bold figures indicate the best performance values. A penalty function was applied in order to address any violations of the constraints which were imposed. Table 8 shows that the same global optimum values were reached by MPA and SMA, at 1268.75, while some of the other algorithms, including HGSO, SOA, MA, and TSA, produced values near the global optimum.

Table 8 Decision variables applied in the problem of optimizing the boiler-turbo generator system (48)

Decision variable	Lower bound	Upper bound
I_1	100000	150000
I_2	200000	250000
HE_1	120000	192000
HE_2	100000	150000
LE_1	0	0
LE_2	100000	150000
C	8000	62000
BF_1	0	0
BF_2	0	0
HPS	380329	380329
MPS	200000	300000
LPS	100000	200000
P_1	2500	6250
P_2	7061	7061
PP	10000	12000
EP	760	1000

SMA and MPA achieved the lowest values for the mean, while SMA's lowest Stdev. occurred. The respective values for I_1, I_2, HE_1, HE_2, LE_1, LE_2, C, BF_1, BF_2, HPS, MPS, LPS, P_1, P_2, PP, and EP at the optimum reached by SMA were 136329, 244000, 128158, 143377, 0, 100623, 8170, 0, 0, 380329, 271536, 100623, 6250, 7061, 11239, and 761.

Table 9 Statistical result of operating cost of example using different algorithms

Algorithm	Best value	Mean	Stdev.
BAS	1276.308	1280.066	4.357
ASO	1269.505	1270.923	1.318
HGSO	1268.800	1279.336	8.429
SFO	1271.474	1274.730	2.604
SOA	1268.770	1269.014	0.216
MA	1268.983	1269.861	0.661
MPA	1268.751	1268.756	0.006
PPA	1271.805	1272.883	1.068
SMA	1268.752	126.755	0.004
TSA	1268.792	1268.888	0.143

5.2 Design of a piston lever

This particular design problem initially appeared in (49). The aim is to determine the location of the piston components given as H, B, D, and X by setting the volume of oil to a minimum while the piston lever is raised to 45° as Figure 10 indicates. The following objective function can be stated for this problem:

Minimize:

$$f(H,B,D,X) = \frac{1}{4}\pi D^2 (L_2 - L_1)$$

Subject to:

$$g_1 = QL\cos\theta - RF \le 0 \ \text{ at } \ \theta = 45^o$$

$$g_2 = Q(L - X) - M_{\max} \le 0$$

$$g_3 = 1.2(L_2 - L_1) - L_1 \le 0$$

$$g_4 = \frac{D}{2} - B \le 0$$

Where

$$R = \frac{\left| -X(X\sin\theta + H) + H(B - X\cos\theta) \right|}{\sqrt{(X-B)^2 + H^2}}$$

$$F = \frac{\pi P D^2}{4}$$

$$L_1 = \sqrt{(X-B)^2 + H^2}$$

$$L_2 = \sqrt{(X\sin 45 + H)^2 + (B - X\cos 45)^2}$$

where P represents the payload (10,000 lbs), L is the lever (240 in), oil pressure is 1,500 psi, and the lever has a maximum permissible bending moment of 6 max M = 1.8x10 lbs in. Several inequality constraints are applied. The calculations also take into consideration the equilibrium force, the minimum piston stroke, the lever's maximum bending moment, and the geometrical conditions.

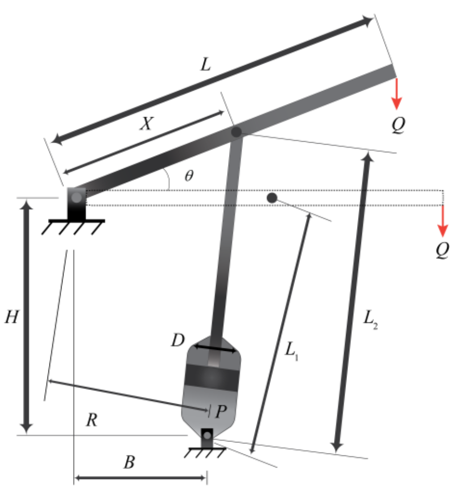


Figure 10 Piston lever design problem

In order to investigate the ten algorithms, a population of size 50 is used, while 100 is set as the maximum number of iterations. The fitness value is fixed at iteration zero to achieve equality during the testing process.

Table 10 presents a summary of the performance of the ten algorithms. For the problem under investigation the SMA, MPA, and PPA algorithms produced the best results in terms of best, mean, and Stdev. values.

In Figure 11 the function values for the piston lever design problem are presented in association with the number of iterations. It is apparent that four algorithms in particular can reach a global solution while offering a fast rate of convergence. These are MPA, SMA, PPA, and HGSO. Among the other algorithms tested, BAS, SOA, SFO, MA, and TSA exhibited slow convergence rates and were unable to avoid local optima, thereby failing to achieve the global solution for the piston lever design problem.

Table 10 Statistical comparison of the different optimization outcomes in the piston lever design problem

Algorithm	Best value	Mean	Stdev.
BAS	120.241	133.372	13.871
ASO	15.283	23.713	14.577
HGSO	9.5874	10.4608	1.359
SFO	12.5863	16.775	8.951
SOA	25.550	42.092	12.803
MA	86.734	90.289	3.322
MPA	8.4268	8.433	0.009
PPA	8.4271	8.428	0.001
SMA	8.4267	8.433	0.009
TSA	50.787	56.798	4.967

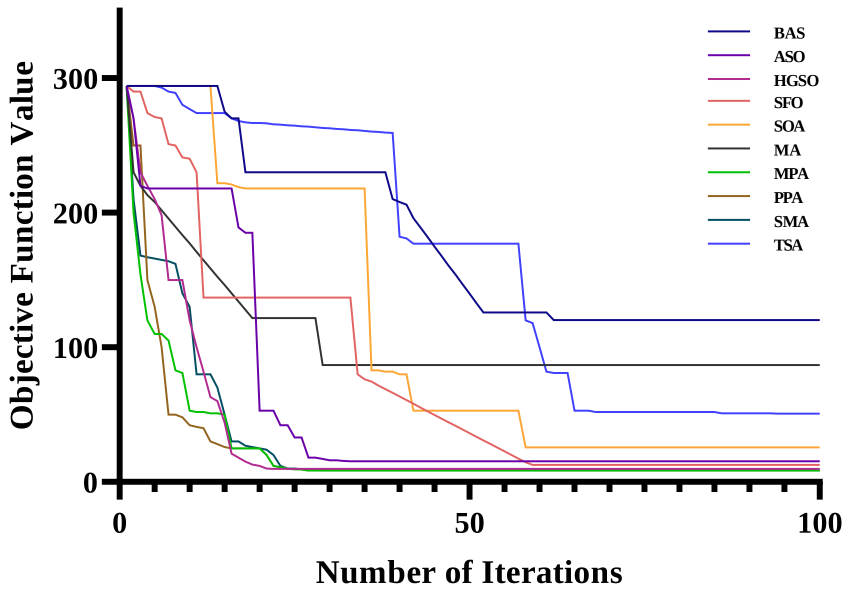


Figure 11 Function values plotted against the number of iterations in the piston lever design problem

6. Conclusion

Metaheuristic optimization algorithms can provide solutions to a wide range of benchmark function problems of different types, whether unimodal, multimodal, continuous, discontinuous and so forth. This ability stems from the inherent advantages the algorithms offer in terms of diversification and exploitation. These factors ultimately explain the capacity of an algorithm to reach a global maximum under different circumstances. This research paper examines ten metaheuristic optimization algorithms introduced during the period from 2018 to 2020, tested them with dimensions set to 100 on 19 benchmark functions, and assessed their problem-solving capacity in scenarios involving constrained chemical and mechanical engineering problems.

From the results of performance testing, it can be observed that each algorithm has different solution advantages and disadvantages based on global search performance, the capability of exploitation and exploration analysis, convergence rate, robustness, and computation time, which can be validated and certified from the Friedman Aligned Ranks test and Quade Ranks test ranking. Both rank tests confirm that the algorithms have different searching approaches to solve with significance at 95%.

In this paper, conclusions confirm that these ten metaheuristic algorithms help solve both unconstrained and constrained mathematical model problems but can still eliminate problems that interfere with the global performance test. Therefore, in the future, metaheuristic optimization algorithms will still need to be developed and improved, with proposed new hybrid algorithms to reach complete solutions to all mathematical problems and solve such problems in multiple scientific and engineering fields involving complex real-world, large-scale challenges for industrial success.

Acknowledgements
This research is supported by personal finance.

Declaration of conflicting interests
The authors declared that they have no conflicts of interest in the research, authorship, and this article's publication.

Appendix
Source code
1. Beetle Antennae Search (BAS)
BAS-Beetle Antennae Search Algorithm for Optimization - File Exchange - MATLAB Central (mathworks.com)
2. Atom Search Optimization (ASO)
Atom Search Optimization (ASO) Algorithm - File Exchange - MATLAB Central (mathworks.com)
3. Henry Gas Solubility Optimization (HGSO)
Henry gas solubility optimization - File Exchange - MATLAB Central (mathworks.com)
4. Sunflower Optimization (SFO)
SUNFLOWER OPTIMIZATION (SFO) ALGORITHM - File Exchange - MATLAB Central (mathworks.com)
5. Seagull Optimization Algorithm (SOA)
Seagull Optimization Algorithm (SOA) - File Exchange - MATLAB Central (mathworks.com)
6. Mayfly Algorithm (MA)
A mayfly optimization algorithm - File Exchange - MATLAB Central (mathworks.com)
7. Marine Predators Algorithm (MPA)
Marine Predators Algorithm (MPA) - File Exchange - MATLAB Central (mathworks.com)
8. Parasitism Predation Algorithm (PPA)
Parasitism – Predation algorithm (PPA) - File Exchange - MATLAB Central (mathworks.com)
9. Slime Mould Algorithm (SMA)
Slime Mould Algorithm (SMA): A Method for Optimization - File Exchange - MATLAB Central (mathworks.com)
10. Tunicate Swarm Algorithm (TSA)
Tunicate Swarm Algorithm (TSA) - File Exchange - MATLAB Central (mathworks.com)

References

1. Abbassi R, Abbassi A, Heidari AA, Mirjalili S. An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. Energ Convers Manage. 2019 Jan 1;179:362-72.
2. Faris H, Ala'M AZ, Heidari AA, Aljarah I, Mafarja M, Hassonah MA, Fujita H. An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. Inform Fusion. 2019 Aug 1;48:67-83.
3. Wu G, Pedrycz W, Suganthan PN, Mallipeddi R. A variable reduction strategy for evolutionary algorithms handling equality constraints. Appl soft comput. 2015 Dec 1;37:774-86.
4. Alba E, Dorronsoro B. The exploration/ exploitation tradeoff in dynamic cellular genetic algorithms. IEEE T Evolut Comput. 2005 Apr 4;9(2):126-42.
5. Lozano M, García-Martínez C. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. Comput Oper Res. 2010 Mar 1;37(3):481-97.
6. Ab Wahab MN, Nefti-Meziani S, Atyabi A. A comprehensive review of swarm optimization algorithms. PloS one. 2015 May 18;10(5):e0122827.
7. Yang XS, editor. Nature-inspired algorithms and applied optimization. Springer; 2017 Oct 8.
8. Cho HJ, Ahmed F, Kim TY, Kim BS, Yeo YK. A comparative study of teaching-learning-self-study algorithms on benchmark function optimization. Korean J Chem Eng. 2017 Mar 1;34(3):628-41.
9. Beheshti Z, Shamsuddin SM. A review of population-based meta-heuristic algorithms. Int. J. Adv. Soft Comput. Appl. 2013 Mar 1;5(1):1-35.
10. Bozorg-Haddad O. Advanced optimization by nature-inspired algorithms. Springer Nature Singapore Pte Ltd.; 2018.
11. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016 May 1;95:51-67.
12. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press; 1992.
13. Holland J. Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control and artificial intelligence. 1975.
14. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim. 1997 Dec;11(4):341-59.
15. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department; 2005 Oct.
16. Kennedy J, Eberhart R. Particle swarm optimization. InProceedings of ICNN'95-international conference on neural networks 1995 Nov 27 (Vol. 4, pp. 1942-1948). IEEE.

17. Yang XS. Flower pollination algorithm for global optimization. InInternational conference on unconventional computing and natural computation 2012 Sep 3 (pp. 240-249). Springer, Berlin, Heidelberg.
18. Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct. 2016 Jun 1;169:1-2.
19. Gandomi AH, Yang XS, Alavi AH, Talatahari S. Bat algorithm for constrained optimization tasks. Neural Comput & Applic. 2013 May;22(6):1239-55.
20. Simon D. Biogeography-based optimization. IEEE transactions on evolutionary computation. 2008 Mar 21;12(6):702-13.
21. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. simulation. 2001 Feb;76(2):60-8.
22. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Design. 2011 Mar 1;43(3):303-15.
23. Savsani P, Savsani V. Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model. 2016 Mar 1;40(5-6):3951-78.
24. Van Laarhoven PJ, Aarts EH. Simulated annealing. InSimulated annealing: Theory and applications 1987 (pp. 7-15). Springer, Dordrecht.
25. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct. 2012 Nov 1;110:151-66.
26. Salimi H. Stochastic fractal search: a powerful metaheuristic algorithm. Knowledge-Based Systems. 2015 Feb 1;75:1-8.
27. Yang XS. Nature-inspired metaheuristic algorithms. Luniver press; 2010.
28. Sukpancharoen S, Srinophakun TR, Aungkulanon P. Grey Wolf Optimizer (GWO) with Multi-Objective Optimization for Biodiesel Production from Waste Cooking Oil Using Central Composite Design (CCD). International Journal of Mechanical Engineering and Robotics Research. 2020 Aug;9(8).
29. Sukpancharoen S, Srinophakun TR, Hirunlabh J, Rattanachoung N. The Use of Factorial Design to Improve a Harmony Search Algorithm to Synthesize Heat-Integrated Distillation Sequences. InKey Engineering Materials 2018 (Vol. 777, pp. 218-225). Trans Tech Publications Ltd.
30. Tariq F, Alelyani S, Abbas G, Qahmash A, Hussain MR. Solving Renewables-Integrated Economic Load Dispatch Problem by Variant of Metaheuristic Bat-Inspired Algorithm. Energies. 2020 Jan;13(23):6225.
31. Jiang X, Li S. Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization. arXiv preprint arXiv:1711.02395. 2017 Nov 7.
32. Zhao W, Wang L, Zhang Z. A novel atom search optimization for dispersion coefficient estimation in groundwater. Future Gener Comp Sy. 2019 Feb 1;91:601-10.
33. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: A novel physics-based algorithm. Future Gener Comp Sy. 2019 Dec 1;101:646-67.
34. Gomes GF, da Cunha SS, Ancelotti AC. A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Eng Comput. 2019 Apr;35(2):619-26.
35. Dhiman G, Kumar V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowl-Based Syst. 2019 Feb 1;165:169-96.
36. Zervoudakis K, Tsafarakis S. A mayfly optimization algorithm. Comput Ind Eng. 2020 Jul 1;145:106559.
37. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. Marine Predators Algorithm: A nature-inspired metaheuristic. Expert Syst Appl. 2020 Aug 15;152:113377.
38. Mohamed AA, Hassan SA, Hemeida AM, Alkhalaf S, Mahmoud MM, Eldin AM. Parasitism–Predation algorithm (PPA): A novel approach for feature selection. Ain Shams Engineering Journal. 2020 Jun 1;11(2):293-308.
39. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: A new method for stochastic optimization. Future Gener Comp Sy. 2020 Oct 1;111:300-23.
40. Kaur S, Awasthi LK, Sangal AL, Dhiman G. Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng Appl Artif Intel. 2020 Apr 1;90:103541.
41. Lennard-Jones JE. On the determination of molecular fields. II. From the equation of state of gas. Proc. Roy. Soc. A. 1924;106:463-77.
42. Macdonald SM, Mason CF. Predation of migrant birds by gulls. Brit Birds. 1973;66:361-3.
43. Howard FL. The life history of Physarum polycephalum. Am J Bot. 1931 Feb 1:116-33.
44. Kessler D. Plasmodial structure and motility. Cell biology of Physarum and Didymium/edited by Henry C. Aldrich, John W. Daniel. 1982.
45. Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE T Evolut Comput. 1997 Apr;1(1):67-82.
46. Jamil M, Yang XS. A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation. 2013 Jan 1;4(2):150-94.

47. Salthouse TA. The complexity of age× complexity functions: Comment on Charness and Campbell (1988).
48. Edgar TF, Himmelblau DM, Lasdon LS. Optimization of chemical processes. 2001.
49. Vanderplaats GN. Design Optimization Tools (DOT) Users Manual, Version 4.20. VR&D, Colorado. 1995.