SAKARYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University |
http://www.saujs.sakarya.edu.tr/en/

Title: A Novel Hybrid Algorithm: Sine Cosine Harmony Search Algorithm for Global Optimization

Authors: Bahadur ALIZADA

Recieved: 2020-07-16 14:16:12

Accepted: 2021-03-17 12:44:43

Article Type: Research Article

Volume: 25
Issue: 2
Month: April
Year: 2021
Pages: 513-529

How to cite
Bahadur ALIZADA; (2021), A Novel Hybrid Algorithm: Sine Cosine Harmony Search Algorithm for Global Optimization. Sakarya University Journal of Science, 25(2), 513-529, DOI: https://doi.org/10.16984/saufenbilder.770367
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/60672/770367

Sakarya University Journal of Science 25(2), 513-529, 2021

A Novel Hybrid Algorithm: Sine Cosine Harmony Search Algorithm for Global Optimization

Bahadur ALIZADA*[1]

Abstract

The study is about the new hybrid optimization algorithm called Sine Cosine Harmony Search (SCHS). SCHS was created by combining the features of Sine Cosine Algorithm (SCA) and Harmony Search (HS) meta-heuristic algorithms. In the research, the stage of creating the model was confirmed by experiments. For this, it has been tested using global optimization techniques. Benchmark Functions (BF) with different parametric properties were used for testing time. For performance measurement, comparisons were made with various optimization algorithms. The improvement of SCHS on the basis of exploitation and exploration shows that the algorithm is competitive.

Keywords: Sine Cosine Algorithm, Harmony Search, Benchmark Functions, Meta-heuristic algorithms

1. INTRODUCTION

Meta-heuristic optimization algorithms are too many to follow. A single algorithm may not always be successful in various problems [1]. Therefore, new optimization algorithms are derived. Apart from this, sometimes hybrid techniques are obtained by combining existing algorithms. Hybrids show more successful results compared to the algorithms they derive from. They achieve this by improving their exploration and exploitation phases. Meta-heuristics are inspired by nature, it has a simple and useful structure, and it can be adapted to real life problems, increasing the interest of scientists in this field.

Some of the most common meta-heuristic optimization techniques and hybrid studies related to them are shown in Table 1.

Table 1
Some popular meta-heuristic optimization algorithms and their hybrids

Algorithms	Hybrids
Genetic Algorithm (GA) [2]	Taguchi – GA [3] GA – Particle Swarm Optimization [4] Grouping – GA [5]
Artificial Bee Colony (ABC) [6]	ABC – Levenberq Marquardt Algorithm [7] ABC – GA [8]

* Corresponding Author: bahadur_alizade@yahoo.com
[1] Erciyes University, Department of Computer Engineering, Kayseri, Turkey, ORCID: https://orcid.org/0000-0001-6587-4057

	Simplex – ABC [9]
Harmony Search (HS) [10]	Bat Algorithm – HS [11] Taguchi – HS [12] HS – ABC [13]
Sine Cosine Algorithm (SCA) [14]	SCA – Crow Search Algorithm [15] SCA – Whale Optimization Algorithm [16] Grey Wolf Optimizer – SCA [17]
Multi-Verse Optimizer (MVO) [18]	PSO –MVO [19] Electricity Generation System – MVO [20] Self-Adaptive – MVO [21]

In this study, 7 different models were first created based on the SCA's position update and based on HS. By comparing these models, final models were decided. Then, hybrid models were evaluated by making comparison with some known optimization problems. Benchmark

Functions (BF) [22-27], which are the most frequently encountered in the literature at the time of comparison, were used. BF are listed in Table 2. In the table, functions, dimension (Dim), range (Lower Bound (LB) - Upper Bound (UB)), the smallest value (f_{min}) that the function takes.

Information about the algorithms inspired by the work done and the models created are shown in part 2. Comparison in various aspects is given in section 3. General interpretation is given in section 4 (conclusion).

Table 2
Benchmark functions list

Function	**Dim**	**[LB UB]**	f_{min}
Unimodal Benchmark Functions			
BF$_1$=$\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
BF$_2$=$\sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
BF$_3$=$\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	30	[-100,100]	0
BF$_4$=$max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[-100,100]	0
BF$_5$= $\sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
BF$_6$=$\sum_{i=1}^{n}([x_i + 0.5])^2$	30	[-100,100]	0
BF$_7$=$\sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-1.28,1.28]	0
Multimodal Benchmark Functions			
BF$_8$=$\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.9829× 30
BF$_9$=$\sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
BF$_{10}$=$-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	[-32,32]	0
BF$_{11}$=$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
BF$_{12}$=$\frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
BF$_{13}$= $0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	30	[-50,50]	0
Fixed-dimension Multimodal Benchmark Functions			

$BF_{14}=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$	2	[-65,65]	1
$BF_{15}=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
$BF_{16}=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
$BF_{17}=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5,5]	0.398
$BF_{18}=[1+(x_1+x_2+1)^2(19-14x_1+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3
$BF_{19}=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1,3]	-3.86
$BF_{20}=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0,1]	-3.32
$BF_{21}=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
$BF_{22}=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
$BF_{23}=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.5363

2. RELATED WORKS

2.1. Harmony search algorithm

HS is a human-based algorithm inspired by the improvised music stage. Musicians improvise the sounds of the instruments seeking the most appropriate fit [10].

The properties of the algorithm are represented by its control parameters. Proper adjustment of these control parameters affects the results of the algorithm. Control parameters are described in Table 3.

Table 3
HS control parameters

Name of Control Parameter	Definition
Maximum Number of Iteration (**MaxIt**)	It shows the number of loops and is often the stopping criterion [28].
Harmony Memory Size (**HMS**)	The number of solution vectors represents the population size [28].
Harmony Memory Consideration Rate (**HMCR**)	It represents the probability of choosing one of the old values in the population [28].
Pitch Adjustment Rate (**PAR**)	It helps to produce different solutions and corresponds to changing pitch adjustment frequencies in music. [10].
Fret Width (**FW**) or Bandwidth (**BW**)	It is a design variable affecting exploration and exploitation [26].

The steps of the algorithm are shown as below with pseudocode.

Step 1: Assign values of control parameters.

Step 2: Create random population (HM) and calculate objective function.

Step 3: Improvise a new harmony ($x_{new}(j)$)

REPEAT

Step 4:

if_1 rand_1 < HMCR

$x_{new}(j)= x_{\propto}(j)$

If_2 rand_2 < PAR

$x_{new}(j)= x_{new}(j) \pm rand() \times BW$

end_if_2

else

$x_{new}(j)= LB(j)+rand()\times(UB(j)-LB(j))$

end_if_1

Where $\alpha \in (1,2,\ldots,HMS), x_{\alpha}(j)$ is selected from HM

Step 5: Update the HM

UNTIL the stop criterion is satisfied (MaxIt).

j represents the instant iteration. Here ±rand() shows uniform distribution and this shows that it changes in the range of [-1,1].

2.2. Sine cosine algorithm

SCA is a meta-heuristic optimization technique graphically inspired by sine and cosine motions. The algorithm creates random solutions first, then determines the best solution with the fitness function [14]. According to this solution, other elements in the population update their position. The update is done with the following function:

$$X_i^{t+1} = \begin{cases} X_{\mathrm{I}}^T + r_1 \times \sin(r_2) \times \left|r_3 P_i^t - X_{\mathrm{I}}^T\right|, r_4 < 0.5 \\ X_{\mathrm{I}}^T + r_1 \times \cos(r_2) \times \left|r_3 P_i^t - X_{\mathrm{I}}^T\right|, r_4 \geq 0.5 \end{cases} \tag{1}$$

X_{I}^T is the current position. P_i^t represents the best solution at the moment. r_1 is the random number that represents the direction of the update calculated by the equation $r_1 = a - t\frac{a}{T}$. Here, a represents the constant in the range [0.2], t is the beginning, and T is the maximum iteration. r_2, r_3, r_4 are random numbers that define the update distance in the range [0, 2π], the weight of the target in the range [0, 2], the balance between the sine and cosine movements ranging from [0, 1].

2.3. The proposed hybrid models

While performing hybridization, the following formulas were obtained by inspiring the position update model in SCA:

$$x_{new}(j) = \begin{cases} r_1 \times \sin(r_2) \times r_3 \times x_\alpha(j), r_4 < 0.5 \\ r_1 \times \cos(r_2) \times r_3 \times x_\alpha(j), \quad r_4 \geq 0.5 \end{cases} \tag{2}$$

$$x_{new}(j) = \begin{cases} x_{new}(j) \pm r_1 \times \sin(r_2) \times r_3 \times rand(\,) \times BW, \\ \qquad r_4 < 0.5 \\ x_{new}(j) \pm r_1 \times \cos(r_2) \times r_3 \times rand(\,) \times BW, \\ \qquad r_4 \geq 0.5 \end{cases} \tag{3}$$

$$x_{new}(j) = \begin{cases} LB(j) + r_1 \times \sin(r_2) \times |r_3 \times LB(J) - UB(J)|, \\ \qquad r_4 < 0.5 \\ LB(j) + r_1 \times \cos(r_2) \times |r_3 \times LB(J) - UB(J)|, \\ \qquad r_4 \geq 0.5 \end{cases} \tag{4}$$

Details about whether the formulas are applied to the models are shown in Table 4 in an explanatory way. In the table, Sine Cosine Harmony Search (SCHS) hybrid models are numbered respectively according to the above formulas.

Table 4
Using formulas in hybrid models

Used Formulas	**SCHS1**	**SCHS2**	**SCHS3**	**SCHS4**	**SCHS5**	**SCHS6**	**SCHS7**
(2)	+	-	-	+	+	-	+
(3)	-	+	-	+	-	+	+
(4)	-	-	+	-	+	+	+

SCHS4 and SCHS7 have been chosen as the final models since they have had close success. These comparisons are more clearly stated in the results section.

2.3.1. SCHS4 pseudocode

r_5 in the model are random numbers in the range [0,1] equal to r_4.

Step 1: Assign values of control parameters.

Step 2: Create random population (HM) and calculate objective function.

Step 3: Improvise a new harmony ($x_{new}(j)$)

REPEAT

Step 4:

if_1 rand_1 < HMCR

if_2 $r_4 < 0.5$

$x_{new}(j)$= $r_1 \times \sin(r_2) \times r_3 \times x_\alpha(j)$

else

$x_{new}(j)$= $r_1 \times \cos(r_2) \times r_3 \times x_\alpha(j)$

end_if_2

If_3 rand_2 < PAR

if_4 $r_5 < 0.5$

$x_{new}(j) = x_{new}(j) \pm r_1 \times \sin(r_2) \times r_3 \times rand() \times BW$

else

$x_{new}(j) = x_{new}(j) \pm r_1 \times \cos(r_2) \times r_3 \times rand() \times BW$

end_if_4

end_if_3

else

$x_{new}(j)$= $LB(j) + rand() \times (UB(j) - LB(j))$

end_if_1

Where $\alpha \in (1,2,\dots,HMS), x_\alpha(j)$ is selected from HM

Step 5: Update the HM

UNTIL the stop criterion is satisfied (MaxIt).

2.3.2. SCHS7 pseudocode

r_5 and r_6 in the model are random numbers in the range [0,1] equal to r_4.

Step 1: Assign values of control parameters.

Step 2: Create random population (HM) and calculate objective function.

Step 3: Improvise a new harmony ($x_{new}(j)$)

REPEAT

Step 4:

if_1 rand_1 < HMCR

if_2 $r_4 < 0.5$

$x_{new}(j)$= $r_1 \times \sin(r_2) \times r_3 \times x_\alpha(j)$

else

$x_{new}(j)$= $r_1 \times \cos(r_2) \times r_3 \times x_\alpha(j)$

end_if_2

If_3 rand_2 < PAR

if_4 $r_5 < 0.5$

$x_{new}(j) = x_{new}(j) \pm r_1 \times \sin(r_2) \times r_3 \times rand() \times BW$

else

$x_{new}(j) = x_{new}(j) \pm r_1 \times \cos(r_2) \times r_3 \times rand() \times BW$

end_if_4

end_if_3

else

if_5 $r_6 < 0.5$

$x_{new}(j)$= $LB(j) + r_1 \times \sin(r_2) \times |r_3 \times LB(J) - UB(J)|$

else

$x_{new}(j)$= $LB(j) + r_1 \times \cos(r_2) \times |r_3 \times LB(J) - UB(J)|$

end_if_5

end_if_1

Where $\alpha \in (1,2,\dots,HMS), x_\alpha(j)$ is selected from HM

Step 5: Update the HM

UNTIL the stop criterion is satisfied (MaxIt).

3. RESULTS AND DISCUSSIONS

Several test cases should be used to demonstrate the performance of optimization algorithms. In this way, various tests are performed to prove that satisfactory results are not accidental. The high number of tests makes the performance of the algorithm more evident.

In the study, seven different models were created first. These models were adapted to 23 familiar BF and compared. These BFs are chosen from two kinds of test functions: unimodal and multimodal functions. Unimodal is locally optimistic and has only one global optima, which tests the speed of convergence and exploitation capability. Multimodal has one global and more than one local solution. This evaluates exploration abilities and avoidance of local consequences.

To solve the specified test functions, the population size, which is the common parameter values in all algorithms, is accepted as 30 and Maxit 500. The codes were run 30 times for each problem. The reason for the high number of run is chosen because the algorithms have a stochastic structure, so the results obtained at once are not very reliable. According to the results, best min, best max, average, standard deviation (S.D.) values were compared. Special control parameters in HS and hybrid models are set as HMCR 0.9, PAR 0.25, and BW 0.01. These specific parametric values are recommended in many studies [28,30]. Comparisons are given in Table 5. In the table, the minimum values were bold by making a comparison over the average.

Table 5
Comparison for the selection of hybrid models

BF1

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	3892,989	12803,9546	7322,6655	2191,8977
SCA	0,097509	308,8207	28,0842	58,0673
SCHS1	5,477E-07	4,1891E-05	1,5360E-05	1,0416E-05
SCHS2	4452,6873	11122,7397	7753,1206	1529,0288
SCHS3	19,1734	417,0179	124,0915	94,7923
SCHS4	**0**	**7,5319E-11**	**2,8383E-12**	**1,3496E-11**
SCHS5	3,9349E-10	5,0071E-05	1,6579E-05	1,3394E-05
SCHS6	24,3369	377,591	111,3353	75,6256
SCHS7	0	3,5452E-10	2,1837E-11	6,8979E-11

BF2

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	16,7249	38,1766	27,9543	5,362864
SCA	6,6656E-05	0,23871	0,030136	0,046223
SCHS1	0,001001	0,012557	0,006608	0,002498
SCHS2	19,1466	35,484	25,8691	4,109259
SCHS3	0,76331	7,0095	3,1899	1,649256
SCHS4	**0**	**1,715E-05**	**1,0579E-06**	**3,2487E-06**
SCHS5	3,9566E-05	0,013136	0,006604969	0,003462
SCHS6	0,58747	5,5668	3,030047	1,227704
SCHS7	0	3,7155E-05	4,3374E-06	9,7959E-06

BF3

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	42778,3951	72911,4585	54844,2568	8007,6166
SCA	1,098E-08	0,10377	0,011794	0,030333
SCHS1	1,5754E-06	0,00017371	7,9154E-05	4,5904E-05
SCHS2	41108,2485	72675,6962	56656,2683	7128,3273
SCHS3	4798,9901	44242,8833	25287,4192	9630,3654

SCHS4	**0**	**1,084E-09**	**4,0015E-11**	**1,9413E-10**
SCHS5	4,951E-09	0,00021399	6,7319E-05	4,8057E-05
SCHS6	9647,2455	41720,7357	23295,7146	8338,3719
SCHS7	0	1,0988E-09	1,1542E-10	2,9091E-10

BF4

Algorithms	Best min	Best Max	Average	S.D.
HS	46,8093	82,3186	66,8731	7,547957
SCA	5,1109E-06	0,01669	0,001494	0,003454
SCHS1	0,00068455	0,0045149	0,002688	0,001014
SCHS2	52,8333	78,3761	66,9844	5,742987
SCHS3	10,3387	23,0841	15,9445	2,717738
SCHS4	**0**	**6,5301E-06**	**3,9403E-07**	**1,2145E-06**
SCHS5	9,6349E-06	0,004835	0,002554	0,001107
SCHS6	10,6076	24,1984	15,9419	3,495881
SCHS7	0	9,7551E-06	1,0691E-06	2,3018E-06

BF5

Algorithms	Best min	Best Max	Average	S.D.
HS	2217460,2225	15905922,3856	7231848,7423	3145552,1831
SCA	**6,3911**	**8,8004**	**7,56233**	**0,531098**
SCHS1	28,717	28,9404	28,8872	0,063852
SCHS2	2788243,1984	19547082,5754	9945070,8493	4155608,2303
SCHS3	529,1656	18757,2572	4904,4418	4503,9617
SCHS4	28,7464	28,9529	28,9279	0,036041
SCHS5	28,7108	28,9453	28,8919	0,050414
SCHS6	498,4125	35196,7695	6035,5670	7178,0631
SCHS7	28,7476	28,9496	28,9103	0,040447

BF6

Algorithms	Best min	Best Max	Average	S.D.
HS	2518,511	11747,9984	7380,9895	2148,0398
SCA	**0,16086**	**0,79996**	**0,437673**	**0,154097**
SCHS1	5,5267	6,8412	6,247517	0,305211
SCHS2	4149,0916	12571,9666	8319,6886	2215,7416
SCHS3	15,2097	407,4046	119,2951	100,9400
SCHS4	5,7956	6,7532	6,34424	0,224476
SCHS5	5,1286	6,5285	6,02554	0,340724
SCHS6	17,7998	463,9078	120,0234	100,8902
SCHS7	4,7331	6,5591	5,984907	0,397911

BF7

Algorithms	Best min	Best Max	Average	S.D.
HS	1,2254	10,5976	5,466523	2,268252
SCA	0,00016383	0,012511	0,003751	0,003455
SCHS1	3,1223E-05	0,0086856	0,002243	0,001928
SCHS2	1,5591	9,804	5,294937	2,064621
SCHS3	0,030193	0,19244	0,089510	0,034497
SCHS4	**3,1062E-05**	**0,0058526**	**0,001664**	**0,001558**
SCHS5	0,00041785	0,0064498	0,002067	0,001534
SCHS6	0,058142	0,23677	0,097752	0,039159
SCHS7	0,00036513	0,0051217	0,001989	0,001330

BF8

Algorithms	Best min	Best Max	Average	S.D.
HS	-10153,2716	-7913,6101	-9075,1808	487,9375
SCA	-2444,2061	-1854,9322	-2144,9223	148,0728
SCHS1	-7182,2703	-4692,1159	-5937,4085	694,3250
SCHS2	-10099,8634	-8485,8737	-9186,2638	400,7526
SCHS3	-12555,5763	-11930,8468	-12461,5521	129,2061

SCHS4	-6972,4301	-4555,3865	-5851,5905	651,2755
SCHS5	-8035,435	-5686,84	-6762,1025	738,6810
SCHS6	**-12559,8936**	**-12258,4787**	**-12481,0026**	**66,2260**
SCHS7	-9277,3491	-5261,9817	-6943,4675	996,4264

BF9

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	64,1678	152,0628	102,0298	18,1078
SCA	0,1658	143,3196	41,9869	30,6500
SCHS1	0,00010829	0,0079029	0,00303246	0,00199915
SCHS2	78,3988	145,3067	106,1607	14,2405
SCHS3	2,6974	42,5112	15,6449	9,53840557
SCHS4	**0**	**1,5035E-08**	**5,6241E-10**	**2,6942E-09**
SCHS5	6,5127E-08	0,0066272	0,00270065	0,00202066
SCHS6	1,8938	30,9549	13,8947	7,67023005
SCHS7	0	3,1922E-08	2,7743E-09	7,22261E-09

BF10

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	15,3455	19,913	19,4762	0,82737886
SCA	5,1076E-10	0,0099231	0,00039209	0,00179942
SCHS1	0,00054147	0,0046738	0,00268940	0,00102041
SCHS2	15,7277	19,9079	19,3959	0,80679996
SCHS3	1,8132	6,3888	3,87572667	1,19379534
SCHS4	**8,8818E-16**	**6,4032E-06**	**3,6103E-07**	**1,1853E-06**
SCHS5	7,4968E-06	0,004716	0,00266100	0,00131918
SCHS6	1,346	7,5954	3,80907667	1,23364947
SCHS7	8,8818E-16	1,3753E-05	1,3807E-06	3,1348E-06

BF11

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	41,6513	117,2951	69,6358	21,4391
SCA	0,51441	3,7794	1,170392	0,575652
SCHS1	1,2188E-08	1,519E-06	6,0945E-07	4,2816E-07
SCHS2	51,6759	103,0083	70,4829	13,0294
SCHS3	1,1777	4,3834	2,112377	0,864032
SCHS4	**0**	**3,1213E-12**	**1,4386E-13**	**5,6899E-13**
SCHS5	3,6235E-11	2,2418E-06	6,8218E-07	5,2416E-07
SCHS6	1,2281	6,2853	2,054643	0,955177
SCHS7	0	1,4267E-11	9,1580E-13	2,8707E-12

BF12

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	446715,7721	42139487,8243	7772057,7222	8393477,1462
SCA	**0,04303**	**0,24875**	**0,116879**	**0,050462**
SCHS1	0,68974	1,3094	0,931589	0,122832
SCHS2	235897,5488	56501724,1297	10991804,1448	11364757,8560
SCHS3	0,054152	9,4979	1,969113	2,056264
SCHS4	0,66057	1,3758	0,990964	0,195818
SCHS5	0,53711	1,0901	0,808253	0,137141
SCHS6	0,07123	9,4034	2,014328	2,181666
SCHS7	0,48584	1,1154	0,749709	0,158297

BF13

Algorithms	**Best min**	**Best Max**	**Average**	**S.D.**
HS	**0,10684**	**0,45856**	**0,306805**	**20940012,2331**
SCA	4333950,6429	110214270,9505	27811430,3828	0,078630
SCHS1	2,7701	2,9933	2,93936	0,051508
SCHS2	3578600,3414	169144889,1309	37779172,4047	35286497,9803
SCHS3	2,2758	42,0018	11,1463	9,580429

SCHS4	2,76	2,9945	2,920243	0,057321
SCHS5	2,0926	2,9937	2,51881	0,184007
SCHS6	1,1087	39,6832	13,2758	9,432823
SCHS7	1,8001	2,9204	2,42839	0,249008

BF14

Algorithms	Best min	Best Max	Average	S.D.
HS	0,998	18,3043	7,360953	6,025582
SCA	0,998	2,9821	1,595877	0,907521
SCHS1	0,99801	12,6705	6,494600	5,213908
SCHS2	0,998	25,6947	8,959773	7,330919
SCHS3	**0,998**	**6,9033**	**1,479651**	**1,369136**
SCHS4	0,99803	12,6705	7,527455	5,173333
SCHS5	0,998	12,6705	2,328715	3,002079
SCHS6	0,998	12,6705	2,167169	2,970434
SCHS7	0,998	3,008	1,73453	0,874200

BF15

Algorithms	Best min	Best Max	Average	S.D.
HS	0,00080187	0,028963	0,00853507	0,00899296
SCA	0,00031894	0,0016633	0,00098219	0,00040097
SCHS1	0,00040062	0,0011645	0,00079686	0,00015385
SCHS2	0,00080366	0,051455	0,00935059	0,01276088
SCHS3	0,00072934	0,020974	0,00404222	0,00649864
SCHS4	0,00035644	0,00087938	0,00075387	0,00012224
SCHS5	0,00034061	0,0010246	0,00072536	0,00013144
SCHS6	0,00077058	0,020916	0,00535980	0,00740826
SCHS7	**0,00034967**	**0,0015024**	**0,00071505**	**0,00022873**

BF16

Algorithms	Best min	Best Max	Average	S.D.
HS	**-1,0316**	**-1,0316**	**-1,0316**	**6,6614E-16**
SCA	-1,0316	-1,0313	-1,0316	7,1802E-05
SCHS1	-1,0316	-1,0248	-1,0310	0,001586
SCHS2	-1,0316	-0.21546	-0.9772	0.203581
SCHS3	**-1,0316**	**-1,0316**	**-1,0316**	**6,6614E-16**
SCHS4	-1,0316	-1,0124	-1,0305	0,003563
SCHS5	-1,0316	-1,0269	-1,0312	0,000987
SCHS6	**-1,0316**	**-1,0316**	**-1,0316**	**6,6614E-16**
SCHS7	-1,0316	-1,0258	-1,0312	0,001244

BF17

Algorithms	Best min	Best Max	Average	S.D.
HS	0,39789	0,53847	0,40274	0,02522
SCA	0,39792	0,40201	0,39881	0,00075
SCHS1	0,39789	0,40179	0,39872	0,00091
SCHS2	0,39789	0,40264	0,39805	0,00085
SCHS3	**0,39789**	**0,39789**	**0,39789**	**1,6653E-16**
SCHS4	0,39791	0,40695	0,39929	0,00201
SCHS5	0,3979	0,41079	0,39972	0,00302
SCHS6	**0,39789**	**0,39789**	**0,39789**	**1,6653E-16**
SCHS7	0,39791	0,40462	0,39911	0,00132

BF18

Algorithms	Best min	Best Max	Average	S.D.
HS	3	84,0105	33,7297	35,4572
SCA	**3**	**3,001**	**3,0001**	**0,00025**
SCHS1	3,0003	30,005	6,21685	7,69182
SCHS2	3	30	6.6	9.17824
SCHS3	3	30	11,1	12,3730

SCHS4	3,0002	30,0267	5,89671	6,79378
SCHS5	3	3,3301	3,03977	0,07827
SCHS6	3	30	9,3	11,4197
SCHS7	3,0001	3,6531	3,08564	0,14505

BF19

Algorithms	Best min	Best Max	Average	S.D.
HS	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**
SCA	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**
SCHS1	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102-16**
SCHS2	-0,30048	-0,25323	-0,29736	0,00888
SCHS3	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**
SCHS4	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**
SCHS5	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**
SCHS6	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**
SCHS7	**-0,30048**	**-0,30048**	**-0,30048**	**1,1102E-16**

BF20

Algorithms	Best min	Best Max	Average	S.D.
HS	-3,322	-3,2031	-3,2744	0,058249
SCA	-3,2205	-1,7941	-2,8824	0,358220
SCHS1	-3,3154	-3,1521	-3,2762	0,035434
SCHS2	-3,322	-3,2031	-3,2824	0,056050
SCHS3	**-3,322**	**-3,2031**	**-3,2982**	**0,04756**
SCHS4	-3,3163	-3,1625	-3,2658	0,047893
SCHS5	-3,3174	-3,1374	-3,2608	0,048837
SCHS6	-3,322	-3,2031	-3,2903	0,052580
SCHS7	-3,3144	-2,929	-3,2525	0,082241

BF21

Algorithms	Best min	Best Max	Average	S.D.
HS	-10,1532	-2,4593	-3,99803	2,789443
SCA	-6,0306	-0,49726	-2,45088	1,936642
SCHS1	-5,0489	-4,9267	-5,00425	0,031241
SCHS2	-10,1532	-1,2646	-5,03592	3,674335
SCHS3	**-10,1532**	**-2,6305**	**-5,07905**	**3,364213**
SCHS4	-5,0507	-3,7408	-4,96542	0,231392
SCHS5	-5,0533	-4,9385	-5,01395	0,028906
SCHS6	-10,1532	-2,6305	-4,42073	3,162511
SCHS7	-5,051	-2,8328	-4,93240	0,395816

BF22

Algorithms	Best min	Best Max	Average	S.D.
HS	-10,4029	-2,7519	-6,20212	3,686260
SCA	-6,0837	-0,90396	-3,61791	1,606093
SCHS1	-5,0832	-4,8636	-5,04436	0,039995
SCHS2	**-10,4029**	**-2,6548**	**-6,49577**	**3,705835**
SCHS3	-10,4029	-1,8376	-5,57897	3,547183
SCHS4	-5,0831	-4,978	-5,04532	0,026063
SCHS5	-5,0857	-4,9767	-5,04650	0,024022
SCHS6	-10,4029	-2,7519	-4,94047	3,152824
SCHS7	-5,0755	-4,8795	-5,02789	0,056975

BF23

Algorithms	Best min	Best Max	Average	S.D.
HS	-10,5364	-2,3558	-5,21524	3,264729
SCA	-7,0357	-0,94159	-4,26876	1,577655
SCHS1	-5,1259	-5,0391	-5,08894	0,025244
SCHS2	-10,5364	-2,4217	-6,89508	3,532485
SCHS3	**-10,5364**	**-2,8066**	**-7,33870**	**3,668120**

SCHS4	-5,1166	-4,9047	-5,08352	0,042022
SCHS5	-5,1251	-4,9137	-5,08013	0,050410
SCHS6	-10,5364	-2,4273	-6,71929	3,583871
SCHS7	-5,1226	-4,9277	-5,07796	0,046998

3.1. Exploitation and Exploration analysis

Looking at the results, whether the unimodal or multimodal benchmarks, SCHS4 and SCHS7 hybrid models have been successful. When viewed on the basis of unimodal, the success of hybrids shows that the exploitation phase is developing. Similarly, the results in terms of multimodal show that hybrid algorithms are also successful in the discovery phase. Hybrids are developed on the basis of HS. Therefore, it is seen that hybrids provide improvement when compared with HS.

According to the results in Table 5, a general ranking table (Table 6) was created. This is because SCHS3 shows first place in some problems. But the hybrid has a lot of bad results. Table 6 shows how many times the Algorithms won. Thus, the ranking that each algorithm earns the most shows its order among all algorithms. In the rankings given, the hybrid algorithms that win the 1st and 2nd places the most are chosen as the final models.

Table 6
Scoring of hybrid models

Algorithms	**Ranks**									**Ranking Won**
	I	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	**IX**	
HS	3	1	-	2	1	-	1	10	5	VIII
SCA	5	2	2	1	6	-	1	1	5	V
SCHS1	1	1	3	9	6	2	1	-	-	IV
SCHS2	1	4	2	-	-	1	1	4	10	IX
SCHS3	7	1	1	-	-	8	5	1	-	VI
SCHS4	9	-	1	2	5	4	-	2	-	I
SCHS5	1	2	9	5	1	1	4	-	-	III
SCHS6	4	1	1	1	1	5	9	1	-	VII
SCHS7	2	11	3	1	2	1	1	2	-	II

According to the results in Table 6, SCHS4 and SCHS7 were compared with well-known algorithms (Ant Lion Optimizer (ALO) [31], Particle Swarm Optimization (PSO) [32], States of Matter Search (SMS) [33,34], Bat Algorithm (BA) [35], Flower Pollination Algorithm (FPA) [36], Cuckoo Search (CS) [37,38], Firefly Algorithm (FA) [39,40], GA [2]) due to their success. Comparison time has been changed to iteration 1000 only. The reason for this is to adapt to the parameters of the ready data used for comparison [31]. Comparison time was compared only through average. The results are discussed in Table 7 over 13 BF.

Table 7
General comparison

F	f_{min}	SCHS4	SCHS7	ALO	PSO	SMS
				Average		
BF1	0	**7,53E-13**	1,65E-12	2.59E-10	2.70E-09	0.056987
BF2	0	**7,21E-07**	1,07E-06	1.84241E-06	7.15E-05	0.006848
BF3	0	**1,03E-11**	1,09E-11	6.06847E-10	4.71E-06	0.959865
BF4	0	2,19E-07	2,51E-07	**1.36061E-08**	3.25E-07	0.276594
BF5	0	28,817207	28,82529	0.346772393	0.123401	0.085348
BF6	0	5,1356	4,877317	**2.56183E-10**	5.23E-07	0.125323
BF7	0	0,000748	0,000739	0.004292492	0.001398	**0.000304**
BF8	-12569.487	-6869,62732	**-7789,52884**	-1606.27643	-1367.01	-4.20735
BF9	0	**1,36E-10**	3,99E-10	7.71411E-06	0.278588	1.32512
BF10	0	2,73E-07	3,34E-07	**3.73035E-15**	1.11E-09	8.88E-06
BF11	0	**3,19E-14**	9,45E-14	0.018604494	0.273674	0.70609
BF12	0	0,716598	0,532728	**9.74645E-12**	9.42E-09	0.12334
BF13	0	2,88286	2,367167	**2.00222E-11**	1.35E-07	1.35E-02

F	f_{min}	BA	FPA	CS	FA	GA
				Average		
BF1	0	0.773622	1.06346E-07	6.50E-03	0.039615	0.118842
BF2	0	0.334583	0.000624246	2.12E-01	0.050346	0.145224
BF3	0	0.115303	5.6682E-08	2.47E-01	0.049273	0.13902
BF4	0	0.192185	0.003837885	1.12E-05	0.145513	0.157951
BF5	0	0.334077	0.781200043	**0.007197**	2.175892	0.714157
BF6	0	0.778849	1.08459E-07	5.95E-05	0.05873	0.167918
BF7	0	0.137483	0.003105276	0.001321	0.000853	0.010073
BF8	-12569.487	-1065.88	-1842.42621	-2094.91	-1245.59	-2091.64
BF9	0	1.233748	0.273294621	0.127328	0.263458	0.659271
BF10	0	0.129359	0.007398721	8.16E-09	0.168306	0.956111
BF11	0	1.451575	0.085021659	0.122678	0.099815	0.487809
BF12	0	0.395977	0.000265711	5.60E-09	0.126076	0.110769
BF13	0	0.386631	3.67E-06	4.88E-06	0.00213	1.29E-01

Looking at the results, both hybrid models were observed to exhibit competitive results. Table 8 has been created to show the comparison more comfortably. Thus, the comparison score of each algorithm was determined. Likewise, the results here show that the hybrid has successful exploitation and exploration features compared to other algorithms.

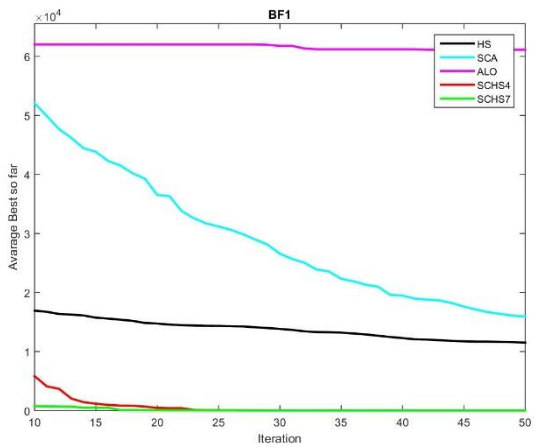
3.2. Convergence behavior analysis

Although the balance provided by the hybrid algorithm in exploitation and exploration phases is exhibited with the above results, the observations of the convergence behavior are not specified. Therefore, the algorithm was inspected from another perspective by preparing the convergence curves.

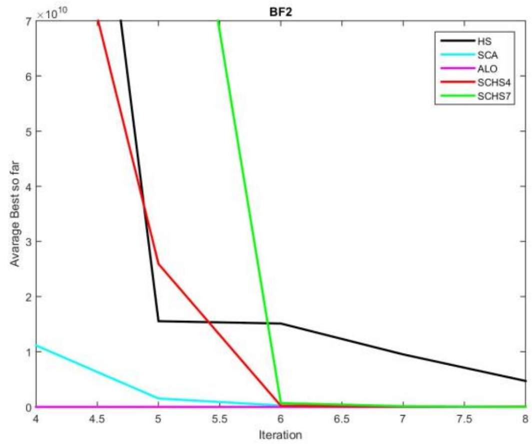
Table 8
Scoring of the compared algorithms

Algorithms	Ranks										Best Score
	I	II	III	IV	V	VI	VII	VIII	IX	X	
SCHS4	5	3	-	1	-	-	-	-	1	3	I
SCHS7	1	5	2	-	1	-	-	-	3	1	II
ALO	5	-	5	-	1	1	-	1	-	-	I, III
PSO	-	2	3	3	1	1	3	-	-	-	III, IV, VII
SMS	1	1	-	-	1	4	-	1	1	4	VI, X
BA	-	-	-	1	-	-	1	4	3	4	VII, X
FPA	-	1	1	3	3	2	3	-	-	-	IV, V, VII
CS	1	1	2	3	2	2	-	-	2	-	IV
FA	-	-	-	1	3	2	4	2	1	-	VII
GA	-	-	-	1	1	1	2	5	2	1	VIII

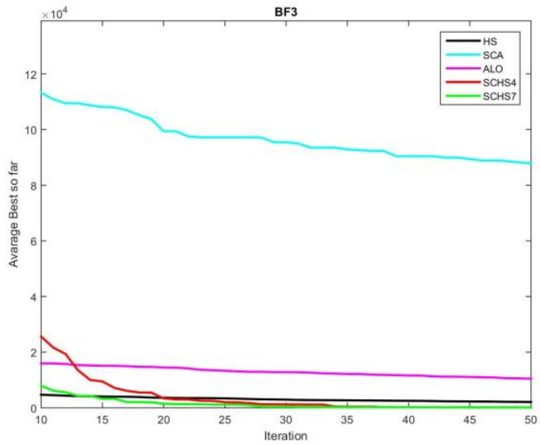
Comparison of selected hybrid algorithms in terms of convergence rate was made. For this, comparison was made with HS, ALO and SCA over BF1, BF2, BF3, BF8, BF9 and BF11. The results are shown in Figure 1. The results are compared with the results obtained over 500 iterations. It was run 30 times independently for each problem. The average was calculated as 30 different results were obtained for each iteration. Convergence graphs were created over these 500 average results.



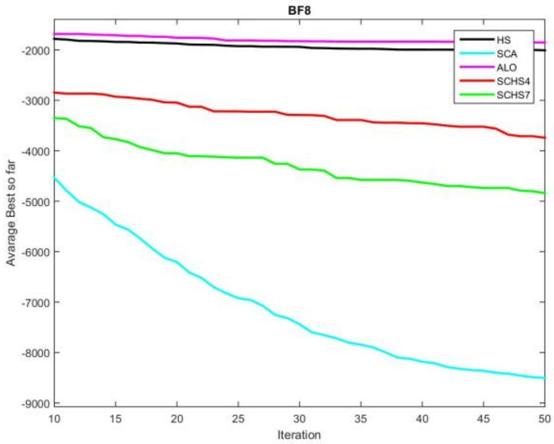
a) BF1



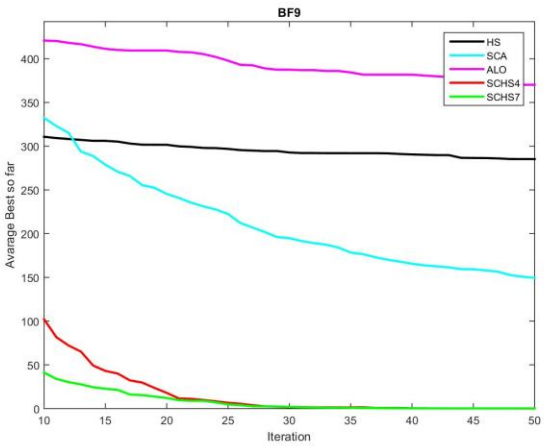
b) BF2



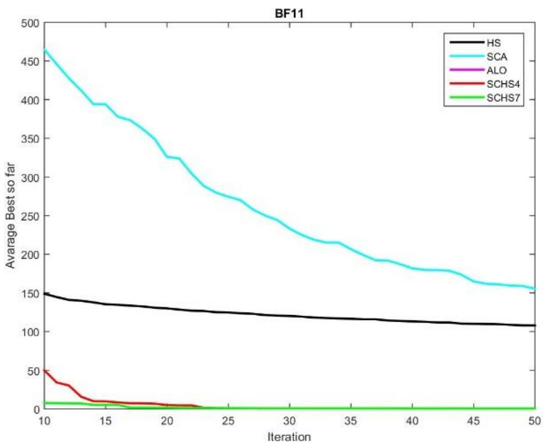
c) BF3



d) BF8



e) BF9



f) BF11

Figure 1 Convergence curves

Looking at the figures, it is observed that SCHS4 and SCHS7 have similar speeds. The iteration interval was kept short in the graphs to notice the comparison between speeds.

In summary, hybrid algorithms show that they are competitive in terms of improvements on the basis of HS, in comparison with other optimization algorithms and in terms of their convergence rate.

4. CONCLUSION

In this study, the new hybrid technique SCHS is presented. BF was used to test the algorithms. Experiments were made to decide on the final models and the optimization algorithms they derive were compared. It has been compared with the known optimization algorithms to demonstrate the performance of the algorithm that has improved. By comparison, it has been observed that the algorithm can compete with other optimization algorithms. In addition, the time of convergence curve comparisons also showed successful performance.

The study is thought to be motivating other researchers to use this algorithm. In future studies, binary and similar models of SCHS can be created. In addition, the performance map can be expanded by adapting the model to various engineering problems.

Acknowledgments

I would like to thank Dr. Seyedali Mirjalili and Seyed Mostapha Kalami Heris for their work in the field and for making them open to us.

Funding

The author received no financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval

Ethics Committee Approval is not required.

The Declaration of Research and Publication Ethics

The author of the paper declare that he comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that he does not make any falsification on the data collected. In addition, he declares that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] Wolpert, D. and Macready, W., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), pp.67-82.

[2] J. H. Holland, “Genetic Algorithms,” Scientific American, vol. 267, no. 1, pp. 66–72, 1992.

[3] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, “Hybrid Taguchi-Genetic Algorithm for Global Numerical Optimization,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 4, pp. 365–377, 2004.

[4] C.-F. Juang, “A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design,” IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 34, no. 2, pp. 997–1006, 2004.

[5] E. Falkenauer, “A hybrid grouping genetic algorithm for bin packing,” Journal of Heuristics, vol. 2, no. 1, pp. 5–30, 1996.

[6] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” Journal of Global Optimization, vol. 39, no. 3, pp. 459–471, 2007.

[7] C. Ozturk and D. Karaboga, “Hybrid Artificial Bee Colony algorithm for neural network training,” 2011 IEEE Congress of Evolutionary Computation (CEC), 2011.

[8] X. Yan, Y. Zhu, W. Zou, and L. Wang, “A new approach for data clustering using hybrid artificial bee colony algorithm,” Neurocomputing, vol. 97, pp. 241–250, 2012.

[9] F. Kang, J. Li, and Q. Xu, “Structural inverse analysis by hybrid simplex artificial bee colony algorithms,” Computers & Structures, vol. 87, no. 13-14, pp. 861–870, 2009.

[10] Z. W. Geem, J. H. Kim, and G. Loganathan, “A New Heuristic Optimization Algorithm: Harmony Search,” Simulation, vol. 76, no. 2, pp. 60–68, 2001.

[11] G. Wang and L. Guo, “A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization,” Journal of Applied Mathematics, vol. 2013, pp. 1–21, 2013.

[12] A. R. Yıldız, «Hybrid Taguchi-Harmony Search Algorithm for Solving,» International Journal of Industrial Engineering Theory, Applications and Practice, cilt 15, no. 3, pp. 286-93, 2008.

[13] B. Wu, C. Qian, W. Ni, and S. Fan, “Hybrid harmony search and artificial bee colony algorithm for global optimization problems,” Computers & Mathematics with Applications, vol. 64, no. 8, pp. 2621–2634, 2012.

[14] S. Mirjalili, “SCA: A Sine Cosine Algorithm for solving optimization problems,” Knowledge-Based Systems, vol. 96, pp. 120–133, 2016.

[15] S. H. R. Pasandideh və S. Khalilpourazari, “SINE COSINE CROW SEARCH ALGORITHM: A POWERFUL HYBRID META HEURISTIC FOR GLOBAL OPTIMIZATION,” arXiv preprint arXiv:1801.08485, 2018.

[16] S. Khalilpourazari and S. Khalilpourazary, “SCWOA: an efficient hybrid algorithm for parameter optimization of multi-pass milling process,” Journal of Industrial and Production Engineering, vol. 35, no. 3, pp. 135–147, 2018.

[17] N. Singh and S. Singh, “A novel hybrid GWO-SCA approach for optimization problems,” Engineering Science and Technology, an International Journal, vol. 20, no. 6, pp. 1586–1601, 2017.

[18] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-Verse Optimizer: a nature-inspired algorithm for global optimization,” Neural

Computing and Applications, vol. 27, no. 2, pp. 495–513, 2015.

[19] P. Jangir, S. A. Parmar, I. N. Trivedi, and R. Bhesdadiya, “A novel hybrid Particle Swarm Optimizer with multi verse optimizer for global numerical optimization and Optimal Reactive Power Dispatch problem,” Engineering Science and Technology, an International Journal, vol. 20, no. 2, pp. 570–586, 2017.

[20] M. Sulaiman, S. Ahmad, J. Iqbal, A. Khan, and R. Khan, “Optimal Operation of the Hybrid Electricity Generation System Using Multiverse Optimization Algorithm,” Computational Intelligence and Neuroscience, vol. 2019, pp. 1–12, 2019.

[21] X. Wang, D. Luo, X. Zhao, and Z. Sun, “Estimates of energy consumption in China using a self-adaptive multi-verse optimizer-based support vector machine with rolling cross-validation,” Energy, vol. 152, pp. 539–548, 2018.

[22] M. Jamil and X. S. Yang, “A literature survey of benchmark functions for global optimisation problems,” International Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, no. 2, p. 150, 2013.

[23] B. Alizada, «Improved Whale Optimization Algorithm Based On Π Number,» International Scientific And Vocational Journal (Isvos Journal), Cilt 4, No. 1, Pp. 21-30, 2020.

[24] S. Mirjalili Ve A. Lewis, «The Whale Optimization Algorithm,» Advances In Engineering Software, Cilt 95, Pp. 51-67, 2016.

[25] M. Danacı And M. A. Dıallo, "A New Hybrid Fruit Fly Optimization Algorithm For Solving Benchmark Problems," Euroasia Journal Of Mathematics-Engineering Natural & Medical Sciences , Vol.2, Pp.23-27, 2019

[26] M. Danacı And Z. Akhdır, "A Novel Hybrid Bat Crow Search Algorithm For Solving Optimization Problems," Euroasia Journal Of Mathematics-Engineering Natural & Medical Sciences , vol.2, pp.40-45, 2019

[27] M. Danacı And B. Alızada, "An Improvement Of Hybrid Whale Optimization Algorithm," Euroasia Journal Of Mathematics-Engineering Natural & Medical Sciences , vol.2, pp.60-68, 2019

[28] P. Chakraborty, G. G. Roy, S. Das, D. Jain, and A. Abraham, “An Improved Harmony Search Algorithm with Differential Mutation Operator,” Fundamenta Informaticae, vol. 95, no. 4, pp. 401–426, 2009.

[29] Q.-K. Pan, P. Suganthan, M. F. Tasgetiren, and J. Liang, “A self-adaptive global best harmony search algorithm for continuous optimization problems,” Applied Mathematics and Computation, vol. 216, no. 3, pp. 830–848, 2010.

[30] X. Gan, E. Jiang, Y. Peng, S. Geng, and M. Kustudic, “Research Optimization on Logistic Distribution Center Location Based on Improved Harmony Search Algorithm,” Lecture Notes in Computer Science Advances in Swarm Intelligence, pp. 410–420, 2018.

[31] S. Mirjalili, “The Ant Lion Optimizer,” Advances in Engineering Software, vol. 83, pp. 80–98, 2015.

[32] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science.

[33] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, “An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation,” Applied Intelligence, vol. 40, no. 2, pp. 256–272, 2013.

[34] E. Cuevas, A. Echavarría, D. Zaldívar, and M. Pérez-Cisneros, “A novel evolutionary algorithm inspired by the states of matter for template matching,” Expert Systems with Applications, vol. 40, no. 16, pp. 6359–6373, 2013.

[35] X. S. Yang, “A New Metaheuristic Bat-Inspired Algorithm,” Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74, 2010.

[36] X. S. Yang, “Flower Pollination Algorithm for Global Optimization,” Unconventional Computation and Natural Computation, pp. 240–249, 2012.

[37] X. S. Yang and S. Deb, “Cuckoo Search via Lévy flights,” 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–4, 2009.
[38] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” International Journal of Mathematical Modelling and Numerical Optimisation, vol. 1, no. 4, p. 330, 2010.
[39] X. S. Yang, “Firefly Algorithm, Lévy Flights and Global Optimization,” Research and Development in Intelligent Systems XXVI, pp. 209–218, 2009.
[40] X. S. Yang, “Firefly algorithm, stochastic test functions and design optimisation,” International Journal of Bio-Inspired Computation, vol. 2, no. 2, p. 78, 2010.