Computational Biomedicine Vol 1 Issue 1 2024
DOI:

ARTICLE

Open Access

Cock-Hen-Chicken Optimizer: A Nature-inspired Algorithm for Real-world Engineering Optimization

Zheng-Ming Gao[1, 2*], and Juan Zhao[3]

[1]School of computer engineering, Jingchu University of Technology, Jingmen, 448000, China
[2]Hubei Engineering Research Center for Specialty Flowers Biological Breeding, Jingmen 448000, China
[3]School of electronics and information engineering, Jingchu University of Technology, Jingmen 448000, China

***Correspondence to:** Zheng-Ming Gao, School of computer engineering, Jingchu University of Technology, Jingmen 448000, China; Email: gaozming@jcut.edu.cn

Received: April 1, 2024; **Accepted:** May 7, 2024; **Published online:** May 8, 2024

How to cite: Gao, Z.M., Zhao, J. Cock-hen-chicken Optimizer: A Nature-inspired Algorithm for Real-world Engineering Optimization. *Computational Biomedicine*, 2024; 1(1). Doi:

Abstract: Nature-inspired algorithms have been a hot spot and proved to be a successive way to handle optimization problems. Due to the No Free Lunch (NFL) theorem, all of the algorithms might fail to solve some of the problems and consequently need to be improved. In order to find a better and efficient way to solve the real-world engineering problems, an algorithm called the Cock-Hen-Chicken (CHC) optimizer was proposed with the inspiration of hunting behaviors of the cocks, hens, and chickens. Simulation experiments on either unimodal, multimodal, IEEE Congress on Evolutionary Computation 2017 (CEC17), or CEC2011 competitive problems were carried out and finally, it was applied to solve five real-world engineering problems. Most of the simulation results except for the CEC17 confirmed the better performance, superiority, and capability of the proposed CHC optimizer comparing with other well-known optimization algorithms such as the ant lion optimizer (ALO), the equilibrium optimizer (EO), the grey wolf optimizer (GWO), the mayfly optimization algorithm (MOA), the particle swarm optimization (PSO), the sine-cosine algorithm (SCA), and the whale optimization algorithm (WOA). Results of real-world engineering problems were also promising. The proposed CHC optimizer reported in this paper would be a better choice for future applications and the code is shared with https://github.com/gaozming/CHCoptimizer for possible future efforts.

Keywords: Nature-inspired algorithms; Global optimization; Swarm intelligence; Benchmark functions; Real-world engineering problems; Simulation experiments; Evolutionary computation

1. Introduction

Along with the development of modern science and technology, we human beings have been living a happy life with intelligent mobile devices, automated cars, and other automated equipment since the time we went into the information era. However, we are facing a more and more complicated world meanwhile. Several hundreds of years ago, the problems raised at that time could be solved with pencils. Yet some of the problems had been related to probabilities, and integrations several dozens of years ago. By now, the satellite has been deployed all over the world, and electronic communications have changed the world into a small country. We can talk to each other with real-time sound and images, just like we are communicating face-to-face. The modern life was deduced by our real and effective understanding of the world. But if one of the problems would be raised in mathematics nowadays, more details would be involved, the numbers of parameters, degree of integration or difference might all be included, and consequently, the modern problems would be quite difficult to solve. And the answer might be no longer possible to find with analytical mathematics. Meta-heuristic algorithms have been a popular choice for such kinds of optimizing problems, including special clustering[1], robot path planning[2], damage assessment[3], even the modal identification of proton exchange membrane fuel cells[4], scheduling of demand response-enabled micro grids[5], distributed generation planning[6], and CFD Analysis and Optimum Design[7], charger placement problem[8], load forecasting[9], and multiple objective optimization[10, 11].

However, due to the No-free Lunch (NFL) Theorem [12], there still does not exist an algorithm that could solve all of the problems with efficiency. Therefore, we are still on the way to find better-performing novel algorithms, even their improvements.

The main contributions of this paper would be:

(1) Literal classifications of the nature-inspired algorithms proposed by now have been made based on the involvements of individuals and the ways to change their behaviors.

(2) A CHC optimizer with swarms of best candidates and multiple updating disciplinary was proposed.

(3) Detailed simulation experiments have been carried out and the superiority of the proposed CHC optimizer was confirmed.

The rest of this paper would be arranged as follows: In section 2, a brief literature review would be given, and the CHC optimizer would be proposed in section 3. Simulation experiments on benchmark functions, together with CEC17 competitive problems, and real-world engineering problems would be carried out in sections 4 and 5. Discussion would be made and conclusions would be drawn in section 6.

2. Related Work

Traditionally, nature-inspired algorithms could be classified into four types according to the source of their inspiration[13]: 1) evolutionary algorithms, such as the genetic algorithm (GA)[14], the evolutionary algorithm (EA)[15]; 2) human-based algorithms, such as the harmony search (HS) algorithm[16], and the group search optimizer (GSO)[17]; 3) physics-based algorithm, including the gravitational search algorithm (GSA)[18], the black hole algorithm (BHA)[19], Ray optimization (RO)[20]; and 4) swarm-based algorithms, including the ant colony optimization (ACO) algorithm[21], particle swarm optimization (PSO) algorithm[22], the grey wolf optimization (GWO) algorithm[23], Aquila optimizer (AO)[24, 25], the gaining-sharing knowledge (GSK) optimizer, and so on. Considering almost all of the inspiration sources being existence in literature, this kind of classification has been popular for several years. In addition, efforts have also been made that the algorithms proposed by now could be divided into nine classes such as physics based, social based, music based, swarm based, chemistry based, biology based, sports based, math based, and the hybrid optimization algorithms[26, 27], more inspirations sources were included.

With a detailed study on the involvement of individuals in swarms, or ways to update their positions, we hereby proposed other two types of classification of the nature-inspired algorithms.

2.1 Classification with Involvements of Individuals

With a glance at the development of the nature-inspired algorithms, we would find that all of the individuals would be involved in updating their positions at the

beginning. They were treated in a same way and simple equations were introduced. For example, all of the information of ants would be equally involved and balanced to change their positions at the next iteration in the ACO swarms[28]:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}(t) \quad (1)$$

Where $\tau_{ij}(t)$ and $\tau_{ij}(t+1)$ represent the amount of pheromone on edge (i, j) for t and $t+1$ time. ρ is the pheromone evaporation and $\Delta\tau_{ij}(t)$ is the total amount of pheromone deposited on edge (i, j).

Also, for individuals in the GA swarms, all of the individuals would have the chance to mutate and play a role in updating their positions at the next iteration. We can call this type of algorithms the **all-involvement algorithms**. For this kind of algorithms, the individuals played the same role either he/she was the best or worst candidate. Consequently, the worse candidates would slow down the convergence and fail to get the global best position sometimes.

For the PSO algorithm, it was a little different because the historical best trajectories $x_p(t)$ along with the global best candidates $x_b(t)$ would be introduced to update individuals' positions at the next iteration:

$$x_i(t+1) = x_i(t)+v_i(t+1) \quad (2)$$

$$v_i(t+1) = v_i(t)+c_1[x_p(t)-x_i]+c_2[x_b(t)-x_i] \quad (3)$$

Where $v_i(t+1)$ and $v_i(t)$ represent the velocity of individuals in the $t+1$ and t iterations respectively for $i-th$ individual, and c_1, c_2 are two fixed numbers.

Same conditions might occur for individuals in swarms of African buffalo algorithm (ABA)[29]:

$$x_i(t+1) = x_i(t)+lp_1\,[bg_{maxk}-x_i(t)]-lp_2[bp_{maxk}-x_i(t)] \quad (4)$$

Where bg_{maxk} and bp_{maxk} represent the best historical and global effective positions of the herd at the current iteration, lp_1 and lp_2 denote the learning rate, which were controlled with randomness and energy along with iterations.

The best candidate was soon paid attention to and the **best-involvement algorithms** were proposed, including the bat algorithm[30], Archerfish hunting optimizer (AHO)[31], and so on. For the best-involvement algorithm, only the positions of the best candidates would be involved to update the positions. The best candidate played more important role during iterations and consequently faster convergence were achieved. However, the best candidates might fall into the local optima and lead to lower diversification capability accordingly. But the overall performance outperforms the **all-involvement algorithms**. This kind of best-involvement algorithms would also include other special examples, such as the GWO algorithm[23]:

$$D_\alpha = |C_1 \cdot X_\alpha(t)-X(t)| \quad (5)$$

$$D_\beta = |C_2 \cdot X_\beta(t)-X(t)| \quad (6)$$

$$D_\delta = |C_3 \cdot X_\delta(t)-X(t)| \quad (7)$$

$$X_1 = X_\alpha(t)-A_1 \cdot D_\alpha \quad (8)$$

$$X_2 = X_\beta(t)-A_2 \cdot D_\beta \quad (9)$$

$$X_3 = X_\delta(t)-A_3 \cdot D_\delta \quad (10)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (11)$$

Where t indicates the current iteration. X is the position vector of the grey wolf, X_α, X_β, X_δ are the position vectors of the alpha, beta, and delta wolf, which would be the top three wolves whose fitness values were the top three best at each iteration. The parameters A and C are combinations of the controlling parameter and random numbers.

The top best three grey wolves would all be involved in updating the positions of individuals in the next iteration. Top best candidates also performed in updating the positions of individuals in swarms of the equilibrium optimizer (EO)[32]:

$$C_i(t+1) = C_{eqs} + (C_{eqs} - C_i) \bullet F + \frac{G}{\lambda}(1-F) \quad (12)$$

Where $C_i(t+1)$ and C_{i} represent the position of individuals at the $t+1$ and t iteration, C_{eqs} is a randomly selected candidate from the equilibrium optimization pool, F, G, and λ are controlling parameters.

More top candidates involved would reduce the capability of being trapped in local optima and diversification capability would be achieved. In addition, the top best candidates would play more important role during iterations, and therefore, faster convergence were also achieved.

In summary, the nature-inspired algorithms could be classified into two types based on the involvements of individuals to update their positons at the next iteration, clearly seen in **Table 1**.

Table 1. Classification of Nature-inspired Algorithm Based on the Involvements of Individuals

Major types	Characteristics	Examples
all-involvement algorithm	all individuals are involved	ES[33], ACO, GA, HS, GSO
best-involvement algorithms	best trajectory and global best candidate	PSO, ABA
	global best candidate	BA, AHO, WOA[13], Hippopotamus optimization algorithm(HO)[34], Puma optimizer (PO)[35]
	top best candidates	GWO, EO

2.2 Classification with Updating Ways of Individuals' Positions

Review on the structure of the nature-inspired algorithms, we can see that the modern algorithms would perform better in most cases along with the complication of equations or parameters involved in the algorithms.

At the literal beginning, all of the individuals would update their positions with a same equation, see the PSO, ACO, BA, ant-lion optimizer (ALO)[36], and so on. All of the individuals were updated with the single equations, which led to less diversification and decreased the capability to find the global best optima. To improve the capability of the algorithms, a multiple updating discipline was soon proposed[37, 38], and better performance was confirmed, therefore, most of the nature-inspired algorithms proposed in the recent years would embrace the multiple updating discipline and the individuals would update their positions with several ways, such as the sine cosine algorithm (SCA)[39], individuals in the SCA swarms would chose the sine or cosine function to update their positions randomly:

$$x_i(t+1)=\begin{cases} x_i(t)+p\cdot\sin(r_1)\left|r_2x_b-x_i(t)\right| & r_3<0.5 \\ x_i(t)+p\cdot\cos(r_1)\left|r_2x_b-x_i(t)\right| & r_3>0.5 \end{cases} \tag{13}$$

Where, r_1, r_2 and r_3 are random numbers in Gauss distribution. p is a random number in uniform distribution with a declination from 2 to 0 linearly.

Individuals in swarms of the Harris Hawk optimization (HHO) algorithm[40] would have four ways to select to update their positions, two ways for individuals in swarms of the arithmetic optimization algorithm (AOA)[41], and four ways for individuals in swarms of the Aquila optimizer (AO), war strategy optimization (WSO) algorithm[42] and so on. The multiple updating discipline introduce multiple ways for individuals to update their positions, and consequently result in diversification and avoid being trapped in local optima. Generally speaking, more ways to update the positions would lead to more diversification capability, and more methods to approach the global optima. This kind of classification was simple and efficient, as shown in **Table 2**.

Table 2. Classification of Nature-inspired Algorithm Based on the Updating Ways of Individuals' Positions

Major types	Examples
Single updating discipline	PSO, ACO, BA, ALO
Multiple updating discipline	SCA, HHO, AOA, AO

3. Cocks Hens Chicken Optimizer

House breeding was very popular for Chinese peasants. During the old days, The Chinese peasants had always bred one or two bulls to plough, one pig for the leftover and also the preparation of pork for the Spring Festival, a swarm of cocks, hens to obtain the eggs and chickens. Talking about the swarm of cocks and hens, we know that if the hens were fertilized, they would produce eggs which could be incubated. Some hens would stay at the prepared lair for 21 days or so in spring and the chickens would be given birth to. After several days of breeding at home, the peasants would open the door and let go all of the cocks, hens, and chickens outside for food and call them back at sun sank. The cocks, hens, chickens would go outside near the village and search for food.

During this period of time searching for food, the cocks would be eager to find a place full of food and coo the hens to come. While the mother hens would be aggressive to secure the chickens and lead them to find food and breed the chickens. Although as a species of the Phasianidae category, the Gallus are not clever,

but the whole searching procedure would be full of interesting behavior including love and education, the cocks would breed the hens sometimes, and they all would breed the chickens, the mother hens would always keep their eyes on the chickens and coo them to follow, they were aggressive when human, snakes or other possible threats are nearby.

Generally, the cocks, hens, and the chickens would find food with their own will and the embedded inheritable characters. The number of the cocks, hens, and the chickens would be more and their hunting behaviors were different.

3.1 Cocks Behavior of Searching and Exploitation

After four to five months, the male chickens would grow up and they would be eager for reproduction of swarms. Therefore, during the searching procedure for food, they are in a rush finding the places full of food. When they found the place, they would coo the hens to come and seek for the chance for fertilization. When a place with food is found, they might breed themselves at first, dig out the food nearby with their beaks and paws. If there is no food at the current place, they may dig nearby or move to another place, as shown in **Figure 1**.

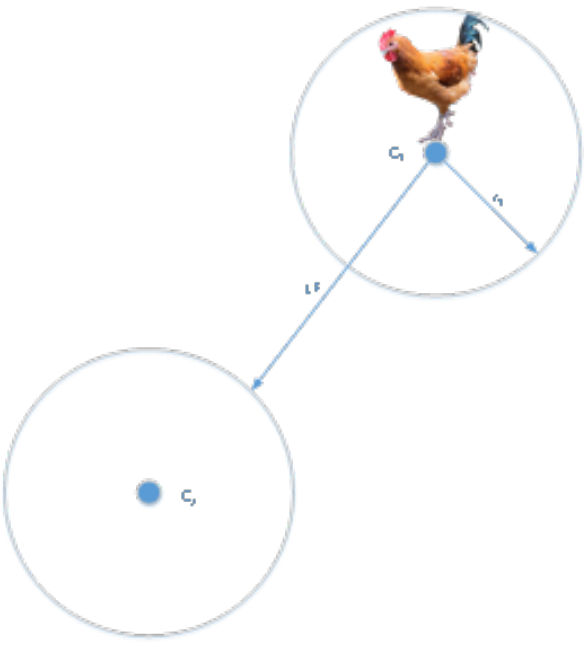


Figure 1. Sketches of finding food for cocks.

When the cocks find some food at the current place, he would want to find more around the current place. In mathematics, this behavior could be described as a random walk from the current position:

$$x_i(t+1) = x_i(t)+dance\cdot r_1 \tag{14}$$

Where $x_i(t+1)$ and $x_i(t)$ represent the position of the i-th cock at the iteration $t+1$ and the current iteration, $dance$ is a constant parameter which would be set 0.8 in this paper. r_1 is a random vector in uniform distribution. With equation (14), the cocks would perform detailed exploration around the current position and find more food with randomness.

There would not be only one cock in swarms. Yet the cocks have the sense of dominions, if one finds the place with food, the others would wander away to find another place. Furthermore, if the cock at the current place did not find food, it would move to another place with random willing. This procedure could be described as random walk with Levy flight. The Levy flights would result in some long-distance research around several smaller one, and cause the individuals to run away from the current position, this could be formulated in mathematics:

$$x_i(t+1) = x_b(t)\cdot LF(D) \tag{15}$$

Where $x_b(t)$ is the best position at the current iteration, and $LF(D)$ is a D dimensional random vector following:

$$LF(D) = 0.01\cdot\frac{\mu}{|v|^{\frac{1}{\beta}}} \tag{16}$$

Where μ and v are global mean value and the standard derivation in Gauss distribution with $\mu = 0$ and

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\cdot\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(1+\frac{\beta}{2}\right)\cdot\beta\cdot 2^{\frac{\beta-1}{2}}}\right]^{\frac{1}{\beta}} \tag{17}$$

Where Γ represents the standard gamma function in mathematics and β is a constant value 1.5.

3.2 Hens Behavior of Searching and Exploitation

Although there is not necessary for peasants to breed cocks and hens together, we did observe that the hens would approach the cocks if they are nearby.

If the cocks are nearby, the hens would follow the cocks to find food. However, there is no morality concept for hens, and consequently, some of the hens would continue to follow a same cock, some of them may lead themselves or the chickens to follow another

one, as shown in **Figure 2**.

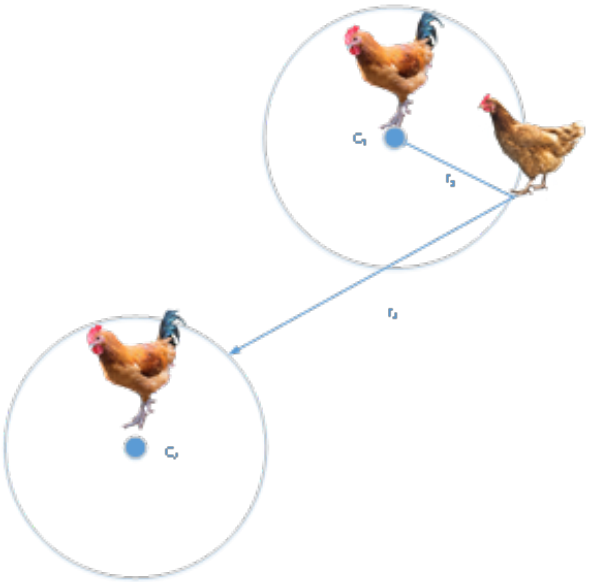


Figure 2. Sketches of finding food for hens

The cocks would continue to coo more hens to follow them and obtain more chance to get fertilization. Therefore, if the food is sufficient, the hens would be cooed to gather around the best cock. That is to say, individuals would continue to find food around the current position and step to the best position. This behavior could be described with the following equation:

$$x_i(t+1) = x_i(t)+a\cdot r_2[x_b(t)-x_i(t)] \quad (18)$$

$$a = a_0\left(1-\frac{t}{maxIter}\right) \quad (19)$$

Where a_0 = 2, and *maxIter* represents the maximum allowed iteration times. r_2 is another random number with an interval of 0 and 1.

Also, the pleasant wanted to breed more hens being eager for eggs and chickens. When searching for food, the mother hens would lead the chickens to find food and avoid meeting each other. In addition, if the place is not full of food for all, some of the hens might walk away and select another random cock to follow:

$$x_i(t+1) = x_c(t)+a\cdot r_3[x_c(t)-x_i(t)] \quad (20)$$

Where $x_c(t)$ represents a random selected candidate of the cocks.

3.3 Chickens Behavior of Searching and Exploitation

Chickens are very young and they are full of courage to try any chance to find food. We have experienced that one young chicken with several days of birth tried to swallow a whole centipede with a length of 6 centimeters or so, the chicken raised its head and swallowed hard, leaving the rest of the centipede crawling extensively, it was really astonishing. **Figure 3** shows that the chickens might follow their mother hens, the cocks, or their own experience.

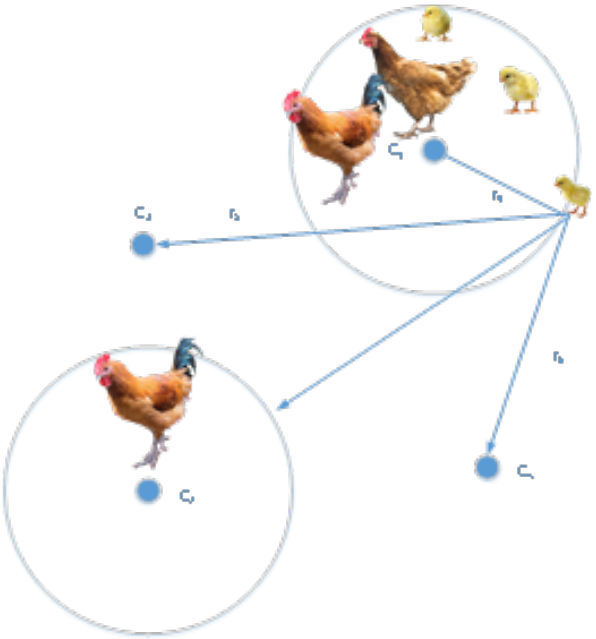


Figure 3. Sketches of finding food for chickens

Considering the observation of the chickens' behavior, there might be four choices for the chickens to find food. At most times, chickens were cooed to follow their mother hens and run to the mother hens with random wills:

$$x_i(t+1) = x_h(t)+a\cdot r_4[x_h(t)-x_i(t)] \quad (21)$$

Note that some of the chickens could not identify their mother and the hens sometimes could not identify which one is not their children either. Therefore, $x_h(t)$ would be a random selected candidate of the hens.

Meanwhile, sometimes chickens would wander away with their own willing, and result in a random initialization of searching with a small proportional number p_1:

$$x_i(t+1) = r_5(ub-lb)+lb \quad (22)$$

Where $[lb, ub]$ is the definitional domain of the given problem, r_5 is another random number in Gauss distribution.

Being young and unwise, the chickens would also risk themselves to follow the best cock, which would be the best candidate as formulated with equation (5). In addition, their small memory could afford some experience, so they would walk away from the worst position or threating target, as described as follows:

$$x_i(t+1) = x_i(t)+a\cdot r_6[x_w(t)-x_i(t)] \quad (23)$$

Where $x_w(t)$ represents the worst position at the current iteration.

3.4 Pseudo Code and Complexity of the Algorithm

Based on the searching behavior of cocks, hens, and chickens, we hereby proposed a new metaheuristic algorithm and called the cock-hen-chicken optimizer with abbreviation CHC optimizer. Supposing there was a swarm of Gallus with N individuals, including m cocks, n hens, and $N-m-n$ chickens, they are searching for a best place with lots of food. After the initialization all around the definitional domain, their positions would be evaluated and an ascending or descending order would be achieved, the top m individuals would be selected as the cocks, and the m+1 to m+n individuals would be selected as the hens, and the rest of them are all treated as chickens. They would perform random search according to their inherited nature and then reach to new positions, until the best place was reached. With a little different to the chicken swarm optimization (CSO) algorithm[43], the swarms would embrace more top candidates to update their positions during iterations. Due to the guidance of multiple best candidates involved in updating the positions, and the multiple updating disciplinary, premature of the convergence might be reduced and better performance might be expected.

With more top candidates being involved in updating their positions and eight updating disciplines, the proposed CHC optimizer would be expected to perform better in convergence and avoid being trapped in local optima.

The flowchart of CHC optimizer was shown in **Figure 4**, and the pseudo code of this algorithms is shown in **Algorithm 1**.

ALGORITHM 1 PSEUDO CODE OF THE CHC OPTIMIZER

Stage	pseudo code
Parameters	given problems *fobj* and its dimensionality D, domain [lb, ub] swarm's population N, m cocks, n hens parameter $p_1 = 0.9$ criterion maxIter
Initialization	initialize the positions with equation (22) calculate the fitness values the top m best candidates are assigned as cocks the top candidates ranked m~m+n are assigned as hens the rest candidates are chicken

Exploration and exploitation	While criterion not met
	for each cock
	  if best one
	  update positions with equation (14)
	  else
	  update positions with equation (15)
	memory saving
	for each hen
	  if $r_1 < 0.5$
	  update positions with equation (18)
	  else
	  update positions with equation (20)
	memory saving
	for each chicken
	  if $r_2 < 0.5$
	  if $r_3 < p_1$
	  update positions with equation (21)
	  else
	  update positions with equation (22)
	  else
	if $r_4 < 0.5$
	  update positions with equation (18)
	  else
	  update positions with equation (23)

Continuation Table:

Stage	pseudo code
Exploration and exploitation	memory saving update swarm the top *m* best candidates are assigned as cocks the top candidates ranked *m*~*m*+*n* are assigned as hens the rest candidates are chicken
Output	Position and fitness of the best cock

We can find that although the individuals in swarms were divided into cocks, hens, and chickens, the updating equations remain simple with multiple updating principle[38]. The computer complexity would be relevant to the number of maximum allowed iterations and the number of individuals. A simple form of the complexity might be formulated as: $O(maxIter \times N)$.

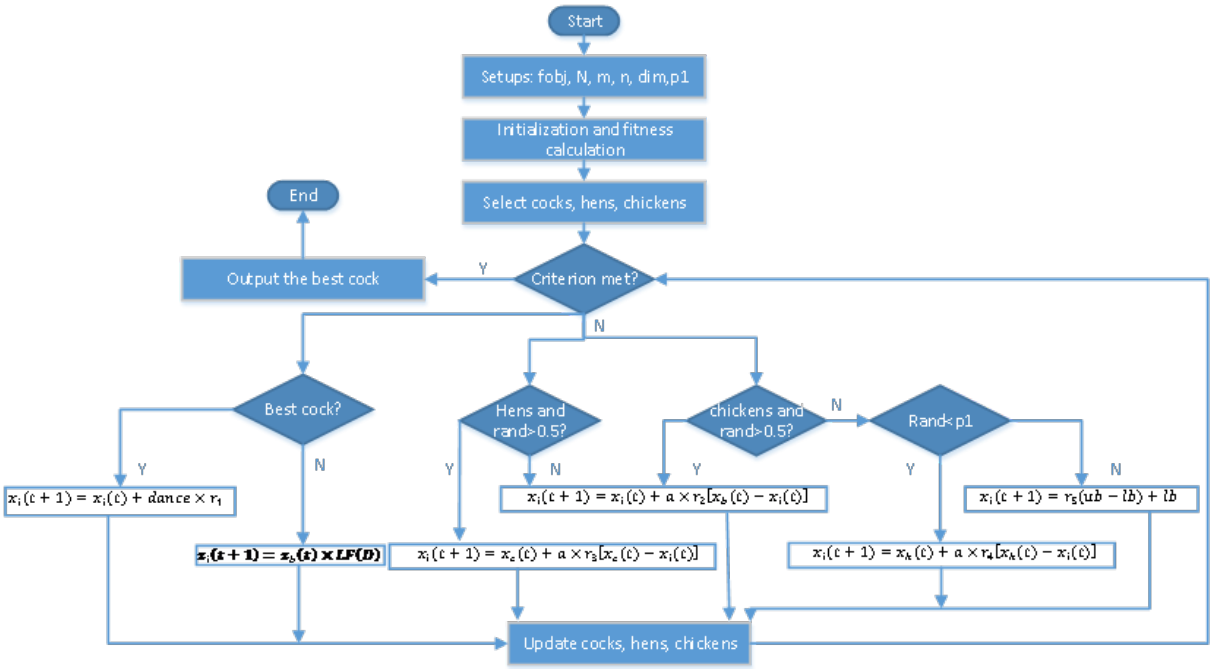


Figure 4. Flowchart of CHC optimizer

In this proposed algorithm, individuals including the cocks, hens, and chicken would have eight ways to update their positions in each iteration, and the best candidate, top and global best historical trajectories would all be involved in the updating. The proposed CHC optimizer absorbs all of the best characteristics of modern optimization algorithms and would be expected to perform quite well in optimization.

4. Results and Discussions

4.1 Experiment and Parameters Setup

Simulation experiments were usually carried out to verify the capabilities of the algorithms, especially the new proposed algorithms. In this section, we would carry out simulations with unimodal, multimodal benchmark functions, composite test functions, together with some real-world engineering problems.

Unimodal benchmark functions are easy to optimized causing the only global optima in the whole domain, while multimodal benchmark functions usually have many local optima with one global optima, so individuals would be usually been trapped in local optima, they would be more difficult to optimize. To confirm the capabilities of the proposed CHC optimizer, five unimodal and six multimodal benchmark functions would be selected as representatives, as shown in **Table 3**, **Table 4**.

Table 3. Unimodal Benchmark Functions

No.	Name	Function	dim	domain
F1	Ackley 1	$f(x)=-20e^{-0.02\sqrt{\frac{\sum_{i=1}^{d}x_i^2}{d}}}-e^{\frac{\sum_{i=1}^{d}\cos(2\pi x_i)}{d}}+20+e$	10	[−100, 100]
F2	Exponential	$f(x)=1-e^{-0.5\sum_{i=1}^{d}x_i^2}$	10	[−3, 3]
F3	Powell Sum	$f(x)=\sum_{i=1}^{d}\lvert x_i\rvert^{i+1}$	10	[−1, 1]
F4	Sargan	$f(x)=\sum_{i=1}^{d}d\left(x_i^2+0.4\sum_{j=1,j\neq i}^{d}x_ix_j\right)$	10	[−100, 100]
F5	Sphere	$f(x)=\sum_{i=1}^{d}x_i^2$	10	[−100, 100]

Table 4. Multimodal Benchmark Functions

No.	Name	Function	dim	domain
F6	Alpine 1	$f(x)=\sum_{i=1}^{d}\lvert x_i\sin(x_i)+0.1x_i\rvert$	10	[−10, 10]
F7	Cosine Mixture	$f(x)=\frac{d}{10}+\sum_{i=1}^{d}x_i^2-\frac{1}{10}\sum_{i=1}^{d}\cos(5\pi x_i)$	10	[−1, 1]
F8	Griewank	$f(x)=\sum_{i=1}^{d}\frac{x_i^2}{4000}-\prod\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	10	[−10, 10]
F9	Inverted Cosine-Wave	$f(x)=d-1-\sum_{i=1}^{d}\left\{e^{\left[\frac{-\left(x_i^2+x_{i+1}^2+0.5x_ix_{i+1}\right)}{8}\right]}cos\left(4\times\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}\right)\right\}$	10	[−5.12, 5.12]
F10	Rastrigin	$f(x)=\sum_{i=1}^{d}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	10	[−5.12, 5.12]
F11	Salomon	$f(x)=1-\cos\left(2\pi\sqrt{\sum_{i=1}^{d}x_i^2}\right)+0.1\sqrt{\sum_{i=1}^{d}x_i^2}$	10	[−100, 100]

All of the simulation experiments were carried out with a HPE ProLiant DL380 Gen10 server, equipped with two Intel Xeon Bronze 3106 CPU and 32GB RAM, the source code was written with Matlab 2021b and shared with address: https://github.com/gaozming/CHCoptimizer.

For each swarms involved in the experiments, the population size would be setup with 50, and in order to reduce the influence of randomness involved in the algorithms, the overall results would be the averaged results with 100 independent Monte Carlo simulation experiment results.

In order to verify the capability of the proposed CHC optimizer, other algorithms in literature would be introduced to make comparisons. In this section, we would carry on the simulation experiments and compare with other famous optimization algorithms such as the ant-lion optimizer (ALO)[36], equilibrium optimization (EO) algorithm[32], grey wolf optimization (GWO) algorithm[23], mayfly optimization algorithm(MOA)[44], particle swarm optimization(PSO) algorithm[22], sine-cosine algorithm(SCA)[39], and the whale optimization algorithm(WOA)[13]. All of the parameters involved in this paper would be listed in **Table 5**.

Table 5. Parameters Setup

Algorithms	Parameters setup
ALO	-
EO	$a_1 = 2$; $a_2 = 1$; $GP = 0.5$
GWO	$a_0 = 2$
MOA	$a_1 = 1$; $a_2 = 1.5$; $a_3 = 1.5$; $\beta = 2$; $da = 5$; $fl = 1$; $dd = 0.8$; $fld = 0.99$; $nc = 20$; $\mu = 0.01$; $gdamp = 0.8$; $g = 0.8$
PSO	$c_1 = 1.5$; $c_2 = 1.5$
SCA	$a_0 = 2$
WOA	$a_0 = 2$
CHC	$m = 10$; $n = 5$

4.2 Empirical Study on the Parameters

Traditionally, the peasants would breed some cocks,

more hens, and many chickens during spring. They would want to breed more hens and less cocks when the chickens grow up. But when we formulate the searching and exploitation procedure for food, the numbers of cocks and hens would be need to be setup at first. The best choice is not known, so we would carry on some empirical research.

In this section we would carry on an empirical study on the influence of numbers of cocks and hens during optimization. 50 individuals would be setup at the beginning, and the numbers of cocks and hens would be changed from 1 to 20. Results of benchmark functions would be optimized with changing numbers of the proposed CHC optimizer and a three-dimensional graph would be drawn to show the influence of numbers of cocks and hens on the optimum under a given maximum number of iterations 100. Again, Monte Carlo method[45] would be chosen to reduce the influence of randomness involved in the algorithm. All of the results would be averaged over 100 independent runs, as shown in **Figure 5-8**. For simplicity, two representative of unimodal and multimodal benchmark functions were selected to demonstrate the results. The best groups of numbers for cocks and hens would result in lowest fitness values and fastest convergence. Near half of them would be evaluated as cocks or hens, the final averaged results would show a balance of their numbers.

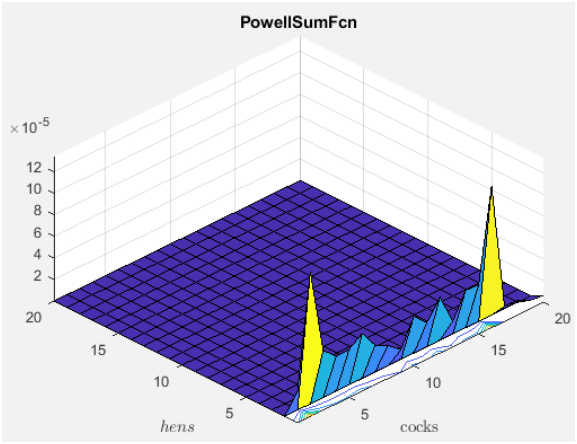


Figure 5. Empirical research on numbers of cocks and hens (PowellSum)

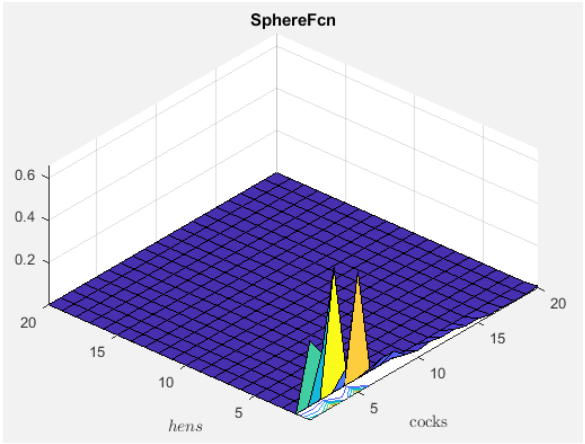


Figure 6. Empirical research on numbers of cocks and hens (Sphere)

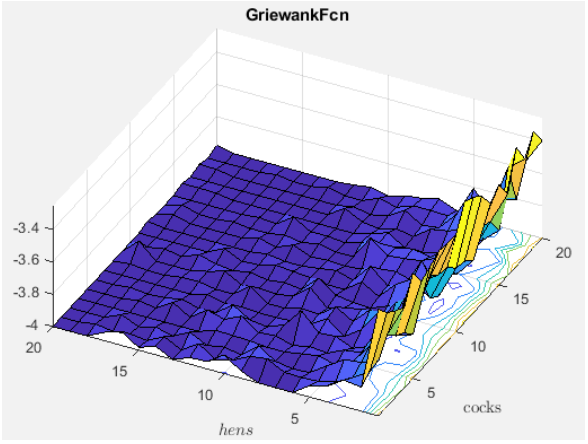


Figure 7. Empirical research on numbers of cocks and hens (Griewank)

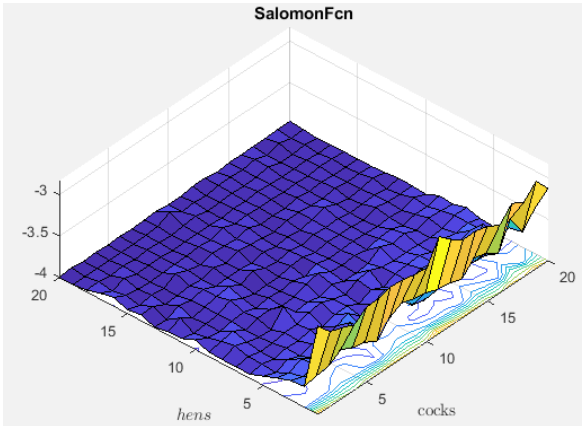


Figure 8. Empirical research on numbers of cocks and hens (Salomon)

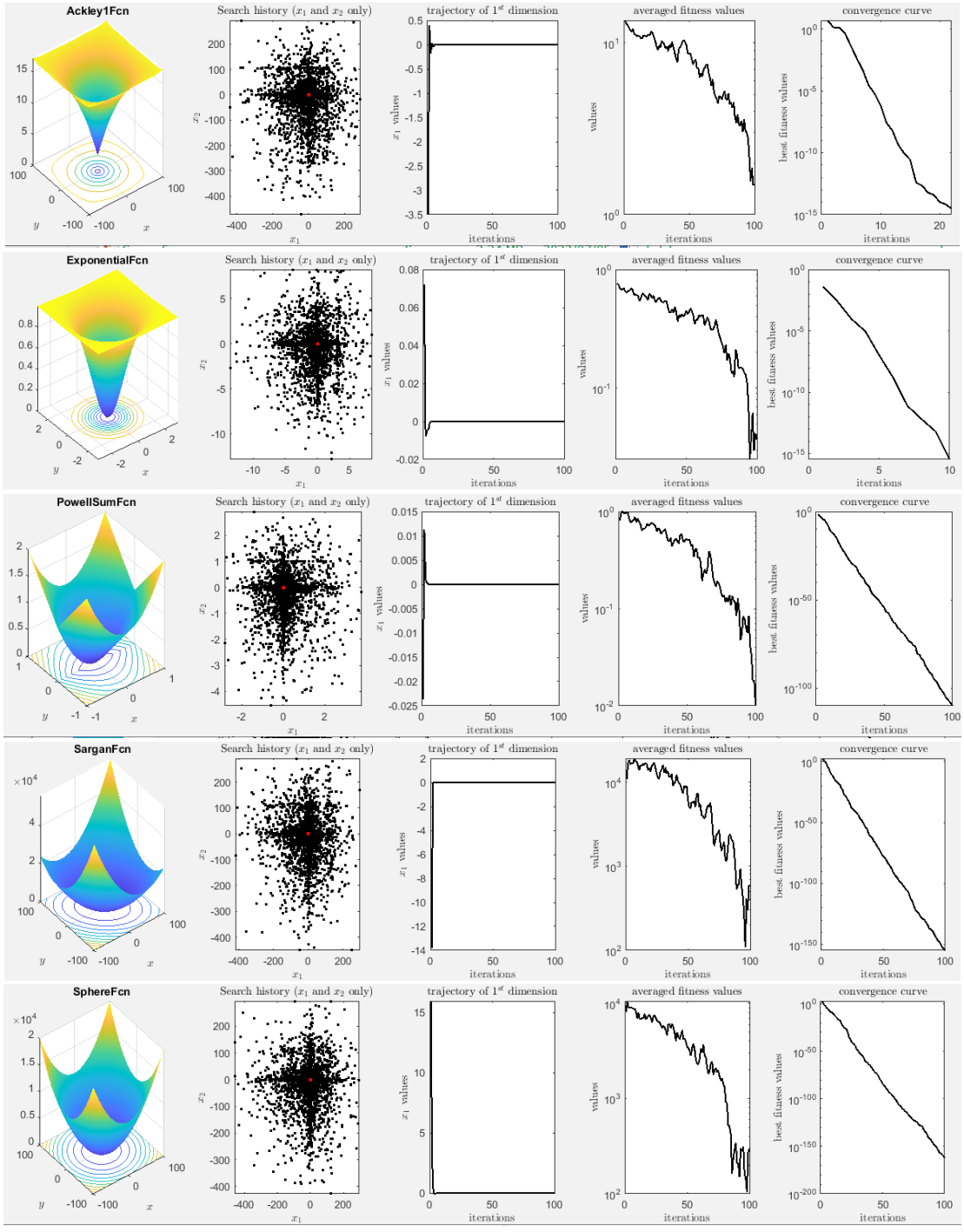
We can see from **Figure 5 to 8** that there were no similar rules for the number of cocks, Sphere function would want 8 cocks for better results, 18 for PowellSum function. But there is not a best number of cocks for multimodal benchmark functions. On the contrary, there is a strong hint that the number of hens should be smaller, which is a little weird and away from the peasants' will. Empirical results recommend

the number of cocks and hens might be 10 and 5 respectively.

Another parameter involved in CHC optimizer is the probability value p_1 for chickens to reinitialize themselves. Considering the inspiration and a same operation with the SMA[46], a same value with $p_1 = 1-z = 0.97$ was given without further empirical research.

4.3 Qualitative Analysis

To get a first glance at capabilities of the proposed CHC algorithm, we would carry on the qualitative experiments, the whole searching procedure would be shown and we can find its capability in optimizing. Qualitative analysis was usually carried out to show the searching history, trajectories, and changes of convergence values along with iterations. Representatives of benchmark functions would be chosen to show the fluctuation of capabilities of the proposed algorithm, results could be seen in **Figure 9**.



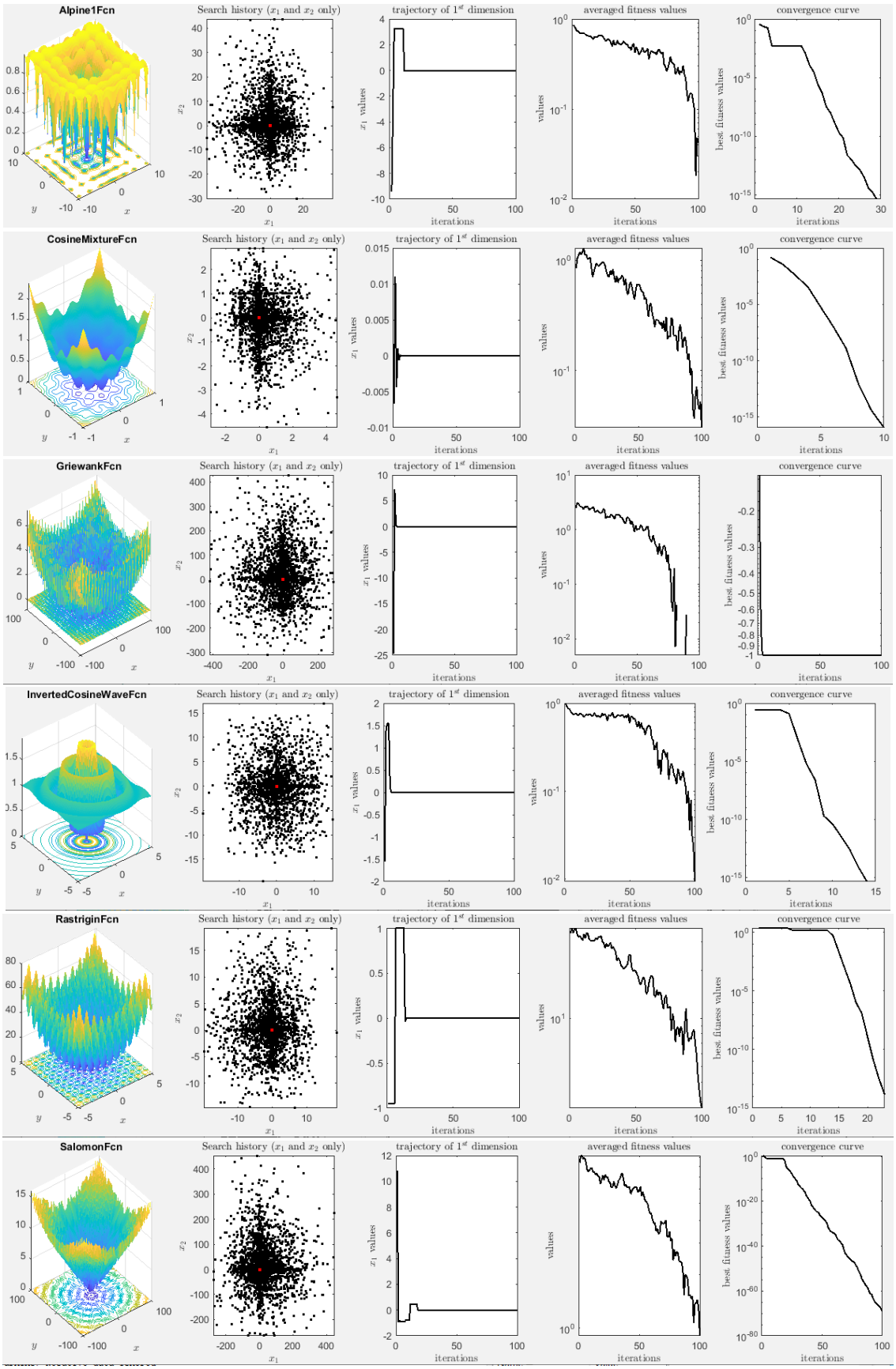


Figure 9. Qualitative results of the proposed CHC optimizer (dim = 10)

We can see that when optimizing the benchmark functions, the proposed CHC optimizer would quickly find the best global optima, either they are unimodal or multimodal. Most of the individuals would approaching the global optima, while some of them remain searching in the whole domain. The trajectory of the first dimensional would change dramatically, and the averaged fitness values remain fluctuate while decreasing rapidly. Convergence curves show that the overall capabilities were quite promising.

4.4. Intensification Analysis on Unimodal Benchmark Functions

All of the unimodal benchmark functions have only one local and also global optima around all of the definitional domains. A better algorithm would lead the individuals in swarms to approach the global optima more quickly in a whole. In order to evaluate the capability of the proposed CHC optimizer, optimization experiments on unimodal benchmark functions would be carried out to confirm the intensification of individuals approaching the global optima. This kind of experiments were usually called intensification analysis. Avoiding the influence of randomness, the algorithms together with the comparisons would be evaluated with 100 independent runs, statistical analysis would be carried out over the 100 results, and the best, worst, medians, means and the standard derivation calculated upon the 100 results would be introduced to balance the capability of the algorithms, as shown in **Table 6**.

Table 6. Intensification Results for Unimodal Benchmark Functions (dim = 10)

Fcn	Items	CHC	ALO	EO	GWO	MOA	PSO	SCA	WOA
	best	-4.44089E-16	0	0	0	0	0	0	0
	worst	**3.10862E-15**	4.493038381	5.63993E-14	0.587499278	1.972709423	0.587288391	0.09935742	1.48392E-11
F1	median	**0**	0	0	0	0	0	0	0
	mean	**1.28786E-16**	0.320937809	3.6815E-15	0.002937497	0.149649509	0.057408594	0.001562399	1.09517E-13
	std	**7.32343E-16**	0.847491335	9.76028E-15	0.041542472	0.385225768	0.150277459	0.00882281	1.07945E-12
	best	**0**	0	0	0	0	0	0	0
	worst	**0**	2.42006E-08	1.11022E-16	2.22045E-16	2.70787E-10	1.25833E-08	9.77745E-06	1.11022E-16
F2	median	**0**	0	0	0	0	0	0	0
	mean	**0**	6.49398E-10	5.55112E-19	1.72085E-17	1.51399E-12	1.90038E-10	8.12503E-08	2.77556E-18
	std	**0**	2.57494E-09	7.85046E-18	4.46558E-17	1.92141E-11	1.07264E-09	7.0983E-07	1.73768E-17
	best	**0**	0	0	0	0	0	0	0
	worst	**1.3644E-188**	0.002882117	7.0561E-41	2.18235E-37	6.73138E-18	6.41762E-09	1.51012E-08	5.52118E-33
F3	median	**0**	0	0	0	0	0	0	0
	mean	**8.8203E-191**	0.000153972	4.79006E-43	1.47413E-39	4.07949E-20	7.06394E-11	1.13902E-10	4.25081E-35
	std	**0**	0.000480291	5.02111E-42	1.57042E-38	4.78651E-19	4.95921E-10	1.10848E-09	4.16179E-34
	best	**0**	0	0	0	0	0	0	0
	worst	**1.8306E-139**	0.006403136	3.44724E-23	4.03977E-19	0.007121995	9.49522E-06	0.949028633	8.0982E-08
F4	median	**0**	0	0	0	0	0	0	0
	mean	**2.287E-141**	0.0001162	5.41733E-25	6.94682E-21	3.69026E-05	1.66648E-07	0.006584187	7.57315E-10
	std	**1.7639E-140**	0.00055635	3.32007E-24	3.89627E-20	0.000503562	8.9812E-07	0.067438429	6.65019E-09
	best	**0**	0	0	0	0	0	0	0
	worst	**8.9056E-146**	1.55407E-05	2.40337E-27	7.65955E-23	2.10377E-07	4.33227E-08	0.011284106	2.2145E-31
F5	median	**0**	0	0	0	0	0	0	0
	mean	**4.4781E-148**	4.21791E-07	2.08527E-29	1.92435E-24	2.96301E-09	5.13357E-10	0.000108135	2.39664E-33
	std	**6.297E-147**	1.61737E-06	1.74949E-28	9.53203E-24	2.11306E-08	3.28124E-09	0.000838642	1.80794E-32

We can see from **Table 6** that the proposed CHC optimizer would work well with all of the five representatives of the unimodal benchmark functions. Although at most times the best and median value would be the same, the worst, mean and standard derivation would always be the best among all of

the involved eight algorithms in this simulation experiments.

4.5 Diversification Analysis on Multimodal Benchmark Functions

Different from the unimodal benchmark functions, the multimodal benchmark functions would have multiple local optima with one global optimum among the definitional domain. Therefore, individuals in swarms might be trapped in local optima, and consequently, result in less capability in finding the global optimum. A better algorithm would give individuals the capability to escape the local optima of multimodal benchmark functions, the diversification capability has been most important and challenging for most of the algorithms, because most of the problems we need to solve are multimodal. Similar to the intensification analysis, diversification analysis would be carried on the multimodal benchmark functions, and it was also evaluated over 100 independent runs. Results of the diversification analysis were also averaged and shown in **Table 7**.

Table 7. Diversification Results for Multimodal Benchmark Functions (dim = 10)

Fcn	Items	CHC	ALO	EO	GWO	MOA	PSO	SCA	WOA
	best	**0**	0	0	0	0	0	0	0
	worst	**1.11022E-16**	0.714315866	2.16469E-07	0.000574877	6.21169E-07	0.006157561	0.650974501	0.958078169
F6	median	**0**	0	0	0	0	0	0	0
	mean	**3.88578E-18**	0.034750612	2.04766E-09	3.44209E-05	1.17461E-08	4.61031E-05	0.005670214	0.043951787
	std	**2.04549E-17**	0.119259201	1.94584E-08	0.00010312	7.21047E-08	0.000440615	0.053252062	0.190531017
	best	**0**	0	0	0	0	0	0	0
	worst	**0**	0.738921259	0	2.22045E-16	0.295627953	2.15529E-05	1.59274E-05	2.22045E-16
F7	median	**0**	0	0	0	0	0	0	0
	mean	**0**	0.047290961	0	7.77156E-18	0.012562025	1.1654E-07	1.8162E-07	2.22045E-18
	std	**0**	0.135592276	0	4.09097E-17	0.046328598	1.52433E-06	1.36923E-06	2.21486E-17
	best	**-9**	-8.291559873	-9	-9	-8.702942871	-8.65534616	-8.996296536	-9
	worst	**0**	0	0	0	0	0	0	0
F8	median	**0**	0	0	0	0	0	0	0
	mean	**-1.349999083**	-1.004630823	-1.336309729	-1.28456375	-1.206849347	-1.221748654	-1.126480052	-1.232913809
	std	**3.221704964**	2.436852208	3.189319233	3.074180805	2.887110107	2.919185029	2.736785953	2.998008179
	best	**0**	0	0	0	0	0	0	0
	worst	**0**	5.999456608	3.04577428	2.648210165	5.692145476	3.570910883	4.879958119	4.613283718
F9	median	**0**	0	0	0	0	0	0	0
	mean	**0**	0.593335511	0.20521879	0.254117746	0.593659569	0.330546931	0.077488714	0.157076811
	std	**0**	1.464468087	0.627736661	0.648621102	1.471589979	0.81439138	0.48537549	0.774552388
	best	**0**	0	0	0	0	0	0	0
	worst	**0**	61.68712991	0.994959883	8.249523747	19.48960963	18.09716113	32.21390213	65.95298852
F10	median	**0**	0	0	0	0	0	0	0
	mean	**0**	3.2634579	0.009949595	0.262892049	1.221346959	1.308472743	0.553854911	0.65006349
	std	**0**	8.847669599	0.099245642	1.086147854	3.408020822	3.397275623	3.34217031	5.634497437
	best	**0**	**0**	0	0	0	0	0	0
	worst	**0.099916083**	**0.799873346**	0.099873347	0.199873348	1.230460125	0.199890576	0.199986666	0.299873346
F11	median	**0**	**0**	0	0	0	0	0	0
	mean	**0.004993896**	**0.046481002**	0.014981002	0.015481002	0.076072716	0.023508325	0.016491524	0.021482268
	std	**0.021822511**	**0.123507126**	0.035751408	0.037596305	0.205148676	0.059284101	0.041042787	0.060008946

Same conclusion might be drawn with **Table 7**. The proposed CHC optimizer only fail to optimize Griewank function, while all of the other compared algorithms did not get much better either. The mean

and standard derivation values of CHC optimizer would be much better than other compared algorithm, which demonstrated that the proposed CHC algorithm would have better diversification capability when handling multimodal benchmark functions.

4.6 Convergence Capability Analysis

Convergence analysis would give a better and convincing review on the capability of the proposed algorithms. The best fitness values changed to be smaller and smaller along with the iterations, comparison might be made with fastest, or steadiest convergence and the least values. In order to show the convergence capabilities of the algorithms, the best fitness values would be obtained during iterations, and all of them would be averaged over 100 independent runs to reduce the influence of randomness. Values of the best fitness along with the iterations would be drawn and would be shown as curves, as shown in **Figure 10-20**, note that the dimensionality was setup with 10 for equal balancing.

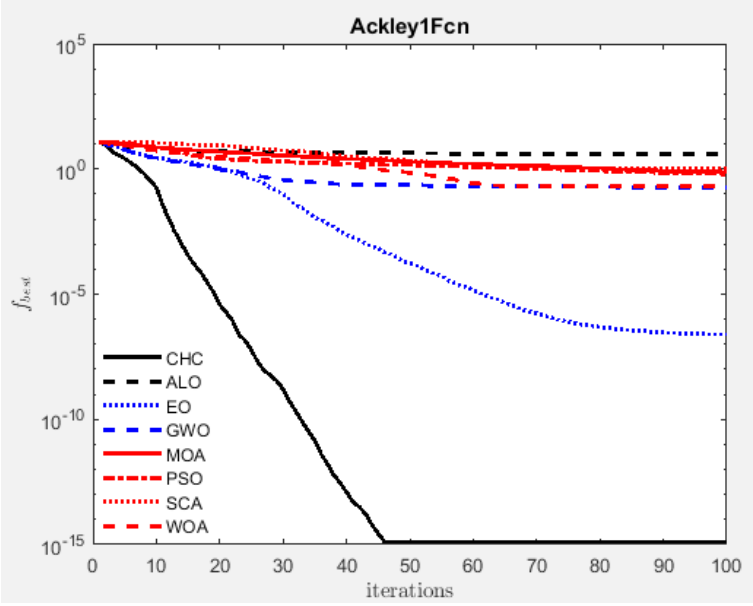


Figure 10. Convergence Analysis for Ackley 1 Function

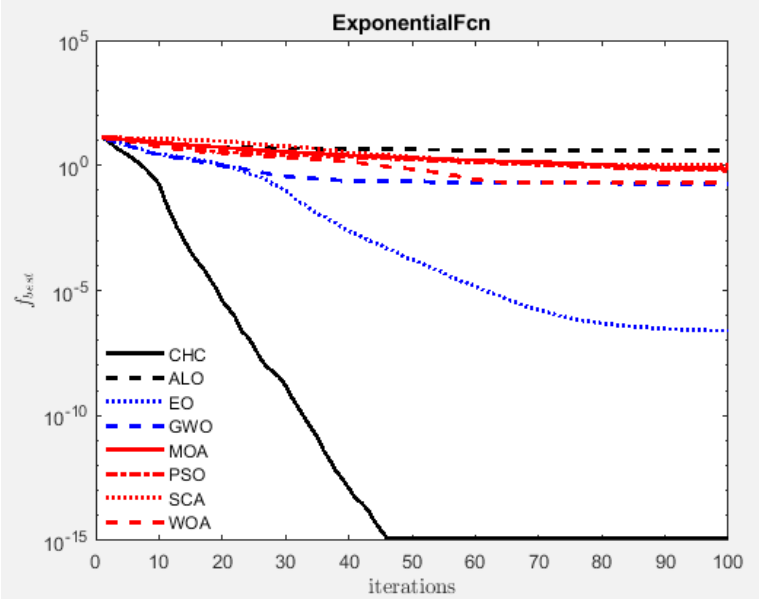


Figure 11. Convergence Analysis for Exponential Function

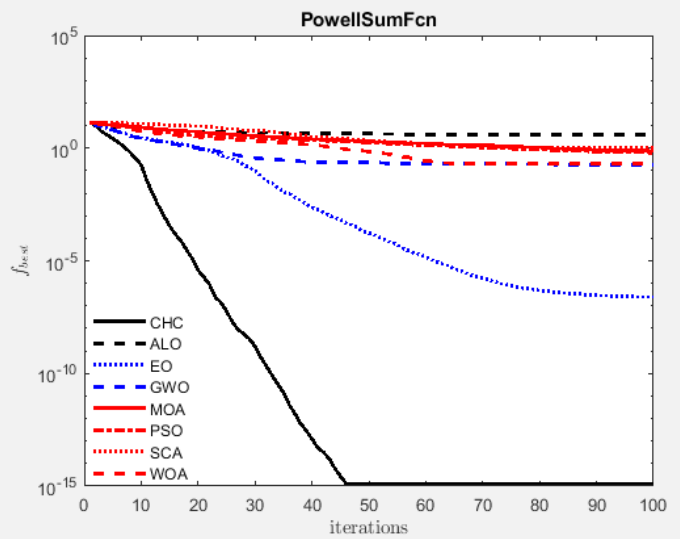


Figure 12. Convergence Analysis for Powell Sum Function

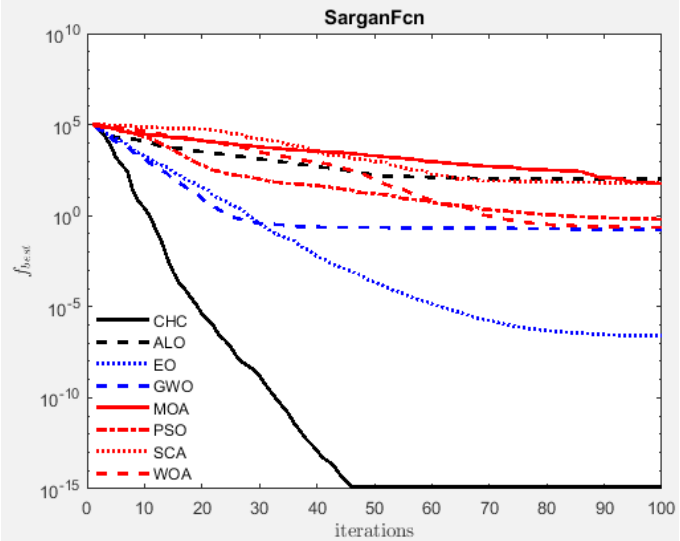


Figure 13. Convergence Analysis for Sargan Function

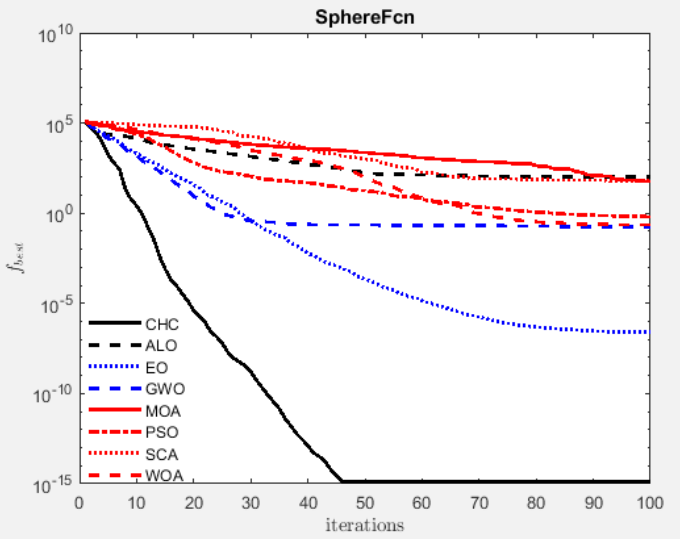


Figure 14. Convergence Analysis for Sphere Function

We can see with **Figure 10-14** that the proposed CHC optimizer would result in faster convergence, lower fitness values for unimodal benchmark functions, and the curves are also very steady. There would be very apparent difference in capability between the proposed CHC optimizer and the second best EO algorithm, and the rest six algorithms would be far away from satisfaction.

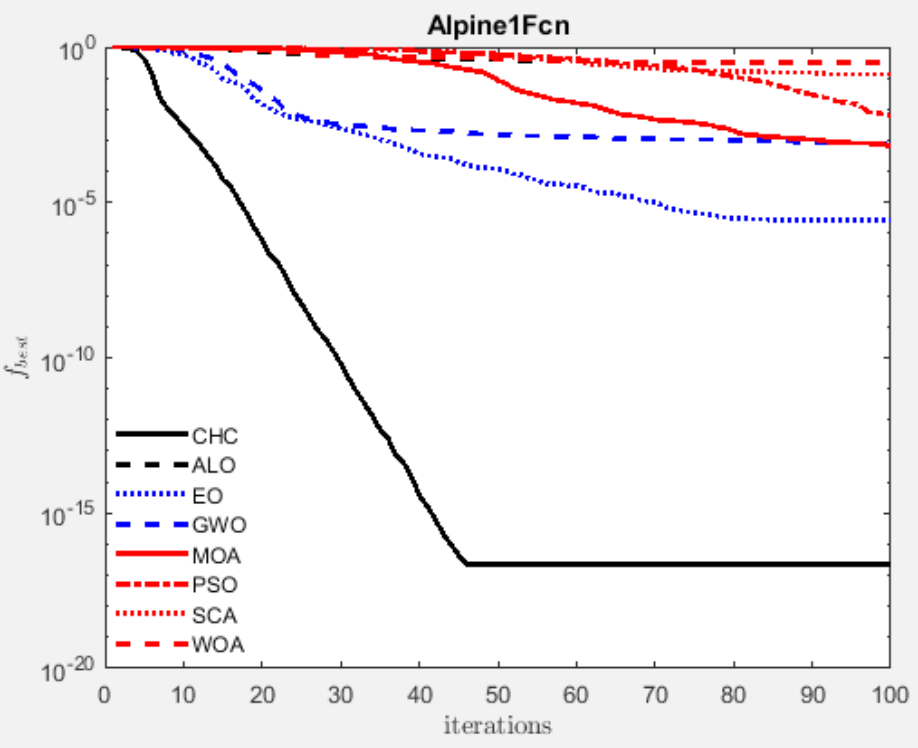


Figure 15. Convergence Analysis for Alpine 1 Function

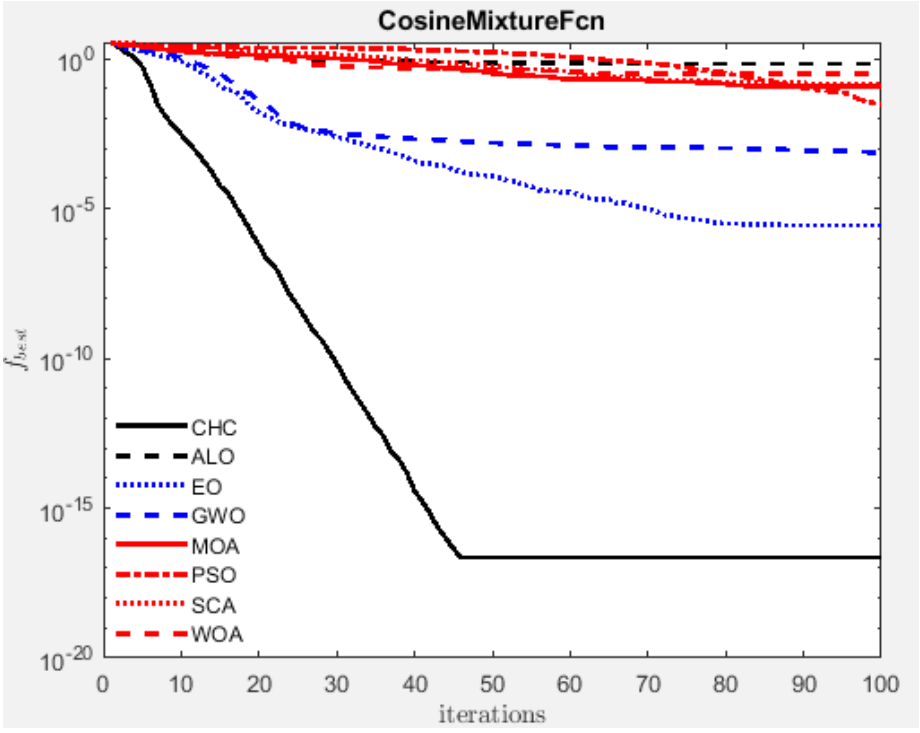


Figure 16. Convergence Analysis for Cosine Mixture Function

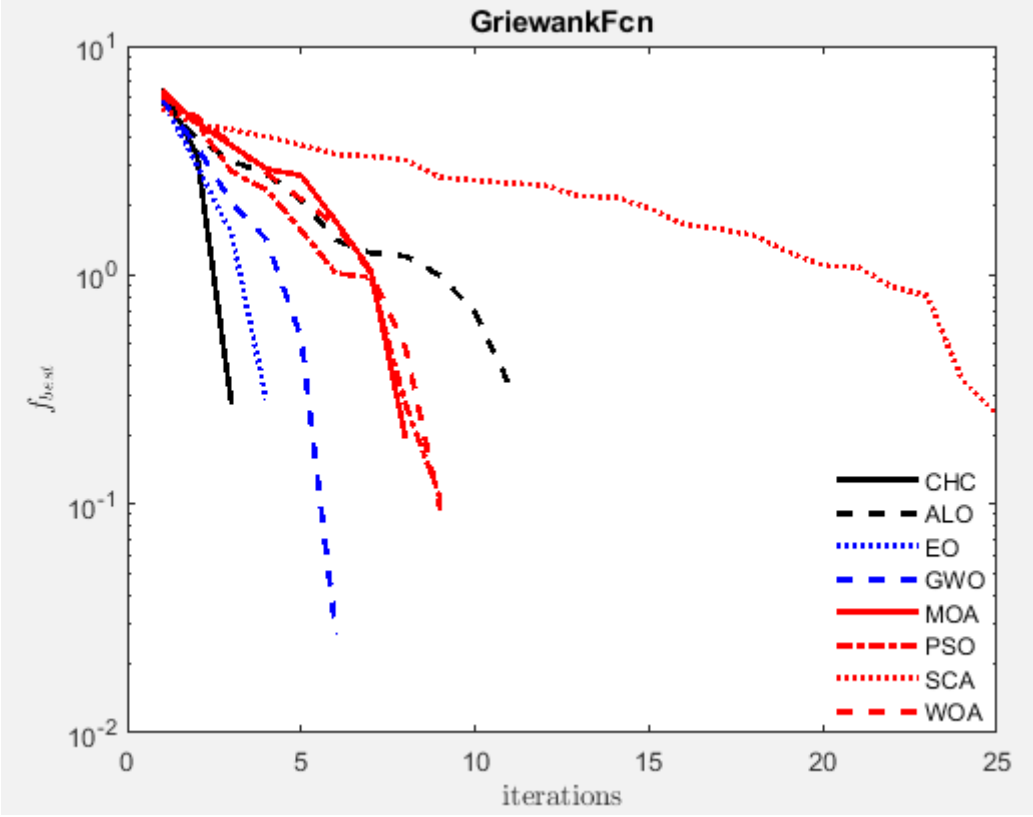


Figure 17. Convergence Analysis for Griewank Function

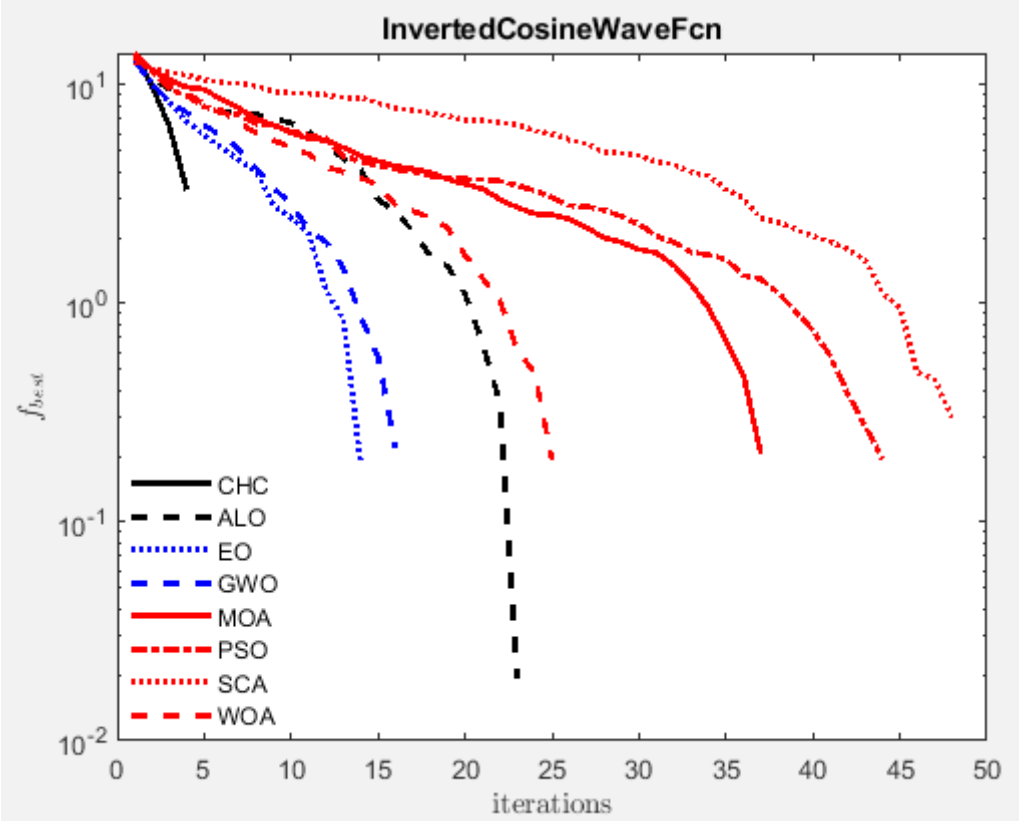


Figure 18. Convergence Analysis for Inverted Cosine-Wave Function

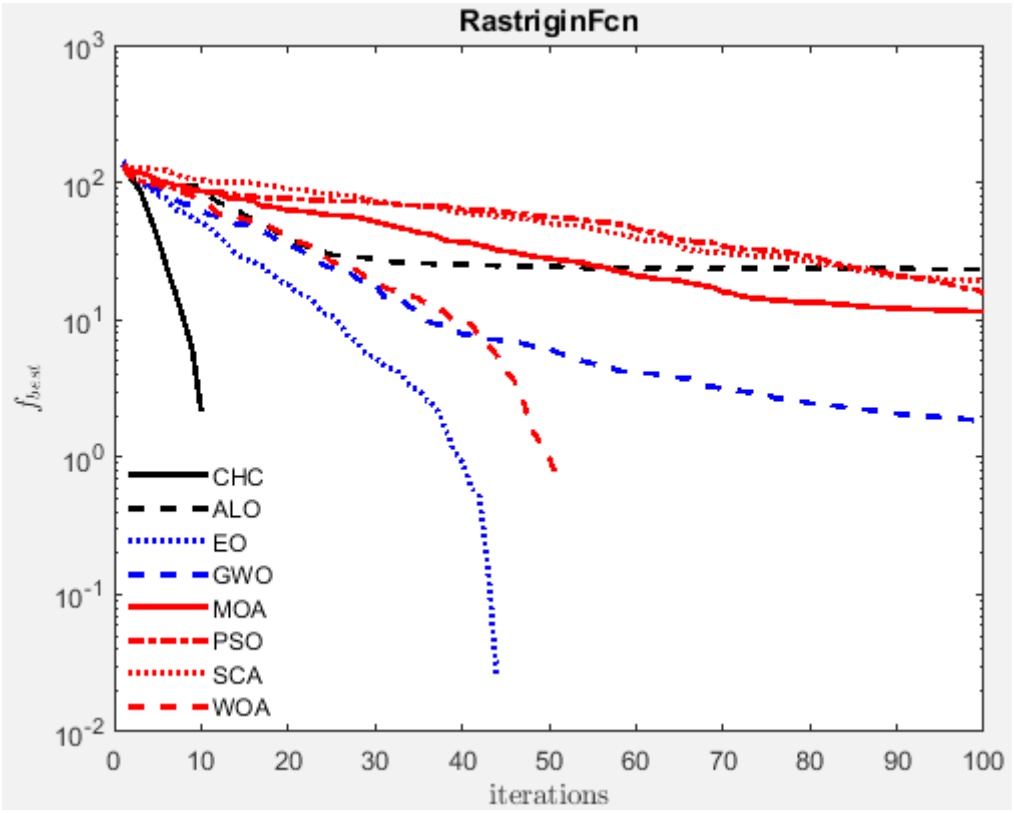


Figure 19. Convergence Analysis for Rastrigin Function

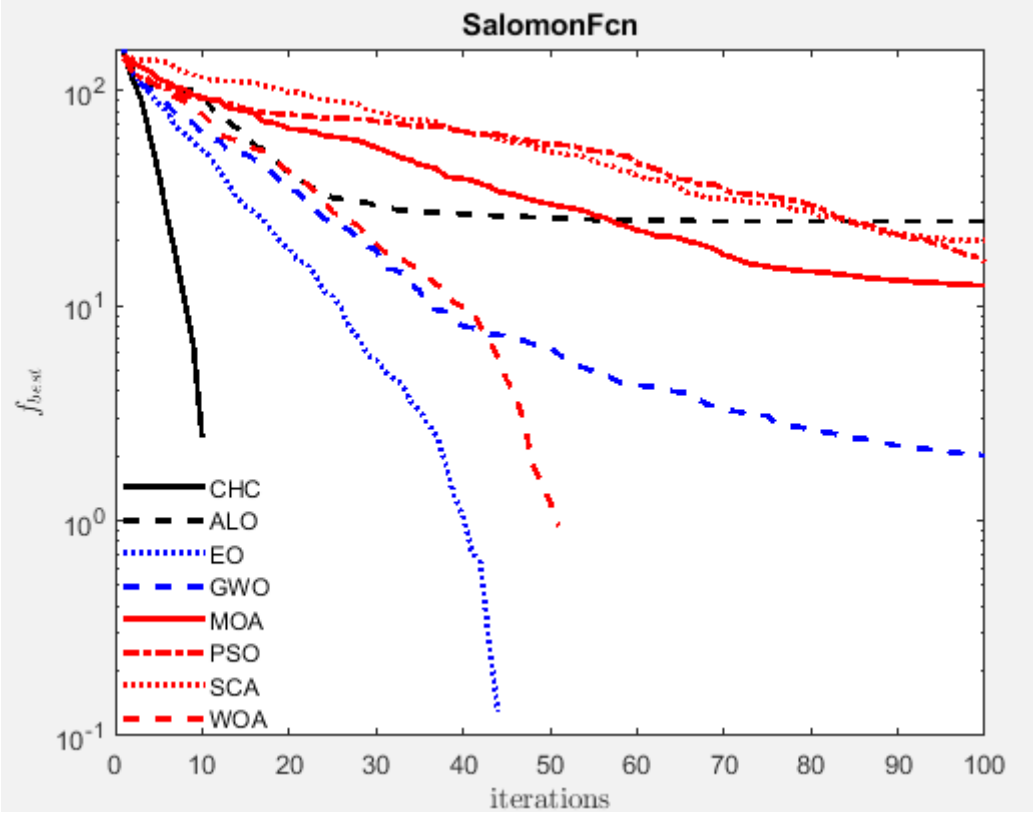


Figure 20. Convergence Analysis for Salomon Function

From **Figure 15 to 20**, we can draw the similar conclusions, the proposed CHC optimizer would also result in faster convergence rate, lower fitness values, and steadier convergence curves for multimodal benchmark functions. Meanwhile, we can also find that the second best algorithm performed in optimization was also the EO algorithm.

On the whole, the convergence capability experiments on benchmark functions, either unimodal or multimodal, confirmed that the proposed CHC optimizer would perform quite better than all of the compared algorithms involved in this paper. Considering the symmetry[47] of these benchmark functions, the proposed CHC optimizer would quickly find the zero optima with several rounds of iterations, which outperform all of the other algorithms significantly.

4.7 Scalability Analysis

Along with the development of modern science and technology, we are facing more and more complex problems, especially the dimensionality of the problems with big data is increasing dramatically. Decreasing the dimensionality would be an efficient way, however, most of the dimensionality remain very large in numbers after decreasing operations. Therefore, scalability capability is very important for an algorithm. We would continue to carry on scalability capability simulation experiments on the unimodal and multimodal benchmark functions to confirm whether the proposed CHC optimizer also have the best scalability or not. Results were shown in **Figure 21-31**.

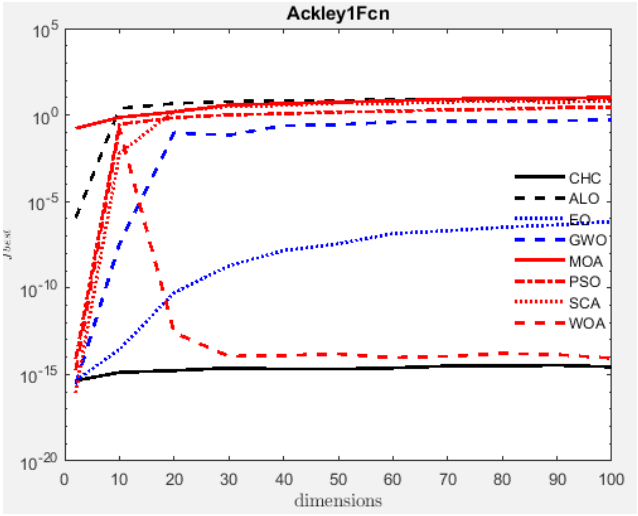


Figure 21. Scalability Experiment Results for Ackley 1 Function

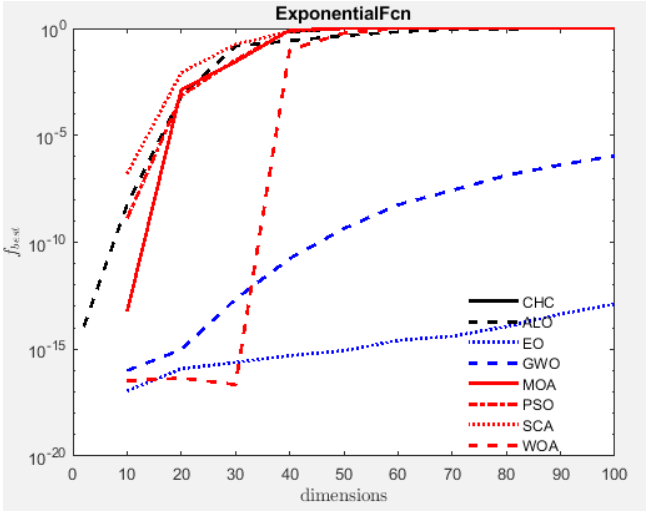


Figure 22. Scalability Experiment Results for Exponential Function

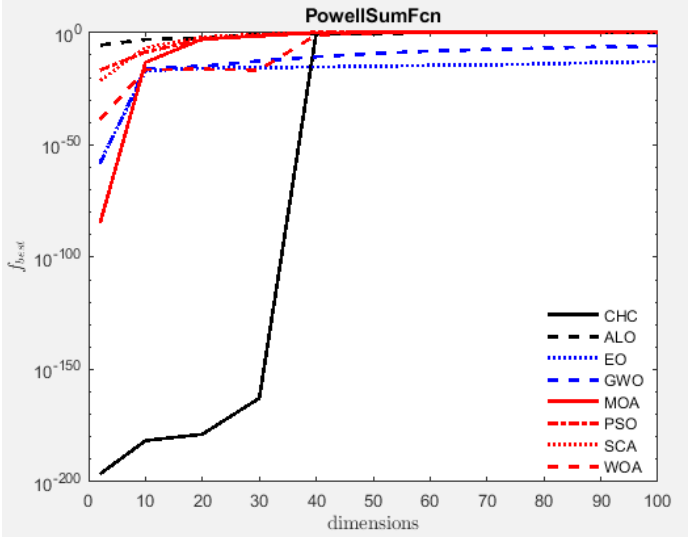


Figure 23. Scalability Experiment Results for Powell Sum Function

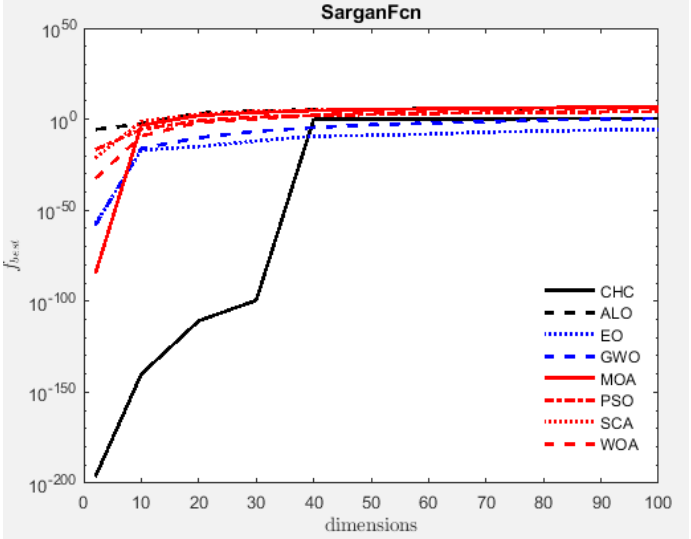


Figure 24. Scalability Experiment Results for Sargan Function

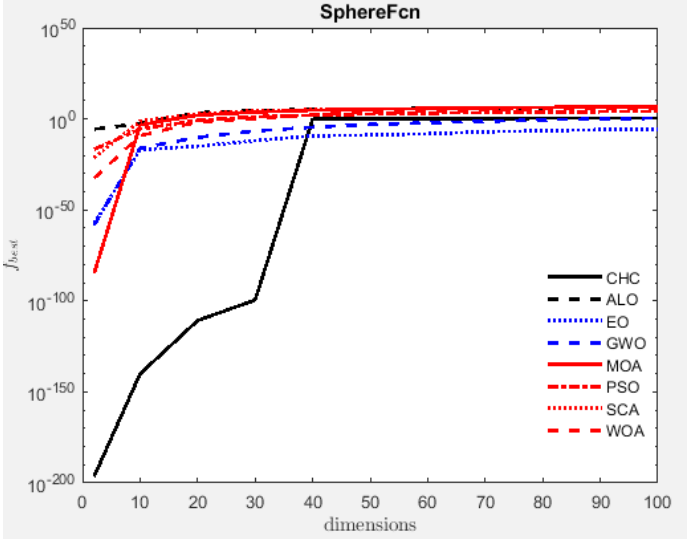


Figure 25. Scalability Experiment Results for Sphere Function

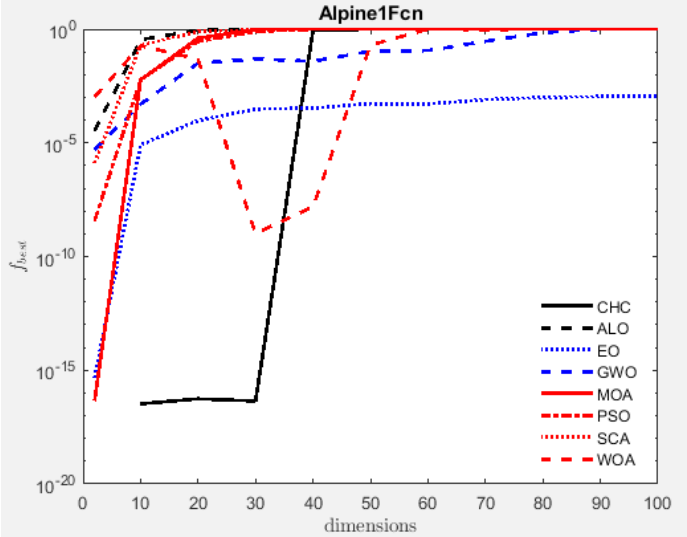


Figure 26. Scalability Experiment Results for Alpine 1 Function

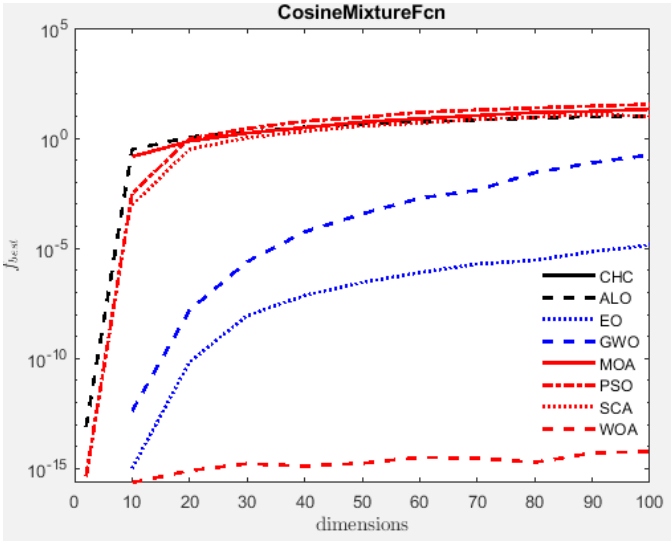


Figure 27. Scalability Experiment Results for Cosine Mixture Function

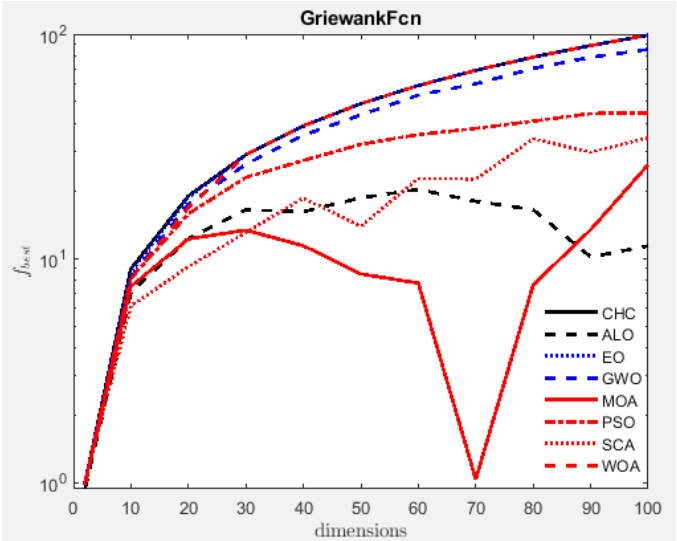


Figure 28. Scalability Experiment Results for Griewank Function

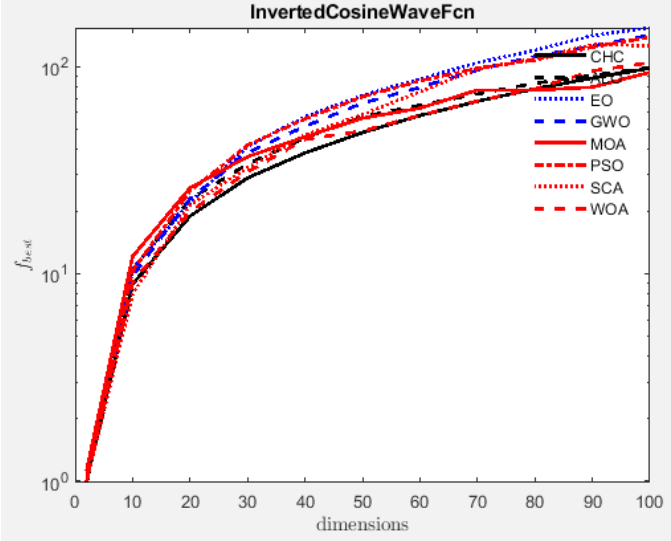


Figure 29. Scalability Experiment Results for Inverted Cosine-wave Function

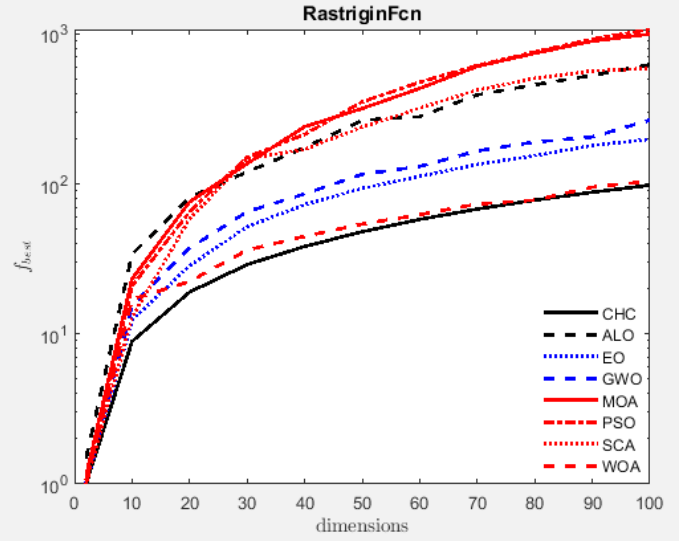


Figure 30. Scalability Experiment Results for Rastrigin Function

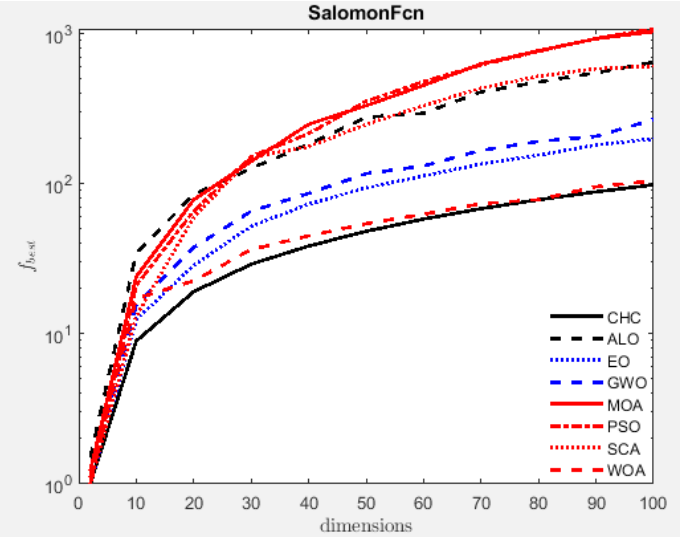


Figure 31. Scalability Experiment Results for Salomon Function

From **Figure 21 to 25**, we can see that the proposed CHC optimizer has the better scalability capability for unimodal benchmark functions. However, the scalability would decrease its superiority along with the increasing in dimensionality.

While when optimizing the multimodal benchmark functions, see **Figure 26 to 31**, the proposed CHC optimizer would perform better at most times, especially on the Inverted Cosine-wave, Rastrigin, and Salomon function, the best fitness values increased along with the increase in dimensionality. Note that for Cosine Mixture function, the proposed CHC optimizer would always find the global optima, thus it could not be shown with semiology graph. And some of fitness values obtained when optimizing Griewank function turned to be smaller than 0,which meant that the individuals are approaching the zero base point, and consequently, the negative values cannot be shown with semiology graph.

We can see that the proposed CHC optimizer perform better at most times, especially when it was applied in optimizing multimodal benchmark functions.

4.8 Wilcoxon Rank Sum Test

Only qualitative and quantitative analysis carried out before could not support a definite conclusion that the proposed CHC optimizer would be the best choice in optimization. The final results including graphs and tables might be following a same distribution statistically. Therefore, statistical analysis should be carried out to confirm the capability for sure. In this experiment, Wilcoxon rank sum test would be carried out. Wilcoxon rank sum test would test the null hypothesis that two data are samples from continues distributions with equal medians, against the alternative that they are not. The confidence level $p = 0.05$ is adopted and used to balance the confidence, results were shown in **Table 8**.

Table 8. Wilcoxon rank sum test results (dim = 10)

Fun	ALO-CHC	EO-CHC	GWO-CHC	MOA-CHC	PSO-CHC	SCA-CHC	WOA-CHC	WOA-CHC
F1	0.000145036	0.000133341	0.000145036	0.000145036	0.000145036	0.000145036	0.000520732	5.93632E-05
F2	6.38644E-05	0.167488756	9.66052E-05	6.38644E-05	6.38644E-05	6.38644E-05	0.033589681	0.368120251
F3	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.427355314
F4	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672
F5	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672	0.000182672
F6	9.28237E-37	9.28237E-37	9.28237E-37	9.28237E-37	9.28237E-37	9.28237E-37	9.28237E-37	9.28237E-37
F7	5.64002E-39	9.23054E-35	5.6185E-39	5.64002E-39	5.64002E-39	5.64002E-39	1.89073E-15	5.64002E-39
F8	2.98488E-37	4.88935E-33	2.5196E-34	1.08585E-36	1.80687E-35	4.3009E-37	1.47886E-33	2.98488E-37
F9	5.64002E-39	5.64002E-39	5.64002E-39	5.64002E-39	5.64002E-39	5.64002E-39	5.17585E-35	5.64002E-39
F10	5.64002E-39	5.6396E-39	5.64002E-39	5.64002E-39	5.64002E-39	5.64002E-39	4.04774E-36	5.64002E-39
F11	2.56214E-34	0.817392297	1.84781E-12	2.562E-34	2.82325E-32	1.21239E-33	1.28813E-06	2.56214E-34

We can see from **Table 8** that when carried on the Wilcoxon rank sum test with the proposed CHC optimizer and other famous algorithm in literature, every p values would be very small and far away from 0.05, therefore, we can confirm that statistically, there is an absolute difference and the proposed CHC optimizer would perform quite better than other compared algorithms involved in this paper.

4.9 Experiments on CEC17 Problems

In this section, we would carry on more simulation experiments to verify whether the proposed CHC algorithm could perform well in optimizing some hard problems reported in competitive evolutionary computation competitions. CEC17 would be involved and 30 functions including 2 unimodal, 7 simple multimodal, 10 hybrid, and 10 composition functions, see reference for details[48]. The best, worst, median, mean, and standard derivation would be reported according to the same limitation with 51 runs. Results were shown in **Table 9** to **Table 10**.

Table 9. The results of CHC algorithm for D = 10 with 100000 FES

Fun	Best	Worst	Median	Mean	Std.
F1	6.667869E+02	1.274083E+04	1.068429E+04	9.266645E+03	3.829495E+03
F3	3.000000E+02	3.000000E+02	3.000000E+02	3.000000E+02	2.464672E-13
F4	4.000045E+02	4.545826E+02	4.008051E+02	4.023625E+02	7.543207E+00
F5	5.059698E+02	5.229091E+02	5.149248E+02	5.148009E+02	4.034218E+00
F6	6.000001E+02	6.002687E+02	6.000039E+02	6.000305E+02	6.038354E-02
F7	7.161178E+02	7.431276E+02	7.224435E+02	7.233280E+02	5.009584E+00
F8	8.039798E+02	8.218891E+02	8.109446E+02	8.113544E+02	3.914231E+00
F9	9.000000E+02	9.004544E+02	9.000000E+02	9.000128E+02	6.546874E-02
F10	1.145317E+03	1.966089E+03	1.485815E+03	1.482085E+03	1.495160E+02
F11	1.100995E+03	1.126950E+03	1.109950E+03	1.110057E+03	6.149058E+00
F12	4.157805E+03	2.000944E+05	5.151565E+04	5.073454E+04	2.504932E+04
F13	1.319419E+03	1.575359E+04	2.308495E+03	2.999680E+03	2.689039E+03
F14	1.404994E+03	1.446290E+03	1.428136E+03	1.427050E+03	1.068323E+01
F15	1.501309E+03	1.517415E+03	1.506051E+03	1.506283E+03	3.339916E+00
F16	1.600463E+03	1.638843E+03	1.611950E+03	1.616533E+03	1.065818E+01
F17	1.701328E+03	1.742377E+03	1.722818E+03	1.720080E+03	1.374882E+01
F18	1.860405E+03	3.461456E+04	9.111965E+03	1.187288E+04	8.492284E+03
F19	1.900205E+03	1.985923E+03	1.903184E+03	1.908941E+03	1.974955E+01
F20	2.000000E+03	2.025288E+03	2.018539E+03	2.013487E+03	9.418143E+00
F21	2.200000E+03	2.203083E+03	2.200000E+03	2.200526E+03	1.082929E+00
F22	2.228946E+03	2.304294E+03	2.302074E+03	2.300808E+03	1.030099E+01
F23	2.613200E+03	2.634860E+03	2.624715E+03	2.622748E+03	5.080640E+00
F24	2.500000E+03	2.774523E+03	2.751840E+03	2.670044E+03	1.215599E+02
F25	2.897743E+03	2.969455E+03	2.943826E+03	2.927652E+03	2.507156E+01
F26	2.600098E+03	3.008774E+03	2.987113E+03	2.958063E+03	7.120268E+01
F27	3.089006E+03	3.093541E+03	3.089940E+03	3.090158E+03	8.239444E-01
F28	3.100000E+03	3.410791E+03	3.165991E+03	3.168106E+03	6.596226E+01
F29	3.132463E+03	3.194244E+03	3.137048E+03	3.138696E+03	8.558521E+00

Table 10. The results of CHC algorithm for D = 30 with 300000 FES

Fun	Best	Worst	Median	Mean	Std.
F1	1.002743E+02	2.094177E+04	2.066590E+03	3.225119E+03	4.726046E+03
F3	3.014690E+02	1.308165E+04	3.056705E+02	1.780976E+03	2.765438E+03
F4	4.041702E+02	6.405106E+02	4.833369E+02	4.801856E+02	2.890381E+01
F5	5.456900E+02	7.062140E+02	5.900548E+02	5.941734E+02	3.352476E+01
F6	6.039508E+02	6.118782E+02	6.073941E+02	6.074558E+02	1.990752E+00
F7	8.341137E+02	9.941719E+02	8.907780E+02	8.961811E+02	3.506911E+01
F8	8.431949E+02	9.949737E+02	9.067794E+02	9.025595E+02	3.000447E+01
F9	1.017562E+03	2.933120E+03	1.294776E+03	1.395438E+03	3.412193E+02
F10	2.804519E+03	4.881721E+03	3.805101E+03	3.844801E+03	4.343582E+02
F11	1.126358E+03	1.299866E+03	1.192322E+03	1.190478E+03	4.064258E+01
F12	2.364703E+04	3.252586E+06	1.604198E+05	3.527546E+05	6.195819E+05
F13	2.879793E+03	6.340141E+04	8.343478E+03	2.183493E+04	2.390305E+04
F14	1.634616E+03	2.108098E+05	6.551532E+03	1.904756E+04	3.636945E+04
F15	1.934578E+03	4.287043E+04	6.053813E+03	8.083546E+03	9.082769E+03

Continuation Table:

Fun	Best	Worst	Median	Mean	Std.
F16	1.974949E+03	3.311952E+03	2.385478E+03	2.437195E+03	4.507006E+02
F17	1.915094E+03	2.521561E+03	2.140978E+03	2.150865E+03	1.531797E+02
F18	1.713541E+04	3.592743E+05	2.017845E+05	1.673665E+05	9.392012E+04
F19	2.104698E+03	5.653034E+04	5.727495E+03	1.030476E+04	1.238350E+04
F20	2.092499E+03	2.734337E+03	2.487346E+03	2.492678E+03	1.673817E+02
F21	2.200004E+03	2.473787E+03	2.402920E+03	2.398025E+03	4.967762E+01
F22	2.300000E+03	7.192550E+03	5.719411E+03	5.317860E+03	1.389934E+03
F23	2.728303E+03	2.807272E+03	2.770932E+03	2.772128E+03	1.830875E+01
F24	2.947777E+03	3.085029E+03	3.002499E+03	3.006595E+03	2.422285E+01
F25	2.883435E+03	2.940893E+03	2.886950E+03	2.887060E+03	8.164145E+00
F26	4.476400E+03	6.098907E+03	5.085284E+03	5.144216E+03	3.956519E+02
F27	3.195614E+03	3.229082E+03	3.200008E+03	3.205124E+03	8.242072E+00
F28	3.218014E+03	6.397741E+03	3.300007E+03	3.606791E+03	9.137416E+02
F29	3.437914E+03	4.246275E+03	3.906158E+03	3.906508E+03	2.346769E+02

Compared with the reported accepted algorithms proposed in CEC17, jSO[49], DES[50], and LSADE_SPACMA[51] were all performed well in finding the best results. However, the proposed CHC optimizer failed at most times. It can only find several best results for composite equations such as F23, F25-F29. Considering the large memory required by this experiments, we do not carry on further simulations, yet the failure could be confirmed for CEC competitive equations at the original version.

5. Experiments on Real-world Engineering Problems

Although we have carried out simulation experiments on unimodal or multimodal benchmark functions, including some composite functions from CEC competitions, the real capability of the proposed algorithms would remain in suspicion due to the ideal disposal of the benchmark functions. A common way to get the capability of the proposed algorithm more convincing is to carry on further experiments on some real-world engineering problems. The mathematics discipline of the real-world engineering problems is known yet without a given solution under some constraints.

In this section, we would carry on further simulation experiments on some classical real-world engineering problems. The real-world engineering problems could be described as the constraint problem with several equal constraints and non-equal constraints:

$$\text{Minimize: } f(x),\ x = \{x_1, x_2, \cdots, x_n\}$$
$$\text{Subject to: } g_i(x) \leqslant 0,\ i = 1, 2, \cdots, m$$
$$h_i(x) = 0,\ i = 1, 2, \cdots, n$$

Where $x_i \in [LB_i, UB_i]$ is the definitional domain of the given problem, note that LB_i, UB_i might be not the same values for every i-th parameter. $g_i(x)$ is called the non-equal constraints, the number of them m would be confirmed for every problem. $h_i(x)$ represents the equal constraints. Note that for all of the real-world engineering problems, m and n could be zero and thus the constraints might be all non-equal or equal.

The simplest way to solve the constraint problems is to formulate a new fitness function with penalty factors Pe_i and Pn_i:

$$F(X) = f(x) + \sum_{1}^{n} Pe_i \max\left\{h_i(x), \varepsilon\right\} + \sum_{1}^{m} Pn_i \times \max\left\{g_i(x), \theta\right\}$$

Where ε is a small parameter.

In order to find the best and convincing results, we would also carry on 10000 times of runs with 50 population size of the CHC swarms, and the best choice would be the final results for every real-world engineering problems.

5.1 Gear Train Design Problem

Gear train design problem is a very popular design optimization problem for verification of the algorithms. The goal of the problem as shown in **Figure 32**[52] is to

find a minimal gear ratio under the given constraint.

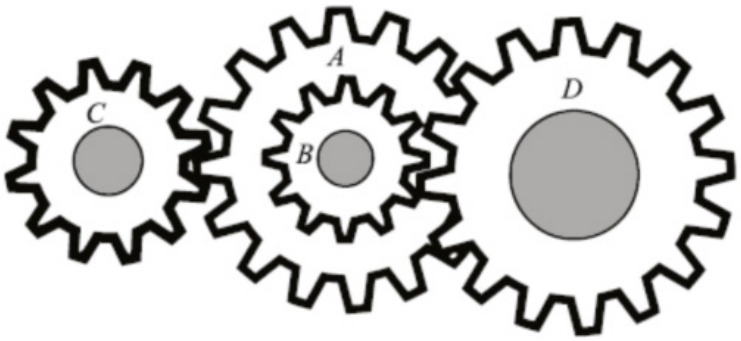


Figure 32. Sketch of the gear train design problem

Consider $X = \{x_1, x_2, x_3, x_4\} = \{n_A, n_B, n_C, n_D\}$

Minimize $f(X) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2$

Subject to $12 \leqslant x_1, x_2, x_3, x_4 \leqslant 60$

This design problem is a discrete problem that all of the parameter should be fixed to integer numbers. Results would also be compared with other related algorithms, as shown in **Table 11**.

Table 11. Comparison results of the gear train design problem

Algorithm	Optimal values for variables				Optimal cost
	n_A	n_B	n_C	n_D	
PSOSCALF[52]	49	19	16	43	2.7009e-12
CS[53]	43	16	19	49	2.7009e-12
ALO[36]	49	19	16	43	2.7009e-12
MFO[54]	43	19	16	49	2.7009e-12
MVO[55]	43	16	19	49	2.7009e-12
ABC[56]	19	16	44	49	2.78e-11
ALM[57]	33	15	13	41	2.1469e-8
CHC	43	19	16	49	2.70086e-12

We can see that the proposed CHC optimizer would find the least gear train ratio among the compared famous algorithm in literature.

5.2 Pressure Vessel Design Problem

The pressure vessel design is another well-known design optimization problem. As shown in **Figure 33**[52], the pressure vessel would store liquids under bigger pressure, the best parameters should result in a lowest cost of manufacturing and keeping safe of the pressure liquids.

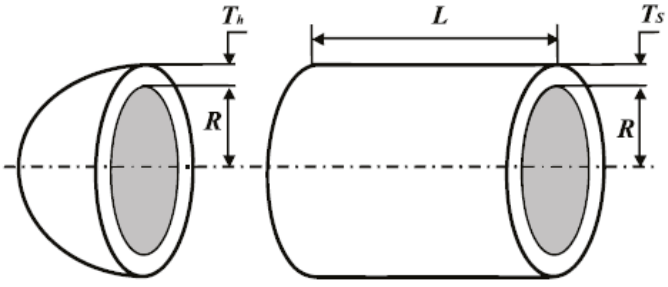


Figure 33. Sketch of the pressure vessel design problem

Consider: $x = \{x_1, x_2, x_3, x_4\} = \{T_s, T_h, R, L\}$

Minimize: $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

Subject to:

$g_1(x) = -x_1+0.0193x_3 \leqslant 0$

$g_2(x) = -x_3+0.00954x_3 \leqslant 0$

$g_3(x) = -\pi x_3^2 x_4-\frac{4}{3}\pi x_3^2+1296000 \leqslant 0$

$g_4(x) = x_4-240 \leqslant 0$

Limit to: $0 \leqslant x_1, x_2 \leqslant 99$, and $10 \leqslant x_3, x_4 \leqslant 200$.

Table 12. Comparison results of the pressure vessel design problem

Algorithms	Optimum values for variables				Optimum cost
	x_1	x_2	x_3	x_4	
CPSO[58]	0.8125	0.4375	42.091266	176.74365	6061.0777
WOA[13]	0.8125	0.4375	42.0982	176.6389	6059.7410
MFO[59]	0.8125	0.4375	42.0984	176.6366	6059.7143
AAO[60]	0.8125	0.4375	42.0985	176.6366	6059.7140
GWO[23]	0.8125	0.4345	42.0892	176.7587	6051.5639
SMA[46]	0.8260	0.4083	42.8155	167.9488	5970.0507
WSA[61]	0.78654289	0.39348835	40.75268075	194.78059812	5929.6218823
SMA[46]	0.7931	0.3932	40.6711	1962178	5994.1857
BSA[62]	0.7788	0.3946	40.3542	199.5188	5886.8000
CHC	**0.774549094**	**0.38320386**	**40.31961872**	**200**	**5870.123977**

We can see from **Table 12** that the proposed CHC optimizer would outperform all of the other compared algorithms.

5.3 Tension/Compression Spring Design Problem

Tension compression spring design optimization is another famous application of modern meta-heuristic algorithm. The goal of this problem is to find the best manufacturing structure that satisfied the given constraints with the minimum weight cost. This problem is described with **Figure 34** and formulated as follows.

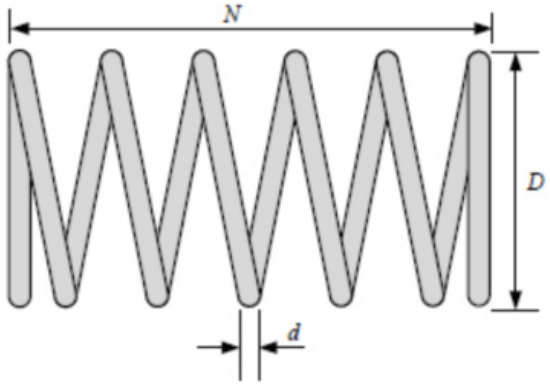


Figure 34. Sketch of the tension/compression design problem

Consider $x = \{x_1, x_2, x_3\} = \{d, D, N\}$

Minimize: $f(x) = (x_3+2)\, x_1^2 x_2$

Subject to: $g_1(x) = 1-\frac{x_2^3 x_3}{71785x_1^4} \leqslant 0$

$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \leqslant 0$

$g_3(x) = 1-\frac{140.45x_1}{x_2^3 x_3} \leqslant 0$

$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leqslant 0$

Optimized with the proposed CHC optimizer, the minimum weight 0.012665265 was obtained, which might be the best results among the reported values in literature, as shown in **Table 13**.

Table 13. Comparison results of the tension/compression spring design problem

Algorithms	Optimum values for variables			Optimum cost
	x_1	x_2	x_3	
GSA[18]	0.050276	0.323680	13.525410	0.0127022
SMA[46]	0.05	0.3174	14.0276	0.0127
WOA[13]	0.0512	0.3452	12.004	0.0127
AAO[60]	0.0517	0.3581	11.2015	0.0127
BSA[62]	0.0528	0.3835	9.8751	0.0127
CPSO[58]	0.051728	0.357644	11.244543	0.0126747
MFO[59]	0.051004457	0.36410932	10.868421862	0.0126669
PSOCSCALF[52]	0.05188366	0.36141614	11.018738	0.012665923
CHC	0.051654981	0.355898385	11.33717563	**0.012665265**

5.4. Three-bar Truss Design Problem

Three bar truss design optimization is a famous structure optimization problem in civil engineering. The basic goal of this problem is to find the minimum truss weight with constraints of tension, deformation, and buckling, as shown in **Figure 35**.

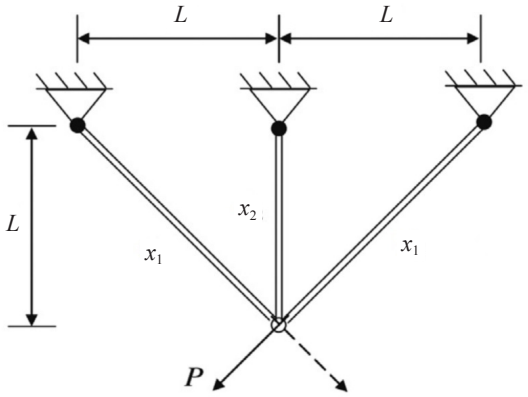


Figure 35. Sketch of the three-bar truss design problem

Consider x = $\{x_1, x_2\} = \{A_1, A_2\}$

Minimize: $f(x) = (2\sqrt{2}x_1+x_2)\times L$

Subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leqslant 0$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leqslant 0$$

$$g_3(x) = \frac{P}{\sqrt{2}x_2 + x_1} - \sigma \leqslant 0$$

And the limitation of x_1, x_2 is [0,1]. **Table 14** showed the latest results reported in literature. Although the efficient numbers were not larger than the results obtained with the GOA, but the overall final fitness value is the minimum best.

Table 14. Comparison results of the three-bar truss design problem

Algorithm	Optimal values for variables		$f(x)$
	A_1	A_2	
RAS[63]	0.795	0.395	264.3
MFO[54]	0.788244771	0.40946690578	263.9859797
CS[64]	0.78867	0.40902	263.9716
AOA[65]	0.79369	0.39426	263.9154
BA[66]	0.78863	0.40838	263.8962
GOA[67]	0.788897555578973	0.407619570115153	263.895881496069

Continuation Table:

Algorithm	Optimal values for variables		$f(x)$
	A_1	A_2	
MVO[55]	0.78860276	0.40845307	263.8958499
ALO[36]	0.788662816	0.40828313383	263.8958434
CHC	**0.78864941**	**0.408234008**	**263.8914911**

5.5. Welded Beam Design Problem

The welded beam design problem is another real-world engineering optimization problem in literature. The goal of this problem is to find the minimum manufacturing cost under the safety constraints, as shown in **Figure 36**.

Consider: $x = \{x_1, x_2, x_3, x_4\} = \{h, l, t, b\}$

Minimize: $f(x) = 1.10471x_1^2x_2+0.04811x_3x_4(14.0+x_2)$

Subject to:

$g_1(x) = \tau(x)-\tau_{max} \leqslant 0$

$g_2(x) = \sigma(x)-\sigma_{max} \leqslant 0$

$g_3(x) = \delta(x)-\delta_{max} \leqslant 0$

$g_4(x) = x_1-x_4 \leqslant 0$

$g_5(x) = P-P_c(x) \leqslant 0$

$g_6(x) = 0.125-x_1 \leqslant 0$

$g_7(x) = 1.104781x_1^2+0.04811x_3x_4(14.0+x_2)-5.0 \leqslant 0$

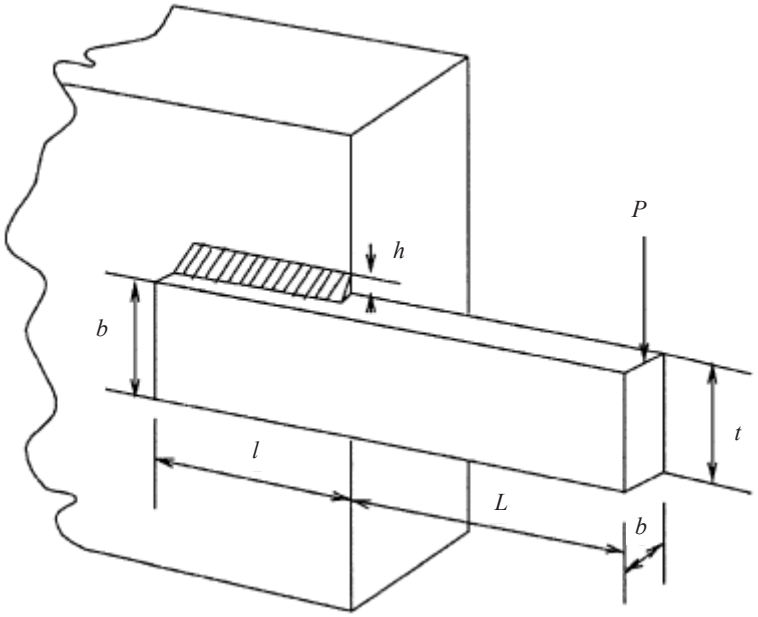


Figure 36. Sketch of the welded beam design problem

And:

$$0.1 \leqslant x_1 \leqslant 2,\ 0.1 \leqslant x_2 \leqslant 10,\ 0.1 \leqslant x_3 \leqslant 10,\ 0.1 \leqslant x_4 \leqslant 2$$

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}\ ,\quad \tau' = \frac{P}{\sqrt{2}x_1x_2},\quad \tau'' = \frac{MR}{J}$$

$$M = P\left(L+\frac{x_2}{2}\right),\quad R = \sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2}\ ,\quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}\qquad \delta(x) = \frac{6PL^3}{Ex_3^2x_4}$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

This optimization is most important because of the wide application in real-world engineering structures. The best fitness values changed dramatically as reported in literature. In this paper, a less fitness value is reported, as shown in **Table 15**.

Table 15. Comparison results of the welded beam design problem

Algorithm	Optimal values for variables				Optimal cost
	h	*l*	*t*	*b*	
FSA[68]	0.2444	6.1258	8.2939	0.2444	2.3811
HS[69]	0.2442	6.2231	8.2915	0.2443	2.3807
GSA[18]	0.182129	3.856979	10.00000	0.202376	1.87995
POS[70]	0.2251501	3.2405498	8.6290848	0.2256212	1.7963199
SCA[39]	0.2046987	3.6555450	9.2759949	0.2047442	1.7824134
SIO[71]	0.3314	2.0174	9.0459	0.2088	1.7621
GA[72]	0.208800	3.420500	8.997500	0.210000	1.748310
ESs[73]	0.199742	3.61206	9.0375	0.206082	1.7373
RO[20]	0.203687	3.528467	9.004233	0.207241	1.735344
CDE[74]	0.203137	3.542998	9.033498	0.206179	1.733462
HHO[40]	0.204039	3.531061	9.027463	0.206147	1.73199057
CPSO[58]	0.202369	3.544214	9.0487210	0.205723	1.73148
FA[75]	0.2015	3.5620	9.0414	0.2057	1.7312
GOA[67]	0.2073334	3.4495889	9.0018462	0.2073224	1.7306820
WOA[13]	0.2054	3.4843	9.0374	0.2063	1.7305
ALO[36]	0.2038936	3.5104491	9.0366268	0.2057296	1.7273787
MVO[55]	0.205463	3.473193	9.044502	0.205695	1.72645
GWO[76]	0.2056760	3.4783770	9.0368100	0.2057780	1.7262400
SSA[77]	0.2057	3.4714	9.0366	0.2057	1.72491
BSA[62]	0.2057	3.4706	9.0366	0.2057	1.7249
WSA[61]	0.20572963	3.47048995	9.03572964	0.20572964	1.72485254
MFO[54]	0.2057	3.4703	9.0364	0.2057	1.72452
SMA[46]	0.2054	3.2589	9.0384	0.2058	1.69604
CHC	0.205730638	3.253104311	9.036624135	0.205729646	**1.69524757**

6. Discussion and Conclusion

Inspired by hunting behaviors of cocks, hens, and chickens, the Cock-Hen-Chicken (CHC) optimizer was proposed in this paper, the proposed CHC optimizer would embrace the multiple updating principle and best candidates would be involved. Simulation experiments were carried out in details, either on unimodal or multimodal, CEC competitive equations or real-world engineering problems. Most of the simulation experiments confirmed the better performance, superiority, and capabilities.

The proposed cock-hen-chicken optimizer introduced multiple top candidates including the cocks and hens to update the positions. Eight updating disciplines were introduced to update their positions. Both of these two methods would increase the capability in convergent rate and diversification. Consequently, individuals in swarms would be more capable to find the global optima, and avoid being trapped in local optima, and fast convergence.

Efforts on the proposed CHC optimizer confirmed that when individuals in swarms would have multiple updating ways, the algorithm would perform well in optimization. Novel algorithms were paid great attention to in literature, experts in computer science were eager to find new ways with fast convergence, low residual errors, and a capability avoiding being trapped in local optima. For engineers, improvements were also important in application, meanwhile, they may also construct a new algorithm with various types

of algorithms and absorb the outstanding instincts.

However, although the overall results were promising, more efforts should be done with the CHC optimizer to find the reason why it failed with CEC17 competitive equations at most times. efforts should be made to clarify whether it was special or not. And furthermore, more applications on real-world optimization problems should be carried out in detail, for example, image segmentation, feature extraction, economic dispatch, discrete optimization problems, and so on.

Acknowledgements

The authors would like to thank the supports of the following projects: The Innovation Fund for Industry-University-Research in Chinese Universities with grant number 2021ZYA11012; the open project of Hubei Engineering Research Center for Specialty Flowers Biological Breeding with grant number 2022ZD006; the science research project of Jingchu university of technology with grant number YY202203, and the Hubei Provincial Central Leading Local Science and Technology Development Project with grant number 2022BGE262.

Data Availability

The code data used to support the findings of this study have been uploaded and deposited in the GitHub repository(https://github.com/gaozming/CHCoptimizer).

Conflict of Interest

No potential conflict of interest was reported by the authors.

References

[1] İnkaya T, Kayalıgil S, Özdemirel NE. Ant colony optimization based clustering methodology. *Applied Soft Computing*. 2015;28:301-11. https://doi.org/10.1016/j.asoc.2014.11.060.

[2] Wang RB, Wang WF, Xu L, Pan JS, Chu SC. An adaptive parallel arithmetic optimization algorithm for robot path planning. *Journal of advanced transportation*. 2021;2021:1-22. https://doi.org/10.1155/2021/3606895.

[3] Khatir S, Tiachacht S, Le Thanh C, Ghandourah E, Mirjalili S, Wahab MA. An improved Artificial Neural Network using Arithmetic Optimization Algorithm for damage assessment in FGM composite plates. *Composite Structures*. 2021; 273:114287. https://doi.org/10.1016/j.compstruct.2021.114287.

[4] Xu YP, Tan JW, Zhu DJ, Ouyang P, Taheri B. Model identification of the proton exchange membrane fuel cells by extreme learning machine and a developed version of arithmetic optimization algorithm. *Energy Reports*. 2021; 7:2332-2342. https://doi.org/10.1016/j.egyr.2021.04.042.

[5] Li Y, Li K, Yang Z, Yu Y, Xu R, Yang M. Stochastic optimal scheduling of demand response-enabled microgrids with renewable generations: An analytical-heuristic approach. *Journal of Cleaner Production*. 2022; 330:129840. https://doi.org/10.1016/j.jclepro.2021.129840.

[6] Li Y, Feng B, Wang B, Sun S. Joint planning of distributed generations and energy storage in active distribution networks: A Bi-Level programming approach. *Energy*. 2022; 245:123226. https://doi.org/10.1016/j.energy.2022.123226.

[7] Wang CN, Yang FC, Nguyen VT, Vo NT. CFD analysis and optimum design for a centrifugal pump using an effectively artificial intelligent algorithm. *Micromachines*. 2022; 13(8):1208. https://doi.org/10.3390/mi13081208.

[8] Deb S, Gao XZ. A hybrid ant lion optimization chicken swarm optimization algorithm for charger placement problem. *Complex & Intelligent Systems*. 2022: 2791–2808. https://doi.org/10.1007/s40747-021-00510-x.

[9] Fan GF, Zhang LZ, Yu M, Hong WC, Dong SQ. Applications of random forest in multivariable response surface for short-term load forecasting. *International Journal of Electrical Power & Energy Systems*. 2022; 139:108073. https://doi.org/10.1016/j.ijepes.2022.108073.

[10] Yuen MC, Ng SC, Leung MF. A competitive mechanism multi-objective particle swarm optimization algorithm and its application to signalized traffic problem. *Cybernetics and Systems*. 2020; 52(1):73-104. https://doi.org/10.1080/01969722.2020.1827795.

[11] Leung MF, Ng SC, Cheung CC, Lui AK. A new algorithm based on PSO for multi-objective optimization. 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan,

2015, 3156-3162, https://doi.org/10.1109/CEC.2015.7257283.

[12] Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*. 1997;1(1):67-82. https://doi.org/10.1109/4235.585893.

[13] Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in engineering software*. 2016; 95:51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

[14] John H. Holland. Genetic algorithms. *Scientific American*. 1992; 267(1):66-73. https://www.jstor.org/stable/24939139.

[15] Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*. 2015; 20(4):606-626. https://doi.org/10.1109/TEVC.2015.2504420.

[16] Zhang T, Geem ZW. Review of harmony search with respect to algorithm structure. *Swarm and Evolutionary Computation*. 2019; 48:31-43. https://doi.org/10.1016/j.swevo.2019.03.012.

[17] Abualigah L. Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Computing and Applications*. 2021; 33(7):2949-2972. https://doi.org/10.1007/s00521-020-05107-y.

[18] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Information Sciences*. 2009; 179(13):2232-2248. https://doi.org/10.1016/j.ins.2009.03.004.

[19] Pashaei E, Aydin N. Binary black hole algorithm for feature selection and classification on biological data. *Applied Soft Computing*. 2017; 56:94-106. https://doi.org/10.1016/j.asoc.2017.03.002.

[20] Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization. *Computers & structures*. 2012; 112:283-294. https://doi.org/10.1016/j.compstruc.2012.09.003.

[21] Li H, Wang S, Ji M. An improved chaotic ant colony algorithm. InAdvances in Neural Networks–ISNN 2012; 633-640. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31346-2_71

[22] Kennedy J, Eberhart R. Particle swarm optimization. InProceedings of ICNN'95-international conference on neural networks. 1995; 1942-1948. https://doi.org/10.1109/ICNN.1995.488968.

[23] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*. 2014; 69:46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[24] Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH. Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*. 2021; 157:107250. https://doi.org/10.1016/j.cie.2021.107250.

[25] Zhao J, Gao ZM, Chen HF. The simplified aquila optimization algorithm. *IEEE Access*. 2022; 10:22487-22515. https://doi.org/10.1109/ACCESS.2022.3153727.

[26] Akyol S, Alatas B. Plant intelligence based metaheuristic optimization algorithms. *Artificial Intelligence Review*. 2017;47:417-462. https://doi.org/10.1007/s10462-016-9486-6.

[27] Bingol H, Alatas B. Chaos based optics inspired optimization algorithms as global solution search approach. *Chaos, Solitons & Fractals*. 2020; 141:110434. https://doi.org/10.1016/j.chaos.2020.110434.

[28] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Computational Intelligence Magazine*. 2006; 1(4):28-39. https://doi.org/10.1109/MCI.2006.329691.

[29] Alweshah M, Rababa L, Ryalat MH, Al Momani A, Ababneh MF. African buffalo algorithm: Training the probabilistic neural network to solve classification problems. *Journal of King Saud University-Computer and Information Sciences*. 2022; 34(5):1808-1818. https://doi.org/10.1016/j.jksuci.2020.07.004.

[30] Yang XS. A new metaheuristic bat-inspired algorithm. InNature inspired cooperative strategies for optimization (NICSO 2010). 2010; 65-74. Berlin, Heidelberg: Springer Berlin Heidelberg.

[31] Zitouni F, Harous S, Belkeram A, Hammou LE. The archerfish hunting optimizer: A novel metaheuristic algorithm for global optimization. *Arabian Journal for Science and Engineering*. 2022; 47(2):2513-2553.

https://doi.org/10.1007/s13369-021-06208-z.

[32] Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-based systems*. 2020; 191:105190. https://doi.org/10.1016/j.knosys.2019.105190.

[33] Beyer HG, Schwefel HP. Evolution strategies–a comprehensive introduction. *Natural computing*. 2002;1:3-52. https://doi.org/10.1023/A:1015059928466.

[34] Amiri MH, Mehrabi Hashjin N, Montazeri M, Mirjalili S, Khodadadi N. Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm. *Scientific Reports*. 2024; 14(1):5032. https://doi.org/10.1038/s41598-024-54910-3.

[35] Abdollahzadeh B, Khodadadi N, Barshandeh S, Trojovský P, Gharehchopogh FS, El-kenawy ES, Abualigah L, Mirjalili S. Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning. *Cluster Computing*. 2024:1-49. https://doi.org/10.1007/s10586-023-04221-5.

[36] Mirjalili S. The ant lion optimizer. Advances in engineering software. 2015; 83:80-98. https://doi.org/10.1016/j.advengsoft.2015.01.010.

[37] Gao ZM, Zhao J, Li XR, Hu YR. An improved sine cosine algorithm with multiple updating ways for individuals. InJournal of Physics: Conference Series. 2020; 1678: 012079). IOP Publishing. https://doi.org/10.1088/1742-6596/1678/1/012079.

[38] Gao ZM, Zhao J, Li SR, Hu YR. The improved equilibrium optimization algorithm with multiple updating discipline. InJournal of Physics: Conference Series. 2020; 1682: 012054. IOP Publishing. https://doi.org/10.1088/1742-6596/1682/1/012054.

[39] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*. 2016; 96:120-133. https://doi.org/10.1016/j.knosys.2015.12.022.

[40] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: Algorithm and applications. *Future generation computer systems*. 2019; 97:849-872. https://doi.org/10.1016/j.future.2019.02.028.

[41] Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*. 2021; 376:113609. https://doi.org/10.1016/j.cma.2020.113609.

[42] Ayyarao TS, Ramakrishna NS, Elavarasan RM, Polumahanthi N, Rambabu M, Saini G, Khan B, Alatas B. War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization. *IEEE Access*. 2022; 10:25073-25105. https://doi.org/10.1109/ACCESS.2022.3153493.

[43] Meng X, Liu Y, Gao X, Zhang H. A new bio-inspired algorithm: chicken swarm optimization. InAdvances in Swarm Intelligence: 5th International Conference. 2014; 86-94. Springer International Publishing. https://doi.org/10.1007/978-3-319-11857-4_10.

[44] Zervoudakis K, Tsafarakis S. A mayfly optimization algorithm. *Computers & Industrial Engineering*. 2020;145:106559. https://doi.org/10.1016/j.cie.2020.106559.

[45] Gao ZM, Zhao J, Li SR. The improved slime mould algorithm with cosine controlling parameters. Journal of Physics: Conference Series, 2020, 1631: 012083. https://doi.org/10.1088/1742-6596/1631/1/012083.

[46] Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: A new method for stochastic optimization. *Future generation computer systems*. 2020; 111:300-323. https://doi.org/10.1016/j.future.2020.03.055.

[47] Gao ZM, Zhao J, Hu YR, Chen HF. The challenge for the nature-inspired global optimization algorithms: Non-symmetric benchmark functions. *IEEE Access*. 2021; 9:106317-106339. https://doi.org/10.1109/ACCESS.2021.3100365.

[48] Awad NH, Ali MZ, Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. InTechnical report 2016 Nov (pp. 1-34). Singapore: Nanyang Technological University Singapore.

[49] Brest J, Maučec MS, Bošković B. Single objective real-parameter optimization: Algorithm jSO. In2017 IEEE congress on evolutionary computation (CEC). 2017; 1311-1318. IEEE.

https://doi.org/10.1109/CEC.2017.7969456

[50] D. Jagodziński, J. Arabas.A differential evolution strategy: 2017 IEEE Congress on Evolutionary Computation (CEC), 5-8 June 2017, 2017[C]. 1872-1876. https://doi.org/10.1109/CEC.2017.7969529.

[51] Mohamed AW, Hadi AA, Fattouh AM, Jambi KM. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In2017 IEEE Congress on evolutionary computation (CEC). 2017; 145-152. IEEE. https://doi.org/10.1109/CEC.2017.7969307.

[52] Chegini SN, Bagheri A, Najafi F. PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. *Applied Soft Computing*. 2018; 73:697-726. https://doi.org/10.1016/j.asoc.2018.09.019.

[53] Bhargava V, Fateen SE, Bonilla-Petriciolet A. Cuckoo search: a new nature-inspired optimization method for phase equilibrium calculations. *Fluid Phase Equilibria*. 2013; 337:191-200. https://doi.org/10.1016/j.fluid.2012.09.018.

[54] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*. 2015; 89:228-249. https://doi.org/10.1016/j.knosys.2015.07.006.

[55] Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*. 2016; 27:495-513. https://doi.org/10.1007/s00521-015-1870-7.

[56] Karaboga D, Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. InInternational fuzzy systems association world congress. 2007; 789-798. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72950-1_77.

[57] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*. 1994; 116 (2): 405-411. https://doi.org/10.1115/1.2919393.

[58] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering applications of artificial intelligence*. 2007; 20(1):89-99. https://doi.org/10.1016/j.engappai.2006.03.003.

[59] Czerniak JM, Zarzycki H, Ewald D. AAO as a new strategy in modeling and simulation of constructional problems optimization. *Simulation Modelling Practice and Theory*. 2017; 76:22-33. https://doi.org/10.1016/j.simpat.2017.04.001.

[60] Baykasoğlu A, Akpinar Ş. Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems–Part 2: Constrained optimization. *Applied Soft Computing*. 2015; 37:396-415. https://doi.org/10.1016/j.asoc.2015.08.052.

[61] Varol Altay E, Alatas B. Bird swarm algorithms with chaotic mapping. *Artificial Intelligence Review*. 2020; 53(2):1373-1414. https://doi.org/10.1007/s10462-019-09704-9.

[62] Ray T, Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization*. 2001; 33(6):735-748. https://doi.org/10.1080/03052150108940941.

[63] Gandomi AH, Yang XS, Alavi AH. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*. 2013; 29:17-35. https://doi.org/10.1007/s00366-011-0241-y.

[64] Zhang YJ, Yan YX, Zhao J, Gao ZM. AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer. *IEEE Access*. 2022; 10:10907-10933. https://doi.org/10.1109/ACCESS.2022.3144431.

[65] Fister Jr I, Fister D, Yang XS. A hybrid bat algorithm. *Elektrotehniški vestnik*. 2013; 80 (3). https://doi.org/10.48550/arXiv.1303.6310.

[66] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*. 2017; 105:30-47. https://doi.org/10.1016/j.advengsoft.2017.01.004.

[67] Hedar AR, Fukushima M. Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of Global optimization*. 2006; 35:521-549. https://doi.org/10.1007/s10898-005-3693-z.

[68] Mahdavi M, Fesanghary M, Damangir E. An

improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*. 2007; 188(2):1567-79. https://doi.org/10.1016/j.amc.2006.11.033.

[69] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. InMHS'95. Proceedings of the sixth international symposium on micro machine and human science. 1995; 39-43. IEEE. https://doi.org/10.1109/MHS.1995.494215.

[70] Tzanetos A, Dounias G. Sonar inspired optimization (SIO) in engineering applications. *Evolving Systems*. 2020; 11(3):531-539. https://doi.org/10.1007/s12530-018-9250-z.

[71] Coello CA. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*. 2000; 41(2):113-127. https://doi.org/10.1016/S0166-3615(99)00046-9.

[72] Mezura-Montes E, Coello CA. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*. 2005; 9(1):1-7. https://doi.org/10.1109/TEVC.2004.836819.

[73] Huang FZ, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*. 2007; 186(1):340-356. https://doi.org/10.1016/j.amc.2006.07.105.

[74] Erdal FE. A firefly algorithm for optimum design of new-generation beams. *Engineering Optimization*. 2017; 49(6):915-931. https://doi.org/10.1080/0305215X.2016.1218003.

[75] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in engineering software*. 2014; 69:46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[76] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*. 2017; 114:163-191. https://doi.org/10.1016/j.advengsoft.2017.07.002.