Soft Computing (2020) 24:12027–12066
https://doi.org/10.1007/s00500-019-04646-4

METHODOLOGIES AND APPLICATION

Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization

Mohammad Zaeimi[1] · Ali Ghoddosian[1]

Published online: 2 January 2020

Abstract
In the last 3 decades, metaheuristic algorithms have received more popularity because of their superior performance to solve large and complex optimization problems. Most of these algorithms are inspired by biological phenomena, social behavior of animals, science and art. Among these four sources, the last one is utilized only by one algorithm. In this paper, we propose another novel art-inspired population-based metaheuristic, called color harmony algorithm (CHA), for solving the global optimization problems. The proposed method models its search behavior through combining harmonic colors based on their relative positions around the hue circle in the Munsell color system and harmonic templates. We utilize simultaneously four different fitness information to construct the hue groups, which improve search ability of the algorithm. CHA has two different phases including the concentration phase and the dispersion phase which are employed to explore and exploit the search space. The performance of the proposed method has been examined using several benchmark test functions commonly used in the literature. To show the effectiveness and robustness of the proposed method, the results are compared with those obtained using ten well-known metaheuristic algorithms. Also, the Wilcoxon Signed-Rank test is conducted to measure the pair-wise statistical performances of the algorithms. The results indicate that besides the simplicity of the proposed algorithm, CHA can outperform the other considered algorithms in terms of the convergence speed and the number of function evaluations.

Keywords Color harmony algorithm · Metaheuristic · Population-based optimization · Global optimization

1 Introduction

In the past few years, different methods have been developed to solve real-world optimization problems, which can roughly be classified as exact methods or metaheuristic algorithms. Exact methods, however, are usually not efficient and practical enough for complex optimization problems with high-dimensional search space or non-differentiable fitness function (Talbi 2009). On the other hand, general applicability and effectiveness are particular advantages of metaheuristics, which are higher-level heuristic algorithms. Moreover, metaheuristics do not necessarily require a proper initial guess which is highly important for convergence of the exact methods toward the optimal solutions (Blum and Roli 2003).

In the last 3 decades, metaheuristic algorithms have received more popularity because of their superior performance to solve large and complex optimization problems. They use iterative search techniques that aim to find near-optimal solutions at a reasonable computational cost, without being able to guarantee either feasibility or optimality.

There are different methods to classify and describe metaheuristic algorithms. Among them, the most widely used is based on the number of solutions these algorithms dealt with. More specifically, they can be classified into two main categories: single-solution-based algorithms and population-based algorithms (Talbi 2009; Xhafa and Abraham 2008).

Single-solution-based algorithms are solution-to-solution search methods in which a single solution is manipulated and transformed during the search process. Notable examples of this class of metaheuristic algorithms

Communicated by V. Loia.

✉ Ali Ghoddosian
aghoddosian@semnan.ac.ir

[1] Faculty of Mechanical Engineering, Semnan University, Semnan, Iran

are Simulated Annealing (SA) (Khachaturyan et al. 1981) and Tabu Search (TS) (Glover 1986). On the other hand, population-based search algorithms that generally incorporate evolutionary algorithms and swarm algorithms work with a set of solutions at the same time and generate new solutions iteratively.

The term evolutionary algorithm (EA) refers to a class of iterative optimization methods that simulate the process of natural evolution. They operate on a population of candidate solutions, called individuals, and provide information based on the fitness function. This population is subsequently modified by the two basic principles of the evolution: selection and variation. Individuals with better fitness value are selected with a higher probability. Then, using some variation operators such as the crossover and mutation, selected individuals are reproduced to form new solutions. This procedure continues until a stopping criterion is satisfied (Talbi 2009). The most common evolutionary algorithms are the evolutionary strategy (ES) (Fogel 1999), genetic algorithms (GA) (Holland 1975), harmony search (HS) (Geem et al. 2001), differential evolution (DE) (Storn and Price 1997) and gravitational search algorithm (GSA) (Rashedi et al. 2009).

The swarm-based algorithms are based on the collective behavior of animals living in groups such as insects, birds or fishes (Gandomi et al. 2013). The main principle of such algorithms is to work with a population of particles. Each particle is able to interact with other particles and also with its environment. Particles can communicate directly or through impacting in surroundings. The most well-known swarm-based algorithm are the particle swarm optimization (PSO) (Kennedy and Eberhart 1995), ant colony optimization (ACO) (Dorigo and Stützle 2004), artificial bee colony (ABC) (Karaboga and Basturk 2007), cuckoo search (CS) (Yang and Deb 2009) and bat algorithm (Yang 2010).

Since there is not a specific algorithm with a superior performance than the others to obtain the global optimal solution for all optimization problems, the proposal of new metaheuristic optimization algorithms is welcomed (Wolpert and Macready 1997). Designing a metaheuristic algorithm, providing a good balance between two contradictory aspects, exploration and exploitation, is an important challenge in finding the global optimum solution. Exploration is the process of visiting different areas of the search space to have a high probability of finding a good optimum. Exploitation, on the other hand, is the ability to find the optima around a good solution. All metaheuristic algorithms employ these aspects but they use different ways and operators (Talbi 2009; Gandomi et al. 2013).

It should be noted that all metaheuristic algorithms in the literature are often inspired by nature. There are different sources of inspiration for metaheuristics, which include biological phenomena, social behavior of animals, science and art (Gandomi et al. 2013). Among these four sources, the last one is utilized only by one algorithm, the harmony search algorithm. The motivation of this study is to propose a new optimization method with the use of art as a source of inspiration which provides acceptable results and efficient performance for a wide range of problems.

In this paper, we propose a novel population-based metaheuristic based on harmonic colors (i.e., color relationships), called color harmony algorithm (CHA), for solving the global unconstrained optimization problems with the mathematical objective function. Harmonic colors have special relationships that are aesthetically pleasing to the eye and are described by their relative positions around the color circle. The proposed method utilizes the Munsell technique and harmonic templates to produce new colors in the dispersion and concentration phases, respectively. The concentration phase is dealt with both exploration and exploitation of the search space, while another phase is introduced to develop only the exploration ability of the proposed algorithm. Switching between phases is done according to a population diversity threshold variable. Each phase is followed by an updating process of the population, and some new colors with some useful fitness information will be added or replaced on the hue circle. The process is repeated until the satisfaction of the termination conditions. The main goal of this process is to find the best agent of the hue circle. By comparing with ten high-quality metaheuristic algorithms, the performance of CHA is investigated by an array of experiments performed on 38 benchmark functions. The results clearly depict that CHA has superior performance in terms of convergence speed and number of function evaluations.

The remainder of this paper is organized as follows. In Sect. 2, the proposed algorithm and its characteristics are described. In Sect. 3, CHA is used for solving several benchmark functions and its performance is compared with ten well-known algorithms. Finally, conclusions are presented in Sect. 4.

2 Color harmony algorithm (CHA)

2.1 Background

The American art instructor and painter Albert Henry Munsell proposed a color system in the early twentieth century. The Munsell System is the first prosperous and generally accepted color system which is an internationally recognized standard to specify all possible colors and to determine the relationships among them (Cochrane 2014). The development of the Munsell System in the art world and the science community is proposed by Cochrane

(Cochrane 2014). By describing the special features of the system, she demonstrated how those features made it suitable for scientific color studies.

It is a color space that describes colors based on the three perceptual attributes including hue, value and chroma. Hue is an attribute which we give names to colors (e.g., red, yellow, green, etc.). Value indicates the lightness or darkness of a color, and chroma is the degree of purity or colorfulness of a color. The independence of these attributes was a novel feature of the system (Cochrane 2014).

Schematic representation of the Munsell System is shown in Fig. 1. The vertical line is called the neutral axis with black and white at the two ends. Colors between black and white along this axis are grays which have no hue. Hue changes around a horizontal circle which is called the hue circle; chroma varies radially from the neutral axis, increasing outward; and value is measured vertically from black to white (Das 2015). It can be said that the Munsell System consists of several hue circles with different value levels. Since certain hues reach their highest amount of the chroma at the different value levels, it has an irregular shape.

As shown in Fig. 2, the hue circle consists of ten hue groups, including five primary groups (red, yellow, green, blue, and purple) and five intermediate ones (yellow–red, green–yellow, blue–green, purple–blue and red–purple), which are placed arbitrarily in equal intervals around the hue circle. Note that the intermediate hues are made by mixing two adjacent primary hues (Cochrane 2014).

Each group contains ten different hues, thus the hue circle is divided into 100 equal sectors corresponding to them. Note that for a certain hue group, hues have different tendencies to their adjacent groups (e.g., in group red, from sector 5 to 10, hues are becoming more yellowish). In addition, each sector is divided into several segments by different chroma zones with different amounts of the chroma.

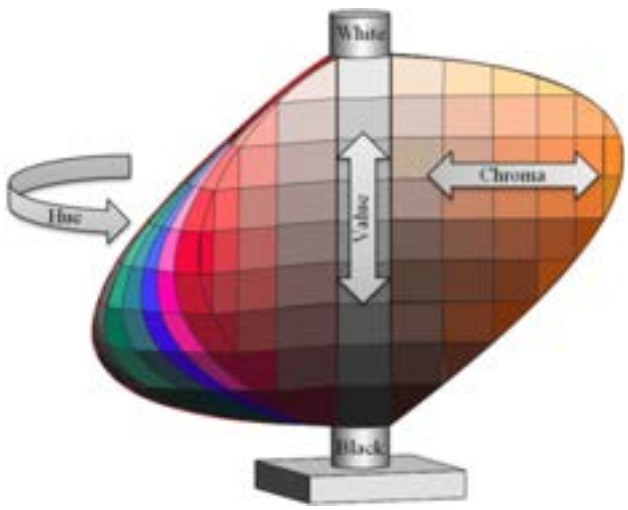

Fig. 1 Schematic representation of the Munsell color system

It should be noted that the purest hue of each group is located on sectors 5, 25, 45, 65 and 85 for the primary hue groups, and sectors 15, 35, 55, 75 and 95 for the intermediate ones (see Fig. 2); They are referred to the purest hue sectors, and colors with hues on these sectors are called agents in this paper. Note that the agents have the highest amount of chroma in each hue group, as shown by the chroma curve in Fig. 2.

In the next sections, we introduce our optimization algorithm, namely the color harmony algorithm (CHA), based on the Munsell System and harmonic colors. Similar to other metaheuristic algorithms, CHA maintains a population of solutions for the optimization problem. Each solution is considered as a color, and different properties of a color (e.g., various psychological/physical effects) are assumed to be described by the design variables of the optimization problem. CHA has two different phases including the concentration phase and the dispersion phase, which are based on the harmonic templates and the Munsell technique, respectively.

2.2 Methodology

For simplicity, we assume that all colors have the same value level. Therefore, the distinction between the colors is only due to the difference between two attributes, hue and chroma. The flowchart of the proposed algorithm is shown in Fig. 3.

This algorithm starts with an initial population of colors which are produced randomly in the search space. First, colors are placed on the hue circle according to their fitness value, and then the concentration phase is performed. At this step, the diversity of the population is decreased gradually. In the next step, if population diversity falls below a threshold value, the dispersion phase is executed to increase the diversity; otherwise, the concentration phase is continued. In each phase, if a purer form of a color is found, the hue circle is updated. Thus, each phase is followed by an updating process of the hue circle. Finally, the diversity threshold value is updated if the dispersion phase is done. In Fig. 3, iter_{cp} is the number of times that the concentration phase is performed continuously until the population diversity reaches D_{th}; and iter_{dp} indicates the number of times that D_{th} is updated or the dispersion phase is performed. CM_{temp} and CM are, respectively, the temporary and main color memory.

Note that, in order to locate the global optimum, the search space should be explored effectively and the premature convergence of the population toward a small region of the search space should be avoided. To achieve

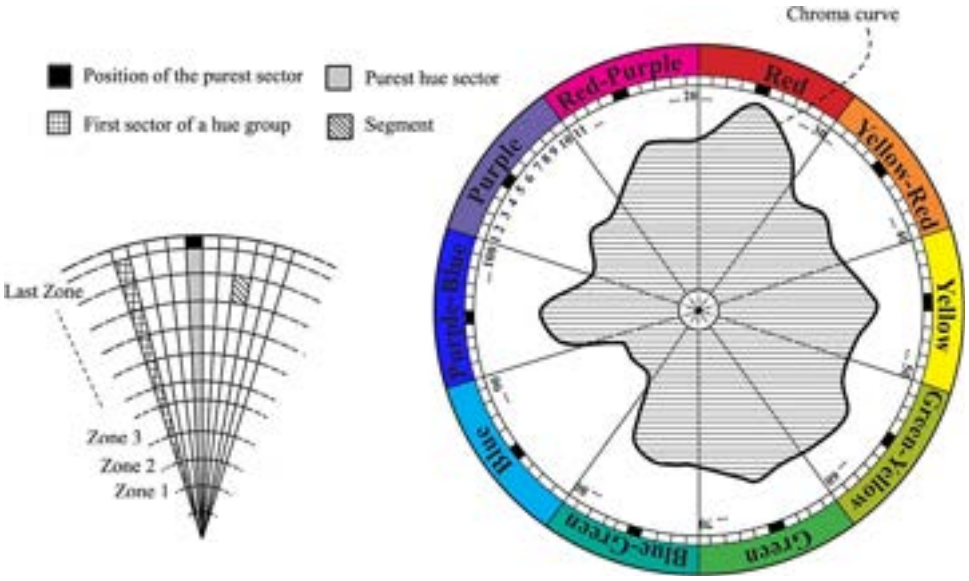

Fig. 2 The hue circle

these purposes, we present a selection strategy for the combination of the colors in the concentration phase and a replacement procedure in the dispersion phase.

The above procedure continues until the termination criterion is satisfied. CHA is terminated when population diversity reaches the final diversity value. However, other common criteria such as the maximum number of fitness evaluations could be applied. The main goal of this procedure is to find the best agent of the hue circle, which corresponds to the global best solution.

2.3 The initial population

As mentioned earlier, each solution candidate $X(i,j)$ is considered as a color containing a number of decision variables. Initially, colors are randomly distributed throughout the search space

$$X^0(i,j) = X^l(i) + \text{rand} \cdot \left(X^u(i) - X^l(i)\right) \quad \begin{matrix} i = 1,2,\ldots,N_v \\ j = 1,2,\ldots,N_c \end{matrix} \tag{1}$$

$X^0(i,j)$, $X^u(i)$ and $X^l(i)$ are, respectively, the initial value for the j_{th} color, upper and lower bound of the i_{th} variable; rand is a uniformly distributed number in the interval [0,1]; N_v and N_c correspond to the total number of variables and colors, respectively.

The initial population is generated in a way that population diversity is greater than or equal to the initial diversity threshold value, D_{th_i}. Population diversity measures the distribution of the solutions (colors) and denotes how much of the search space is covered. Satisfying the initial population diversity is essential to execute the next steps of CHA. In this paper, we utilize a modified definition of population diversity, proposed by Cheng and Shi (Cheng and Shi 2011). It is based on L_1 norm and some useful diversity information, including the solution diversity along each dimension and the diversity of the whole population. The definition is as follows:

$$C = \frac{1}{m}\sum_{i=1}^{m} X(i,j) \tag{2}$$

$$\boldsymbol{d} = \frac{1}{m}\sum_{i=1}^{m} |X(i,j) - C| \tag{3}$$

$$D = \frac{1}{n}\sum_{j=1}^{n} d_j \tag{4}$$

where C is the mean of the current solutions on each dimension; $\boldsymbol{d} = [d_1, d_2, \ldots, d_n]$, which measures solutions diversity based on L_1 norm for each dimension; and D determines the whole population diversity. Geometrically, C and $\boldsymbol{d}$ are the center of the population and the average distance of the solutions from the center in j dimension, respectively.

After generating the initial population, the distribution of the colors on the first chroma zone is performed. First, colors are sorted according to their fitness value,F_j, in increasing order. The best of them are selected as the agents and located randomly on the purest hue sectors. The remaining sorted colors are distributed randomly in such a way that each hue group contains ten colors.

2.4 Concentration phase

This phase deals with both the exploration and the exploitation of the search space and utilizes a selection strategy to make a balance between the local and global searches. In this phase, different harmonic colors are

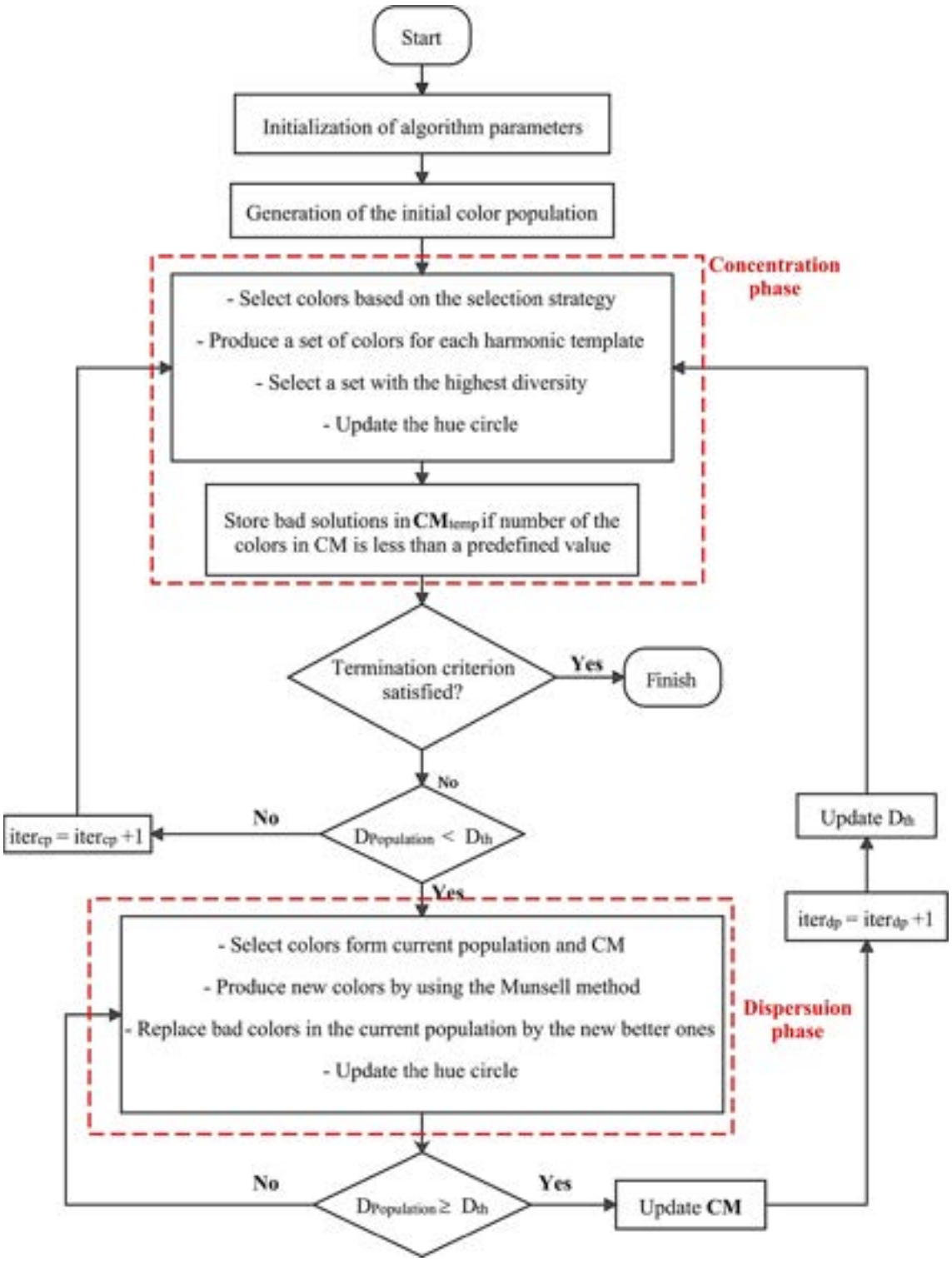

Fig. 3 Flowchart of the color harmony algorithm

selected based on the harmonic templates, and then the different contributions of their properties are combined together to produce new colors. In fact, our goal in this section is to find the properties of some new colors with better chroma than their components (i.e., two harmonic colors which are combined to produce a new color).

The word 'harmony' is taken from the Greek word harmonia, which means fit together. Harmonic colors are

sets of two or more colors which provide a pleasant human visual perception (Cohen-Or et al. 2006; Westland et al. 2007). Harmony among colors is highly affected by their relative positions around the hue circle and can be described by a set of harmonic templates. Figure 4a shows eight harmonic templates defined by hues of the hue circle. These templates are proposed by Matsuda based on an experimental investigation (Tokumaru et al. 2002; Matsuda 1995). Colors with hues that fall into the gray areas are considered to be harmonic. The precise sizes of the gray areas are proposed in Appendix A.

Since harmonic colors are not particular colors, the templates can be rotated by an arbitrary angle while the radial relationship stays fixed; Thus, a large variety of harmonic color sets would be available (Cohen-Or et al. 2006). Due to a respective similarity between *i* and *I* with *V* and *X* on the one hand and the lack of hue in template *N* on the other hand, templates *i*, *I* and *N* are ignored in this work.

In order to cover all hues by the gray areas, each template rotates ten times according to the position of the agents on the hue circle. In this way, all colors can have a chance to be present in the harmonic colors combinations. At each time, the amount of rotation is such that the sector containing an agent (the purest hue sector) coincides with a certain sector of the gray areas; this sector is defined as follows for each template (see Fig. 4b for template type V):

- The middle sector of the gray area in T and V type;
- The middle sector of the small gray area in L and Y type;
- The middle sector of the top (or bottom) gray area in type X.

2.4.1 Selection and combination procedure of colors

As mentioned earlier, this phase starts with the harmonic colors combination. For this purpose, after rotation of all the templates according to the position of a specific agent, selection of the harmonic colors from the gray areas is performed, and then N_{comb} new colors are produced for each template as follows:

$$X'(m,i) = r_1.X(a_m,i) + r_2.X(b_m,i) \quad \begin{matrix} m = 1,2,\ldots,N_{\text{comb}} \\ i = 1,2,\ldots,N_v \end{matrix} \tag{5}$$

and

$$r_2 = 1 - r_1 \tag{6}$$

where $\boldsymbol{X}'$ is the combination set; r_1 and r_2 are random numbers in (0, 1); N_{comb} is the total number of combinations; $X(a_m,i)$ and $X(b_m,i)$ are components of m_{th} combination;a_m and b_m indicate the sector numbers of each component; as will be seen later in this section, they are used for updating the hue circle. We store them in the index matrix $\boldsymbol{I}$ as follows:

$$\boldsymbol{I} = [a_m, b_m] \tag{7}$$

As mentioned before, this phase utilizes a selection strategy to control the exploration and exploitation of the search space, which is based on three types of color combinations as follows: (i) combination of the agents with the non-agents colors, (ii) combination of the best agent with the other agents, and (iii) combination of the non-agents colors. Using the first two types ensures that the solutions will converge to the optimal solution, while the diversification through the last one avoids the solutions being trapped into local optima. In other words, types (i) and (ii) increase the exploitation of the solutions, while type (iii) improves the exploration ability.

In order to achieve these types of combinations and also to control exploration and explanation, we define the power of agent (PA) parameter in the following form, which indicates how many times an agent should be appeared as a component in the combination set, $\boldsymbol{X}'$ [see Eq. (5)]:

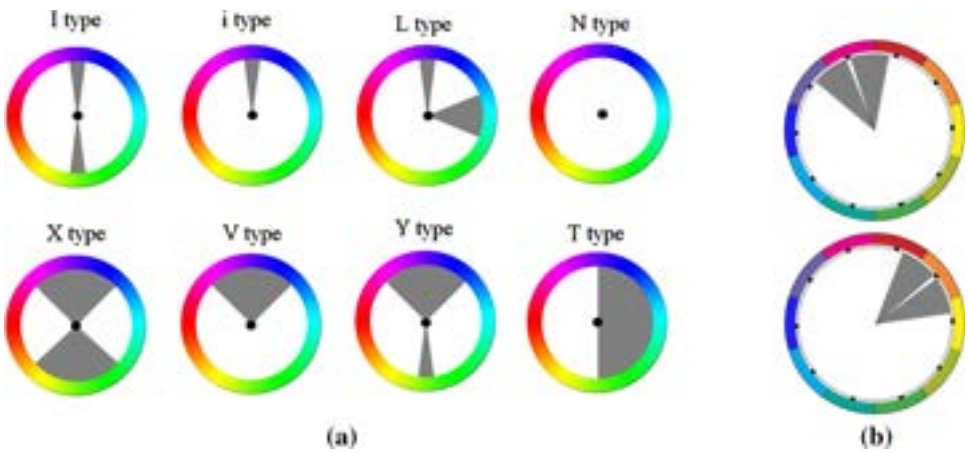

Fig. 4 **a** Harmonic templates on the hue circle; **b** rotation of the gray area

$$\mathrm{PA} = \min\left(N_{\mathrm{comb}}, \mathrm{floor}\left(\frac{\mathrm{iter}_{\mathrm{cp}}}{\mathrm{step}}\right)\right) \tag{8}$$

where

$$\mathrm{step} = \frac{N_{\mathrm{comb}} + 1}{N_{\mathrm{comb}} + N_i} \tag{9}$$

and

$$N_i = \min\left(N_{\mathrm{comb}}, \mathrm{floor}\left(\frac{\mathrm{iter}_{\mathrm{dp}}}{N_{D_{\mathrm{th}}}/N_{\mathrm{comb}}}\right)\right) \tag{10}$$

As can be seen, PA depends on four parameters: N_{comb}, $N_{D_{\mathrm{th}}}$, $\mathrm{iter}_{\mathrm{dp}}$ and $\mathrm{iter}_{\mathrm{cp}}$. $N_{D_{\mathrm{th}}}$ is the number of times that D_{th_i} (the initial diversity value) is multiplied by a constant reduction factor, damp, to be equal to the final diversity value, D_{th_f}; it can be defined as follows:

$$N_{D_{\mathrm{th}}} = floor\left(\frac{\log\left(\frac{D_{\mathrm{th}_f}}{D_{\mathrm{th}_i}}\right)}{\log(\mathrm{damp})}\right) \tag{11}$$

where D_{th_i} is experimentally set to $0.5\max\left(\boldsymbol{X}^{\boldsymbol{u}} - \boldsymbol{X}^{\boldsymbol{l}}\right)$. It can be seen from Eqs. (8) to (10) that for smaller values of $\mathrm{iter}_{\mathrm{dp}}$ (or in early iterations of the algorithm), switching from the exploration to the exploitation is done slowly. Therefore, the exploration power of the algorithm is high in the early iterations due to performing more global searches. By increasing $\mathrm{iter}_{\mathrm{dp}}$, the exploration power is decreased while exploitation power is increased. It should be noted that more exploitation is guaranteed in the last iterations because the diversity of the population becomes smaller.

After producing different combination sets using the above templates, the diversity of each combination set is determined using Eqs. (3) and (4) with respect to the above specific agent (i.e., $C = X_{\mathrm{agent}}$ in Eq. (3)). Then, a set with the highest diversity value is selected along with its index matrix to store in the new colors set $\boldsymbol{X}_{\mathbf{new}}$ and the new colors index set $\boldsymbol{I}_{\mathbf{new}}$, respectively.

By repeating this procedure for the other agents, the total number of new colors will be finally equal to $10N_{\mathrm{comb}}$ and the generation of $\boldsymbol{X}_{\mathbf{new}}$ and $\boldsymbol{I}_{\mathbf{new}}$ is completed. The fitness value of the new colors $\boldsymbol{F}_{\mathbf{new}}$ is then determined.

2.4.2 Updating of the hue circle in the dispersion phase

At the end of this phase, updating of the hue circle is performed by adding some new colors with some useful fitness information. As shown in Fig. 5, the following rules are used for locating a new color which is qualified to be added on the hue circle (i.e., for the next population):

i. It can only be placed on segments with better chroma zone than their components.

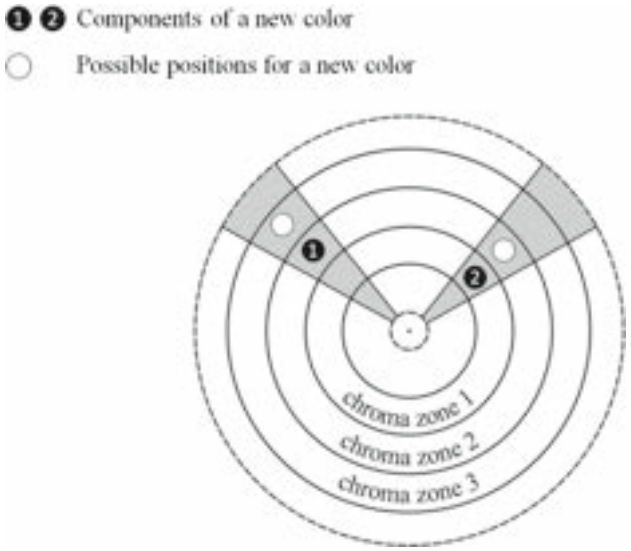

Fig. 5 Moving colors to a better chroma zone

ii. It can only be placed on one of the sectors of its components.

Updating process of the hue circle is described in the following steps:

Step (1) Delete new colors which have the fitness value worse than the worst value in the current population.
Step (2) Set $j = 1$
Step (3) Set $S = \emptyset$, $P = \emptyset$ and $\mathrm{sub} = \emptyset$.
Step (4) Using $\boldsymbol{I}_{\mathbf{new}}$ and $\boldsymbol{F}_{\mathbf{new}}$, find new colors which have a component on sector j and a better fitness value than $F(j)$; store them in S.
Step (5) If $S = \emptyset$, go to step (13); otherwise, for each member in S, find the sector number of another component using $\boldsymbol{I}_{\mathbf{new}}$ and then store them in P.
Step (6) For each sector number in P, if its related group have a worse agent than the group containing sector j, delete it.
Step (7) If $P = \emptyset$, go to step (13); otherwise, go to the next step.
Step (8) If P has more than one sector number, keep only the one belonging to the group with the lowest average fitness value; otherwise, go to step (9).
Step (9) Set p equal to the only sector number in P.
Step (10) Set sub equal to the member of S that has a component on sector p.
Step (11) Update the current population as follows: $X(j,:) = X_{new}(sub,:)$ and $F(j) = F_{new}(sub)$.
Step (12) Update $\boldsymbol{X}_{\mathbf{new}}$, $\boldsymbol{F}_{\mathbf{new}}$ and $\boldsymbol{I}_{\mathbf{new}}$ by deleting $\mathrm{sub}_{\mathrm{th}}$ row.
Step (13) If $j < N_c$, set $j = j + 1$ and go to step)3); otherwise, go to the next step.

Step (14) If $\boldsymbol{X}_{\mathbf{new}} \neq \emptyset$ and $N_{\mathrm{CM}} < k \times N_s$, add the rest of the new colors to the temporary color memory, $\boldsymbol{CM}_{\mathrm{temp}}$; otherwise, go to the next step.

Step (15) Update the agents of each hue group and then stop.

Since steps 3–12 are repeated for different values of j, a schematic representation of these steps is shown in Fig. 6.

As mentioned before, $\boldsymbol{CM}_{\mathrm{temp}}$ and $\boldsymbol{CM}$ are, respectively, the temporary and main color memories; and N_{CM} is the number of colors in $\boldsymbol{CM}$. k and N_s are integer numbers which are described in the next subsection. As can be seen in step (14), the new colors that have no opportunity to be present in the next population are added into $\boldsymbol{CM}_{\mathrm{temp}}$. This process will be continued until N_{CM} is less than $k \times N_s$. Note that the temporary color memory is used to constitute the main color memory which plays a very important role for enhancing the population diversity in the dispersion phase.

Since there are two possible positions for each new color (see Fig. 5), we utilize the following fitness information to choose one of them;

- Fitness value of the new color (in step (4));
- Fitness value of the components of the new color (in step (4));
- Fitness value of the agent of each hue group related to the components of the new color (in step (6));
- Average fitness value of each hue group related to the components of the new color (in step (8)).

Using these values in the way described above, new colors are distributed on more different sectors, and thus more groups will be updated. They help to utilize the information of more new colors in the next iteration of the algorithm.

It should be noted that there is a relationship between the fitness value of colors and harmonic templates. It is made through the rotation of the hue circle (or gray area in Fig. 4) and the selection strategy. In other words, colors with better fitness value (agents) are selected with a higher probability for producing new colors.

2.5 Dispersion phase

This phase deals with only the exploration of the search space. As shown in Fig. 3, when population diversity is less than the diversity threshold value (D_{th}), the dispersion phase is performed to increase the population diversity and thus to explore the search space more comprehensively.

For this purpose, colors that are very close to the center of the population will be replaced by new ones which are produced by using the Munsell method and colors in $\boldsymbol{CM}$. Based on the Munsell method definition, different colors should occupy different areas to look harmonious in an image. As he proposed, the areas (or amount of color) should be inversely proportional to the value and chroma (Glover 1986). Note that, in the following equation, r_{cm} corresponds to the amount of colors. The replacement process is defined as follows:

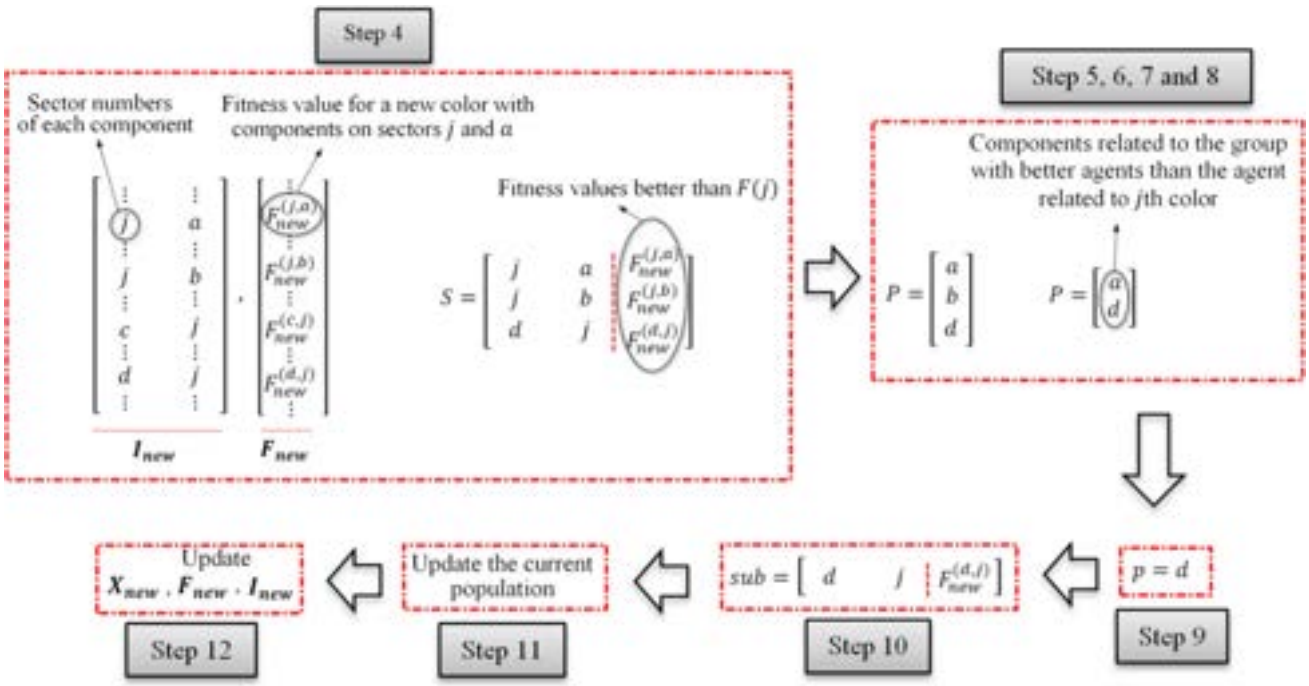

Fig. 6 Schematic representation of steps 3–12 for updating of the hue circle

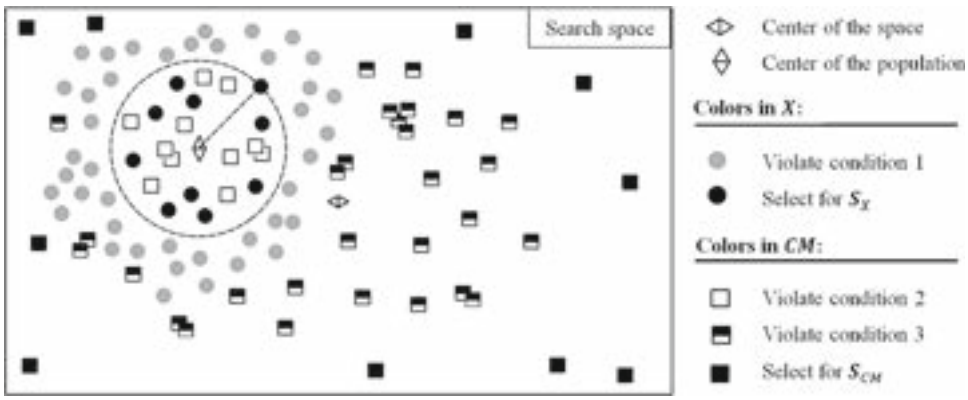

Fig. 7 Schematic representation of the selection procedure of colors in the dispersion phase

$$S_X^{\text{new}}(i,j) = r_{\text{cm}} S_{\text{CM}}(i,j) + (1 - r_{\text{cm}}) S_X(i,j) \quad \begin{matrix} i = 1,2,\ldots,N_v \\ j = 1,2,\ldots,N_s \end{matrix} \tag{12}$$

where S_{CM} and $S_X(j)$ are, respectively, the selected colors from ***CM*** and ***X***; N_s is the number of the replaced colors; r_{cm} is a random number uniformly distributed in the range of $(r_{\text{cm}_0}, 1)$, respectively. Since colors in ***CM*** have not been used in the hue circle, they have low amounts of the chroma. Therefore, according to the Munsell method, they should have more contribution in combination with colors in the hue circle. To fulfill this requirement, r_{cm_0} is experimentally set to be 0.7.

This phase takes the following steps:

Step (1) Choose colors for Eq. (12) (see Fig. 7):

- Select N_s colors from ***X***, except the agents, with the smallest distance from the center of the population (condition 1), and store them in $\boldsymbol{S}_X$.
- Select N_s colors from ***CM*** if $N_{\text{CM}} \geq k \times N_s$ (or from $\boldsymbol{CM}_{temp}$ if $N_{\text{CM}} < k \times N_s$), which have not only more distance from the center of the population than the colors in $\boldsymbol{S}_X$ (condition 2) but also the most diversity with respect to the center of the search space (condition 3); store them in $\boldsymbol{S}_{\text{CM}}$.

Step (2) If the number of the different sets in ***CM*** is less than k, add $\boldsymbol{S}_{CM}$ to ***CM***; otherwise, find a set in ***CM*** with less diversity than $\boldsymbol{S}_{CM}$ then replace it by $\boldsymbol{S}_{CM}$.
Step (3) Produce new colors using Eq. (12), then replace them on the hue circle.

Note that, in step (2), k is the number of different sets in ***CM***. Each set has N_s colors with different diversity. When ***CM*** is updated by adding or replacing a new set, its diversity is increased, and therefore helps to explore the search space more.

At the end of this phase, as mentioned before, the updating process of the hue circle is performed. For this aim, if replaced colors have better fitness value than their related agents, their positions on the hue circle will be exchanged.

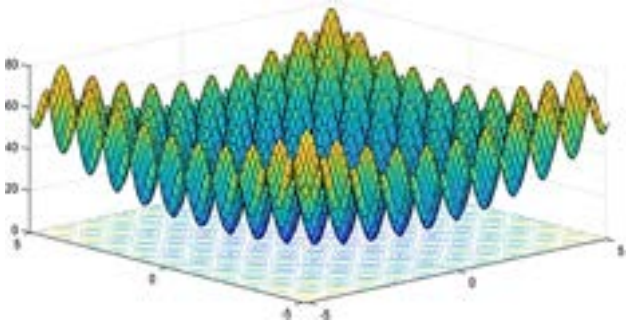

Fig. 8 2D Rastrigin function

2.6 An example of optimization using CHA

In this section, a highly multimodal function, namely the Rastrigin function, is optimized to present the movement of the colors toward the global minimum. It is defined as follows:

$$f(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$$

where the global optimizer is located at $x^* = 0, f(x^*) = 0$ for $x_i \in [-5.12, 5.12]$ and $n = 2$. It can be seen from Fig. 8 that this function has several local minima. For the optimization process, the mathematical formula of the objective function, number of the variables, search space bounds and algorithm parameters should be defined beforehand. The number of colors (or population) is equal to 100, and the CHA parameters are set as follows: $D_{\text{th}_f} = 0.01$, $N_s = 20$, $k = 4$ $N_{\text{comb}} = 10$, damp $= 0.5$. It should be noted that all these parameters are fixed except damp which should be set experimentally to 0.5 and 0.96 for low- and high-dimensional functions, respectively. If either the lower or upper bound of a variable is violated by a color,

the algorithm utilizes the nearest limit values for its violated variable.

We ran CHA and captured the position of the colors for five different steps of the algorithm, as shown in Figs. 9, 10, 11, 12 and 13. It should be noted that different marker symbols show colors in different hue groups, and each filled marker indicates the agent of each group.

As shown in Fig. 9, the algorithm begins with a population of potential solutions (colors) and then the concentration phase is done. From the movements of the colors in Figs. 9 and 10, it can be seen that the diversity of the initial population is decreased after the concentration phase is completed. To describe the effect of the dispersion phase on the population diversity, the positions of colors before and after performing this phase are shown in Figs. 11 and 12 for $\text{iter}_{\text{dp}} = 2$. It can be seen that after the dispersion phase, some colors are replaced by new ones to enhance the diversity of the population and to increase the exploration ability of the CHA. From Fig. 13, at the end of the optimization process in CHA, all colors are moved in close vicinity of the global optimum for $\text{iter}_{\text{dp}} = 8$.

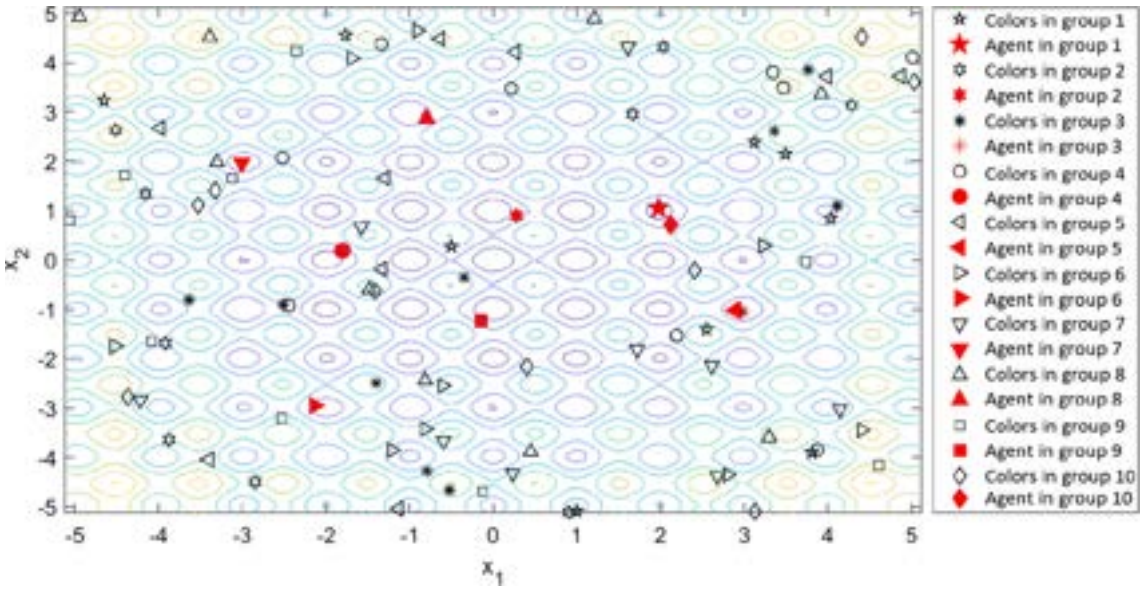

Fig. 9 Position of colors before the concentration phase for $\boldsymbol{iter_{dp}} = 0$

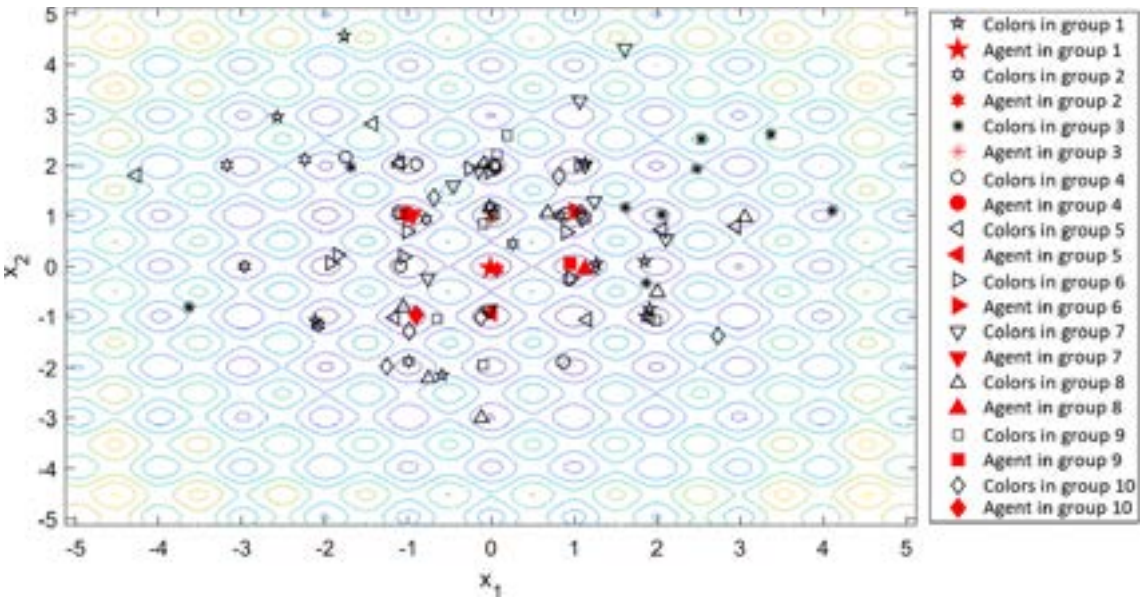

Fig. 10 Position of colors after the concentration phase for $\boldsymbol{iter_{dp}} = 0$

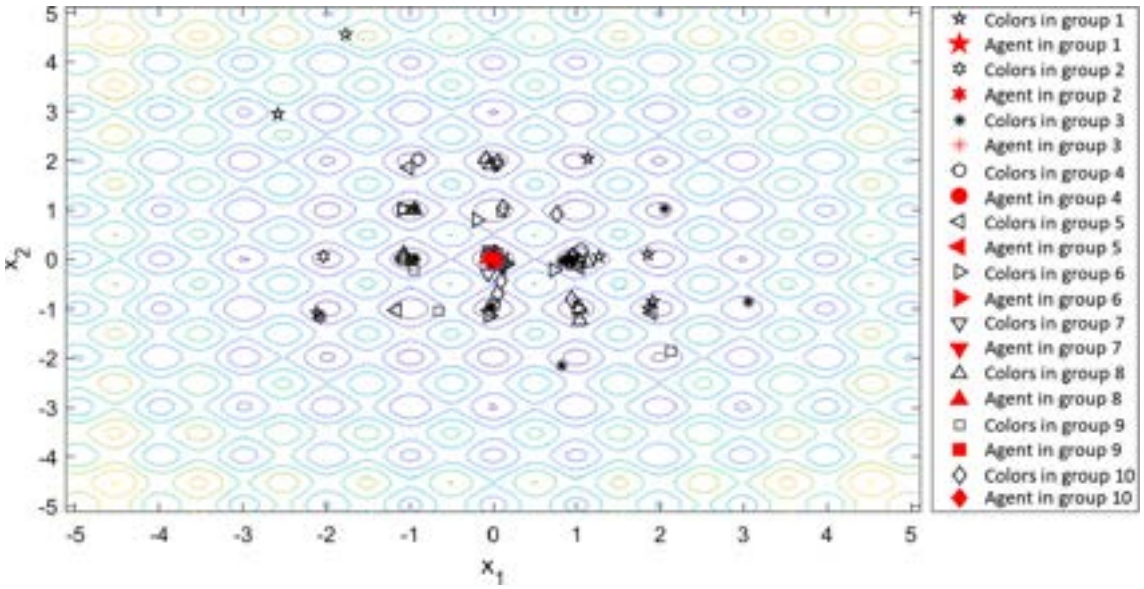

Fig. 11 Position of colors before the dispersion phase for $iter_{dp} = 2$

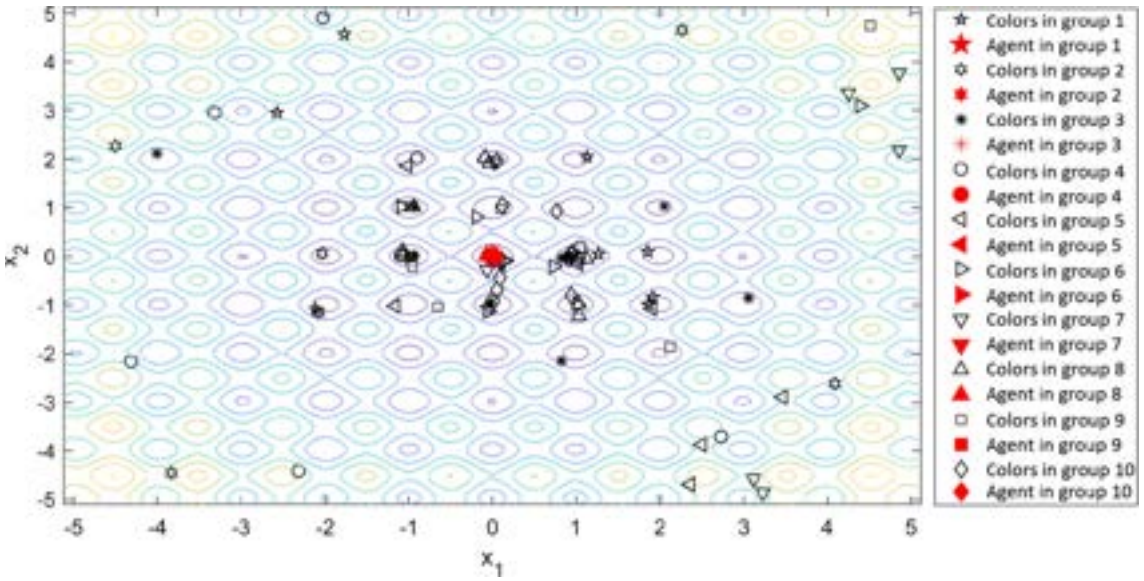

Fig. 12 Position of colors after the dispersion phase for $iter_{dp} = 2$

3 Experimental results

In this section, the performance of CHA is tested using different unimodal and multimodal benchmark functions, including both low- and high-dimensional problems (Jamil and Yang 2013). The functions are listed in Table 1, and the related constant parameters used in some functions are described in Appendix B. It is worth mentioning that the source code of CEC 2017 benchmark functions is available, but they should be treated as black-box problems, i.e., without explicit knowledge of their structure (Awad 2016).

In unimodal functions, as their names imply, there is only one optimum which is called the global optimum. In this class of functions, the exploitation ability and convergence of an algorithm is more important than the final results because there are other methods which have been specially designed to optimize unimodal functions. In contrast, multimodal functions have more than one optimum. One of them is the global optimum, and the rest are the local optima. Optimizing this class of test functions is more challenging than unimodal functions, because the number of local optima exponentially increases by increasing the dimension of the search space. Therefore,

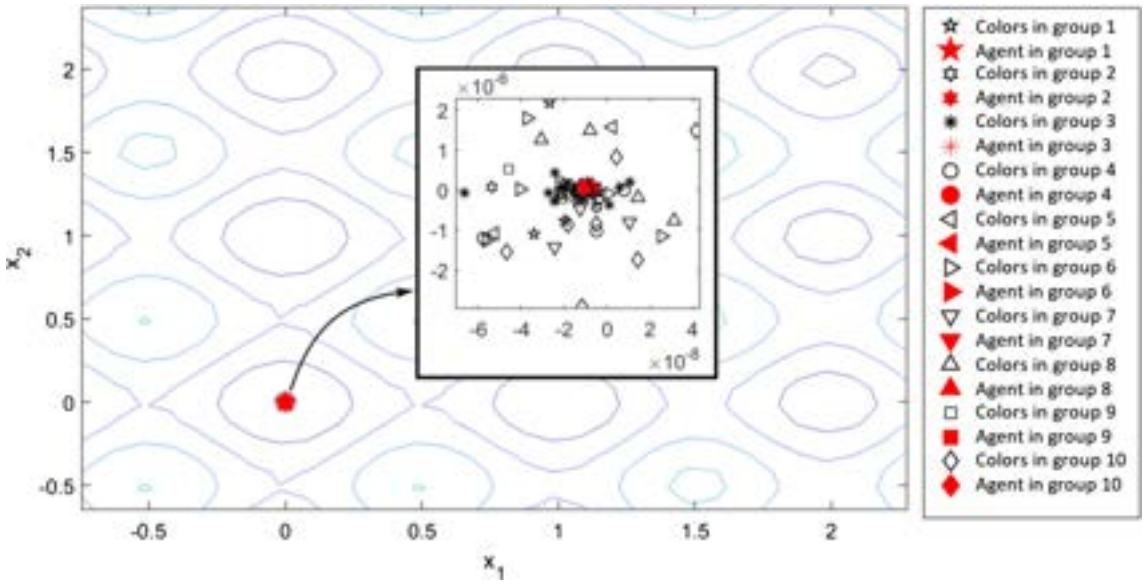

Fig. 13 Position of colors after the dispersion phase for $iter_{dp} = 8$

they are employed to test the exploration ability and local optima avoidance of the algorithm. Algorithms with high exploration ability should not become trapped in a local optimum and should be able to find better final results.

For verification of the results, the proposed CHA algorithm is compared to ten well-known optimization algorithms including, ACOR (Dorigo and Stützle 2004), HS, DE, GA, PSO, CS, GSA, ABC, KH (Gandomi and Alavi 2012) and TLBO (Rao et al. 2012a; Rao et al. 2012b). For this purpose, the overall performance, the convergence behavior and the average number of fitness evaluations to find the global optimum are studied. Because of the stochastic nature of CHA and other algorithms, for each test function, we have carried out 30 independent runs with independent population initializations to obtain a useful conclusion about the performance of each method. During our experiments, the population size of all algorithms is kept equal to 100. The parameter settings in all experiments for each algorithm are shown in Table 2 (Wang et al. 2014; Wang and Guo 2013). Note that TLBO and GSA are parameter-less algorithms, only population size and number of generations need to be specified.

3.1 Overall performance comparison

In this section, the comparison between CHA and the other optimization algorithms is performed by using the 38 benchmark functions. The maximum number of fitness evaluations is selected as $D \times 10^3$ to evaluate the overall performances of the algorithms.

The mean, best and standard deviation of the results found after 30 independent runs of each algorithm are reported in Table 3. In addition to the statistical results in Table 3, a pair-wise statistical test is used for a better insight into the performance of the algorithms. For this aim, the Wilcoxon Signed-Rank test is performed at the 95% significant level ($\alpha = 0.05$); the statistical testing of the algorithms is shown in Table 4.

The null hypothesis H_0 for this test is: 'There is no difference between the medians of the solutions obtained by two different algorithms,' i.e., median (algorithm A) = median (algorithm B). To determine whether algorithm A reached a statistically better solution than algorithm B, or if not, whether the alternative hypothesis is valid, the sizes of the ranks determined by the Wilcoxon Signed-Rank Test (i.e., T + and T as defined in (Derrac et al. 2011)) are examined. The symbols $(+, -, =)$ in Table 4 show that CHA performed significantly better $(+)$, worse $(-)$, or not significantly different $(=)$ compared to a given algorithm. From the total count of $(+, -, =)$ in the last row of Table 4, it can be found that CHA outperforms all of the compared algorithms. Further, the performance of TLBO and PSO is competitive.

3.2 Convergence behavior comparison

The convergence behavior of each algorithm is given in Figs. 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 and 51. The values shown in these figures are the average best solution achieved from 30 runs. The maximum numbers of the fitness evaluation are set to 1000 and 5000 for the low- and high-dimensional functions, respectively.

Table 1 Benchmark functions

No.	Name	D	Bounds	Definition	C	Min.
Low-dimensional function						
F1	Beale	2	$[-4.5, 4.5]$	$f(x) = (1.5 - x_1 - x_1x_2)^2 + (2.5 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_3^2)^2$	U	0
F2	Three-hump camel back	2	$[-5, 5]$	$f(x) = 2x_1^2 - 1.052x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	M	0
F3	Six-hump camel back	2	$[-5, 5]$	$f(\boldsymbol{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + \left(4x_2^2 - 4\right)x_2^2$	M	− 1.03163
F4	Colville	4	$[-10, 10]$	$f(x) = 100\left(x_1 - x_2^2\right)^2 + (1 - x_1)^2 + 90\left(x_4 - x_3^2\right)^2 + (1 - x_3)^2 + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1)$	U	0
F5	Easom	2	$[-100, 100]$	$f(x) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1 - \pi)^2 - (x_2 - \pi)^2\right)$	U	− 1
F6	Exponential 2	2	$[-1, 1]$	$f(x) = -\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right)$	U	− 1
F7	Exponential 4	4	$[-1, 1]$	$f(x) = -\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right)$	U	− 1
F8	Exponential 8	8	$[-1, 1]$	$f(x) = -\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right)$	U	− 1
F9	Hartman 4	4	$[0, 1]$	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{n} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	M	− 3.86
F10	Hartman 6	6	$[0, 1]$	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{n} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	M	− 3.32236
F11	Kowalik	4	$[-5, 5]$	$f(x) = \sum_{i=1}^{11}\left(a_i - \frac{x_1(1 + x_2b_i)}{\left(1 + x_3b_i + x_4b_i^2\right)}\right)^2$	M	0.000030748
F12	Matyas	2	$[-10, 10]$	$f(x) = 0.26\left(x_1^2 + x_2^2\right) - 0.48x_1x_2$	U	0
F13	Schaffer	2	$[-100, 100]$	$f(x) = 0.5 + \frac{\left(\sin\sqrt{x_1^2 + x_1^2} - 0.5\right)}{\left(1 + 0.001\left(x_1^2 + x_1^2\right)\right)^2}$	M	0
F14	Trid 6	6	$[-36, 36]$	$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_ix_{i-1}$	U	− 50
F15	Zakharov	10	$[-5, 10]$	$f(x) = \sum_{i=1}^{n} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^4$	U	0
High-dimensional function						
F16	Ackley	20	$[-32, 32]$	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)\right) + 20 + \varepsilon$	M	0
F17	Dixon-price	20	$[-10, 10]$	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i\left(\left[2x_i^2 - x_{i-1}\right]\right)^2$	U	0
F18	Griewank	20	$[-100, 100]$	$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n}\left(\frac{x_i}{\sqrt{i}}\right) + 1$	M	0

Table 1 (continued)

No.	Name	D	Bounds	Definition	C	Min.
F19	Penalized 1	20	$[-50, 50]$	$f(x) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left(1+10\sin^2(\pi y_{i+1})\right) + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = a + \frac{1}{4}(x_i+1), u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	M	0
F20	Penalized 2	20	$[-50, 50]$	$f(x) = 0.1\left\{\sin^2(\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2\left(1+\sin^2(3\pi x_{i+1})\right) + (x_n-1)^2\left(1+\sin^2(2\pi x_n)\right)\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	M	0
F21	Powell	20	$[-4, 5]$	$f(x) = \sum_{i=1}^{\frac{n}{4}}(x_{4i-3}+10x_{4i-2})^2 + 5(x_{4i-1}+10x_{4i})^2 + (x_{4i-2}+10x_{4i-1})^4 + 10(x_{4i-3}-10x_{4i})^4$	U	0
F22	Rosenbrock	20	$[-30, 30]$	$f(x) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1}-x_i^2\right)^2 + (x_i-1)^2\right)$	U	0
F23	Schwefel 2.22	20	$[-100, 100]$	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	U	0
F24	Schwefel 2.21	20	$[-100, 100]$	$f(x) = \max_{1 \le i \le n}\lvert x_i\rvert$	U	0
F25	Schwefel 1.2	20	$[-100, 100]$	$f(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	U	0
F26	sphere	20	$[-100, 100]$	$f(x) = \sum_{i=1}^{n} x_i^2$	U	0
F27	Step	20	$[-100, 100]$	$f(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	U	0
F28	Sum-squares	20	$[-10, 10]$	$f(x) = \sum_{i=1}^{n} i x_i^2$	U	0
F29	Quartic	20	$[-1.28, 1.28]$	$f(x) = \sum_{i=1}^{n} i x_i^4 + rand[0, 1)$	U	0
F30	Rastrigin	20	$[-5.12, 5.12]$	$f(x) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right)$	M	0
High-dimensional function, CEC2017 (Awad 2016)						
F31	f1-CEC2017	30	$[-100, 100]$	Shifted and rotated bent Cigar function	U	100
F32	f3-CEC2017	30	$[-100, 100]$	Shifted and rotated Zakharov function	U	300
F33	f4-CEC2017	30	$[-100, 100]$	Shifted and rotated Rosenbrock's function	M	400
F34	f10-CEC2017	30	$[-100, 100]$	Shifted and rotated Schwefel's function	M	− 1000
F35	f11-CEC2017	30	$[-100, 100]$	Hybrid function 1	M	1100
F36	f20-CEC2017	30	$[-100, 100]$	Hybrid function 6	M	2000
F37	f21-CEC2017	30	$[-100, 100]$	Composition function 1	M	2100
F38	f30-CEC2017	30	$[-100, 100]$	Composition function 10	M	3000

D dimension, C characteristics, U unimodal, M multimodal

Table 2 Parameter setting

Algorithms	Parameters
ABC	Limit value limit = SN × D, where SN and D are, respectively, the number of food sources and the dimension
ACOR	Initial pheromone value $\tau_0 = 1E-6$, exploration constant $q_0 = 1$, local pheromone decay rate $\rho_l = 0.9$, global pheromone decay rate $\rho_g = 0.9$, pheromone update constant $Q = 20$, pheromone sensitivity rate $s = 1$ and visibility sensitivity $\beta = 5$
CHA	$D_{th_f} = 0.01$, $N_s = 20$, $k = 4$ and $N_{comb} = 10$. *damp* is set to 0.5 and 0.96 for low- and high-dimensional functions
CS	Discovery rate $p_a = 0.25$
DE	Crossover constant CR = 0.5 and a weighting factor $F = 0.5$
GA	Roulette wheel selection, single-point crossover with a crossover probability of 1 and a mutation probability of 0.01
GSA	–
HS	Harmony memory consideration rate HMCR = 0.5, the pitch adjustment rate PAR = 0.1
KH	The foraging speed $V_f = 0.02$, the maximum diffusion speed $D^{max} = 0.0005$, the maximum induced speed $N^{max} = 0.01$
PSO	Inertial constant $w = 0.3$, cognitive constant $c_1 = 1$, and social constant $c_2 = 1$
TLBO	–

Figures 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 and 28 show the optimization values obtained by eleven algorithms for low-dimensional functions (F1–15). For F5, F7–10 and F14, CHA performance is noticeably superior to the other ten methods. It converges to better optimum solutions with the fastest convergence rate. Although slower, PSO performs the second best in the optimization process. Moreover, GA is the third most effective, performing the best on these problems. For F1, F4 and F15, CHA is outperformed by other algorithms in the speed of convergence. It is followed by GA for F1 and F4 and by PSO for F15. Further, the performance of PSO is different for function F1 which is weaker than GA, ABC, DE, TLBO and HS. For F12, the performance of CHA and PSO is quite similar and better than the other algorithms. For F2–3, F6, F11 and F13, CHA is outperformed only by PSO in terms of the convergence speed at the beginning of the search. However, CHA reaches the same minimum value as PSO at last.

Figures 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42 and 43 show the convergence characteristics of the algorithms solving high-dimensional functions (F16–F38). For F16, F24 and F29–30, CHA overtakes all other approaches; not only it shows the fastest convergence rate, but also it can reach a better optimum at last. Figures 42 and 43 show that HS has the fifth and second best performance for F29 and F30, respectively. Moreover, GSA clearly outperforms GA and PSO in Fig. 37. For function F16, It can be seen that ACOR, CS and ABC seem to be trapped into sub-optimal values.

Figures 31 and 34 show that CHA has the fastest convergence rate initially toward the global optimum for F18 and F21. As can be seen, PSO and GA are the second and third most effective, respectively. Further, GSA shows better performance than GA for F25.

For F17, F19, F20, F22 and F26–27, PSO and GSA perform better than CHA at the beginning of the search, but CHA and GA eventually converge to the same values of them. Moreover, the performance HS, TLBO and KH are very good for these functions. For F23, all algorithms follow the same convergence behavior except ACOR and KH, which have slower convergence rate compared to others. From Figs. 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42 and 43, the performance of ACOR for high-dimensional problems is similar to low-dimensional ones. As can be observed, it is outperformed by all other algorithms.

In Figs. 44, 45, 46, 47, 48, 49, 50 and 51, comparison between CHA and the other optimization algorithms is carried out based on eight CEC 2017 benchmark functions. Figure 44 illustrates the values for the shifted and rotated bent cigar function. Figure 44, apparently, shows that CKH has better performance in terms of the convergence rate and average final results for this unimodal function. Although slower, HS performs the second best in the optimization process.

For F32, F34 and F36, it is clear that CHA significantly overtakes all other approaches. Moreover, PSO and GA work very well because they rank 2 and 3, respectively, among all considered algorithms for F32 and F36. Figure 47 shows that ACOR, GSA, KH, TLBO, HS and DE seem to be trapped in a locally optimal solution.

For F33 and F35, CHA and PSO exhibit almost the same convergence behavior after about 1000 function evaluations; however, CHA reaches better fitness value at the maximum number of fitness evaluations.

Table 3 Statistical results obtained by different algorithms for 30 independent runs

	CHA	ACOR	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F1											
Best	2.97E−06	1.02E−04	7.64E−06	2.15E−05	2.02E−08	2.86E−12	8.43E−05	2.99E−04	2.17E−05	5.96E−06	1.44E−04
Mean	2.63E−04	7.02E−03	5.66E−03	4.57E−04	7.58E−07	5.08E−02	5.71E−03	2.39E−02	2.82E−03	4.04E−04	3.52E−02
SD	4.56E−04	8.91E−03	9.98E−03	4.32E−04	1.21E−06	1.93E−01	6.04E−03	2.23E−02	3.31E−03	5.31E−04	5.91E−02
F2											
Best	5.97E−08	1.28E−05	8.03E−07	1.37E−06	2.04E−11	2.00E−13	5.73E−05	9.42E−04	4.59E−06	1.96E−06	1.62E−04
Mean	3.68E−06	1.50E−03	4.22E−04	1.71E−05	9.74E−08	3.16E−11	2.19E−03	2.02E−02	2.84E−04	9.32E−05	1.78E−02
SD	3.83E−06	1.39E−03	1.11E−03	1.03E−05	1.37E−07	3.87E−11	2.52E−03	1.57E−02	2.30E−04	1.84E−04	2.62E−02
F3											
Best	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00
Mean	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 9.92E−01	− 1.03E+00	− 1.03E+00	− 1.02E+00
SD	2.23E−04	3.53E−03	3.58E−04	3.70E−04	7.03E−05	3.15E−10	1.94E−03	4.66E−02	7.57E−04	4.87E−04	1.56E−02
F4											
Best	2.32E−02	1.64E+00	1.18E−01	5.64E−01	4.25E−06	1.60E−04	2.53E+00	9.03E+00	4.80E−01	1.67E−01	7.44E−01
Mean	1.74E+00	9.97E+00	4.88E+00	3.66E+00	1.76E+00	1.94E+00	7.06E+00	8.43E+01	4.97E+00	1.96E+00	7.37E+00
SD	2.18E+00	6.24E+00	9.49E+00	2.01E+00	2.07E+00	2.86E+00	4.15E+00	5.27E+01	2.47E+00	9.15E−01	3.01E+00
F5											
Best	− 9.99E−01	− 9.80E−01	− 9.97E−01	− 8.69E−01	− 9.97E−01	− 1.00E+00	− 9.81E−01	− 1.00E+00	− 5.11E−01	− 9.26E−01	− 9.94E−01
Mean	− 8.52E−01	− 1.16E−01	− 5.03E−01	− 2.69E−01	− 3.56E−01	− 1.00E+00	− 1.27E−01	− 7.41E−01	− 5.08E−02	− 2.04E−01	− 7.58E−01
SD	1.78E−01	2.61E−01	4.40E−01	2.98E−01	3.35E−01	1.43E−07	2.41E−01	2.57E−01	1.32E−01	3.11E−01	2.63E−01
F6											
Best	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00
Mean	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00
SD	5.89E−08	4.86E−06	1.03E−06	9.69E−08	1.13E−09	4.15E−13	1.93E−05	2.98E−04	1.34E−06	7.31E−07	4.77E−05
F7											
Best	− 1.00E+00	− 9.99E−01	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 9.87E−01	− 8.59E−01	− 1.00E+00	− 1.00E+00	− 1.00E+00
Mean	− 1.00E+00	− 9.93E−01	− 1.00E+00	− 1.00E+00	− 9.99E−01	− 1.00E+00	− 9.53E−01	− 4.52E−01	− 9.98E−01	− 1.00E+00	− 9.96E−01
SD	7.42E−07	3.73E−03	8.30E−05	5.28E−05	2.97E−03	2.06E−17	2.55E−02	2.20E−01	1.40E−03	1.24E−05	2.56E−03
F8											
Best	− 1.00E+00	− 9.99E−01	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 9.98E−01	− 9.55E−01	− 1.00E+00	− 1.00E+00	− 1.00E+00
Mean	− 9.00E−01	− 9.98E−01	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 1.00E+00	− 9.96E−01	− 8.74E−01	− 1.00E+00	− 1.00E+00	− 1.00E+00
SD	3.05E−01	6.54E−04	1.14E−06	1.55E−06	3.21E−04	5.05E−17	1.60E−03	3.47E−02	1.16E−04	1.00E−10	8.97E−06
F9											
Best	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00
Mean	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00	− 3.84E+00	− 3.86E+00	− 3.86E+00	− 3.86E+00
SD	2.54E−03	3.84E−04	1.34E−05	3.33E−06	6.98E−04	3.03E−14	2.60E−03	1.35E−02	3.43E−04	5.67E−05	8.43E−03

Table 3 (continued)

	CHA	ACOR	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F10											
Best	− 3.32E+00	− 3.32E+00	− 3.32E+00	− 3.32E+00	− 3.32E+00	− 3.32E+00	− 3.31E+00	− 3.21E+00	− 3.32E+00	− 3.32E+00	− 3.32E+00
Mean	− 3.31E+00	− 3.24E+00	− 3.32E+00	− 3.26E+00	− 3.27E+00	− 3.28E+00	− 3.28E+00	− 2.94E+00	− 3.30E+00	− 3.29E+00	− 3.27E+00
SD	8.65E−03	7.07E−02	2.18E−02	4.84E−02	5.62E−02	5.84E−02	2.33E−02	1.19E−01	1.59E−02	4.75E−02	6.30E−02
F11											
Best	3.08E−04	3.11E−04	3.09E−04	3.09E−04	3.07E−04	3.07E−04	3.08E−04	3.32E−04	3.11E−04	3.08E−04	3.08E−04
Mean	3.08E−04	3.57E−04	3.21E−04	3.13E−04	3.10E−04	3.09E−04	3.17E−04	3.80E−04	3.22E−04	3.09E−04	3.24E−04
SD	8.37E−07	2.70E−05	9.92E−06	2.78E−06	3.68E−06	2.35E−06	5.01E−06	3.22E−05	8.64E−06	1.01E−06	1.54E−05
F12											
Best	7.25E−08	7.17E−05	1.71E−05	1.45E−06	6.96E−11	1.30E−13	5.99E−06	1.10E−05	4.40E−07	6.52E−07	3.46E−05
Mean	1.70E−05	1.88E−03	2.96E−03	2.82E−05	1.49E−08	2.84E−11	4.24E−04	7.88E−03	3.25E−04	3.18E−05	1.11E−03
SD	2.90E−05	2.01E−03	5.33E−03	2.57E−05	1.86E−08	4.08E−11	4.52E−04	5.50E−03	3.64E−04	3.80E−05	9.92E−04
F13											
Best	4.82E−06	1.57E−03	2.39E−05	9.37E−06	4.74E−07	3.26E−12	6.91E−04	4.12E−06	6.87E−05	2.94E−05	1.44E−06
Mean	3.36E−03	3.06E−02	1.49E−02	8.13E−03	1.67E−02	2.93E−09	1.54E−02	2.32E−03	1.55E−02	9.99E−03	3.78E−03
SD	4.21E−03	2.01E−02	1.53E−02	6.28E−03	1.91E−02	5.46E−09	1.09E−02	3.09E−03	1.31E−02	1.09E−02	4.23E−03
F14											
Best	− 5.00E+01	− 4.73E+01	− 4.99E+01	− 4.97E+01	− 5.00E+01	− 5.00E+01	− 4.92E+01	− 4.77E+01	− 4.92E+01	− 5.00E+01	− 4.99E+01
Mean	− 4.58E+01	− 4.01E+01	− 4.74E+01	− 4.91E+01	− 4.70E+01	− 5.00E+01	− 4.62E+01	− 4.54E+01	− 4.54E+01	− 5.00E+01	− 4.93E+01
SD	3.78E+00	6.10E+00	2.71E+00	4.23E−01	4.61E+00	6.59E−02	1.96E+00	1.42E+00	3.31E+00	4.61E−03	4.35E−01
F15											
Best	1.60E−09	1.96E+01	1.80E−03	3.03E−03	3.51E−01	4.15E−30	7.79E+00	8.97E+02	3.45E+00	1.84E−09	1.43E−02
Mean	3.91E−08	8.34E+01	5.45E−03	1.14E−02	8.29E+00	3.31E−06	2.02E+01	2.78E+03	8.64E+00	6.92E−09	5.72E−02
SD	4.14E−08	5.70E+01	2.43E−03	4.19E−03	7.70E+00	1.81E−05	8.48E+00	1.32E+03	3.20E+00	3.61E−09	3.04E−02
F16											
Best	8.05E−03	1.38E+01	2.06E−01	3.01E−01	1.08E+00	7.25E−03	9.77E+00	3.85E+00	8.04E+00	1.25E−03	1.50E−01
Mean	5.19E−02	1.58E+01	2.90E−01	5.41E−01	2.48E+00	2.49E+00	1.41E+01	4.48E+00	9.70E+00	7.16E−02	2.41E−01
SD	4.00E−02	1.00E+00	5.97E−02	1.14E−01	5.78E−01	9.12E−01	1.61E+00	2.17E−01	8.61E−01	9.85E−02	6.30E−02
F17											
Best	6.67E−01	1.40E+04	9.82E−01	1.59E+00	1.19E+00	6.67E−01	1.04E+02	8.77E+02	9.21E+02	6.67E−01	6.81E−01
Mean	6.67E−01	4.74E+04	2.14E+00	2.72E+00	4.91E+00	1.50E+00	3.01E+02	1.77E+03	2.75E+03	6.67E−01	7.76E−01
SD	2.09E−04	2.57E+04	8.82E−01	6.46E−01	2.61E+00	9.20E−01	1.16E+02	6.07E+02	1.13E+03	1.77E−02	9.86E−02
F18											
Best	5.20E−07	1.92E+01	6.70E−01	5.65E−01	1.01E+00	1.32E−02	2.95E+00	4.43E−01	3.32E+00	8.66E−08	6.80E−01
Mean	4.13E−03	4.25E+01	8.43E−01	8.03E−01	1.10E+00	1.50E−01	5.54E+00	5.62E−01	4.95E+00	1.88E−04	8.42E−01
SD	1.14E−02	1.06E+01	7.79E−02	7.69E−02	5.61E−02	1.39E−01	1.14E+00	6.91E−02	7.39E−01	1.02E−03	8.64E−02

Table 3 (continued)

	CHA	ACOR	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F19											
Best	1.47E−02	2.14E+06	1.69E−03	1.50E−01	7.17E−03	1.57E−03	6.34E+00	1.01E+00	5.08E+01	1.85E−07	7.19E−04
Mean	3.66E−02	2.16E+07	9.36E−03	4.07E−01	3.77E−01	1.14E+00	1.88E+01	1.52E+00	1.53E+05	1.99E−06	2.45E−02
SD	1.71E−02	1.34E+07	2.90E−02	1.73E−01	3.59E−01	9.77E−01	6.10E+00	2.93E−01	2.58E+05	2.21E−06	4.79E−02
F20											
Best	2.12E−01	1.68E+07	1.39E−02	1.88E−01	1.84E−01	1.32E−01	8.51E+01	2.87E+00	1.78E+05	6.81E−01	1.13E−02
Mean	4.80E−01	5.51E+07	3.51E−02	5.52E−01	1.91E+00	4.80E+00	5.74E+03	4.27E+00	1.20E+06	6.42E−01	1.26E−01
SD	1.55E−01	4.47E+07	1.38E−02	1.67E−01	2.93E+00	4.47E+00	6.05E+03	8.26E−01	6.59E+05	2.44E−01	1.57E−01
F21											
Best	6.15E−14	3.11E−06	8.37E−06	5.03E−08	9.14E−14	1.00E−08	2.01E−10	1.50E−02	2.45E−06	2.63E−09	8.28E−06
Mean	1.77E−09	4.32E−04	5.39E−03	2.54E−07	3.82E−05	1.83E−05	2.93E−09	1.03E+00	6.83E−05	1.47E−07	1.27E−04
SD	5.16E−09	6.92E−04	5.43E−03	2.07E−07	9.52E−05	2.55E−05	2.41E−09	7.09E−01	9.25E−05	1.64E−07	1.94E−04
F22											
Best	1.85E+01	2.46E+06	3.56E+01	1.71E+02	1.15E+02	1.90E+01	1.24E+04	1.93E+03	1.93E+03	1.93E+01	2.02E+01
Mean	1.87E+01	1.03E+07	1.44E+02	3.54E+02	4.93E+02	9.57E+01	5.60E+04	4.46E+03	5.36E+03	1.99E+01	2.38E+01
SD	9.49E−02	4.81E+06	9.43E+01	1.29E+02	3.54E+02	5.41E+01	2.28E+04	1.26E+03	1.40E+03	3.06E−01	8.77E+00
F23											
Best	7.32E−03	5.10E+01	9.90E−02	1.47E−01	2.13E−01	1.93E−02	1.68E+01	1.28E+01	1.73E+01	1.42E−08	7.03E+00
Mean	4.19E−02	2.96E+02	1.55E−01	2.16E−01	5.63E−01	3.81E−01	2.32E+01	1.55E+01	2.42E+01	3.18E−08	1.44E+01
SD	2.77E−02	4.46E+02	2.69E−02	3.21E−02	2.20E−01	4.17E−01	3.67E+00	1.05E+00	4.57E+00	7.62E−09	5.17E+00
F24											
Best	4.29E−05	6.21E+01	2.39E+00	8.85E+00	5.20E+00	1.65E+00	1.82E+01	1.37E+00	3.08E+01	8.37E−04	1.97E−01
Mean	1.97E−04	7.07E+01	3.95E+00	1.25E+01	7.92E+00	4.91E+00	2.61E+01	1.58E+00	4.27E+01	1.47E−03	2.94E−01
SD	1.35E−04	4.62E+00	1.11E+00	1.55E+00	1.43E+00	1.60E+00	2.96E+00	1.29E−01	4.65E+00	4.69E−05	4.89E−02
F25											
Best	8.99E−04	2.29E+05	2.74E+01	1.69E+01	2.96E+02	2.05E−02	2.24E+04	9.87E+02	1.47E+04	5.92E−03	2.51E+01
Mean	2.24E−01	4.16E+05	5.61E+01	3.71E+01	8.62E+02	4.75E+01	3.50E+04	1.56E+03	3.32E+04	2.39E−01	1.81E+02
SD	2.27E−01	9.60E+04	1.45E+01	1.16E+01	4.11E+02	8.21E+01	8.49E+03	3.59E+02	8.03E+03	1.70E−01	1.22E+02
F26											
Best	4.53E−04	2.31E+03	4.10E−01	2.46E−01	2.17E+00	2.80E−05	2.68E+02	9.19E+00	2.12E+02	9.27E−07	1.41E−01
Mean	1.09E−02	4.65E+03	7.26E−01	5.79E−01	1.07E+01	8.41E−02	4.85E+02	1.24E+01	4.28E+02	2.58E−05	4.93E−01
SD	1.10E−02	1.39E+03	1.72E−01	1.64E−01	8.08E+00	1.46E−01	1.38E+02	1.34E+00	9.86E+01	1.29E−05	1.90E−01
F27											
Best	0.00E+00	3.43E+03	0.00E+00	0.00E+00	2.00E+00	3.00E+00	2.42E+02	7.00E+00	2.51E+02	0.00E+00	0.00E+00
Mean	0.00E+00	4.99E+03	0.00E+00	2.00E−01	1.61E+01	2.23E+01	5.52E+02	1.29E+01	4.11E+02	0.00E+00	1.00E−01
SD	0.00E+00	1.02E+03	0.00E+00	4.07E−01	1.10E+01	1.50E+01	1.70E+02	2.18E+00	8.59E+01	0.00E+00	3.05E−01

Table 3 (continued)

	CHA	ACOR	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F28											
Best	1.34E−03	2.17E+03	2.84E−01	2.53E−01	3.61E+00	4.36E−03	2.16E+02	1.07E+03	1.74E+02	6.17E−05	2.55E−01
Mean	1.12E−02	4.33E+03	5.73E−01	4.10E−01	9.55E+00	4.34E−01	3.59E+02	1.79E+03	3.36E+02	2.68E−06	1.31E+00
SD	9.47E−03	1.03E+03	1.60E−01	1.11E−01	6.84E+00	9.77E−01	8.15E+01	2.71E+02	9.60E+01	1.50E−05	1.56E+00
F29											
Best	4.50E+00	1.02E+01	5.14E+00	5.85E+00	4.62E+00	4.42E+00	5.97E+00	1.41E+01	6.21E+00	4.16E+00	3.93E+00
Mean	5.39E+00	1.35E+01	5.72E+00	6.37E+00	5.20E+00	5.37E+00	7.24E+00	2.05E+01	7.83E+00	5.26E+00	5.12E+00
SD	3.43E−01	1.64E+00	2.86E−01	3.14E−01	2.71E−01	3.52E−01	5.10E−01	2.24E+00	6.55E−01	4.49E−01	3.61E−01
F30											
Best	1.38E−01	1.47E+02	2.82E−01	5.75E+01	4.11E+00	8.96E+00	6.62E+01	1.41E+02	1.16E+02	4.60E+00	1.22E+00
Mean	5.27E+00	1.82E+02	1.93E+00	8.79E+01	1.11E+01	1.71E+01	8.14E+01	1.67E+02	1.45E+02	3.39E+01	7.28E+00
SD	8.30E+00	1.63E+01	1.04E+00	8.82E+00	4.56E+00	5.42E+00	6.99E+00	1.11E+01	1.02E+01	2.01E+01	4.81E+00
F31											
Best	3.63E+04	3.34E+10	1.83E+06	2.13E+07	1.36E+08	6.89E+08	5.14E+08	4.65E+06	5.11E+09	4.08E+04	1.02E+05
Mean	1.03E+05	4.86E+10	3.17E+06	7.16E+07	3.15E+08	1.95E+09	1.06E+09	6.41E+06	8.33E+09	4.56E+05	2.96E+05
SD	8.73E+04	9.58E+09	6.19E+05	2.74E+07	1.24E+08	1.23E+09	2.43E+08	1.00E+06	1.49E+09	5.16E+05	1.61E+05
F32											
Best	1.93E+04	2.88E+05	8.70E+04	1.11E+05	3.84E+04	2.04E+04	9.89E+04	7.16E+04	1.82E+05	5.13E+04	2.41E+04
Mean	4.08E+04	7.47E+05	1.35E+05	1.63E+05	5.96E+04	5.52E+04	1.35E+05	8.74E+04	3.19E+05	8.04E+04	4.99E+04
SD	1.65 E+04	3.98E+05	2.35E+04	2.68E+04	1.36E+04	1.93E+04	1.79E+04	6.35E+03	8.26E+04	1.36E+04	1.47E+04
F33											
Best	2.96E+02	8.73E+03	5.02E+02	5.06E+02	5.29E+02	5.95E+02	6.12E+02	5.34E+02	1.89E+03	4.74E+02	4.68E+02
Mean	4.17E+02	1.35E+04	5.23E+02	5.17E+02	6.05E+02	8.83E+02	6.73E+02	5.91E+02	2.69E+03	5.13E+02	5.06E+02
SD	2.30E+01	2.81E+03	1.03E+01	6.19E+00	5.15E+01	1.94E+02	4.04E+01	5.45E+01	5.57E+02	2.79E+01	1.68E+01
F34											
Best	3.35E+03	8.52E+03	7.74E+03	7.11E+03	3.10E+03	4.16E+03	5.31E+03	3.87E+03	8.45E+03	7.69E+03	3.87E+03
Mean	3.22E+03	9.69E+03	8.27E+03	8.45E+03	4.52E+03	5.20E+03	5.71E+03	4.82E+03	9.21E+03	8.55E+03	5.45E+03
SD	1.75E+02	3.73E+02	2.87E+02	3.92E+02	6.52E+02	5.09E+02	1.94E+02	4.18E+02	2.97E+02	3.83E+02	1.44E+03
F35											
Best	2.91E+03	1.09E+04	1.26E+03	1.60E+03	1.28E+03	1.25E+03	1.43E+03	6.00E+03	9.59E+03	1.17E+03	1.67E+03
Mean	3.01E+03	4.38E+04	1.90E+03	1.75E+03	1.90E+03	1.48E+03	1.53E+03	8.92E+03	1.51E+04	1.27E+03	2.13E+03
SD	4.27E+01	1.71E+04	7.96E+02	9.05E+01	6.05E+02	1.96E+02	6.64E+01	1.13E+03	3.81E+03	5.25E+01	4.06E+02
F36											
Best	2.03E+03	2.87E+03	2.03E+03	2.49E+03	2.09E+03	2.28E+03	2.42E+03	2.56E+03	2.87E+03	2.36E+03	2.30E+03
Mean	2.38E+03	3.33E+03	2.46E+03	2.73E+03	2.42E+03	2.55E+03	2.55E+03	2.98E+03	3.14E+03	2.69E+03	2.66E+03
SD	5.19E+01	1.65E+02	2.92E+02	1.08E+02	1.66E+02	1.84E+02	7.43E+01	1.77E+02	1.27E+02	1.22E+02	2.28E+02

Table 3 (continued)

	CHA	ACOR	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F37											
Best	2.71E+03	2.65E+03	2.43E+03	2.48E+03	2.38E+03	2.41E+03	2.44E+03	2.47E+03	2.55E+03	2.36E+03	2.36E+03
Mean	2.36E+03	2.72E+03	2.47E+03	2.52E+03	2.43E+03	2.47E+03	2.51E+03	2.53E+03	2.60E+03	2.45E+03	2.40E+03
SD	9.31E+00	2.83E+01	1.59E+01	1.55E+01	2.87E+01	3.87E+01	1.86E+01	2.35E+01	1.76E+01	3.84E+01	2.13E+01
F38											
Best	8.10E+03	1.35E+07	1.12E+04	3.61E+05	2.60E+04	3.91E+04	2.53E+05	5.44E+06	6.40E+05	9.80E+03	1.02E+05
Mean	1.04E+04	5.10E+07	2.42E+04	9.94E+05	1.56E+05	9.97E+05	4.47E+05	8.37E+06	3.00E+06	5.09E+04	2.39E+06
SD	9.93E+03	2.75E+07	1.25E+04	4.23E+05	1.36E+05	1.23E+06	1.15E+05	1.84E+06	1.71E+06	3.50E+04	1.30E+06

Figure 50 shows the average fitness values obtained by different algorithms for F37. Figure 50 shows that TLBO and DE have a stable convergence rate toward the better final solution and overtake all other methods. CHA is the third best in this multimodal function.

For F38, PSO shows a faster convergence rate at the initial stages of optimization procedure; however, it gets trapped in local optima. It is clear that TLBO slightly outperforms CHA in terms of the average final results for this function.

Moreover, a comparative study is performed to analyze the performance of all considered algorithms shown in Figs. 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 and 51. In Table 5, the algorithms are ranked based on the convergence speed for unimodal functions; and Table 6 depicts a comparison in terms of the average final solution obtained by each algorithm within a certain number of function evaluations for multimodal problems. As noted earlier, contrary to the unimodal functions, the final results are more important than the convergence rate for multimodal functions.

To obtain a general conclusion, the sum of the normalized scores and rank of each algorithm are proposed in these tables. It can be seen that CHA algorithm has lower rank compared to other algorithms. Therefore, we can conclude that CHA outperforms all of the compared algorithms in terms of the convergence speed and final solution. Further, PSO, GA and TLBO work very well because they rank 2, 3 and 4, respectively, among 11 algorithms.

3.3 The study of the number of fitness evaluations

As reported in Table 7, the average number of fitness evaluations to find the global optimum is used to compare the computational cost of each algorithm. The best results are highlighted in boldface. In this investigation, the maximum fitness evaluation is set to 500,000. In other words, if a method cannot converge to the global value before the maximum number of fitness evaluations, it will stop. Note that numbers in parentheses show the final solution for those algorithms that are not converged to the global optimum.

The results show that, for 20 benchmark functions, CHA converges to the global optimum with the least fitness evaluations. Moreover, CHA has better average final solutions in five out of eight CEC functions compared to other methods. As can be seen, KH and TLBO are the second and third most effective, respectively. Moreover, GA and PSO are superior to the other algorithms only for one function. It should be noted that ABC, GSA and

Table 4 Pair-wise statistical comparison of the algorithms by Wilcoxon Signed-Rank test ($\alpha = 0.05$)

	CHA versus ACOR				CHA versus HS				CHA versus DE				CHA versusGA				CHA versus PSO			
	p value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	*W*
F1	0.000002	3	462	+	0.000034	31	434	+	0.023038	122	343	+	0.000002	465	0	−	0.000359	406	59	−
F2	0.000002	0	465	+	0.000006	13	452	+	0.000002	3	462	+	0.000002	464	1	−	0.000002	465	0	−
F3	0.000002	2	463	+	0.017518	117	348	+	0.020671	120	345	+	0.000125	419	46	−	0.000002	0	465	−
F4	0.000002	0	465	+	0.158855	164	301	=	0.002765	87	378	+	0.829	243	222	=	0.416534	272	193	=
F5	0.000002	1	464	+	0.003379	90	375	+	0.000004	8	457	+	0.000053	36	429	+	0.000002	465	0	−
F6	0.000003	0	435	+	0.000002	1	464	+	0.195031	169.5	295.5	=	0.000003	435	0	−	0.000002	0	465	−
F7	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000022	26	439	+	0.000002	465	0	−
F8	0.002765	87	378	+	0.000002	465	0	−	0.000002	464	1	−	0.002765	87	378	+	0.000002	465	0	−
F9	0.120445	308	157	=	0.000016	442	23	−	0.000008	449	16	−	0.000388	405	60	−	0.000002	465	0	−
F10	0.000028	29	436	+	0.000031	435	30	−	0.000049	35	430	+	0.004992	96	369	+	0.382034	190	275	=
F11	0.000002	0	465	+	0.000002	0	465	+	0.000003	6	459	+	0.024308	123	342	=	0.877403	240	250	=
F12	0.000002	0	465	+	0.000002	0	465	+	0.011748	110	355	+	0.000002	465	0	−	0.000002	465	0	−
F13	0.000003	6	459	+	0.000771	69	396	+	0.001197	75	390	+	0.000963	72	393	+	0.000002	465	0	−
F14	0.000022	26	439	+	0.057096	325	140	=	0.000049	430	35	−	0.057096	325	140	=	0.000002	465	0	−
F15	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000031	435	30	−
F16	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	1	464	+
F17	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	1	464	+
F18	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F19	0.000002	0	465	+	0.000031	435	30	−	0.000002	0	465	+	0.000008	15	450	+	0.000005	10	455	+
F20	0.000002	0	465	+	0.000002	465	0	−	0.191522	169	296	=	0.000034	31	434	+	0.000049	35	435	+
F21	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000014	21	444	+	0.000002	0	465	+
F22	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F23	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000008	16	449	+
F24	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F25	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	2	463	+
F26	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.011748	110	355	+
F27	0.000002	0	465	+	1.000000	0	0	=	0.014306	0	21	+	0.000002	0	465	+	0.000002	0	465	+
F28	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000010	18	447	+
F29	0.000002	0	465	+	0.000453	62	403	+	0.000002	1	464	+	0.028486	339	126	−	0.975387	231	234	=
F30	0.000002	0	465	+	0.085896	316	149	=	0.000002	0	465	+	0.000894	71	394	+	0.000020	25	440	+
F31	0.000002	0	465	+	0.000002	0	465	+	0.000002	1	464	+	0.000002	0	465	+	0.000002	0	465	+
F32	0.000002	1	464	+	0.000771	69	396	+	0.000002	3	462	+	0.000002	0	465	+	0.199031	171	294	=
F33	0.000003	6	459	+	0.002765	87	378	+	0.011710	109	356	+	0.000636	67	398	+	0.000189	51	414	+
F34	0.00002	25	440	+	0.000049	35	430	+	0.000008	15	450	+	0.000388	405	60	−	0.020025	121	344	+
F35	0.000002	0	465	+	0.000031	435	30	−	0.136583	160	305	=	0.000008	449	16	−	0.000049	430	35	−
F36	0.000002	0	465	+	0.382034	275	190	=	0.000771	69	396	+	0.749871	248	217	=	0.147040	162	303	=
F37	0.000002	0	465	+	0.015658	115	350	+	0.000002	1	464	+	0.191522	169	296	=	0.009842	107	358	+
F38	0.000002	0	465	+	0.000261	410	55	−	0.000002	0	465	+	0.000044	34	441	+	0.000002	1	464	+
(+/=/−)	37/1/0				26/5/7				32/3/3				22/5/9				19/6/13			

Table 4 (continued)

	CHA versus CS				CHA versus GSA				CHA versus ABC				CHA versus TLBO				CHA versus KH			
	p value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	W	*p* value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	*W*	*p* value	*T*+	*T*−	*W*
F1	0.000007	14	451	+	0.000002	0	465	+	0.000018	24	441	+	0.011748	110	355	+	0.000002	1	465	+
F2	0.000002	0	465	+	0.000002	0	465	+	0.000002	1	464	+	0.000002	3	462	+	0.000002	0	465	+
F3	0.000002	0	465	+	0.000002	0	465	+	1.000000	0	0	=	0.141390	161	304	=	0.000007	14	451	+
F4	0.000002	1	464	+	0.000002	0	465	+	0.000075	40	425	+	0.000894	71	394	+	0.000005	11	454	+
F5	0.000002	3	462	+	0.027029	125	340	+	0.000002	0	465	+	0.688359	252	231	=	0.059836	141	324	=
F6	0.000002	0	465	+	0.000002	0	465	+	0.000006	13	425	+	0.000058	31.5	403.5	+	0.000002	0	465	+
F7	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F8	0.002765	87	378	+	0.002765	78	387	+	0.002765	87	378	+	0.000002	465	0	−	0.416534	193	272	=
F9	0.002585	86	379	+	0.000002	0	465	+	0.102011	153	312	=	0.000069	426	39	−	0.007731	103	362	+
F10	0.000003	6	459	+	0.000002	0	465	+	0.000053	36	429	+	0.135908	160	305	=	0.205888	171	294	=
F11	0.000002	2	463	+	0.000002	0	465	+	0.000002	0	465	+	0.110926	155	310	=	0.000002	0	465	+
F12	0.000002	1	464	+	0.000002	1	464	+	0.000009	17	448	+	0.000002	0	465	+	0.000002	0	465	+
F13	0.000008	15	450	+	0.942611	229	236	=	0.000189	51	414	+	0.000002	0	465	+	0.298944	182	283	=
F14	0.926255	237	228	=	0.393334	191	274	=	0.530440	202	263	=	0.014795	114	351	=	0.000034	434	31	−
F15	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000097	422	43	−	0.000002	0	465	+
F16	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.308615	183	282	=
F17	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.007271	102	363	=	0.000002	0	465	+
F18	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	465	0	−	0.000002	0	465	+
F19	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	465	0	−	0.003609	374	91	−
F20	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000075	40	425	+	0.000006	452	13	−
F21	0.001593	79	386	+	0.000002	0	465	+	0.000002	0	465	+	0.000004	7	458	+	0.000002	0	465	+
F22	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F23	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	465	0	−	0.000002	0	465	+
F24	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	465	0	−	0.075213	141	319	=
F25	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F26	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	465	0	−	0.000002	0	465	+
F27	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.942611	229	236	=	0.002765	378	87	−
F28	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	465	0	−	0.000002	462	3	−
F29	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.082206	148	317	=	0.093676	151	314	=
F30	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+
F31	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.012518	111	354	+	0.003185	89	376	+
F32	0.000189	51	414	+	0.017508	116	348	+	0.000002	0	465	+	0.000963	72	393	+	0.102011	153	312	=
F33	0.000371	58	407	+	0.000049	35	435	+	0.000014	21	444	+	0.102011	153	312	=	0.942411	228	235	=
F34	0.002585	86	379	+	0.027029	125	340	+	0.000034	31	434	+	0.000065	38	427	+	0.020671	120	345	+
F35	0.000031	435	30	−	0.382034	190	275	=	0.011738	109	356	+	0.000002	465	0	−	0.000124	420	45	−
F36	0.089718	150	350	=	0.000003	6	459	+	0.000002	0	465	+	0.000963	72	393	+	0.006836	101	364	+
F37	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.926255	237	228	=	0.000053	429	36	−
F38	0.000002	0	465	+	0.000002	0	465	+	0.000002	0	465	+	0.416534	272	193	=	0.000002	0	465	+
(+/=/−)	35/2/1				35/3/0				35/3/0				17/11/10				22/9/7			

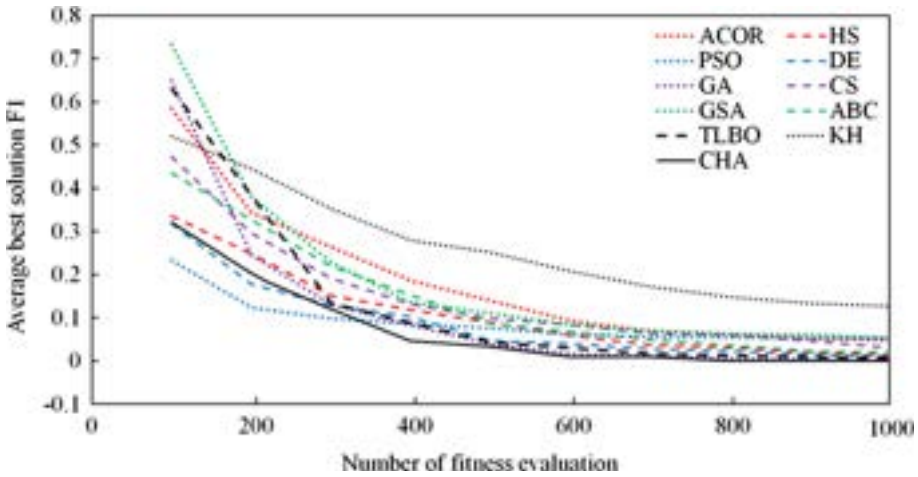

Fig. 14 Convergence of algorithms for the test function F1

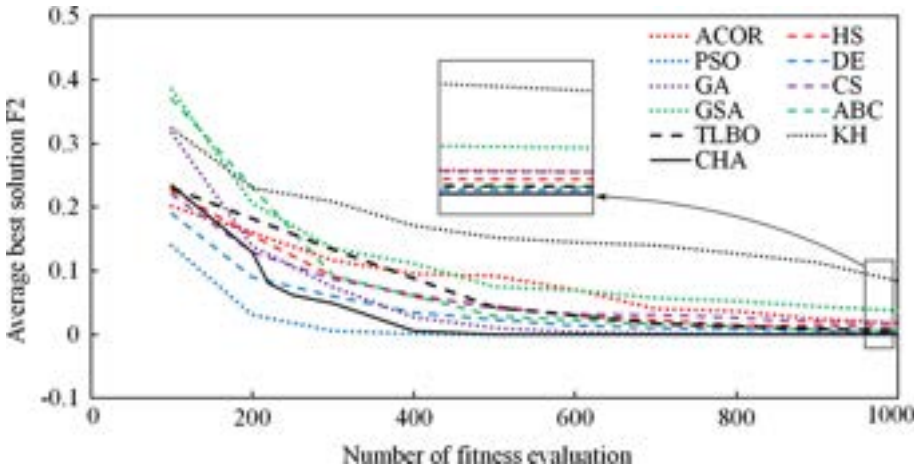

Fig. 15 Convergence of algorithms for the test function F2

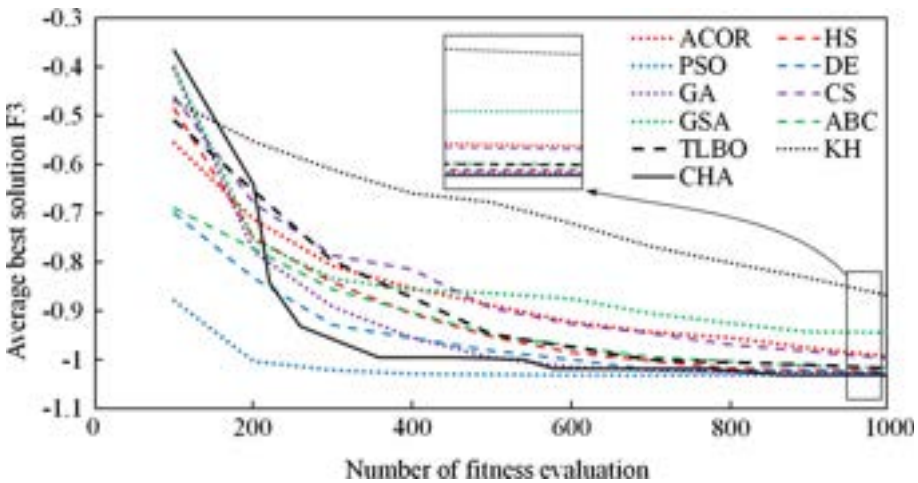

Fig. 16 Convergence of algorithms for the test function F3

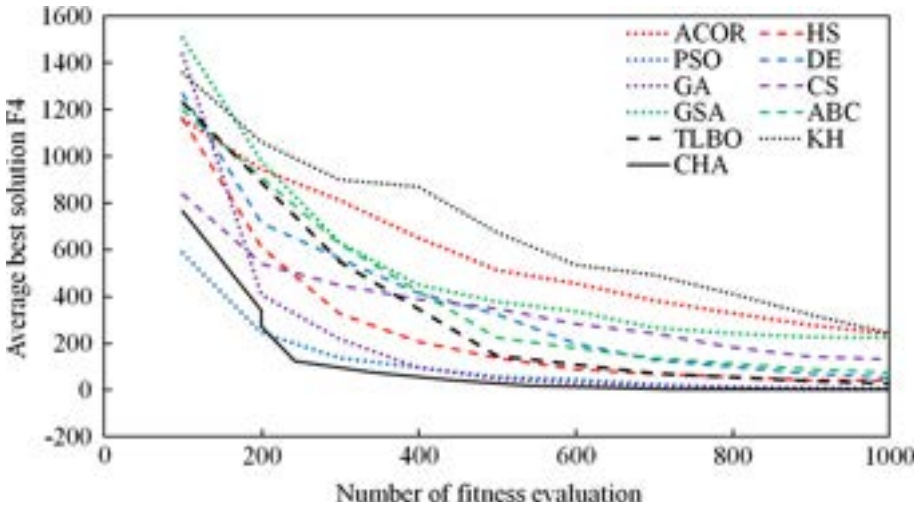

Fig. 17 Convergence of algorithms for the test function F4

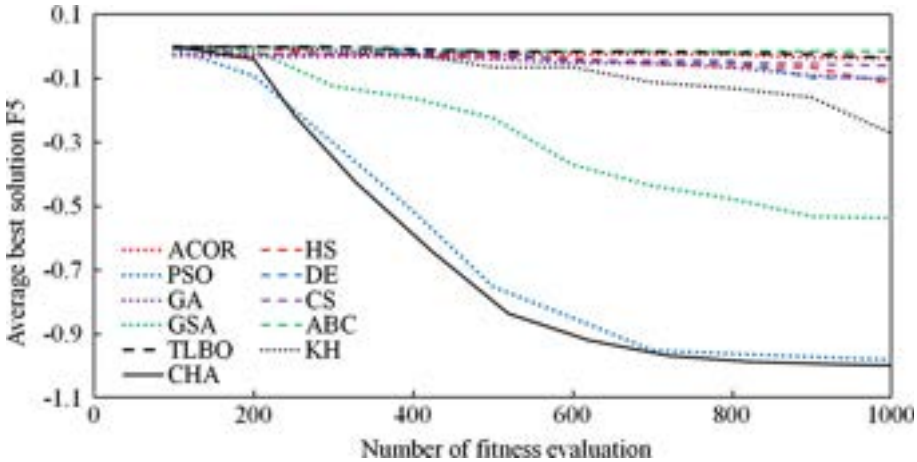

Fig. 18 Convergence of algorithms for the test function F5

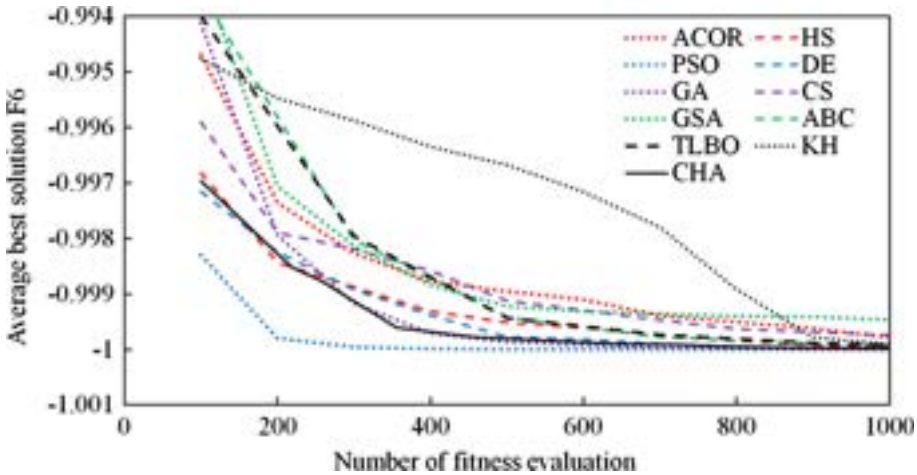

Fig. 19 Convergence of algorithms for the test function F6

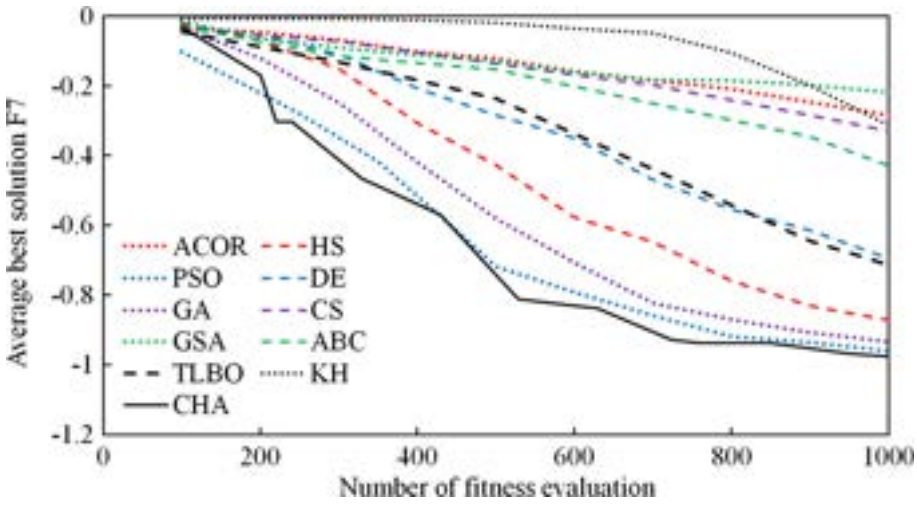

Fig. 20 Convergence of algorithms for the test function F7

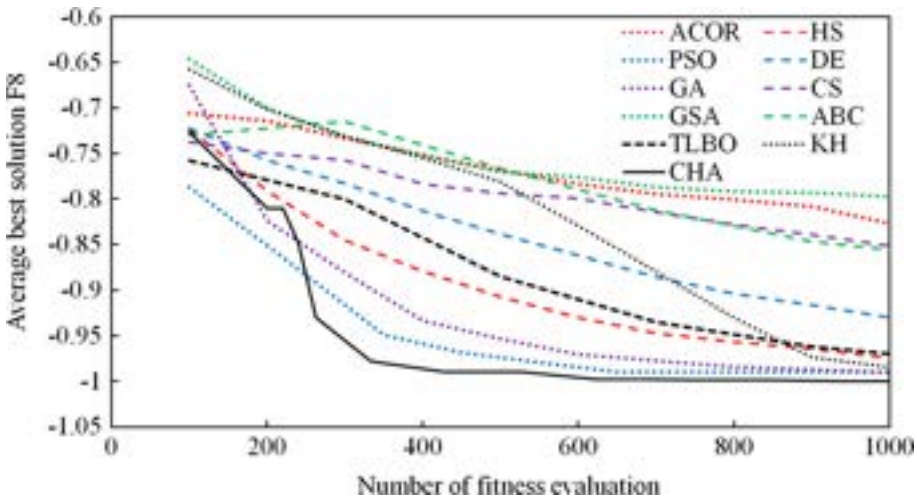

Fig. 21 Convergence of algorithms for the test function F8

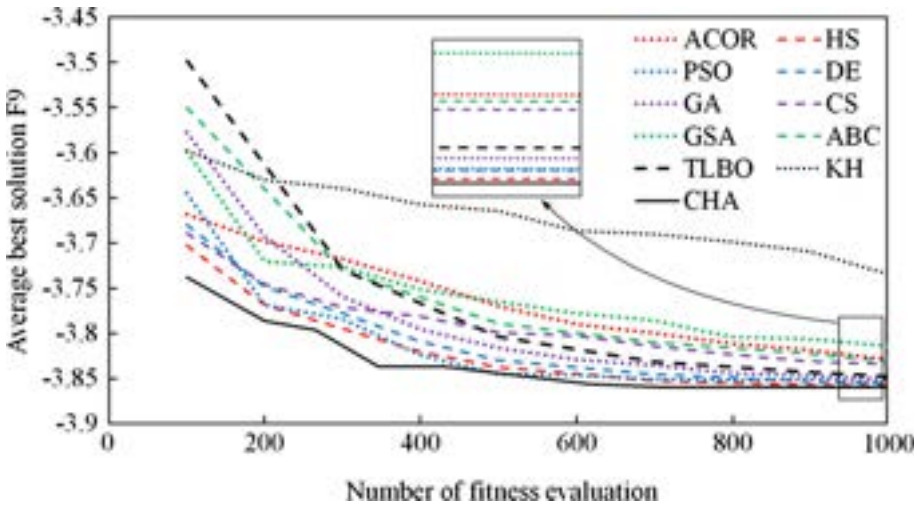

Fig. 22 Convergence of algorithms for the test function F9

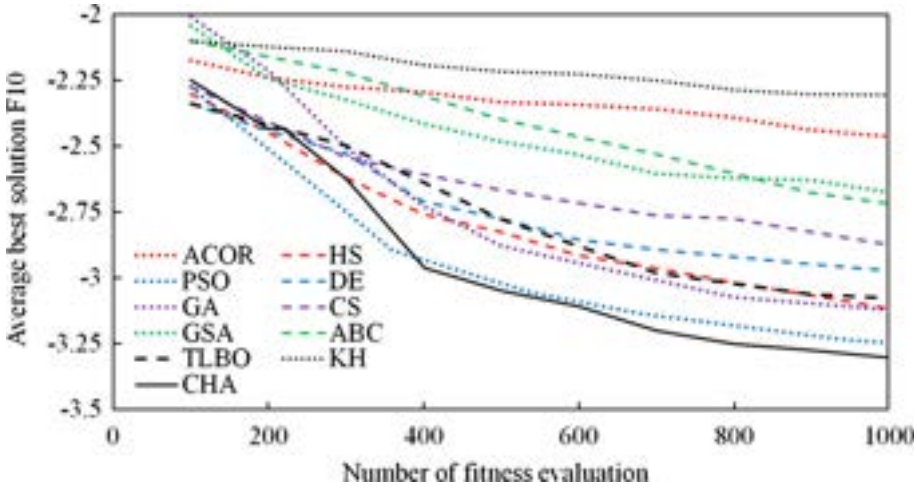

Fig. 23 Convergence of algorithms for the test function F10

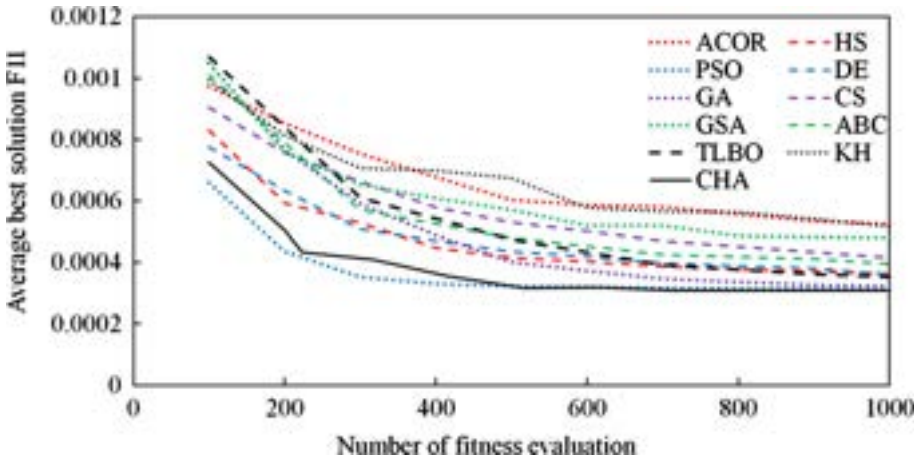

Fig. 24 Convergence of algorithms for the test function F11

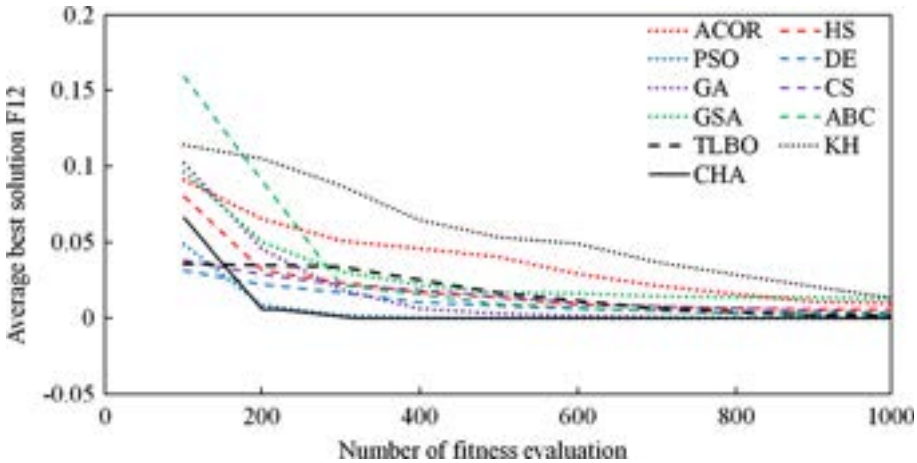

Fig. 25 Convergence of algorithms for the test function F12

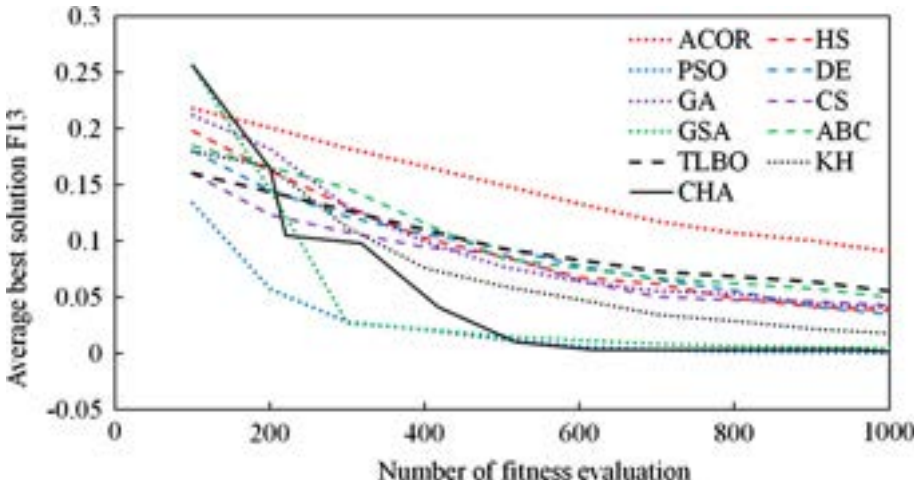

Fig. 26 Convergence of algorithms for the test function F13

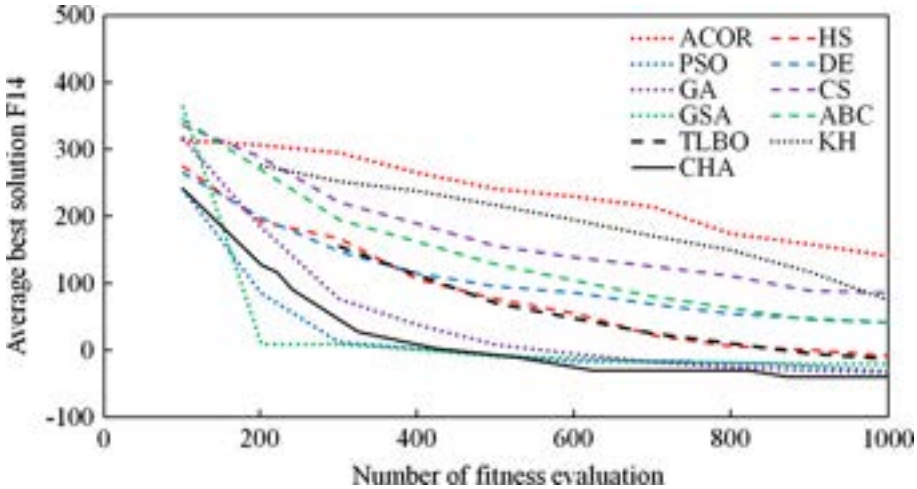

Fig. 27 Convergence of algorithms for the test function F14

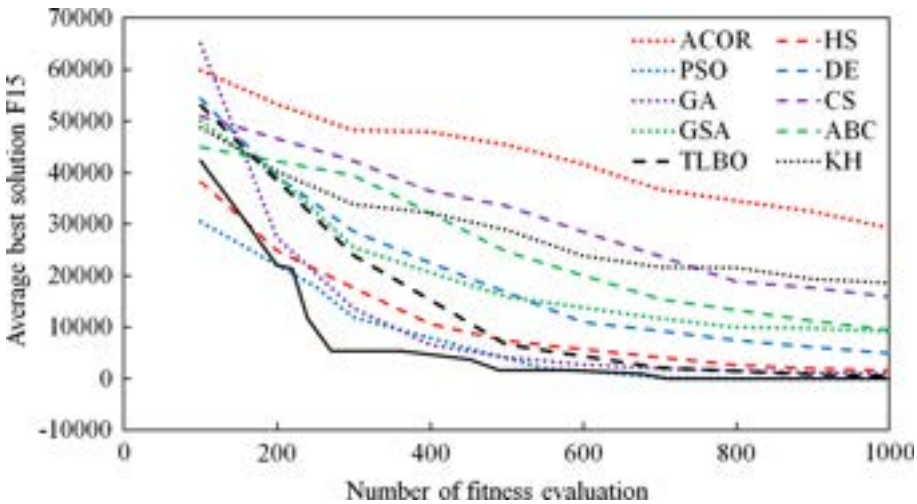

Fig. 28 Convergence of algorithms for the test function F15

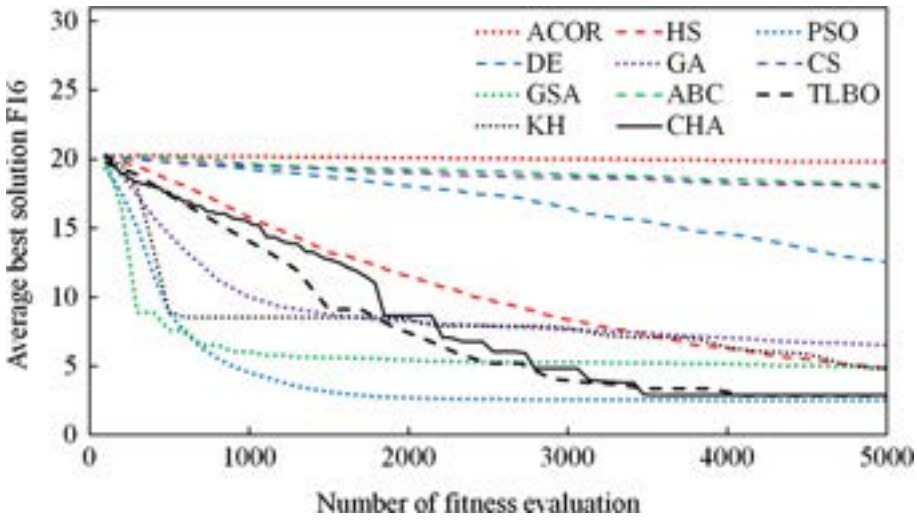

Fig. 29 Convergence of algorithms for the test function F16

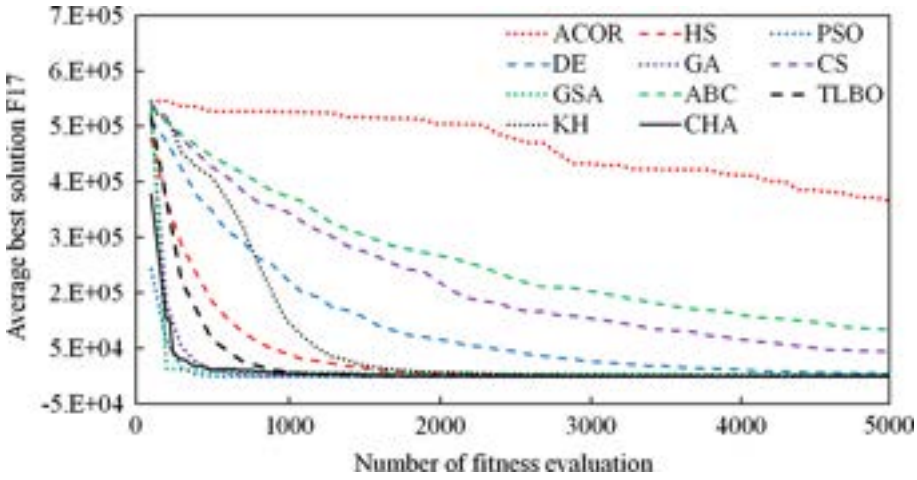

Fig. 30 Convergence of algorithms for the test function F17

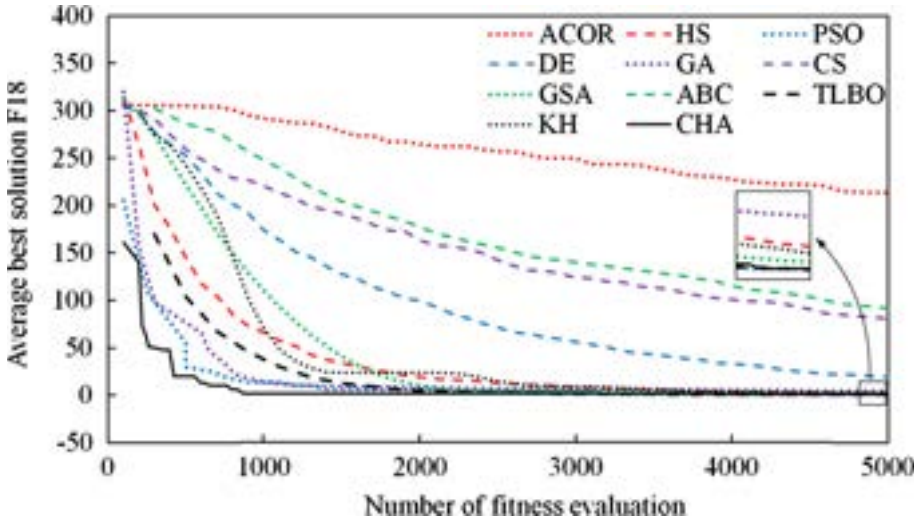

Fig. 31 Convergence of algorithms for the test function F18

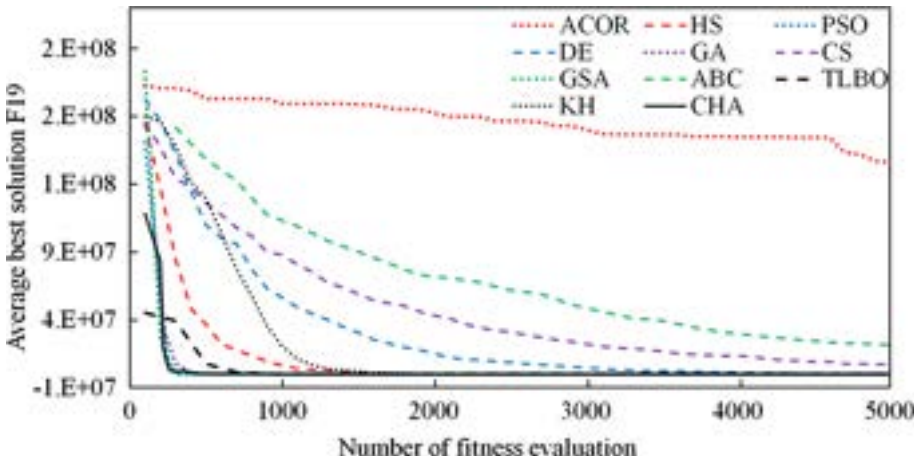

Fig. 32 Convergence of algorithms for the test function F19

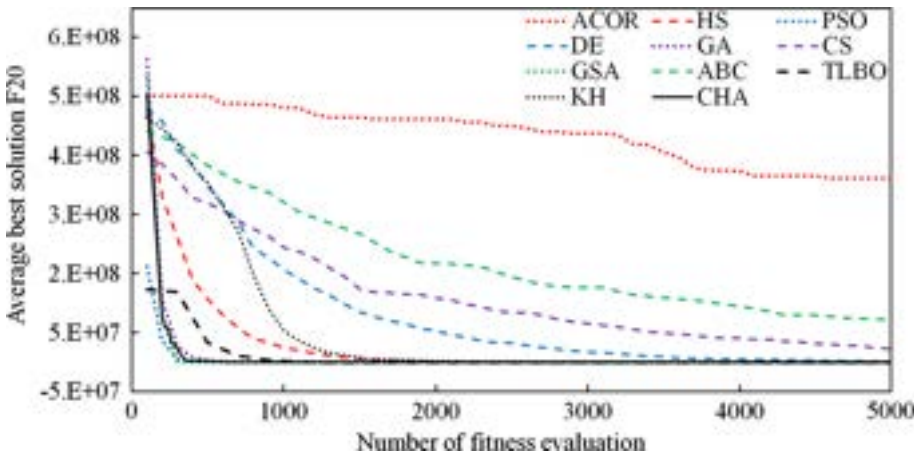

Fig. 33 Convergence of algorithms for the test function F20

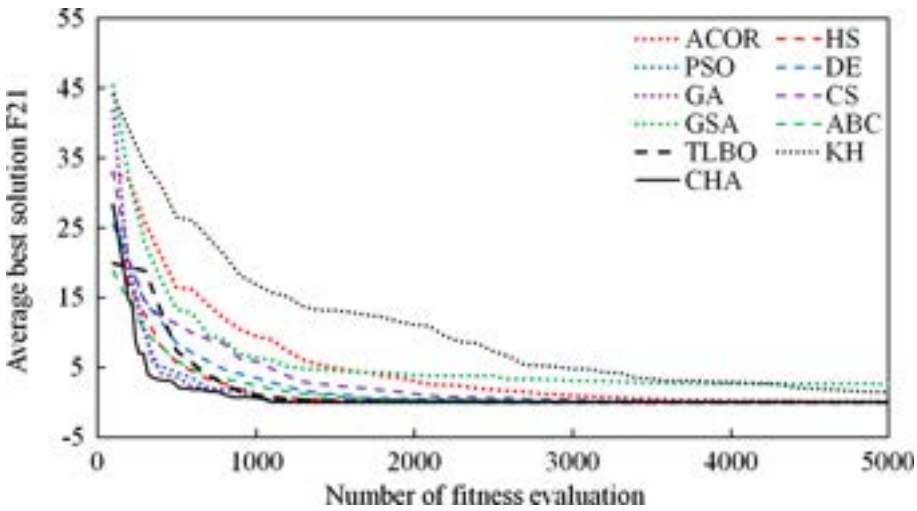

Fig. 34 Convergence of algorithms for the test function F21

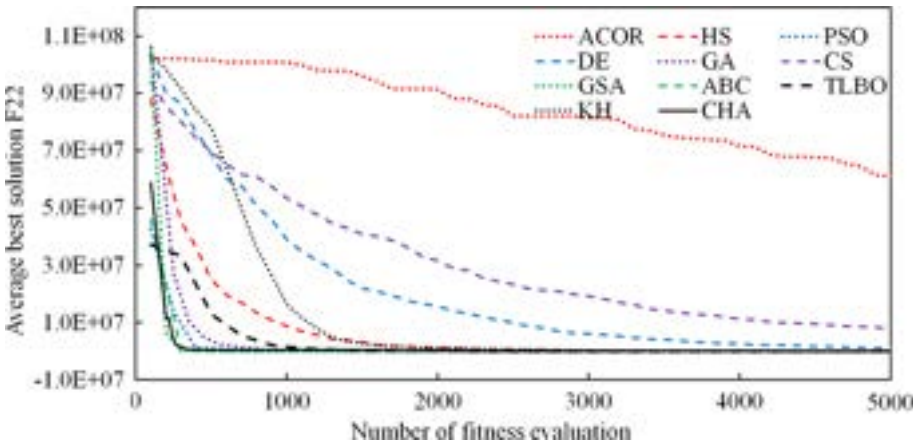

Fig. 35 Convergence of algorithms for the test function F22

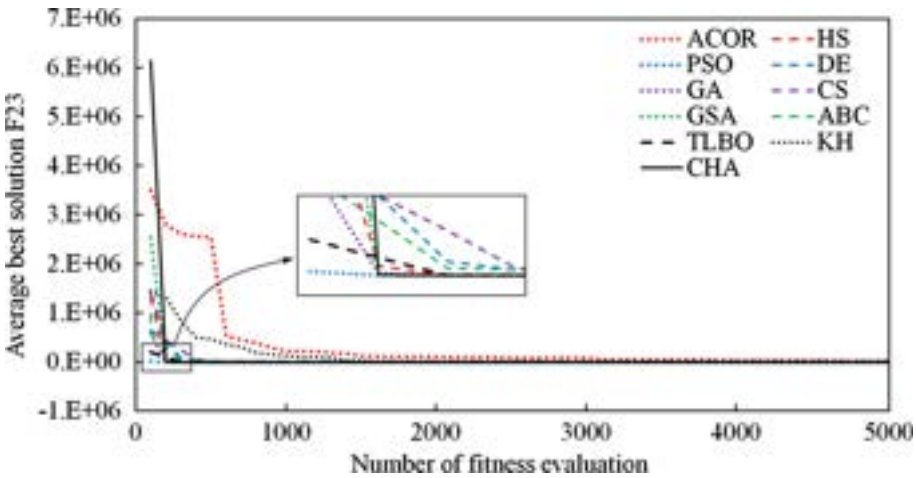

Fig. 36 Convergence of algorithms for the test function F23

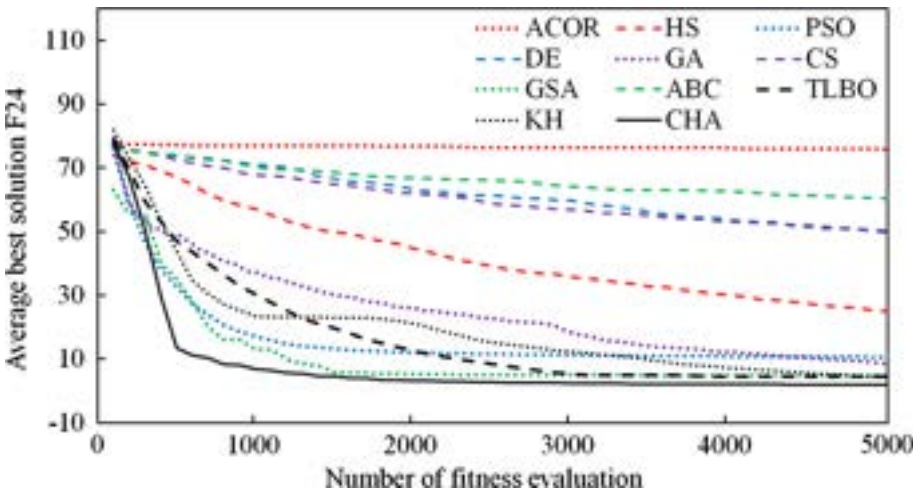

Fig. 37 Convergence of algorithms for the test function F24

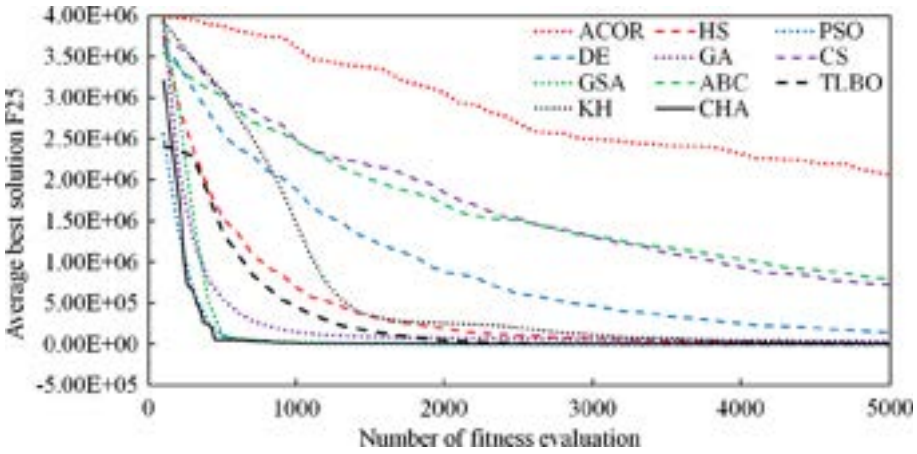

Fig. 38 Convergence of algorithms for the test function F25

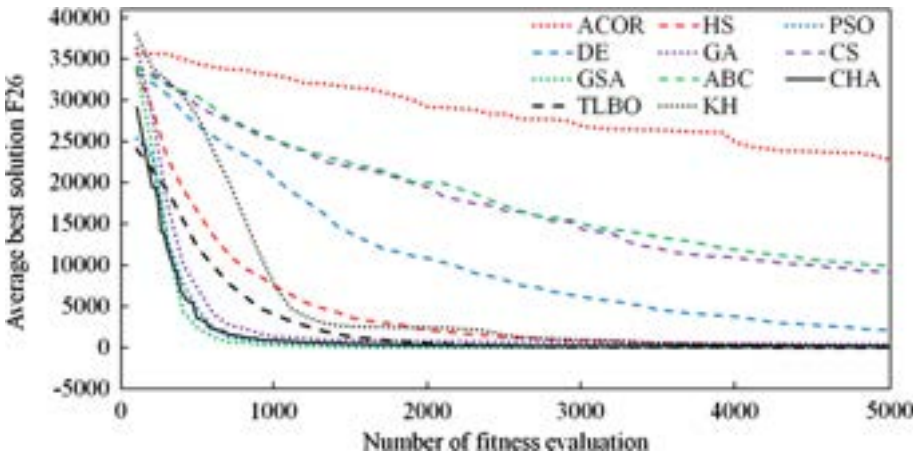

Fig. 39 Convergence of algorithms for the test function F26

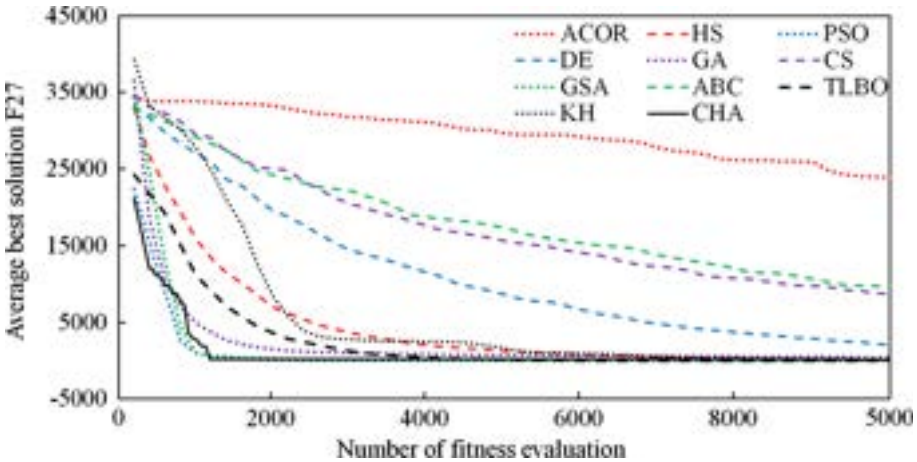

Fig. 40 Convergence of algorithms for the test function F27

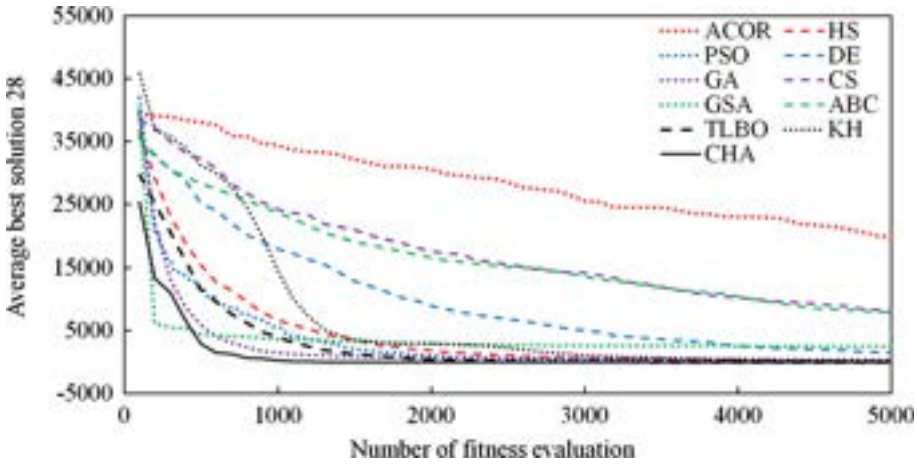

Fig. 41 Convergence of algorithms for the test function F28

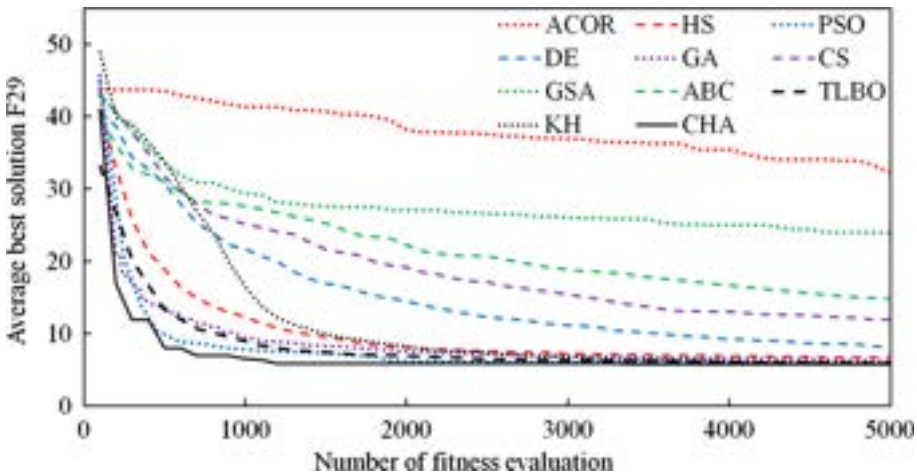

Fig. 42 Convergence of algorithms for the test function F29

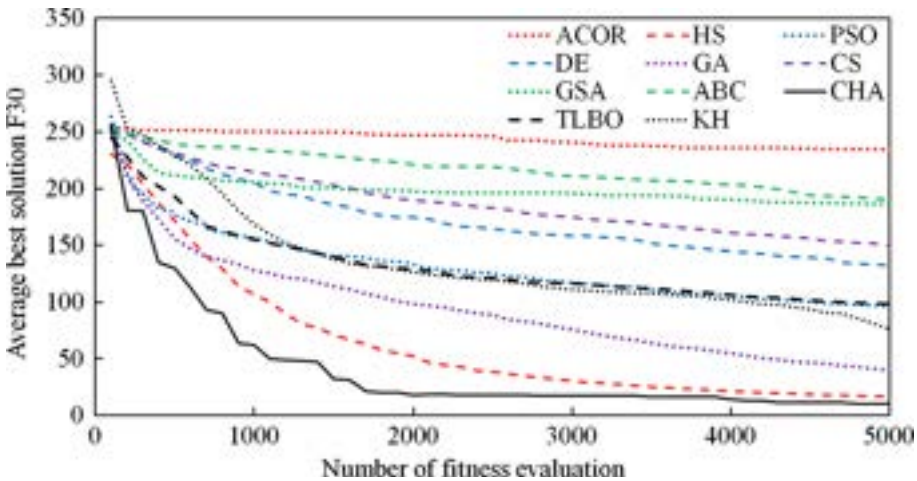

Fig. 43 Convergence of algorithms for the test function F30

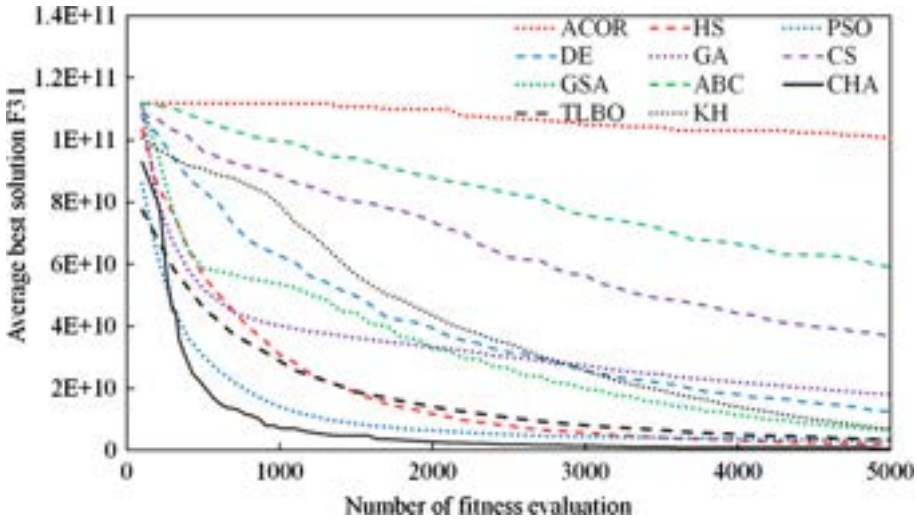

Fig. 44 Convergence of algorithms for the test function F31

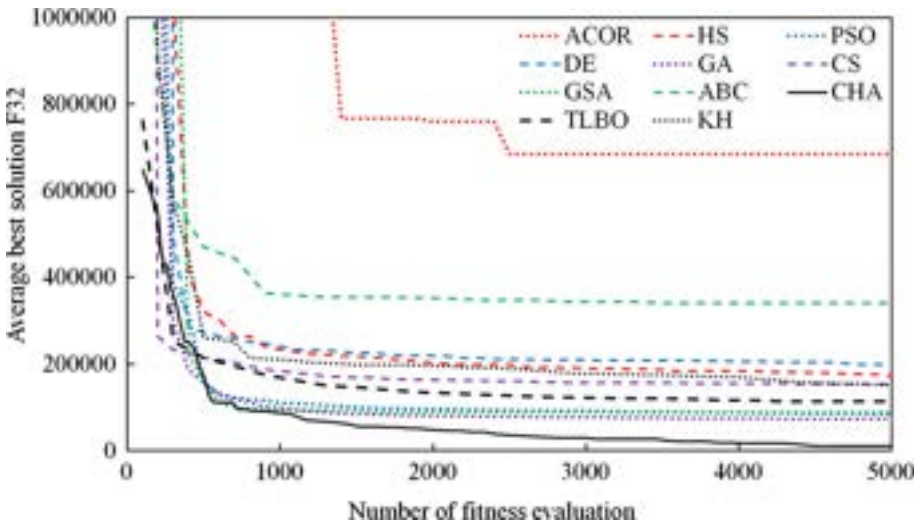

Fig. 45 Convergence of algorithms for the test function F32

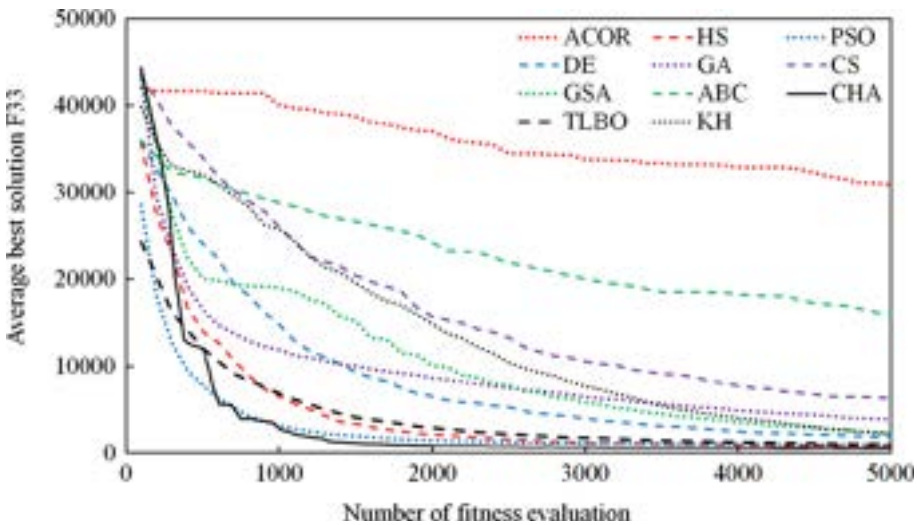

Fig. 46 Convergence of algorithms for the test function F33

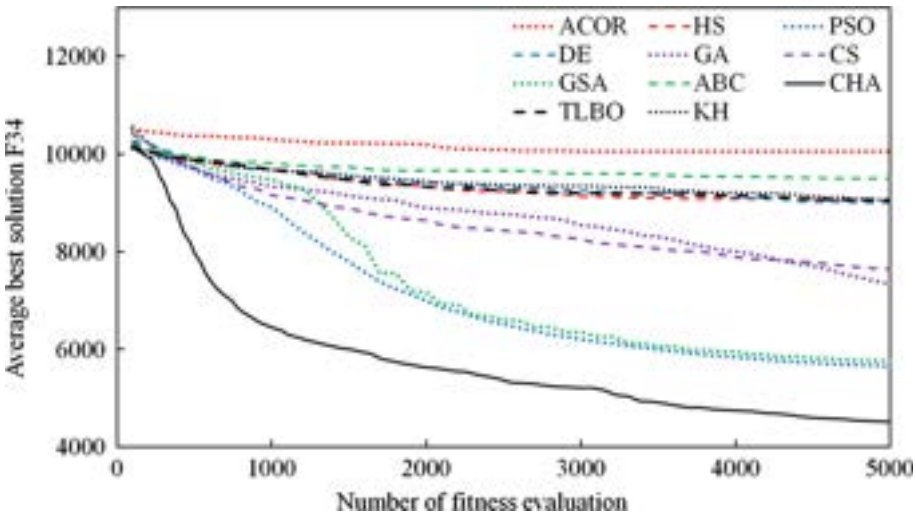

Fig. 47 Convergence of algorithms for the test function F34

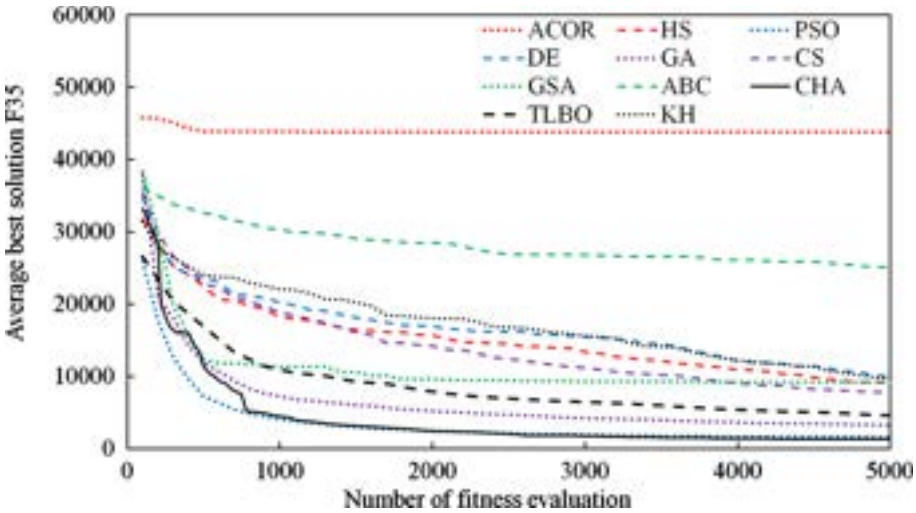

Fig. 48 Convergence of algorithms for the test function F35

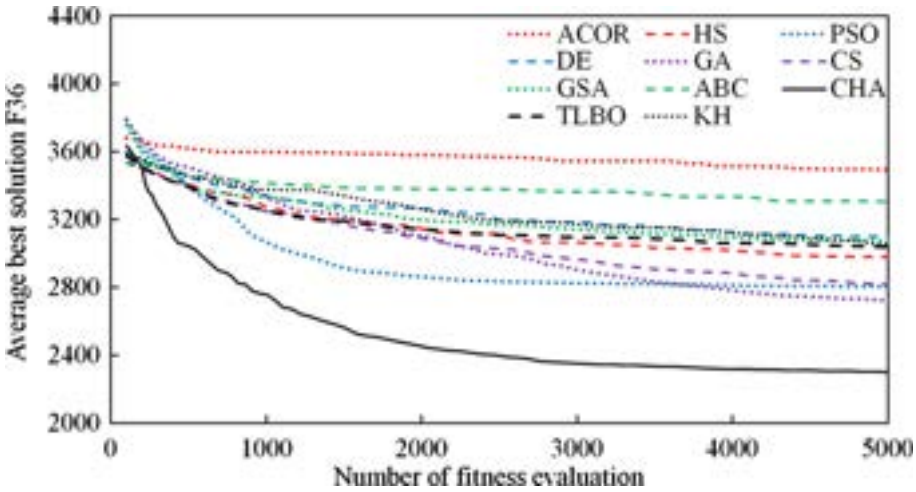

Fig. 49 Convergence of algorithms for the test function F36

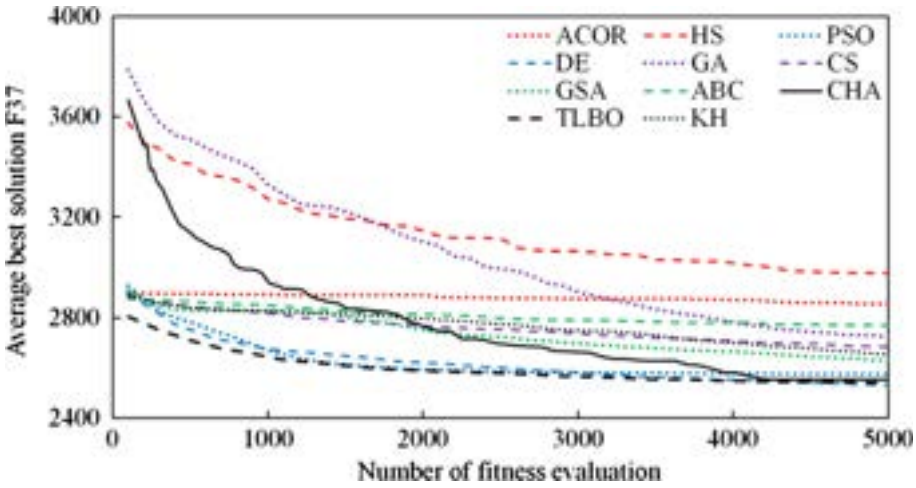

Fig. 50 Convergence of algorithms for the test function F37

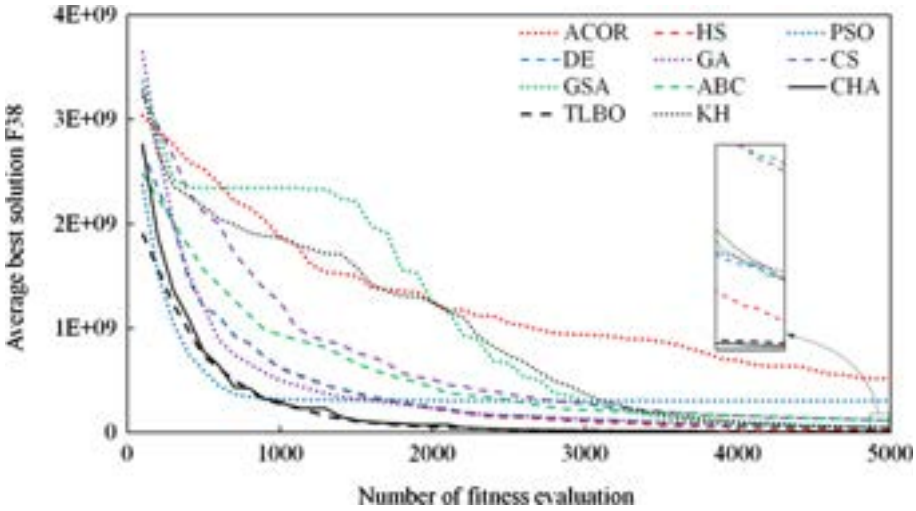

Fig. 51 Convergence of algorithms for the test function F38

ACOR show poor results among the algorithms for all considered functions.

4 Conclusion

In this study, we propose an art-inspired algorithm for solving optimization problems called CHA, which is a population-based metaheuristic method. CHA is developed based on the Munsell Color System and harmony between colors. The performance of CHA has been examined using various well-known benchmark test functions. For assessing the effectiveness and the robustness of the proposed method, 30 independent runs have been performed and the results compared with those obtained using ten well-known metaheuristic algorithms including ACOR, HS, DE, GA, PSO, CS, GSA and ABC. The results show that besides the simplicity of the proposed algorithm, CHA can outperform these algorithms in terms of the convergence speed and the number of function evaluations. As future works, CHA

Table 5 Comparison of CHA with other algorithms for unimodal functions in terms of the convergence speed

	CHA	ACO	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F1	1.00	0.10	0.60	0.70	0.90	0.40	0.30	0.20	0.50	0.80	0.00
F4	1.00	0.10	0.60	0.50	0.90	0.80	0.30	0.20	0.40	0.70	0.00
F5	1.00	0.22	0.56	0.44	0.56	0.89	0.33	0.78	0.00	0.11	0.67
F6	0.80	0.10	0.60	0.70	0.90	1.00	0.20	0.00	0.50	0.40	0.30
F7	1.00	0.10	0.70	0.50	0.80	0.90	0.30	0.00	0.40	0.60	0.20
F8	1.00	0.10	0.60	0.40	0.80	0.90	0.20	0.00	0.30	0.50	0.70
F12	1.00	0.22	0.33	0.78	0.89	1.00	0.44	0.11	0.56	0.67	0.00
F14	1.00	0.00	0.56	0.44	0.89	0.78	0.11	0.78	0.33	0.67	0.22
F15	1.00	0.00	0.60	0.50	0.70	0.90	0.20	0.40	0.30	0.80	0.10
F17	0.70	0.00	0.50	0.30	0.80	1.00	0.20	0.90	0.10	0.60	0.40
F21	1.00	0.20	0.60	0.40	0.90	0.80	0.30	0.10	0.60	0.70	0.00
F22	0.40	0.00	0.30	0.20	0.70	0.80	0.10	0.90	1.00	0.60	0.50
F23	0.89	0.00	0.67	0.33	0.89	1.00	0.22	0.78	0.44	0.56	0.11
F24	1.00	0.00	0.44	0.22	0.67	0.56	0.33	0.89	0.11	0.78	0.67
F25	1.00	0.00	0.50	0.30	0.60	0.90	0.20	0.80	0.10	0.70	0.40
F26	0.70	0.00	0.50	0.30	0.60	0.90	0.20	1.00	0.10	0.80	0.40
F27	1.00	0.00	0.40	0.30	0.70	0.90	0.20	0.80	0.10	0.60	0.50
F28	1.00	0.00	0.56	0.22	0.78	0.67	0.11	0.33	0.11	0.89	0.44
F29	1.00	0.00	0.50	0.40	0.60	0.90	0.30	0.10	0.20	0.80	0.70
F31	1.00	0.00	0.90	0.40	0.30	0.80	0.20	0.60	0.10	0.70	0.50
F32	1.00	0.00	0.30	0.20	0.90	0.70	0.50	0.80	0.10	0.60	0.40
Sum	19.49	1.14	11.31	8.54	15.77	17.49	5.26	10.47	6.36	13.57	7.21
Rank	1	11	5	7	3	2	10	6	9	4	8

Table 6 Comparison of CHA with other algorithms for multimodal functions in terms of the final solution

	CHA	ACO	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F2	1.00	0.33	0.50	0.83	1.00	1.00	0.33	0.17	0.67	0.67	0.00
F3	1.00	0.25	0.63	0.75	0.88	1.00	0.38	0.13	0.50	0.50	0.00
F9	1.00	0.22	0.89	0.78	0.67	0.78	0.44	0.11	0.33	0.56	0.00
F10	1.00	0.10	0.70	0.50	0.80	0.90	0.40	0.20	0.30	0.60	0.00
F11	1.00	0.00	0.63	0.63	0.88	1.00	0.38	0.25	0.50	0.75	0.13
F13	0.90	0.00	0.50	0.60	0.40	1.00	0.30	0.80	0.20	0.10	0.70
F16	0.86	0.00	0.71	0.43	0.57	1.00	0.29	0.71	0.14	0.86	0.71
F18	1.00	0.00	0.63	0.38	0.50	1.00	0.25	0.88	0.13	1.00	0.75
F19	1.00	0.00	1.00	1.00	1.00	1.00	0.67	1.00	0.33	1.00	1.00
F20	1.00	0.00	0.75	1.00	1.00	1.00	0.50	1.00	0.25	1.00	1.00
F22	1.00	0.00	1.00	0.67	1.00	1.00	0.33	1.00	1.00	1.00	1.00
F30	1.00	0.00	0.89	0.44	0.78	0.56	0.33	0.22	0.11	0.56	0.67
F33	1.00	0.00	0.86	0.57	0.29	0.71	0.14	0.43	0.14	0.71	0.43
F34	1.00	0.00	0.29	0.29	0.57	0.86	0.43	0.71	0.14	0.29	0.29
F35	1.00	0.00	0.44	0.22	0.78	0.89	0.56	0.44	0.11	0.67	0.33
F36	1.00	0.00	0.60	0.20	0.90	0.80	0.70	0.30	0.10	0.50	0.40
F37	0.89	0.11	0.00	1.00	0.33	0.78	0.44	0.67	0.22	1.00	0.56
F38	1.00	0.00	0.71	0.57	0.57	0.14	0.43	0.57	0.29	0.86	0.57
Sum	17.65	1.02	11.72	10.85	12.91	15.41	7.30	9.59	5.47	12.61	8.53
Rank	1	11	5	6	3	2	9	7	10	4	8

Table 7 The average number of fitness evaluations to find the global optimum

	CHA	ACO	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
F1	5640	21,600	298,500	10,600	4400	**3800**	41,300	108,200	500,000 (1.58e−13)	16,000	18,350
F2	**4040**	11,000	193,400	6900	4300	4700	29,800	4300	413,039	8800	67,260
F3	**3860**	16,600	227,400	10,900	6600	4200	55,500	123,100	417,819	160,567	122,150
F4	**6600**	500,000 (1.2e−3)	500,000 (1.4e−3)	70,200	395,500	66,500	164,200	343,300	500,000 (2.1e−5)	332,600	138,960
F5	11,060	14,500	283,800	10,300	27,000	**7200**	59,800	143,000	402,011	390,227	75,233
F6	**2420**	9000	158,200	5400	3900	2800	21,700	83,400	408,916	273,733	132,087
F7	**3960**	24,100	215,100	13,300	208,800	4000	52,600	169,700	420,964	118,893	141,427
F8	**2960**	55,300	271,200	24,700	285,100	4600	84,000	136,600	442,233	3400	225,793
F9	4100	17,100	294,500	12,100	22,700	3900	55,100	142,700	418,210	**3600**	114,877
F10	**2980**	42,600	269,600	32,800	249,000	3900	164,600	108,200	475,575	500,000 (3.9e−7)	117,327
F11	**3080**	500,000 (3.1e−4)	500,000 (3.1e−4)	38,400	20,6400	41,300	52,100	14,7000	500,000 (3.e−4)	68,413	19,9180
F12	4980	16,800	263,800	9300	**4300**	4400	26,900	30,100	445,328	11,800	14,3260
F13	20,760	18,700	18,9300	14,000	24,5100	18,500	65,200	77,100	42,3279	17,000	**13,700**
F14	**8120**	225,800	344,900	60,700	500,000 (− 49.9995)	138,800	11,4900	203,600	500,000 (− 49.9997)	442,680	142,527
F15	**16,680**	97,700	27,6000	60,700	303,400	17,300	121,700	500,000 (4.1)	500,000 (5.6e−14)	17,200	247,860
F16	**70,420**	500,000 (2.7e−10)	500,000 (4.2e−12)	147,600	500,000 (1.0e−8)	77,200	500,000 (7.9e−7)	500,000 (0.91099)	500,000 (3.1e−9)	72,100	138,307
F17	**64,880**	500,000 (0.66667)	336,700	137,600	500,000 (0.62)	73,700	500,000 (0.63)	294,900	500,000 (0.66)	500,000	442,057
F18	43,600	500,000 (0.13904)	368,200	92,000	433,900	34,900	500,000 (2.0e−12)	500,000 (1.5e−02)	500,000 (1.6e−02)	**21,800**	203,067
F19	**64,900**	496,200	309,200	84,800	304,100	85,500	457,200	500,000 (3.9e−03)	500,000 (2.14e−12)	**68,200**	192,930
F20	62,460	500,000 (1.3e−15)	325,600	85,600	331,600	53,200	287,000	256,700	500,000 (1.9e−13)	59,200	**51,100**
F21	**44,520**	224,400	284,900	59,000	202,700	60,600	47,800	209,100	500,000 (8.81e−08)	500,000 (0.67)	182,880
F22	**61,220**	499,600	500,000 (46.5)	500,000 (2.4)	500,000 (12.8)	62,700	500,000 (6.6e−02)	500,000 (56.1)	500,000 (428.3)	500,000 (0.68)	307,647
F23	115,040	500,000	500,000	135,500	500,000	87,400	500,000	500,000	500,000	**38,200**	262,770

Table 7 (continued)

	CHA	ACO	HS	DE	GA	PSO	CS	GSA	ABC	TLBO	KH
		(1.8e−10)	(3.5e−12)		(1.4e−09)		(1.4e−14)	(1.5)	(3.8e−13)		
F24	**78,040**	500,000	500,000	435,900	500,000	81,200	500,000	500,000	500,000	96,400	334,673
		(0.28033)	(2.7e−11)		(0.0067405)		(6.1e−06)	(0.23229)	(4.6635)		
F25	257,860	487,800	410,100	93,800	500,000	163,300	278,400	500,000	500,000	296,040	**23,200**
					(3.4e−13)			(35.4)	(8.9e−11)		
F26	263,360	447,800	368,400	86,300	426,600	113,600	256,500	500,000	500,000	140,697	**21,400**
								(0.27457)	(1.2e−15)		
F27	20,460	111,900	15,100	19,200	38,600	27,000	71,100	331,500	500,000	9300	**4800**
									(6.8e−15)		
F28	**20,460**	457,200	366,600	85,900	412,200	36,400	261,000	500,000	500,000	21,200	276,407
								(30.6177)	(6.1e−15)		
F29	**34,800**	227,200	195,800	116,700	105,900	199,500	161,100	120,500	425,600	425,600	131,077
F30	**38,000**	172,300	320,600	253,300	136,300	43,800	500,000	278,100	450,000	500,000	135,870
							(14.7796)			(1.519)	
F31	**500,000**	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
	(100.1)	(2,431,593.9)	(1857.6)	(105.4)	(2831.7)	(1,941,264,397.5)	(156.0)	(1691.5)	(5884.1)	(3434.1)	(12,490.7)
F32	**500,000**	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
	(300.35)	(746,683.6)	(4907.0)	(47,292.3)	(18,727.4)	(49,680.4)	(10,066.4)	(73,548.5)	(270,501.5)	(336.35)	(33,651.9)
F33	**500,000**	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
	(429.1)	(429.7)	(508.3)	(482.9)	(482.5)	(875.7)	(441.3)	(532.7)	(469.1)	(451.5)	(504.3)
F34	**500,000**	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
	(4113.4)	(9486.7)	(3224.7)	(7454.3)	(7473.2)	(5162.1)	(4383.5)	(2547.7)	(8854.1)	(4100.6)	(4949.7)
F35	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	**500,000**
	(1182.8)	(19,161.7)	(1415.7)	(1209.9)	(1216.6)	(1472.7)	(1166.1)	(1387.7)	(4108.0)	(1223.7)	**(1175.1)**
F36	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	**500,000**	500,000
	(2189.3)	(3053.3)	(2197.9)	(2245.2)	(2377.7)	(2534.2)	(2327.0)	(2952.3)	(2882.3)	**(2163.7)**	(2625.7)
F37	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	**500,000**
	(2351.2)	(2546.8)	(2402.3)	(2461.2)	(2422.1)	(2466.9)	(2376.7)	(2518.7)	(2517.4)	(2738.7)	**(2319.7)**
F38	**500,000**	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
	(6247.1)	(6923.8)	(7866.7)	(18,603.2)	(9530.8)	(980,874.2)	(7612.9)	(40,795.7)	(66,036.9)	(7848.6)	(730,084.1)
NSP[a]	25	0	0	0	1	1	0	0	0	4	7

[a]NSP: total number of superior performances of an algorithm comparing to the other algorithms

should be combined with an efficient constraint handling method to validate the applicability of this algorithm for real-world problems. Additionally, a study can be conducted to investigate the impact of different population sizes on the performance of this algorithm.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest

Human and animal rights This article does not contain any studies with human participants performed by any of the authors.

Appendix A

The precise sizes of gray areas in Fig. 4 are as follows: the large areas of X, V and Y type cover 26% of the hues; the small areas of i, L, I and Y type cover 5% of the hues; the large area of L type covers 22%; the area of T type covers 50%; The angle between the bisectors of the two areas of I, X and Y type is 180 , and for L type it is 90° (Cohen-Or et al. 2006).

Appendix B

See Tables 8, 9, B.3.

Table 8 Constant parameters for Hartman-4 function

i	$a_{ij}, j = 1,2,3$			c_i	$p_{ij}, j = 1,2,3$		
1	3	10	30	1	0.3689	0.117	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.747
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828

Table 9 Constant parameters for Hartman-6 function

i	$a_{ij}, j = 1,2,\ldots,6$						c_i	$a_{ij}, j = 1,2,\ldots,6$					
1	10	3	17	3.5	1.7	8	1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10	17	0.1	8	14	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3	3.5	1.7	10	17	8	3	0.2348	0.1451	0.3522	0.2883	0.3047	0.665
4	17	8	0.05	10	0.1	14	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

Table 10 Constant parameters for Kowalik function

i	1	2	3	4	5	6	7	8	9	10	11
a_i	0.1957	0.1947	0.1735	0.16	0.0844	0.0627	0.0456	0.0342	0.0323	0.0235	0.0246
b_i	0.25	0.5	1	2	4	6	8	10	12	14	16

References

Awad NH et al (2016) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization

Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv 35(3):268–308

Cheng S, Shi Y (2011) Diversity control in particle swarm optimization. In: 2011 IEEE symposium on swarm intelligence (SIS). IEEE

Cochrane S (2014) The Munsell color system: a scientific compromise from the world of art. Stud Hist Philos Sci Part A 47:26–41

Cohen-Or D et al (2006) Color harmonization. ACM Trans Graph 25(3):624–630

Das A (2015) Guide to signals and patterns in image processing. Springer, Berlin

Derrac J et al (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1(1):3–18

Dorigo M, Stützle T (2004) Ant colony optimization. MIT Press, Cambridge

Fogel LJ (1999) Intelligence through simulated evolution: forty years of evolutionary programming. Wiley, New York, p 162

Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845

Gandomi AH et al (2013) Metaheuristic applications in structures and infrastructures, 1st edn. Elsevier, Amsterdam

Geem Z, Kim J, Loganathan GV (2001) A new Heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13(5):533–549

Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, AnnAnbor

Jamil M, Yang X-S (2013) A literature survey of benchmark functions for global optimisation problems. Int J Math Model Numer Optim 4(2):150–194

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE international conference on neural networks, proceedings. IEEE, Perth, WA, Australia

Khachaturyan A, Semenovsovskaya S, Vainshtein B (1981) The thermodynamic approach to the structure analysis of crystals. Acta Crystallogr Sect A 37(5):742–754

Matsuda Y (1995) Color design. Asakura Shoten 2(4):10

Rao RV, Savsani VJ, Balic J (2012a) Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Eng Optim 44(12):1447–1462

Rao RV, Savsani VJ, Vakharia DP (2012b) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248

Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

Talbi E-G (2009) Metaheuristics: from design to implementation. Wiley, New York, p 593

Tokumaru M, Muranaka N, Imanishi S (2002) Color design support system considering color harmony. In: Proceedings of the 2002 ieee international conference on fuzzy systems, 2002. FUZZ-IEEE'02. IEEE

Wang G, Guo L (2013) A novel hybrid bat algorithm with harmony search for global numerical optimization. J Appl Math 2013:21

Wang GG et al (2014) Chaotic krill herd algorithm. Inf Sci 274(1):17–34

Westland S et al (2007) Colour harmony. Colour Des Creat 1(1):1–15

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Xhafa F, Abraham A (2008) Metaheuristics for scheduling in industrial and manufacturing applications, vol 128, 1st edn. Springer, Berlin

Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: González J et al (eds) Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, pp 65–74

Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: World congress on nature & biologically inspired computing, 2009. NaBIC 2009. IEEE