Journal of Computers Vol. 31 No. 6, 2020, pp. 116-126
doi:10.3966/199115992020123106010

A Particle Swarm Optimization Algorithm Based on Multi-Subgroup Harmony Search

Qi-Wen Zhang, Fang-Fang Zhang*

School of Computer and Communication, Lanzhou University of Technology University,
Lanzhou 730050, China
tdyy2010@126.com, 840778814@qq.com

Received 19 September 2019; Revised 5 January 2020; Accepted 1 March 2020

Abstract. In order to address the problem that the harmony search and particle swarm optimization algorithms are easy to fall into local optimum when solving high-dimensional complex problems, a particle swarm optimization algorithm based on multi-subgroup harmony search (i.e. PSO-LHS) is proposed. With respect to the PSO-LHS, we introduce the Levy flight and give the parameter adaptive adjustment method in the harmony search algorithm. By the aid of a hierarchical search strategy for enhancing the global search ability of PSO-LHS, the bottom layer is composed of a series of sub-populations with the Levy flight harmony search algorithm. Moreover, the upper layer consists of the optimal individuals of each sub-population to form an elite group, and the particle swarm algorithm is used for improving the accuracy of local search. In the process of searching, the sub-population exchanges information with the upper elite to improve the diversity and search efficiency of the population. The experimental results demonstrate that our PSO-LHS algorithm has better efficiency and global convergence compared with HS, PSO and improved HS algorithm.

Keywords: harmony search, hierarchical optimization, multi-subgroup, particle swarm optimization

1 Introduction

Optimization problems generally exist in many fields such as scientific research, engineering technology, and economic management. In view of the disadvantages of traditional optimization methods such as low computational efficiency and easily fall into local optimization, in recent years, the intelligent optimization algorithm represented by harmony search and particle swarm optimization algorithms have been rapidly developed and widely used [1]. The intelligent optimization algorithm is easy to realize and has a wide range of applicability. However, the intelligent optimization algorithm is easy to fall into the local optimal solution in the optimization process, because it does not rely on the analysis of the mathematical model of the objective function [2]. In order to overcome the shortcomings of intelligent optimization algorithms, various improved methods have been studied to enhance their search performance, and can be roughly categorized into two types. One is to modify the algorithm itself, including improvement of algorithm parameters [3], improved incremental structure, strengthen local search [4], improve mutation operator to enhance population diversity [5] and so on. The other is fusion algorithm. For example, Chegini et al. [6] combined the PSO algorithm, the sine and cosine algorithm, and the position update equation in the Levy flight method to improve the exploration ability of the PSO algorithm; Assad et al. [7] proposed a new algorithm based on a mixture of harmony search and simulated annealing and inherited its advantages through complementary algorithms; Ouyang et al. [8] introduced a new global speed update strategy and proposed a hybrid harmony search particle swarm optimization algorithm with global dimension selection, which accelerates the convergence speed of the HS algorithm.

* Corresponding Author

Generally, both HS and PSO algorithms have well performance in the face of small-scale optimization problems. However, as the actual optimization problem becomes more and more complex, their optimization efficiency and the solution quality appear to be "inadequate." In view of the almost complementary advantages of HS and PSO, many scholars have combined these two algorithms to learn from each other's strengths and weaknesses, and have developed algorithms with the better performance [9]. At present, there are two main ways of combining HS with PSO.

(1) Introduce the basic operation of the harmony search algorithm into the PSO algorithm to improve the performance of the PSO algorithm [10], or the basic operation of the PSO algorithm is introduced into HS calculation [12] to improve the performance of HS algorithm.

(2) Mix the two algorithms with equal status. For example, Zhao et al. [15] adopted the evolutionary method of group multiples, combining harmony search with particle swarm optimization; Chen et al. [16] introduced a dynamic crossover operation and proposed a dynamic crossover particle swarm optimization algorithm based on harmony search combined with the random search capability of HS.

Based on the above review of improved intelligent optimization algorithms and the development trend of HS and PSO hybrid optimization algorithms, we propose a new PSO-LHS algorithm that combines with the advantages of these two improved methods of optimization algorithms. Not only the parameters of the algorithm are modified, but HS and PSO are combined in a new way to make overall improvements. The technical achievements main contributions of this article are as follows:

(1) In order to improve the empirical value of traditional HS algorithm parameters, levy flight is introduced to improve HS parameters, and a harmony search algorithm based on levy flight is proposed.

(2) In order to enhance the diversity of the population, the levy-based flight HS algorithm is divided into multiple sub-populations; the hierarchical search strategy is used, and combined with the particle swarm optimization algorithm, the local search and global search of the effective coordination algorithm are effectively coordinated.

(3) The fusion mode of PSO-LHS algorithm can not only combine PSO and HS but also theoretically combine any two or more evolutionary algorithms, which can derive many hybrid evolutionary optimization algorithms.

(4) Experimental results simulated on benchmark functions demonstrate the effectiveness and superiority of the PSO-LHS algorithm.

The rest of this paper is organized as follows. Section 2 reviews the basic theories of HS and PSO optimization, the advantages and shortcomings of the existing methods are analyzed in depth as well. Then the detail of the improved algorithm is shown in section 3. Finally, experimental results and conclusions are made in section 4 and section 5, respectively.

2 Related Work

Harmony search (HS) is a new type of intelligent optimization algorithm proposed by Geem in 2001 [17]. It simulates the principle and process of music tuning. In the HS algorithm, a solution component is randomly generated in the harmony memory HM if a random in the range of (0,1) is less than the memory access probability HMCR, otherwise, a variable is randomly generated in the search space. If the generated solution is a solution variable selected from HM, fine-tuning is performed using the bandwidth BW. The update process of the traditional harmony search algorithm is defined as follows:

$$x_i^{new} = \begin{cases} x_i^{new} \pm r * BW, & \text{r<PAR} \\ x_i^{new}, & \text{else} \end{cases}, r \in (0,1) . \quad \textbf{(1)}$$

Particle swarm optimization (PSO) is a new swarm intelligence optimization algorithm proposed by Kennedy and Eberhart in 1995 [18]. In the PSO algorithm, the position of each particle represents a candidate solution in the search space, and the particle has two characteristics of position and velocity. The update equations for standard particle swarm optimization speed and position are defined as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_{ib}(t) - x_i(t)) + c_2 r_2 (p_{gb}(t) - x_i(t)) . \quad \textbf{(2)}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) . \quad \textbf{(3)}$$

Where v_i and x_i represent the velocity vector and position vector of the *i-th* particle, respectively; ω is the inertia weight; p_{ib} represents the individual extreme value; c_1 and c_2 are acceleration constants; r_1 and r_2 are two random numbers uniformly distributed in the range of (0, 1).

Harmony search and particle swarm optimization are heuristic intelligent optimization algorithms. They have their own characteristics and advantages but also have some shortcomings. The harmony search algorithm owns a strong global search ability, but has the poor local search ability, slow convergence rate and low search efficiency in the late evolution. By contrast, the particle swarm optimization has a fast convergence speed, but its global search capability is poor, since the fast convergence speed results in the quick decrease of population diversity decreases and the issue of premature convergence.

According to the introduction, there are two main ways to combine HS and PSO. One way is to use one algorithm to improve the other [10]. To some extent, this method improves the convergence performance of the algorithm, but it is not enough to change the essential characteristics of the algorithm optimization. Another way is to mix the two algorithms with equal status [15]. This hybrid approach faces the same problem, that is, the two algorithms are in the same position, the division of labor between the two is not clear, and their respective advantages have not been fully utilized. Therefore, there is still a lot of space worth digging in the fusion of harmony search and particle swarm optimization. This paper proposes a new particle swarm optimization algorithm for multi-subgroup harmony search. Starting from the organizational structure of the individual, the PSO-LHS algorithm adopts the hierarchical structure to separate the global search and the local search, which not only speeds up the convergence speed, but also effectively balances the local search ability of the global search.

3 Harmony Search Algorithm Based on Levy Flight (LHS)

In the traditional HS algorithm, the harmony memory considering rate (HMCR), pitch adjustment rate (PAR), and bandwidth (BW) are fixed based on experience, which affects the search performance of the algorithm. In order to overcome the shortcomings resulting from the constant parameter, the levy flight is introduced to improve the performance of the HMCR and BW algorithms.

In the traditional harmony search algorithm, HMCR, PAR and BW are selected according to experience. But when the algorithm starts to run, the smaller HMCR, PAR and larger BW are beneficial to the global search of this algorithm. When the algorithm is iterated multiple times, the larger HMCR, PA, and smaller BW make it easier for the algorithm to find the optimal solution in the local range. According to the above considerations, the dynamic adaptive adjustment of HMCR, PAR and BW are carried out. The expressions are defined as follows:

$$HMCR_t = HMCR_{\min} + \alpha(\frac{HMCR_{\max} - HMCR_{\min}}{T_{\max}} t) + (1-\alpha)levy(s). \quad \textbf{(4)}$$

Where t is the current iteration number; $HMCR_{\max}$ is the maximum of harmony memory considering rate; $HMCR_{\min}$ is the minimum of harmony memory considering rate; $T_{\max}$ is the maximum number of iterations; $Levy(s)$ is Levy Flight Random Walk Strategy, s is the step size, i.e. $levy(s) = \frac{\mu}{|\nu|^{1/\beta}}$, where μ and ν are drawn from normal distributions. That is, where $\mu \sim N(0,\sigma_\mu^2), \nu \sim N(0,1)$, $\sigma_\mu^2 = \{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[((1+\beta)/2]\beta 2^{(\beta-1)/2}}\}^{\frac{1}{\beta}}$, $\beta = \frac{3}{2}$, is the Gamma function; α represents a normal distribution density function that satisfies the variance of 1, i.e. $\alpha \sim N(\mu,1)$, where the expected value μ is determined by the number of iterations of the algorithm.

$$PAR_t = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{\pi/2} ar\tan(t) \quad \textbf{(5)}$$

$$BW = a_0 \oplus Levy(s). \quad \textbf{(6)}$$

Where $PAR_{\max}$ represents the maximum of pitch adjustment rate and $PAR_{\min}$ represents the minimum of pitch adjustment rate; $\oplus$ is point multiplication; a_0 represents step size control, the value is calculated by using the square root of the expected population mean of HS algorithm, which satisfies the requirement $a_0 = \sqrt{E(\bar{x})}$.

4 Particle Swarm Optimization Algorithm Based on Multi-subgroup Harmony Search

Based on a full analysis of the harmony search and particle swarm optimization search mechanism and information flow mechanism, the two algorithms are combined in a hierarchical structure, so that the new algorithm can not only jump out of the local optimum but also coordinate well local search and global search.

The organization of individuals in the PSO-LHS algorithm is shown in Fig. 1. The bottom layer is composed of n sub-populations with the Levy flight harmony search algorithm ($LHS_1, LHS_2, ..., LHS_n$), which is mainly responsible for global search; the optimal individual of each sub-group of the bottom layer constitutes the upper elite group, which is evolved by PSO algorithm. It is mainly responsible for the local search of the elite group and accelerates the convergence speed of the algorithm. The PSO-LHS algorithm can be divided into the following two stages.

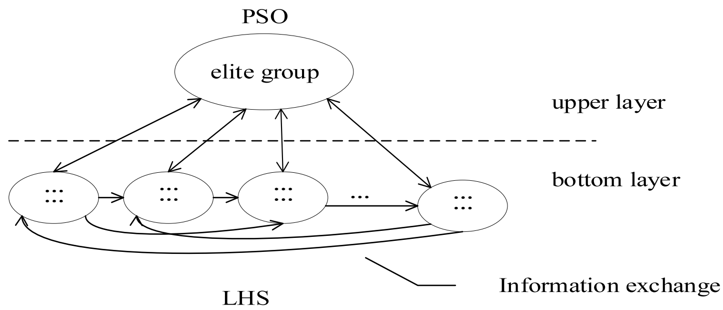

Fig. 1. PSO-LHS combination mode

Phase 1: Use the LHS algorithm for global search while providing initial values for the upper PSO algorithm

Firstly, randomly initialize n-population, denoted as $LHS_1, LHS_2, ..., LHS_n$, and each sub-population runs the LHS algorithm independently. Secondly, after iterating a certain number of algebras, information interaction between different sub-populations is performed, and one individual is randomly selected from each sub-population and communicates with randomly generated individuals in the other two sub-populations. Finally, optimal individuals of each sub-population are selected to form the upper elite group as the initial value of the upper PSO algorithm.

Phase 2: Co-evolution of upper PSO algorithm and LHS sub-population

In the optimization process, two algorithms are iterated alternately and judged by whether the stopping criterion is satisfied or not. That is, the optimal individual of each sub-group at the bottom layer constitutes the upper elite group. After several algebras carried out by the PSO algorithm, the stopping criterion is judged. If the stopping criterion is satisfied, the result is output and the algorithm stops; otherwise, the worst individual in each sub-population is selected to communicate with the best individual in the elite group, and the worse individual is accelerated to the optimal individual. At this point, the entire algorithm ends. The n sub-groups restart the operation of the LHS algorithm and continue to cycle until the stop criterion is satisfied.

The evolutionary process design of the PSO-LHS algorithm is shown in Fig. 2.

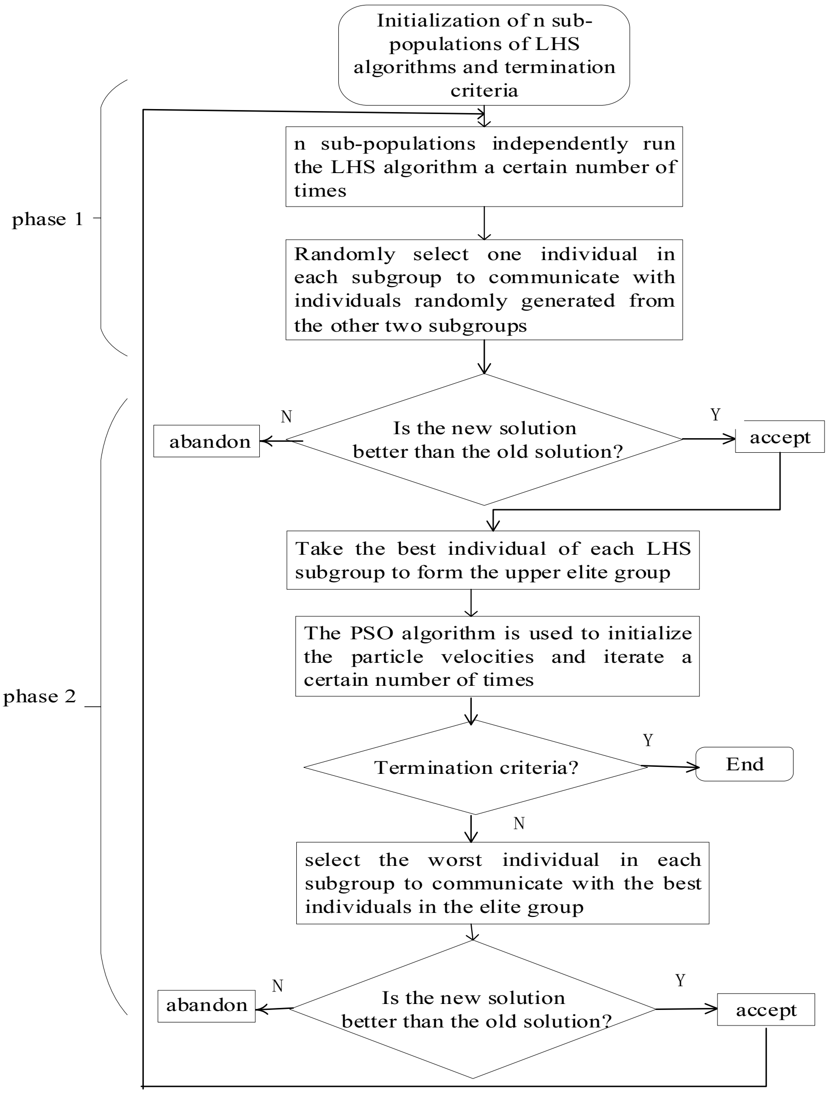

Fig. 2. PSO-LHS flow chart

4.1 Information Exchange between the Sub-populations at the Bottom

In order to strengthen the information interaction between different sub-populations in the evolution process and enhance the diversity of the population, one individual in each population is randomly exchanged with the individuals in the other two sub-populations randomly generated. The specific steps are defined as follows:

Step 1: In the *t-th* iteration, randomly select an individual x_{LHS_i} from the sub-group LHS_i;

Step 2: In other two sub-groups $LHS_{i\%n+1}$, $LHS_{i\%n+2}$, randomly generate an individual $x_{LHS_{i\%n+1}}$, $x_{LHS_{i\%n+2}}$;

Step 3: Update x_{LHS_i} as follows:

$$\begin{cases} x_{LHS_i}^{new} = x_{LHS_i} + r(x_{LHS_{i\%n+1}} - x_{LHS_{i\%n+2}}) \ , \ \text{if} f(x_{LHS_{i\%n+1}}) \geq f(x_{LHS_{i\%n+2}}) \\ x_{LHS_i}^{new} = x_{LHS_i} + r(x_{LHS_{i\%n+2}} - x_{LHS_{i\%n+1}}) \ , \text{if} f(x_{LHS_{i\%n+1}}) < f(x_{LHS_{i\%n+2}}) \end{cases} r \in (0,1). \quad \textbf{(7)}$$

Step 4: If $f(x_{LHS_i}^{new})$ is better than $f(x_{LHS_i})$, accept $x_{LHS_i}^{new}$, otherwise x_{LHS_i} will not change.

4.2 Information Exchange between the Sub-populations at the Bottom

In order to speed up the searching speed of the algorithm, the worst individuals in each sub-population are selected to communicate with the best individuals in the elite population, and the poorer individuals are accelerated to approach the best individuals. The specific steps are defined as follows:

Step 1: For the *t-th* iteration, find the best individual *Gbest* in the elite group;

Step 2: Find the worst individual $x_{worst_i (i=1,2,...,n)}$ in each sub-population;

Step 3: Make the worst individual learn from the best individuals in the elite group, as follows:

$$x_{\text{new}_i} = x_{worst_i} + r(\text{Gbest}^{\text{t}} - x_{worst_i}) \ \ , r \in (0,1). \quad \textbf{(8)}$$

Step 4: If $f(x_{\text{new}_i})$ is better than $f(x_{worst_i})$, accept x_{new_i}, otherwise x_{worst_i} will not change.

4.3 Global Convergence of PSO-LHS Algorithm

In reference [19], Solis and Wets prove that the stochastic optimization algorithm converges to the global optimal solution with probability 1. On this basis, this paper proves the global convergence of the PSO-LHS algorithm.

Hypothesis. 1: for the optimization problem $< x, f >$, if $f(D(x,\xi)) \le f(x)$, and $\xi \in \text{X}$, then $f(D(x,\xi)) \le \min\{f(x), f(\xi)\}$. Where X is the feasible solution space, f is the fitness function, and ξ is the solution that *D* the algorithm has searched in this iteration process.

Hypothesis. 2: for any Borel subset A in X, if its Lebesgue measure $\upsilon(A) > 0$, then there is $\prod_{\text{t}=1}^{\infty}(1\text{-}\mu_{\text{t}}(A)) = 0$ among them, $\mu_{\text{t}}(A)$ is the probability obtained by measuring μ_{t}.

Lemma assumes that the objective function f solved by the algorithm is a measurable function, and its solution space X is a measurable subset and satisfies hypothesis 1 and hypothesis 2. Let $\{\text{P}_{g,t}\}_{t=1}^{+\infty}$ be the sequence of solutions generated by the algorithm, then $\lim_{t\to\infty}[P_{g,t} \in R_g] = 1$ among them, $P[P_{g,t} \in R_g]$ is the probability of the solution $P_{g,t} \in R_g$ generated by the *t-th* iteration of the algorithm, and R_g is the set of global optimal points.

Theorem. LHS-PSO algorithm converges to the global optimal solution with probability 1.

Proof. Let the iterative function D of the LHS-PSO algorithm is:

$$D(P_{g,t}, x_{i,t}) = \begin{cases} P_{g,t} & , f(P_{g,t}) \le f(x_{i,t}) \\ x_{i,t} & , f(P_{g,t}) > f(x_{i,t}) \end{cases}. \quad \textbf{(9)}$$

Where *t* is the number of iterations, $P_{g,t}$ is the global best positions for the *t-th* iteration update of PSO and $x_{i,t}$ is the global best positions for the *t-th* iteration update of LHS.

Prove that it satisfies hypothesis 1:

$$D(P_{g,t}, x_{i,t}) \notin R_g \Rightarrow D(P_{g,t}, x_{i,t}) \notin R_g, \forall l < t. \quad \textbf{(10)}$$

$$P(D(P_{g,t}, x_{i,t}) \in X \setminus R_g) \le \prod_{t=0}^{t-1}(1 - \mu_l(R_g)). \quad \textbf{(11)}$$

$$P(D(P_{g,t}, x_{i,t}) \in X) \text{=1-} P(D(P_{g,t}, x_{i,t}) \in X \setminus \text{R}_g) \ge 1 - \prod_{t=0}^{t-1}(1 - \mu_l(R_g)). \quad \textbf{(12)}$$

It can be proved that it satisfies hypothesis 1.

Prove that it satisfies hypothesis 2: Let Y_t for the LHS algorithm in the *t-th* iteration to search solution. If the LHS algorithm is run alone, Y_t converges to the global optimal solution R with 1. In the LHS-PSO algorithm, for Y_t satisfying $f(Y_t) > f(P_{g,t})$, the next generation is $P_{g,t}$, which is easy to prove: $\lim_{t\to\infty} P\{Y_t \in R\} = 1$.

$$1 \ge \lim_{t\to\infty} P\{Y_t \in R\} \ge 1\text{-}\prod_{t=1}^{\infty}(1-\mu_t(A)) = 1. \quad \textbf{(13)}$$

When $f(Y_t) < f(P_{g,t})$, the union of the particle swarm sample space must contain the solution space X, i.e. $X \subseteq \bigcup_{i=1}^{n} M_{i,t}$, where $M_{i,t}$ is the support set of the particle i sample space in the t-th generation. For particles i_0, $M_{i_0,t} = X$ satisfying $x_{i_0,t} = P_t = P_g$, for other particles i satisfying:

$$M_{i,t} = x_i(t-1) + \omega(x_i(t-1) - x_i(t-2)) + \varnothing_1(P_{i,t-1} - x_i(t-1)) + \varnothing_2(P_{g,t-1} - x_i(t-1)). \quad \textbf{(14)}$$

Where $0 \le \varnothing_1 \le c_1$, $0 \le \varnothing_2 \le c_2$. . When formula (11) is established, obviously there is $\upsilon(M_{i,t} \cap X) < \upsilon(X)$. $\max(c_1 | P_{i,j,t-1} - x_{i,j,t-1} |, c_2 | P_{g,j,t-1} - x_{i,j,t-1} |) < 0.5 * diam_j(X)$, Where $diam_j(X)$ represents the length of X in the j-th dimension component. Because $x_i \to c_1 P_i + c_2 P_g / c_2 + c_2$ $\lim_{t\to\infty} M_{i,t} = 0$. As the number of iterations t increases, $\upsilon(M_{i,t} \cap X) < \upsilon(X)$. But because of $M_{i_0,t} = X$, so $\bigcup_{i=1}^{n} M_{i,t} = X$, The Borel subset A=$M_{i,t}$, which defines X, has $\upsilon(A) = \sum_{i=1}^{n} \mu_{i,t}(A) = 1$. Thus it can be proved that hypothesis 2 is satisfied.

According to the lemma, the PSO-LHS algorithm converges to the global optimal solution with probability 1, so the theorem holds.

5 Experimental Results and Analysis

In order to verify the advantages of the proposed algorithm in global search ability, search speed and stability, the harmony search algorithm (HS), particle swarm optimization algorithm (PSO) and the harmony search algorithm with Levy flight (LHS) are selected for experimental comparison. In this paper, 3 unimodal functions, 7 simple multimodal functions and 10 hybrid functions in CEC2017 are introduced to be tested in the case of high dimensions D=50 and 100, respectively. The search scope of the function is $[100, -100]^D$, and the experimental results are given in Table 2.

Table 1 lists the parameters of each algorithm, among which HS, PSO, and LHS adopt a single population, the number of individuals is set to 60, the number of iterations $T_{\max}$=1000, $\alpha \sim N(25.5, 1)$. The PSO-LHS algorithm proposed in this paper uses 6 sub-groups, 10 individuals per sub-population, and the optimal individuals of each sub-population constitute the upper elite group. The iteration stops for 50 rounds, and the LHS and PSO run for 20 generations in each round. During optimization, the algorithm randomly initializes individuals in the search space. In order to exclude randomness, the experiment is run 25 times independently, and the optimal value, average value, worst value, and variance are recorded as comparative evaluation indexes. The experiment is simulated based on the python3.7+pycharm2018 platform.

Table 1. Algorithm parameter settings

Algorithm	$\omega_{\max}$	$\omega_{\min}$	c_1	c_2	BW	PAR	$HMCR$	$PAR_{\max}$	$PAR_{\min}$	$HMCR_{\max}$	$HMCR_{\min}$	pop num	pop size	$T_{\max}$
PSO	0.9	0.4	1.4962	1.4962	—	—	—	—	—	—	—	1	60	1000
HS	—	—	—	—	0.01	0.99	0.2	—	—	—	—	1	60	1000
LHS	—	—	—	—	—	—	—	0.99	0.01	0.99	0.1	1	60	1000
PSO-LHS	0.9	0.4	1.4962	1.4962	—	—	—	0.99	0.01	0.99	0.1	6	10	50/20

In Table 2, the optimal value and the average value are used as evaluation items. The smaller the optimal value is, the better the optimization effect will be. If the optimal value is the same, compare the mean value. The smaller the mean value is, the better the optimization effect will be. It can be seen from the situations marked in bold in Table 2 that PSO-LHS has a better optimization effect for unimodal functions, simple multimodal functions, and hybrid functions. Compared with the 50 and 100 dimensions, the results of LHS algorithm in the $F_1 \sim F_4$, F_{11}, F_{14}, F_{16}, F_{18} and F_{20} functions are better than those in the 50 dimension, and the optimization results of other functions in the 50 dimension are better than those in the 100 dimension. Therefore, in the practical application, we should choose the appropriate dimensions for different optimization problems, so as to obtain better optimization results. In general, the PSO-LHS algorithm has strong global search ability. The optimization performance of a single peak, multi-peak and mixed function is better than that of a single algorithm. It is not easy to fall into a local optimum, has a faster optimization speed, and also obtains a better solution than a single algorithm.

Table 2. Comparison results of HS, PSO, LHS and PSO-LHS algorithms

F	D	HS				PSO				LHS				PSO-LHS			
		Best	Worst	Mean	Var	Best	Worst	Mean	Var	Best	Worst	Mean	Var	Best	Worst	Mean	Var
F_1	50	8.73E+01	9.56E+01	9.39E+01	1.75E+02	7.19E+00	2.13E+01	1.39E+01	3.01E+01	9.93E+00	2.87E+01	1.41E+01	2.42E+01	**5.14E+00**	**1.32E+02**	**2.10E+01**	**1.17E+02**
	100	9.27E+01	9.53E+01	9.53E+01	1.93E+02	1.68E+01	4.59E+01	1.79E+01	2.21E+01	3.82E+01	8.57E+01	6.92E+01	8.29E+01	**1.18E+01**	**4.59E+01**	**1.92E+01**	**1.94E+01**
F_2	50	1.07E+02	1.24E+02	1.06E+02	2.10E+02	1.11E+02	1.29E+02	1.06E+02	2.21E+02	7.19E+00	5.92E+01	1.27E+01	1.17E+02	**6.83E+00**	**2.10E+02**	**1.50E+01**	**1.04E+02**
	100	9.22E+01	1.07E+02	9.22E+01	9.22E+01	1.05E+02	1.28E+02	1.04E+02	1.24E+02	1.48E+01	1.40E+02	1.41E+02	1.43E+02	**1.48E+00**	**1.42E+02**	**1.41E+02**	**1.40E+02**
F_3	50	7.65E+00	4.39E+01	2.42E+01	1.33E+01	1.89E+01	1.15E+01	9.60E+01	1.34E+02	3.26E+00	2.01E+01	7.48E+00	1.66E+01	**5.26E+00**	**4.72E+01**	**3.09E+01**	**2.52E+01**
	100	3.75E+00	2.38E+01	9.65E+00	1.47E+01	2.83E+01	8.91E+01	8.44E+01	9.55E+01	3.62E+00	1.65E+01	1.95E+01	1.20E+01	**1.91E+00**	**4.43E+01**	**3.11E+01**	**2.76E+01**
F_4	50	1.43E+01	2.37E+01	1.44E+01	2.19E+01	4.89E+01	4.90E+01	4.89E+01	2.93E+01	9.73E+00	2.90E+01	1.44E+01	2.35E+01	**6.21E+00**	**4.91E+01**	**4.91E+01**	**4.89E+01**
	100	7.98E+00	1.80E+01	2.28E+01	1.12E+01	9.89E+01	9.90E+01	9.89E+01	9.76E+01	1.03E+01	2.79E+01	1.62E+01	1.62E+01	**1.61E+00**	**2.79E+01**	**2.69E+01**	**2.79E+01**
F_5	50	6.19E+00	1.86E+01	2.28E+01	1.05E+01	5.62E+00	1.56E+01	1.31E+01	1.02E+01	3.33E+00	2.54E+01	1.15E+01	2.30E+01	**3.09E+00**	**4.55E+01**	**5.01E+00**	**5.23E+00**
	100	1.59E+01	2.71E+01	2.45E+01	1.69E+01	5.13E+00	1.18E+01	8.28E+00	1.02E+01	8.70E+00	8.73E+00	6.83E+00	1.42E+01	**3.36E+00**	**4.13E+00**	**3.52E+00**	**3.89E+00**
F_6	50	7.68E-01	2.60E+01	2.13E+01	1.50E+01	7.08E-01	2.10E-01	7.08E-01	7.08E-01	7.08E-01	1.02E+01	6.81E+00	6.60E+00	**6.26E-01**	**2.13E+01**	**2.06E+01**	**1.83E+01**
	100	5.62E-01	5.13E+01	2.38E+01	2.96E+01	7.08E-01	2.98E-01	7.08E-01	7.08E-01	3.00E+00	2.30E+01	1.45E+01	3.32E+01	**4.92E-01**	**4.14E+01**	**3.91E+01**	**4.10E+01**
F_7	50	4.68E-02	5.09E+05	1.70E-01	8.67E+03	6.96E+00	7.54E+05	2.23E+01	7.54E+05	3.87E-03	7.12E+03	9.51E+01	2.50E+02	**4.81E-04**	**9.65E+01**	**8.76E+00**	**8.15E+00**
	100	2.78E+00	1.17E+06	5.61E+00	1.68E+04	7.80E+00	1.38E+06	5.37E+01	7.21E+00	1.68E+00	3.38E+04	3.71E+02	6.53E+02	**1.56E+00**	**8.72E+02**	**5.37E+00**	**6.02E+00**
F_8	50	8.63E+01	9.70E+01	9.74E+01	8.84E+01	6.57E+01	7.89E+01	6.78E+01	7.88E+01	8.17E+01	9.12E+01	8.26E+01	9.15E+01	**4.65E+01**	**6.48E+01**	**6.35E+01**	**5.64E+01**
	100	8.53E+01	9.34E+01	9.63E+01	8.53E+01	6.24E+01	6.06E+01	6.69E+01	6.34E+01	7.41E+01	9.05E+01	8.00E+01	9.00E+01	**5.42E+01**	**6.92E+01**	**6.62E+01**	**6.83E+01**
F_9	50	5.04E+00	1.09E+08	6.12E+00	8.59E+04	4.62E+00	8.63E+07	5.14E+00	5.49E+00	5.33E+00	5.26E+05	8.80E+02	2.14E+03	**5.07E+00**	**5.85E+00**	**5.34E+00**	**5.45E+00**
	100	7.74E+01	1.46E+08	2.99E+02	1.53E+05	8.35E+00	1.61E+08	9.77E+00	2.07E+01	9.91E+00	1.95E+06	2.14E+03	4.40E+03	**8.13E+00**	**1.22E+01**	**1.03E+01**	**8.72E+00**
F_{10}	50	2.09E+04	4.00E+04	2.09E+04	2.16E+04	2.09E+04	3.36E+04	2.10E+04	2.10E+04	1.95E+01	2.10E+04	2.10E+04	2.09E+04	**1.17E-02**	**2.09E+04**	**2.09E+04**	**2.09E+04**
	100	4.19E+04	4.25E+04	4.19E+04	4.18E+04	4.19E+04	4.25E+04	4.19E+04	4.19E+04	5.82E+00	4.19E+04	4.19E+04	4.19E+04	**2.89E-02**	**4.19E+04**	**4.19E+04**	**4.19E+04**
F_{11}	50	8.97E+01	1.19E+02	8.91E+01	1.75E+02	2.19E+01	9.23E+01	9.49E+01	9.52E+01	7.13E+00	2.51E+01	7.57E+00	2.10E+01	**6.59E+00**	**3.32E+01**	**1.53E+01**	**1.49E+01**
	100	9.82E+01	1.08E+02	9.55E+01	1.93E+01	2.56E+01	8.80E+01	7.69E+01	1.04E+02	1.15E+01	1.88E+01	1.31E+01	1.69E+01	**1.05E+01**	**2.63E+01**	**1.74E+01**	**1.99E+01**
F_{12}	50	7.21E+01	9.51E+01	7.19E+01	1.73E+01	3.92E+01	1.14E+02	9.32E+01	9.66E+01	9.16E+00	2.23E+01	1.05E+01	2.37E+01	**5.99E-01**	**3.47E+01**	**3.05E+00**	**4.05E+01**
	100	9.57E+01	9.68E+01	9.47E+01	1.93E+01	3.53E+01	9.57E+01	8.67E+01	9.60E+01	1.35E+01	3.56E+01	3.07E+01	2.20E+01	**7.04E+00**	**1.29E+01**	**2.55E+01**	**8.27E+00**

Table 2. Comparison results of HS, PSO, LHS and PSO-LHS algorithms (continue)

F	D	HS				PSO				LHS				PSO-LHS			
		Best	Worst	Mean	Var	Best	Worst	Mean	Var	Best	Worst	Mean	Var	Best	Worst	Mean	Var
F_{13}	50	2.07E+01	2.19E+01	2.09E+01	1.17E-02	9.00E-03	2.19E+01	2.17E+01	2.12E+01	3.41E+00	2.11E+01	1.95E+01	1.69E+01	**8.96E-03**	**2.16E+01**	**2.17E+01**	**2.13E+01**
	100	2.09E+01	2.19E+01	2.10E+01	3.80E-03	4.56E-03	2.18E+01	2.17E+01	2.13E+01	9.24E-02	2.12E+01	2.04E+01	8.48E+00	**2.94E-03**	**2.16E+01**	**2.14E+01**	**2.13E+01**
F_{14}	50	4.63E+03	5.48E+03	4.85E+03	3.48E+04	4.46E+03	5.56E+03	5.29E+03	2.34E+04	4.40E+03	4.95E+03	4.85E+03	1.09E+04	**2.83E+03**	**5.11E+03**	**4.96E+03**	**1.07E+04**
	100	9.19E+00	9.65E+00	9.41E+00	9.24E+00	9.03E+00	9.87E+00	9.68E+00	1.10E+01	9.14E+00	1.74E+01	9.30E+00	9.38E+00	**8.07E+00**	**9.69E+00**	**8.56E+00**	**8.60E+00**
F_{15}	50	7.80E-01	8.91E+02	9.70E-01	3.20E+02	5.61E+00	7.27E+02	7.95E+00	1.30E+01	1.59E-01	1.01E+01	4.53E+00	9.24E+00	**1.24E-01**	**4.56E+01**	**2.67E+00**	**3.92E+00**
	100	1.17E+00	1.22E+03	1.46E+00	5.73E+02	2.08E+00	1.97E+03	2.22E+00	2.22E+00	8.86E-01	7.49E+01	1.53E+01	3.01E+01	**5.28E-01**	**2.10E+01**	**1.18E+00**	**1.24E+00**
F_{16}	50	1.06E-14	3.87E-03	3.87E-03	3.87E-03	8.24E-15	3.87E-03	3.87E-03	3.87E-03	3.04E-15	3.87E-03	3.87E-03	3.87E-03	**1.44E-15**	**3.87E-03**	**3.87E-03**	**3.87E-03**
	100	3.56E-18	9.82E-04	9.82E-04	9.82E-04	4.02E-18	9.82E-04	9.82E-04	9.82E-04	1.22E-18	9.82E-04	9.82E-04	9.82E-04	**1.09E-18**	**9.82E-04**	**9.82E-04**	**9.82E-04**
F_{17}	50	1.20E+00	6.53E+04	1.67E+00	2.24E+03	8.41E-01	7.86E+04	2.56E+00	3.18E+00	2.71E+00	4.52E+02	2.49E+01	6.24E+01	**6.46E-02**	**2.64E+00**	**1.94E+00**	**1.35E+00**
	100	1.73E+00	5.48E+04	2.33E+00	2.21E+03	1.07E+00	4.50E+04	2.98E+00	5.03E+00	2.47E+00	7.28E+02	3.20E+01	7.59E+01	**3.77E-01**	**2.83E+00**	**1.73E+00**	**1.45E+00**
F_{18}	50	1.93E+00	3.95E+08	5.35E+00	2.02E+05	5.00E-01	6.97E+08	5.00E-01	5.01E-01	8.50E-01	5.46E+06	2.82E+03	7.14E+03	**4.47E-03**	**5.96E-01**	**5.27E-01**	**5.04E-01**
	100	1.00E+02	7.16E+08	2.29E+02	3.87E+05	5.00E-01	9.03E+08	5.00E-01	5.00E-01	7.59E+01	2.32E+07	8.71E+03	1.69E+04	**1.29E-03**	**5.45E-01**	**5.11E-01**	**3.47E+02**
F_{19}	50	4.05E+00	3.60E+01	4.19E+00	2.24E+01	6.05E+00	3.83E+01	6.35E+00	6.51E+00	6.51E+00	1.23E+01	8.43E+00	8.02E+00	**1.65E-01**	**1.72E+02**	**1.67E+02**	**2.86E+01**
	100	1.15E+01	4.11E+01	1.22E+01	2.24E+01	8.86E+00	4.26E+01	9.13E+00	1.14E+01	9.02E+00	1.83E+01	1.13E+01	1.60E+01	**3.34E-01**	**3.63E+02**	**3.44E+02**	**3.62E+02**
F_{20}	50	1.14E+00	3.83E+02	2.38E+00	1.53E+02	9.67E+00	3.69E+02	2.66E+00	1.62E+01	6.82E-02	5.97E+00	2.50E+00	3.11E+00	**4.27E-04**	**3.34E+00**	**8.64E-04**	**2.37E-03**
	100	2.91E+00	1.58E+02	4.44E+00	1.39E+02	1.67E+00	1.70E+02	2.80E+00	1.81E+00	3.57E-01	8.54E+00	4.95E+00	4.39E+00	**1.84E-14**	**3.80E-07**	**3.17E-07**	**1.53E-07**

In addition to numerical comparison, we illustrate the operation of the algorithm in specific functions to get the convergence of each function more clearly. The data used in the figure is the average value of 25 independent optimizations of the algorithm. The abscissa is the evolutionary algebra and the ordinate is the optimal value. Fig. 3 and Fig. 4 show the optimization effect of each algorithm in different dimensions of the unimodal function F_1. Fig. 5 and Fig. 6 show the optimization effect of each algorithm in dimension 100 of the simple multimodal function F_9 and hybrid function F_{15}. From Figures 3 to 6, it can be clearly seen that PSO-LHS can converge with fewer iterations in different dimensions, and has fast optimization speed as well as high stability. Compared with LHS, the convergence of the LHS algorithm is better than the HS algorithm. Therefore, the use of levy flight has a good effect on the improvement of parameters, and combined with the PSO algorithm, the proposed PSO-LHS algorithm not only maintains the original fast optimization speed, but also has better stability in the later stages.

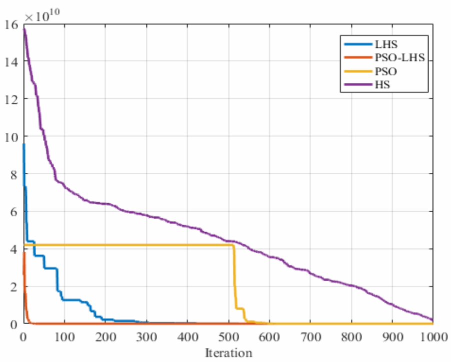

Fig. 3. F_1 (D=50, T_{max}=1000)

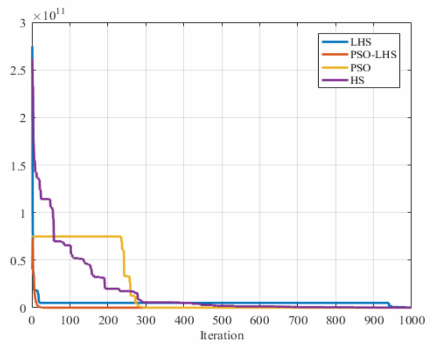

Fig. 4. F_1 (D=100, T_{max}=1000)

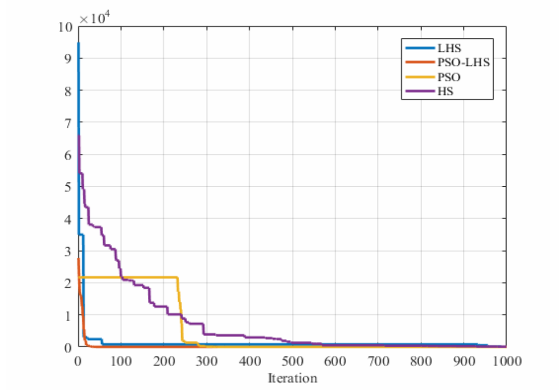

Fig. 5. F_9 (D=100, T_{max} =1000)

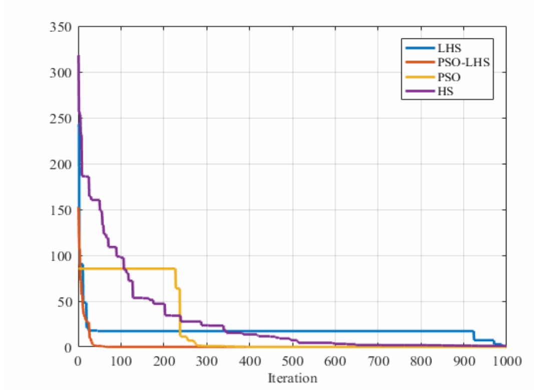

Fig. 6. F_{15} (D=100, T_{max} =1000)

6 Conclusions

This paper proposes a particle swarm optimization algorithm (PSO-LHS) based on multi-subgroup and harmony search. The multi-subgroup structure can better maintain the diversity of the population and is beneficial to the global search. Levy flight is introduced to perturb the parameters of the harmony search algorithm to get rid of the problem that falls into local optimum. Particle swarm optimization is used to search local elite groups to speed up convergence. The use of hierarchical structure effectively separates global search from local search, which can not only accelerate the search speed but also avoid the decline of population diversity caused by too fast convergence and weaken the global search ability. The experimental result shows that the PSO-LHS algorithm has superior performance in global search ability search speed, and stability. The future work is to apply the PSO-LHS algorithm to complex practical problems.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (61663023). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] U. Boryczka, K. Szwarc, The harmony search algorithm with additional improvement of harmony memory for asymmetric traveling salesman problem, Expert Systems With Applications 122(2019) 43-53.

[2] T.H. Zhang, Z.W. Geem, Review of harmony search with respect to algorithm structure, Swarm and Evolutionary Computation 48(8)(2019) 31-43.

[3] Z.-L. Guo, H.-G. Yang, S.-W. Wang, C.-Y. Zhou, X.-S. Liu, Adaptive harmony search with best-based search strategy, Soft Computing 22(4)(2018) 1335-1349.

[4] N.S. Jaddi, S. Abdullah, A cooperative-competitive master-slave global-best harmony search for ANN optimization and water-quality prediction, Applied Soft Computing 51(2)(2017) 209-224.

[5] W.-X. Ye, W.-Y. Feng, S.-H. Fan, A novel multi-swarm particle swarm optimization with dynamic learning strategy, Applied Soft Computing 61(12)(2017) 832-843.

[6] S.N. Chegini, A. Bagheri, F. Najafi, PSOSCALF: a new hybrid PSO based on sine cosine algorithm and levy flight for solving optimization problems, Applied Soft Computing 73(12)(2018) 697-726.

[7] A. Assad, K. Deep, A hybrid harmony search and Simulated Annealing algorithm for continuous optimization, Information Sciences 450(2018) 246-266.

[8] H.-B. Ouyang, L.-Q. Gao, X.-Y. Kong, S. Li, D.-X. Zou, Hybrid harmony search particle swarm optimization with global dimension selection, Information Sciences 346-347(2016) 318-337.

[9] Q.-J. Qiu, Y.-J. Xiao, Y. Cao, S.-Z. Yu, An intelligent distributed algorithm of multi-subgroup hierarchical hybrid of simulated annealing algorithm and particle swarm optimization, Journal of Ordnance Equipment Engineering 12(38)(2017) 261-266.

[10] X.H. Yan, H.E. Fazhi, Y.L. Chen, A parallel hardware/software partitioning method based on conformity particle-swarm optimization with harmony search, Scientia Sinica 46(9)(2016) 1321-1338.

[11] F. Qian, Chaotic particle swarm optimization algorithm with harmony search for industrial applications, Journal of Tsinghua University 53(3)(2012) 325-330.

[12] L.-Q. Gao, Y.-F. Ge, Z. Kong, D.X. Zou, Adaptive harmony PSO search algorithm, Control and Decision, Control and Decision 25(7)(2010) 1101-1104.

[13] D.U. Ezgi, A PSO/HS based Algorithm for Optimization Tasks, in: Proc. 2017 Computing Conference, 2017.

[14] X.-W. Gao, J.-Z. Zhou, Research on economic operation problem of power plant based on binary harmonic particle swarm optimization algorithm, Journal of Yangtze River Scientific Research Institute 35(11)(2018) 133-139.

[15] S.Z. Zhao, P.N. Suganthan, Q.K. Pan, M. F. Tasgetiren, Dynamic multi-swarm particle swarm optimizer with harmony search, IEEE Congress on Evolutionary Computation 38(4)(2011) 3735-3742.

[16] J.-Y. Chen, D.-Y. Yang, Dynamic crossover particle swarm optimization based on harmony search, Journal Chinese Computer Systems 34(9)(2013) 2152-2157.

[17] R.K. Meena, M. Jain, S.S. Sanga, A. Assad, Fuzzy modeling and harmony search optimization for machining system with general repair, standby support and vacation, Applied Mathematics and Computation 361(21)(2019) 858-873.

[18] Y.-G. Chen, L.-X. Li, J.-H. Xiao, Y.-X. Yang, J. Liang, T. Li, Particle swarm optimizer with crossover operation, Engineering Applications of Artificial Intelligence 70(4)(2018) 159-169.

[19] F. Solis, R. Wets, Minimization by random search technique, Mathematics of Operations research 6(5)(1981) 19-30.