Expert Systems with Applications 42 (2015) 7436–7455

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A self-adaptive harmony PSO search algorithm and its performance analysis

Fuqing Zhao [a,c,*], Yang Liu [a], Chuck Zhang [b], Junbiao Wang [c]

[a] School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China
[b] H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[c] Key Laboratory of Contemporary Design & Integrated Manufacturing Technology, Ministry of Education, Northwestern Polytechnical University, 710072, China

ARTICLE INFO

Article history:
Available online 27 May 2015

Keywords:
Harmony Search algorithm
PSO algorithm
Self-adaptive scheme
Mutation strategy
Markov model

ABSTRACT

Harmony Search (HS) algorithm is a new population-based meta-heuristic which imitates the music improvisation process and has been successfully applied to a variety of combination optimization problems. In this paper, a self-adaptive harmony particle swarm optimization search algorithm, named SHPSOS, is proposed to solve global continuous optimization problems. Firstly, an efficient initialization scheme based on the PSO algorithm is presented for improving the solution quality of the initial harmony memory (HM). Secondly, a new self-adaptive adjusting scheme for pitch adjusting rate (PAR) and distance bandwidth (BW), which can balance fast convergence and large diversity during the improvisation step, are designed. PAR is dynamically adapted by symmetrical sigmoid curve, and BW is dynamically adjusted by the median of the harmony vector at each generation. Meanwhile, a new effective improvisation scheme based on differential evolution and the best harmony (best individual) is developed to accelerate convergence performance and to improve solution accuracy. Besides, Gaussian mutation strategy is presented and embedded in the SHPSOS algorithm to reinforce the robustness and avoid premature convergence in the evolution process of candidates. Finally, the global convergence performance of the SHPSOS is analyzed with the Markov model to testify the stability of algorithm. Experimental results on thirty-two standard benchmark functions demonstrate that SHPSOS outperforms original HS and the other related algorithms in terms of the solution quality and the stability.

1. Introduction

In the real world production and engineering fields, many practical engineering problems can be transformed into optimization problems (Costantino, Di Gravio, Shaban, & Tronci, 2014;Kadambur & Kotecha, 2015; Xiao, Shao, Gao, & Luo, 2014). However, due to the traditional mathematical methods have certain limitations such as premature convergence, poor global search ability and require derivable fitness functions, so it is difficult to address complex optimization problems. After extensive research and exploration on optimization problems, researchers simulate a variety of evolutionary laws in nature, and many meta-heuristics are put forward. Classical meta-heuristics are described as follows, such as genetic algorithm (Chung, Chan, & Chan, 2013;Goldberg & Holland, 1988), simulated annealing (Arce, Román, Velásquez, & Parada, 2014;Kirkpatrick, 1984), particle swarm optimization (Eberhart & Kennedy, 1995;Leung, Tang, & Wong, 2012), ant colony (Xiao, Ao, & Tang, 2013), artificial bee colony (Gao & Liu, 2012), shuffled complex evolution (Zhao, Jiang, Zhang, & Wang, 2014a, 2014b, 2014c, 2014d, 2015) and other typical hybrid evolution computation algorithm (Zhao et al., 2014a, Zhao, Tang, Wang, and Jonrinaldi, 2014b, 2014c, 2014d; Zhao et al., 2014a, 2014b, Zhao, Tang, Wang, and Jonrinaldi, 2014c, 2014d; Zhao et al., 2014a, 2014b, 2014c, Zhao, Zhang, Wang, & Zhang, 2014d).

Harmony Search (HS) algorithm is a new meta-heuristic algorithm proposed by Geem, Kim, and Loganathan (2001), which is one of the effective methods for solving optimization problems. The basic idea of HS stems from the music improvisation, which mimics the process that musicians repeatedly adjust the pitches of different instruments so as to reach a pleasing harmony eventually. Compared with the earlier meta-heuristics, HS has many advantages such as simple concept, few parameters to be tuned and easy to implement, and also owns a particular way of exploring and exploiting the search space, which has made it very

* Corresponding author at: School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China.

E-mail addresses: Fzhao2000@hotmail.com (F. Zhao), 2476938557@qq.com (Y. Liu), chuck.zhang@isye.gatech.edu (C. Zhang), 827585311@qq.com (J. Wang).

http://dx.doi.org/10.1016/j.eswa.2015.05.035
0957-4174/

successfully in the optimization world, such as Dam system scheduling (Geem, 2007), Cluster analysis (Mehrdad Mahdavi & Abolhassani, 2009) and others (Li, Li, & Gupta, 2014;Salcedo-Sanz et al., 2013; Zou, Gao, Li, & Wu, 2011). However, the main drawback of the original HS is that the parameters are set to fixed values, and it is difficult to suggest values that work well in different situations. In order to enhance the accuracy and convergence performance of solution, a wide variety of modifications have been proposed.

Modifications of the HS mainly include two aspects: (1) by changing the way HS behaves (including HM initialization, parameters setting, the way of generating a new solution, algorithm structure and others); (2) by hybridizing HS with other meta-heuristic algorithms.

The parameters setting greatly affect the performance of HS, but tuning the parameters through experiments to obtain optimal setting is quite time-consuming. Therefore, a dynamic scheme is adopted by the researchers. Mahdavi, Fesanghary, and Damangir (2007) discussed the impacts of constant parameters on HS algorithm and then introduced dynamic parameters PAR and BW into HS, where PAR increased linearly and BW decreased exponentially with the number of generations. Experimental results show that performance of the IHS algorithm has been explicitly improved compared with original HS Omran and Mahdavi (2008) put forward global-best harmony search (GHS) inspired by PSO (Eberhart & Kennedy, 1995), which made full use of the direction information of the best harmony (best individual) in HM to guide the search. Experimental results show that the performance of GHS is superior to HS and IHS. Wang and Huang (2010) proposed a self-adaptive harmony search algorithm (SAHS) which excludes the selection of the values of parameters PAR and BW. That is, SAHS generated a new solution vector according to the maximum and minimum values of the decision variables in HM with some associated probability Pan, Suganthan, Tasgetiren, and Liang (2010) presented a self-adaptive global best harmony search (SGHS) algorithm inspired by GHS. Unlike GHS, SGHS employed a new improvisation scheme. HMCR and PAR are dynamically adapted by the learning mechanism. BW is dynamically tuned to favor exploration in the early stages and exploitation for the final stages Zou, Gao, Wu, Li, and Li (2010) developed a novel global harmony search (NGHS) algorithm inspired by the swarm intelligence of particle swarm. NGHS included two important mechanisms with a small probability, which are position updating and genetic mutation. The former enables the worst harmony in HM to move to the global best harmony rapidly, and the latter can effectively prevent the NGHS from trapping into the local optimum. Geem and Sim (2010) proposed a novel technique, named parameter-setting-free (PSF) Harmony Search algorithm, which eliminates tedious and experience-requiring parameter assigning efforts. PSF technique contains one additional matrix for each variable in harmony memory according to the certain operation type including random selection, memory consideration and pitch adjustment Pan, Suganthan, Liang, and Tasgetiren (2011) presented a local-best HS with dynamic sub-harmony memory, which randomly divided HM into sub-harmony memories. Sub-harmony memories recombine and exchange information each other after a certain generations. Meanwhile, a chaotic sequence to produce decision variables for harmony vectors and a mutation scheme are utilized to enhance the diversity of the HM. The kind of strategy can keep the algorithm diversity well. Chen, Pan, and Li (2012) proposed an effective HS which took advantage of tournament selection rule to generate a new harmony. PAR and BW are adjusted dynamically with respect to the evolution of the search process and the different search spaces of the optimization problems Ashrafi and Dariane (2013) presented an innovative improved version of HS, named Melody Search (MS) Algorithm. MS algorithm mimics performance processes of the group improvisation for finding the best succession of pitches within a melody. A novel Alternative Improvisation Procedure (AIP) was employed by Valian, Tavakoli, and Mohanna (2014) introduced a novel improvisation scheme, called an intelligent global Harmony Search, which employs the swarm intelligence technique Khalili, Kharrat, Salahshoor, and Sefat (2014) developed a global dynamic harmony search (GDHS) algorithm which eliminates setting parameters that have to be defined before optimization process. All the key parameters are changed to dynamic mode and there is no need to predefine any parameters. Kumar, Chhabra, and Kumar (2014) presented a parameter adaptive harmony search (PAHS) algorithm, where two key parameters HMCR and PAR are both being allowed to change dynamically. Four different cases of linear and exponential changes are explored in the PAHS algorithm. Castelli, Silva, Manzoni, and Vanneschi (2014) introduced a new variant of the Harmony Search algorithm, called Geometric Selective Harmony Search. A selection procedure was adopted in the improvisation phase. A recombination operator and a new mutation operator were employed in the memory consideration process. In short, different kinds of modifications (El-Abd, 2013;Hasan, Abu Doush, Al Maghayreh, Alkhateeb, & Hamdan, 2014; Mun & Cho, 2012;Wang et al., 2013; Yadav, Kumar, Panda, & Chang, 2012) emerge in endlessly.

The other aspect is hybridization with other meta-heuristics. The characteristics of two algorithms are extracted to improve the performance of HS. According to the strategy, Wang and Li (2012) put forward hybridization of DE/HS, where two populations evolve simultaneously and cooperatively, one population for the continuous part evolves by means of differential evolution while another population for the integer part evolves by means of Harmony Search. The DE/HS has been applied to address reliability–redundancy optimization problems. Wang and Guo (2013) presented HS/BA which betters the performance respectively. In the HS/BA, pitch adjustment operation in HS is served as a mutation operator during the process of the bat updating to speed up convergence. In order to accelerate search efficiency and performance, HS hybridized with PSO, called PSO-CE-GHS, was proposed by Wang and Yan (2013). In the PSO-CE-GHS, Harmony Search operators are applied to evolve the original population. PSO is applied to co-evolve the symbiotic population. Thus, the symbiotic population is dynamically and self-adaptively adjusted with the evolution of the original population Xiang, An, Li, He, and Zhang (2014) proposed an improved global-best harmony search (IGHS) algorithm. In IGHS, a new improvisation scheme based on differential evolution was employed to enhance the local search ability and a modified random consideration based on artificial bee colony algorithm for reducing randomness of the global-best harmony search (GHS) algorithm are integrated. HMCR and PAR are designed as a periodic function and a sign function in view of approximate periodicity of evolution in nature. Hosseini, Akbarpour Shirazi, and Karimi (2014) proposed a hybrid of Harmony Search (HS) and simulated annealing (SA) based heuristics for consolidation network. Moslehi and Khorasanian (2014) presented a hybrid variable neighborhood search (HVNS) algorithm hybridized with the simulated annealing algorithm, which is used to solve the flow shop problem. Literatures (Moh'd Alia & Mandava, 2011), (Manjarres et al., 2013) and (Yoo, Kim, & Geem, 2014) make a detailed survey on the development of HS, which are comprehensive references in recent years.

Although the performance of HS has been improved through the aforementioned methods, there are still some intrinsic problems in the original HS. Therefore, a self-adaptive harmony particle swarm optimization search algorithm, named SHPSOS, is proposed in this paper. Usually, PAR and BW are described either linear or exponential type. In SHPSOS, a new self-adaptive adjusting scheme for parameters PAR and BW is designed with symmetrical sigmoid type, which can lead to fast convergence during the improvisation step. Many heuristic algorithms have been employed to enhance the local search ability of HS in the literatures. In the proposed algorithms, PSO algorithm is utilized to initial the harmony memory to improve the quality of initial solutions. A new improvisation scheme based on differential evolution and the best harmony (best individual), which guides the candidates close to the best harmony with fast speed, is employed. Gaussian mutation strategy is added to enhance the robustness and avoid premature convergence during the evolutionary process of populations. Additionally, the global convergence of the proposed algorithm based on Markov model has also been analyzed. By testing thirty-two standard benchmark functions, the numerical results demonstrate that SHPSOS has a strong global optimization ability compared with the well-known HS variants such as IHS, GHS, SGHS, NGHS and AIP_MS.

The remainder of the paper is organized as follows. Section 2 summarizes the original HS algorithm. The SHPSOS algorithm is described in Section 3. Section 4 analyzes the convergence of the SHPSOS algorithm. And some experimental studies regarding the numerical benchmark functions, along with their analysis and discussions are stated in Section 5. Finally, we end the paper with some conclusions and future work in Section 6.

2. Harmony Search algorithm

HS algorithm is one of the latest evolutionary computation meta-heuristics in the literature. Musical performances seek a best state (fantastic harmony) determined by aesthetic evaluation, which is a set of the sounds played by joined instruments, as the meta-heuristics seek a best state (global optimum) determined by objective function evaluation, which is a set of values produced by component variables; the sounds for better evaluation can be improved through practice after practice, the values for better objective function estimation can be improved iteration by iteration.

The HS consists of the following main steps:

Step1: Optimization problem description and initialization of HS parameters.

The initialization of the original HS aims to set the control parameters and to fill in the HM. The main control parameters of HS are specified as follows, harmony memory size (HMS) is similar to the population size in EA; harmony memory considering rate (HMCR) determines the rate of selecting the value from the HM; pitch adjusting rate (PAR) determines the probability of local improvement; bandwidth (BW) determines the distance of adjustment; number of improvisations (NI). The optimization problem is characterized as a function$f$to be minimized:

$\min f(X), X = (x_1, x_2 \ldots, x_n), x_i \in [LB_i, UB_i], i = 1, 2 \ldots n.$ where $f(X)$ is the objective function. X is a solution vector composed of decision variable x_i. LB_i and UB_i are lower and upper bounds for the decision variable x_i, respectively.

Step 2: HM initialization.

In this step, the HM matrix, shown in Eq. (1), is filled with randomly generated solution vectors between lower and upper bounds and ranked by the values of the objective function.

$$HM = \begin{bmatrix} X^1 & f(X^1) \\ X^2 & f(X^2) \\ \vdots & \vdots \\ X^{HMS} & f(X^{HMS}) \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_n^1 & f(X^1) \\ x_1^2 & x_2^2 & \ldots & x_n^2 & f(X^2) \\ \vdots & \vdots & & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \ldots & x_n^{HMS} & f(X^{HMS}) \end{bmatrix} \tag{1}$$

Step3. Improvise a new harmony

In this step, a new harmony vector $X^{new} = (x_1^{new}, x_2^{new}, x_i^{new} \ldots x_n^{new})$ is generated based on three operators: (1) memory consideration; (2) pitch adjustment; (3) random selection. For a new harmony vector $X^{new} = (x_1^{new}, x_2^{new}, x_i^{new} \ldots x_n^{new})$. If a component (x_i^{new}) is randomly selected from the historical values stored in HM, it is performed as Eq. (2)

$$x_i^{new} = \begin{cases} x_i \in \{x_i^1\ x_i^2\ \ldots\ x_i^{HMS}\} & \text{if } rand < HMCR \\ x_i \in X_i & otherwise \end{cases} \tag{2}$$

where $X_i (i = 1, 2, \ldots n)$ is the ith search space. The HMCR is the probability of assigning one value based on historical values stored in the HM, and (1-HMCR) is the probability of randomly assigning one value according to their possible range. A HMCR value of 1.0 is not recommended because it eliminates the possibility that the solution may be improved by the values which are not stored in the HM. If the component x_i^{new} comes from HM, it will be tuned with a possibility of PAR which is performed as Eq. (3)

$$x_i^{new} = \begin{cases} x_i^{new} \pm rand * bw & \text{if } rand < PAR \\ x_i^{new} & otherwise \end{cases} \tag{3}$$

where rand is uniformly distributed number between 0 and 1 and BW is an arbitrary distance bandwidth for the continuous decision variables.

Step 4: Update the HM.

In this step, if the newly generated harmony vector outperforms the worst one in the HM in terms of the objective function value, then the new harmony will substitute the existing worst one. The HM is sorted by objective function values again.

Step 5: Check the termination criterion.

In this step, check whether the termination criterion is satisfied, if not, return Step 3 and 4 until the maximum number of iterations is met.

3. Self-adaptive harmony PSO search algorithm (SHPSOS)

3.1. self-adaptive parameters and mutation strategy

(1) Parameter PAR in the literature is either linear or exponential type. In the paper, a new self-adaptive strategy which dynamically adjusts the parameters with respect to the evolution process of population (Fan & Yan, 2014) were designed, it is described as Eq. (4) and PAR graph is shown in Fig. 1.

$$PAR(t) = PAR_{\min} + (PAR_{\max} - PAR_{\min})/(1 + \exp(20 * t/NI - 10)) \tag{4}$$

where PAR (t) is the pitch adjusting rate for t generation, PAR_{min} and PAR_{max} are the minimum and maximum pitch adjusting rates, respectively; NI is the number of solution vector generations, t is the generation number. From Fig. 1, it can be observed that PAR drops as symmetrical sigmoid curve, which can increase the diversity of solutions at the beginning of iterations, as PAR decreases, the diversity of solution decreases, but the convergence speed and solutions accuracy are improved.

(2) BW takes the median of each dimension and is not needed to set initial values which reduce the impact of human factors on the algorithm performance. It is expressed in Eq. (5)

$$bw(i) = median(HM(:,\ i)) \quad i \in [1, D] \tag{5}$$

where $bw(i)$ is the bandwidth for dimension$i$. D represents the dimension of the fitness function.

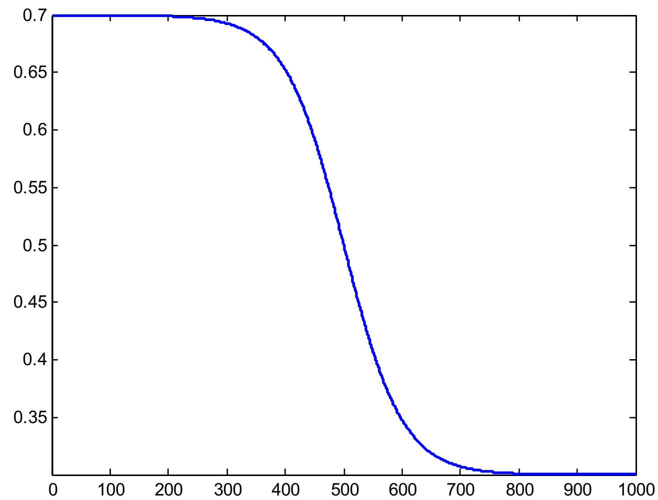


Fig. 1. PAR graph in the SHPSOS.

(3) Mutation strategy of SHPSOS is given bellow.

for each i $\in$[1:D] **do**
  if rand > 1 – PAR(t) **then**
    $R = X^i_{new}$ – HM(:, i)
    if max(R) < X^i_{new} × (le − 3) **then**
      $X^i_{new} = X^i_{new} \times (1 \pm randn \times (5e-2))$
      if max(R) < X^i_{new} × (le − 5) **then**
        $X^i_{new} = X^i_{new} \times (1 \pm randn \times (1e-2))$
      endif
    endif
  endif
Done

3.2. The steps of SHPSOS

Step 1: Initialization of the SHPSOS parameters.

In this step, initialization includes two parts: (1) parameters of HS: HMS, HMCR, PAR_{min}, PAR_{max}, BW, NI; (2) parameters of PSO: w_{max}, w_{min}, c1, c2, T, v_{max}, p(the number of particles).

Step 2: HM initialization.

HM is filled with HMS best solution vectors according to objective function values which are generated by PSO algorithm, it is described as follows.

$$w(t) = w_{min} + (w_{max} - w_{min})/(1 + \exp(20^* t/T - 10)) \tag{6}$$

$$c_1 = (c_1 f - c_1 s)/(t/T) + c_1 s \tag{7}$$

$$c_2 = (c_2 f - c_2 s)/(t/T) + c_2 s \tag{8}$$

$$v_{id}(t+1) = w v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \tag{9}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{10}$$

Where p_{id} and p_{gd} are local optimal place and global optimal place, respectively.w is the inertia weight, c_1 and c_2 are learning factors. T is the number of iterations in the PSO algorithm. r_1 and r_2 are uniformly distributed numbers between 0 and 1.

Step 3: Improvise a new harmony.

The procedure of improvising a new harmony is shown as follows.

for t < NI **do**
  for each i$\in$[1:D] **do**
    if rand < HMCR **then**
      $X^i_{new} = X^i_a$ where a $\in$ (1,2,...,HMS)
      $X^j_{new} = X^j_a + randn(X^j_{best} - X^j_a)$ *where* $j \leftarrow (1:D)$
      if $f(X^j_{new}) \leqslant f(X^j_a)$ **then**
        $X^j_{worst} = X^j_{new}$
        Replace X^j_{worst} with X^j_{new}
      endif
        if rand < PAR(t) then //*See Eq.* (4)
          $X^i_{new} = X^i_{new} \pm rand \times bw(i)$ //*See Eq.* (5)
        if $f(X^j_{new}) \leqslant f(X^j_a)$ **then**
          $X^i_a = X^i_{new}$
        endif
      endif
    else
      $X^i_{new} = LB_i + rand \times (UB_i - LB_i)$
    endif
    Add mutation strategy
    $X^{i'}_{new} = X^i_{new}$
  endfor
Done

Step 4: Update harmony memory.

In this step, it is similar to the original HS. The pseudo code is stated as follows.

$X^j_{new} = X^{i'}_{new} + rand(1, D) \times (X^i_{best} - X^{i'}_{new})$
Judge the range of $X^{i'}_{new}$
if $f(X^j_{new}) \leqslant f(X^{i'}_{new})$ **then**
  $X^{i'}_{new} = X^j_{new}$
endif
  if $f(X^{i'}_{new}) \leqslant f(X^j_{worst})$ **then**
    $X^j_{worst} = X^{i'}_{new}$
  endif

Step 5: Check the termination criterion.

This step is the same as the original HS.

3.3. The parameters setting of different HS algorithms

- HS: HMS = 5, HMCR = 0.9, PAR = 0.3, BW = 0.01;
- IHS:HMS = 5, HMCR = 0.9, PAR_{min} = 0.01, PAR_{max} = 0.99, BW_{min} = 0.0001, $BW_{max} = (x_u - x_l)/20$;
- GHS:HMS = 5, HMCR = 0.9, PAR_{min} = 0.01, PAR_{max} = 0.99;
- SGHS:HMS = 5, HMCRm = 0.98, PARm = 0.9, BW_{min} = 0.0005, $BW_{max} = (x_u - x_l)/10$, LP = 100;
- NGHS: HMS = 5, P_m = 0.005;
- AIP_MS: PMS = 5, PMN = 5, PMCR = 0.98, PAR_{min} = 0.01, PAR_{max} = 0.99, $BW(j) = (U_j - L_j)/200$;
- SHPSOS: HMS = 5, HMCR = 0.9, PAR_{min} = 0.3, PAR_{max} = 0.7, BW(i) = median(HM(:,i)), w_{max} = 0.9, w_{min} = 0.4, p = 40, T = 1000, NI = 15000, c1s = c2s = 1.5, c1f = c2f = 2.5, v_{max} = 0.3;

4. Analysis of the convergence in SHPSOS

Since the development of HS, only numerical results illustrate its convergence performance and it lacks theory analysis. In this paper, the convergence performance of SHPSOS through Markov model was analyzed.

4.1. The convergence criterion of random search algorithm

SHPSOS is a random search algorithm, whether the algorithm converged to global optima can be examined through the convergence criterion.

Solis and Wets (1981) presented the general convergence criterion of random search algorithm. For the optimization problems $< S,f >$,the result of the $k$thgeneration is x_k, next generation is $x_{k+1} = B(x_k, \xi)$,where$S$ is search space,f is the fitness function and ξ are the searched solutions in the process of the algorithm$B$.

The infimum of Lebesgue measure volume, which can be defined as follows,

$$\alpha = \inf(t : v[x \in S|f(x) < t] > 0) \tag{11}$$

where $v[X]$ denotes the Lebesgue measure on set of X. Eq. (11) indicates that there exist non-empty subsets in the search space, and its members corresponding to the fitness value can be infinitely close to β. Therefore, the optimality region is defined as follow.

$$R\varepsilon, M = \begin{cases} \{x \in S|f(x) < \alpha + \varepsilon\}, & \alpha \text{ is finite} \\ \{x \in S|f(x) < C\}, & \alpha = -\infty \end{cases}$$

where ε >0,C < 0. If the random search algorithm has found a point in $R\varepsilon, M$, it can be considered that the algorithm finds the global optimal solution or approximate global optimal solution.

Assumption 1. For algorithm$B$, let $\{f(x^k)\}_{k=0}^{\infty}$ be monotonous non-increasing. If $f(B(x,\xi)) \leqslant f(x)$, where $\xi \in S$, then $f(B(x,\xi)) \leqslant \min\{f(x), f(\xi)\}$.

Assumption 2. For $\forall L \in S, s.t.\ v(L) > 0$, we have that $\prod_{k=0}^{\infty}(1 - \mu_k(L)) = 0$

Where $\mu_k(L)$ is the probability measure on set of Lwhich is generated by algorithm$B$at step k.

Lemma 1. Suppose that$f$ is a measurable function, and$S$ is the measurable subset of R^n.The algorithm$B$meets the condition $(H1)$ as well as condition $(H2)$. Let $\{x_k\}_{k=0}^{\infty}$ be a sequence generated by algorithm$B$. Then $\lim_{k\to\infty} P(x_k \in R\varepsilon, M) = 1$, namely, algorithm$B$can find global optima, where $P(x_k \in R\varepsilon, M)$ is the probability measure of the point x_k generated by algorithm$B$at step k in $R\varepsilon, M$.

Proof. From $H1, x^k \notin R\varepsilon, M \Rightarrow x^l \notin R\varepsilon, M, \forall l < k$,

$$P(x^k \in S \backslash R\varepsilon, M) \leqslant \prod_{l=0}^{k-1}(1 - u_l(R\varepsilon, M)),$$

$$P(x^k \in R\varepsilon, M) = 1 - P(x^k \in S \backslash R\varepsilon, M) \geqslant 1 - \prod_{l=0}^{k-1}(1 - u_l(R\varepsilon, M)),$$

$$\text{From } H2, 1 \geqslant \lim_{k\to\infty} p(x_k \in R\varepsilon, M) \geqslant 1 - \lim_{k\to\infty}\prod_{l=0}^{k-1}(1 - u_l(R\varepsilon, M)) = 1$$

□

4.2. Markov model and convergence performance of SHPSOS

The harmony memory in the HS is similar to the best individual in Genetic algorithm, which ensures that the best individual in HM can transfer into next generation. Then the pitch of each instrument is adjusted, whose process can be seen as a change of each pitch state.

Definition 1. For random sequence $\{Xn\ , Xn \in S, n \geqslant 0\}$. S is the state space. For $\forall n \geqslant 0$ and any state $i, j, i_0, \ldots, i_{n-1}$, if

$$P(Xn+1 = j|Xn = i, Xn-1 = i_{n-1}, \ldots, X0 = i_0) = P(Xn+1 = j|Xn = i) \tag{12}$$

$\{Xn\ , Xn \in S, n \geqslant 0\}$ is a Markov chain. Eq. (12) may be interpreted as stating that for a Markov chain, the conditional distribution of any future state $Xn+1$ is independent of the past states and depends only on the present state X_n.

Definition 2. If $f_{ii} = 1, i$ is a recurrent state; if $f_{ii} < 1, i$ is a transient state.

Definition 3. On the condition of $f_{ii} = 1$, the $\mu_{ii} = \sum_{n=1}^{\infty} t f_{ii}^{(t)}$ sets as the average visit time starting from state$i$and return to state$i$. If $\mu_{ii} < \infty$, state$i$ is positive recurrent; if $\mu_{ii} = \infty$, state$i$ is null recurrent.

Definition 4. If the set $\{t : t \geqslant 1, p_{ii}^{(t)} > 0\} \neq \emptyset$, greatest common divisor $d(i)$ is the period of the state$i$. If $d(i) > 1$, state$i$ is periodic; and if $d(i) = 1$, state$i$ is aperiodic.

Theorem 1. *In the SHPSOS algorithm, suppose that the sequence $\{X(t), t \in T\}$ is the best harmony (best individual) in the evolutionary process, where t is the evolution generation. We can have that sequence $\{X(t), t \in T\}$ is monotonically decreasing (for the minimum).*

Proof. In the SHSPSOS algorithm, the best harmony is sorted in ascending order in harmony memory. The next new individual relies on three cases: (1) memory consideration; (2) pitch adjustment; (3) random selection. No matter which case of the new individual depends on, there is only individual changed in harmony memory at each generation. Thus, the function value for the next new individual is either becoming better or remaining unchanged than that of the previous generation, which can be expressed as $f(X(t+1)) \leqslant f(X(t))$. So we can say that the sequence $\{X(t), t \in T\}$ in the SHPSOS algorithm is monotonically decreasing. □

Theorem 2. *In the SHPSOS algorithm, the SHPSOS generates new best harmony (best individual) with Markov chain. Let $\zeta_T = \min\{f(X_1), f(X_2), \cdots, f(X_T)\}$, where ζ_T denotes the state of the harmony memory at time T, the sequence $\{\zeta_T\ , t \geqslant 0\}$ is a Markov chain.*

Proof. To prove Theorem 2 is true, only prove $\zeta = \{\zeta_T, t \geqslant 0\}$ satisfies the Eq. (12).

The left side of equation:

$$\begin{aligned} &P(\zeta_{t+1} = j|\zeta_t = i_t, \zeta_{t-1} = i_{t-1}, \ldots, \zeta_1 = i_1, \zeta_0 = i_0) \\ &\quad = P(\min\{f(X_0), f(X_1), \ldots, f(X_{t-1}), f(X_t), \zeta_{t+1}\} = j|\zeta_t = i_t, \zeta_{t-1} \\ &\quad = i_{t-1}, \ldots, \zeta_1 = i_1, \zeta_0 = i_0) = P(\min\{f(X_0), f(X_1), \ldots, i, \zeta_{t+1}\} = j|\zeta_t \\ &\quad = i_t, \zeta_{t-1} = i_{t-1}, \ldots, \zeta_1 = i_1, \zeta_0 = i_0) = \begin{cases} p(f(X_{t+1}) = j) & j < i \\ p(f(X_{t+1}) \geqslant i) & j = i \\ 0 & j > i \end{cases} \end{aligned}$$

The right side of equation:

$$P(\zeta_{t+1} = j|\zeta_t = i) = P(\min\{f(X_{t+1}), i\} = j) = \begin{cases} p(f(X_{t+1}) = j) & j < i \\ p(f(X_{t+1}) \geqslant i) & j = i \\ 0 & j > i \end{cases}$$

The right side equals the left. From the reasoning above, we can draw a conclusion that the state X_{t+1} in SHPSOS is only related with X_t According to Definition 1, the solution sequence $\{\zeta_T\ , t \geqslant 0\}$ can be described as a Markov chain. □

Table 1
Benchmark problems.

Test functions	Domain range	Optimum
$f1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
$f2(x) = \sum_{i=1}^{D} \lvert xi \rvert + \prod_{i=1}^{D} \lvert xi \rvert$	$[-10, 10]^D$	0
$f3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$	0
$f4(x) = \max i^D \{\lvert xi \rvert\}$	$[-100, 100]^D$	0
$f5(x) = \sum_{i=1}^{D-1} [100(xi+1-xi^2)^2 + (xi-1)^2]$	$[-30, 30]^D$	0
$f6(x) = \sum_{i=1}^{D} (\lfloor xi + 0.5 \rfloor)^2$	$[-100, 100]^D$	0
$f7(x) = \sum_{i}^{D} ix_i^4 + random[0, 1)$	$[-1.28, 1.28]^D$	0
$f8(x) = 418.9829D - \sum_{i=1}^{D} (x_i \sin \sqrt{\lvert x_i \rvert})$	$[-500, 500]^D$	0
$f9(x) = \sum_{i=1}^{D} (xi^2 - 10\cos(2\pi xi) + 10)$	$[-5.12, 5.12]^D$	0
$f10(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} xi^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi xi)\right)$	$[-32, 32]^D$	0
$f11(x) = \frac{1}{4000}\sum_{i=1}^{D} xi^2 - \prod_{i=1}^{D} \cos\left(\frac{xi}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	0
$f12(x) = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})](y_D + 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)10\sin^2(\pi y_1)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leqslant x_i \leqslant a \\ k(-x_i - a)^m & x_i \leqslant a \end{cases}$	$[-50, 50]^D$	0
$f13(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$	0
$f14(x) = 0.5 + \frac{\sin^2\sqrt{\sum_{i=1}^{D} xi^2} - 0.5}{(1 + 0.001\sum_{i=1}^{D} xi^2)^2}$	$[-100, 100]^D$	0
$f15(x) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$ where $y_i = \begin{cases} x_i & \lvert x_i \rvert < 0.5 \\ round(2x_i)/2 & \lvert x_i \rvert \geqslant 0.5 \end{cases}$	$[-5.12, 5.12]^D$	0
$f16(x) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{1/4} (\sin(50(x_i^2 + x_{i+1}^2)^{1/10})^2 + 1)$	$[-100, 100]^D$	0
$f17(x) = \sum_{i=1}^{D} ix_i^2$	$[-10, 10]^D$	0
$f18(x) = x_1^2 + 10^6\sum_{i=2}^{D} x_i^2$	$[-100, 100]^D$	0
$f19(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]^D$	0
$f20(x) = \sum_{i=1}^{D} ix_i^4$	$[-1.28, 1.28]^D$	0
$f21(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$	$[-100, 100]^D$	0
$f22(x) = \sum_{i=1}^{D} z_i^2 \quad z = x - o$	$[-100, 100]^D$	0
$f23(x) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) \quad z = x - o$	$[-5, 5]^D$	0
$f24(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi z_i)) \quad z = x - o$	$[-32, 32]^D$	0
$f25(x) = \frac{1}{4000}\sum_{i=1}^{D} z_i^2 - \prod_{i=1}^{D} \cos(\frac{z_i}{\sqrt{i}}) + 1 \quad z = x - o$	$[-600, 600]^D$	0
$f26(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} z_i)^2 \quad z = x - o$	$[-100, 100]^D$	0
$f27(x) = \sum_{i=1}^{D-1} [100(zi + 1 - zi^2)^2 + (zi - 1)^2] \quad z = x - o + 1$	$[-100, 100]^D$	0
$f28(x) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) \quad z = (x - o)M$	$[-5, 5]^D$	0
$f29(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} zi^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi zi)\right) \quad z = (x - o)M$	$[-32, 32]^D$	0
$f30(x) = \frac{1}{4000}\sum_{i=1}^{D} z_i^2 - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \; z = (x - o)M$	$[-600, 600]^D$	0
$f31(x) = f8(f2(z_1, z_2)) + f8(f2(z_2, z_3)) + \ldots + f8(f2(z_{D-1}, z_D)) + f8(f2(z_D, z_1)) \quad z = x - o + 1$	$[-3, 1]^D$	0
$f32(x) = f(z_1, z_2) + f(z_2, z_3) + \cdots + f(z_{D-1}, z_D) + f(z_D, z_1)$ $z = (x - o)M$	$[-100, 100]^D$	0

Theorem 3. *The SHPSOS algorithm has unique positive recurrence.*

Proof. According to Theorem 2, the solutions in HM have been the best so far. Therefore, the one-step transition probability matrix of SHPSOS at t th generation is described as follows.

$$P^{(t)} = (p_{ij}^{(t)})|n||n| = \begin{bmatrix} 1 & & & \\ p_{21}^{(t)} & p_{22}^{(t)} & & \\ \vdots & \vdots & \ddots & \\ p_{|n|1}^{(t)} & p_{|n|2}^{(t)} & \cdots & p_{|n||n|}^{(t)} \end{bmatrix} \tag{13}$$

In Eq. (13), $P^{(t)}$ is a lower triangular matrix, where $p_{ij}^{(t)} > 0, i > j$. According to Theorem 1, the solution of state ith is better than that of state $(i+1)$th.The first state has been testified to be a global optimal solution. For the first state, the probability of arriving for first time is as follows.

$$f_{11} = \sum_{t=1}^{\infty} f_{11}^{(t)} = 1 + 0 + 0 + \cdots = 1 \tag{14}$$

For the other states, the probability of arriving for first time is as follows,

$$f_{ii} = \sum_{t=1}^{\infty} f_{ii}^{(t)} = p_{ii} + 0 + 0 + \cdots = p_{ii} < 1 \tag{15}$$

From the Definition 2, we can say that the first state is the recurrent state and others are transient states. For the first state, the average return time is

$$\mu_{11} = \sum_{t=1}^{\infty} tf_{11}^{(t)} = 1 + 0 + 0 + \cdots = 1 < \infty \tag{16}$$

According to the Definition 3, we can summarize that the first state is the unique positive recurrence.

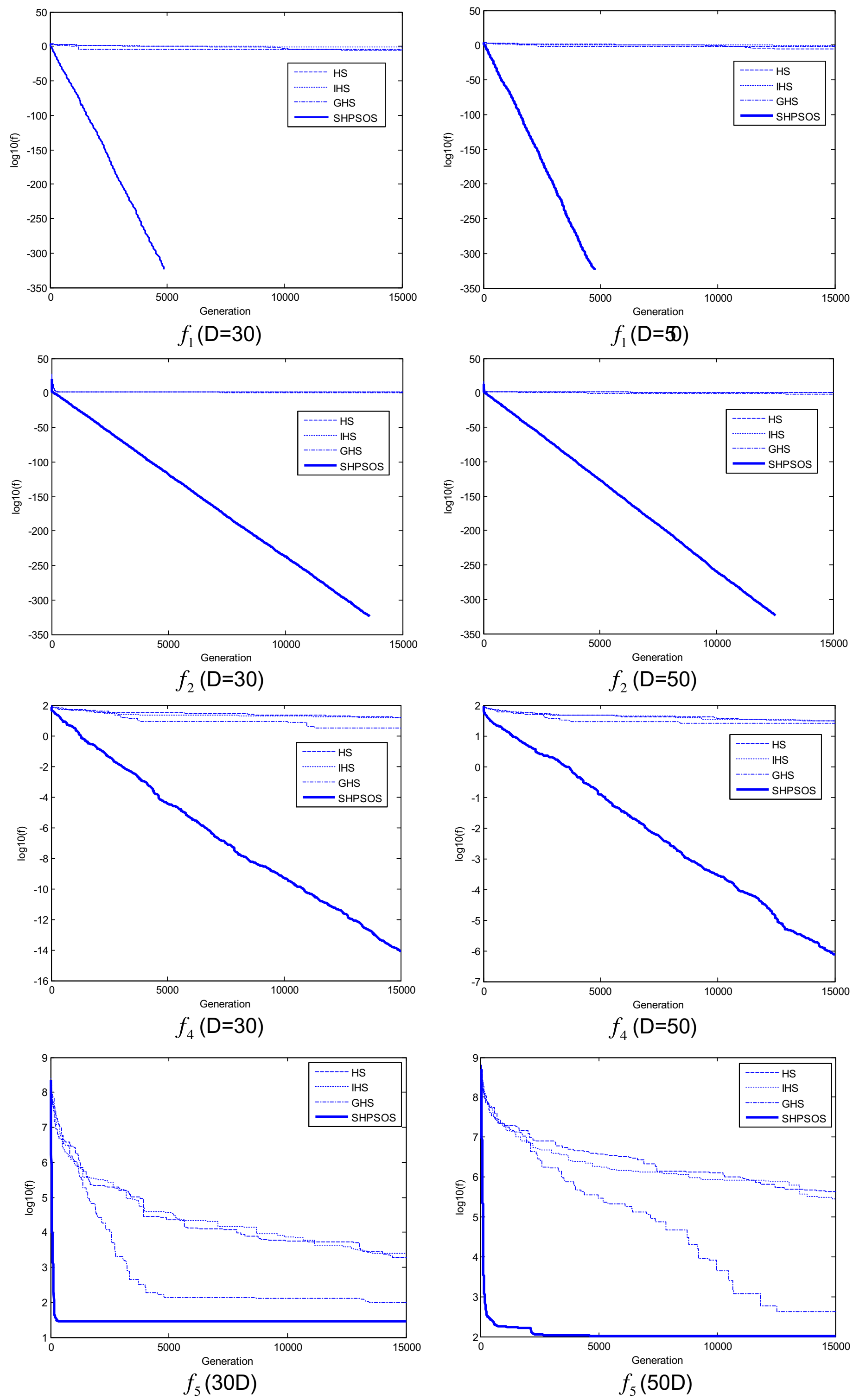


Fig. 2. Convergence graphs of some typical benchmark functions.

Theorem 4. *The SHPSOS algorithm can converge to global optimal set B^* with probability of 1. Namely,* $\lim_{t\to\infty} P(X(t) \in B^* | X(0) = i_0) = 1$.

Proof. Suppose that the closed set B^* is the global optimal states. The return time of the global optimal state constituted a set, which can be described as $\{t : t \geqslant 1, p_{11}^{(t)}\} = \{t : t = 1, 2, 3, 4 \ldots\}$. The greatest common divisor (gcd) of the set is one. According to the Definition 4, we can say that the set closed B^* is aperiodic. Then for any initial distribution π_0, there exists

$$\lim_{t\to\infty} P(X(t) = j | X(0) = i_0) = \begin{cases} \pi(j) & j \in B^* \\ 0 & j \notin B^* \end{cases} \tag{17}$$

That is, $\lim_{t\to\infty} P(X(t) \in B^* | X(0) = i_0) = 1$.

From the Theorem 1, we can say that the SHPSOS meets the Assumption 1. Meanwhile, the SHPSOS satisfies the SHPSOS meets

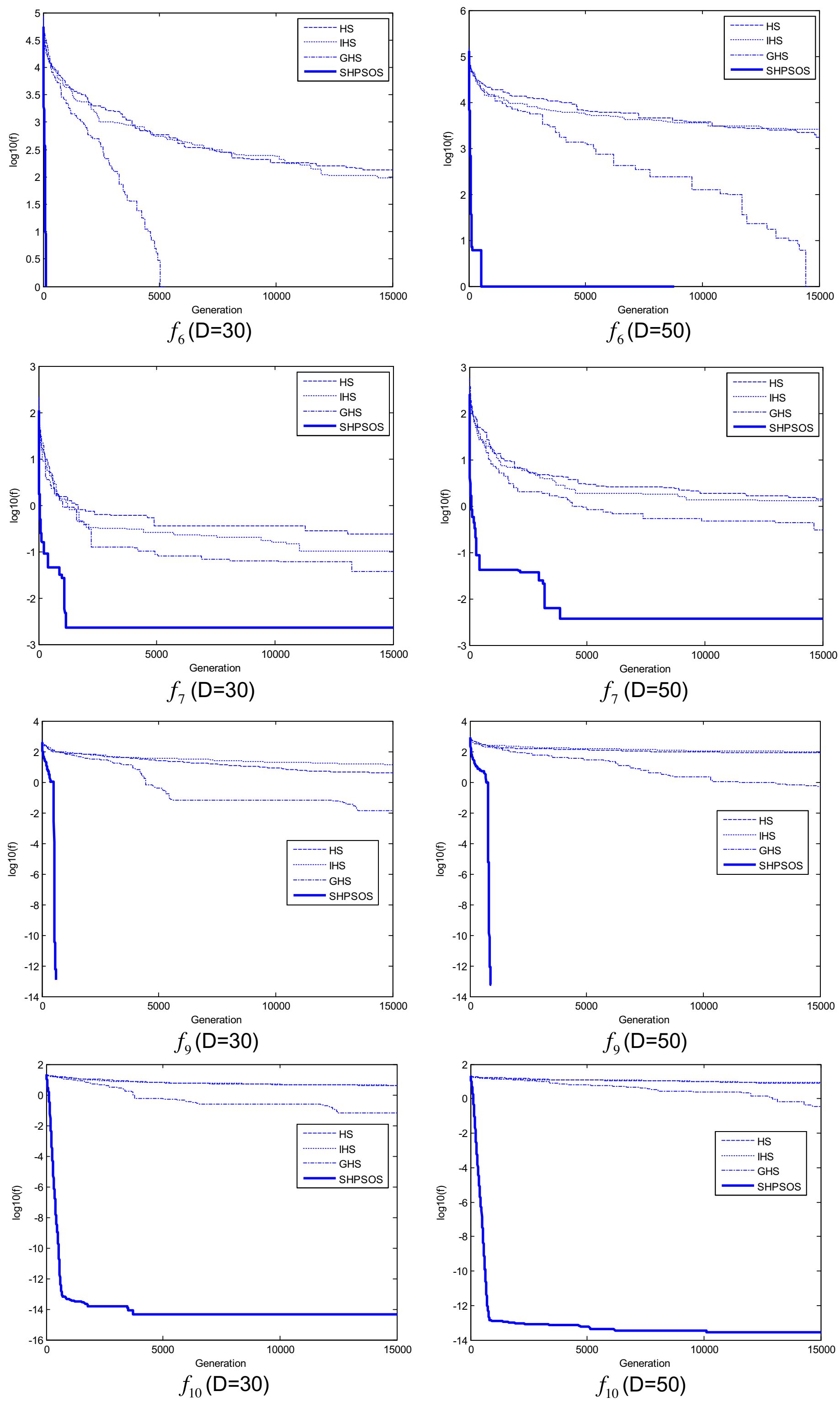


Fig. 2 (*continued*)

the Assumption 2 by the Theorem 4. According to the Lemma 1, SHPSOS can converge to the global optimum. □

5. The experiments and comparisons

In this paper, thirty-two typical benchmark functions, which are selected from Liang, Qu, and Suganthan (2013), Suganthan et al. (2005) and Ashrafi and Dariane (2013), are employed to test the SHPSOS performance. These benchmark functions with different characteristics are listed briefly in Table 1 and are divided into four groups: the first group $f_1 - f_5$ and $f_{17} - f_{21}$ are unimodal continuous functions. Function f_{06} is a step function which is discontinuous and has one minimum. Function f_{07} is a quartic function with white noise. Functions $f_{08} - f_{16}$ in the second group are high-dimensional multimodal functions which have many local

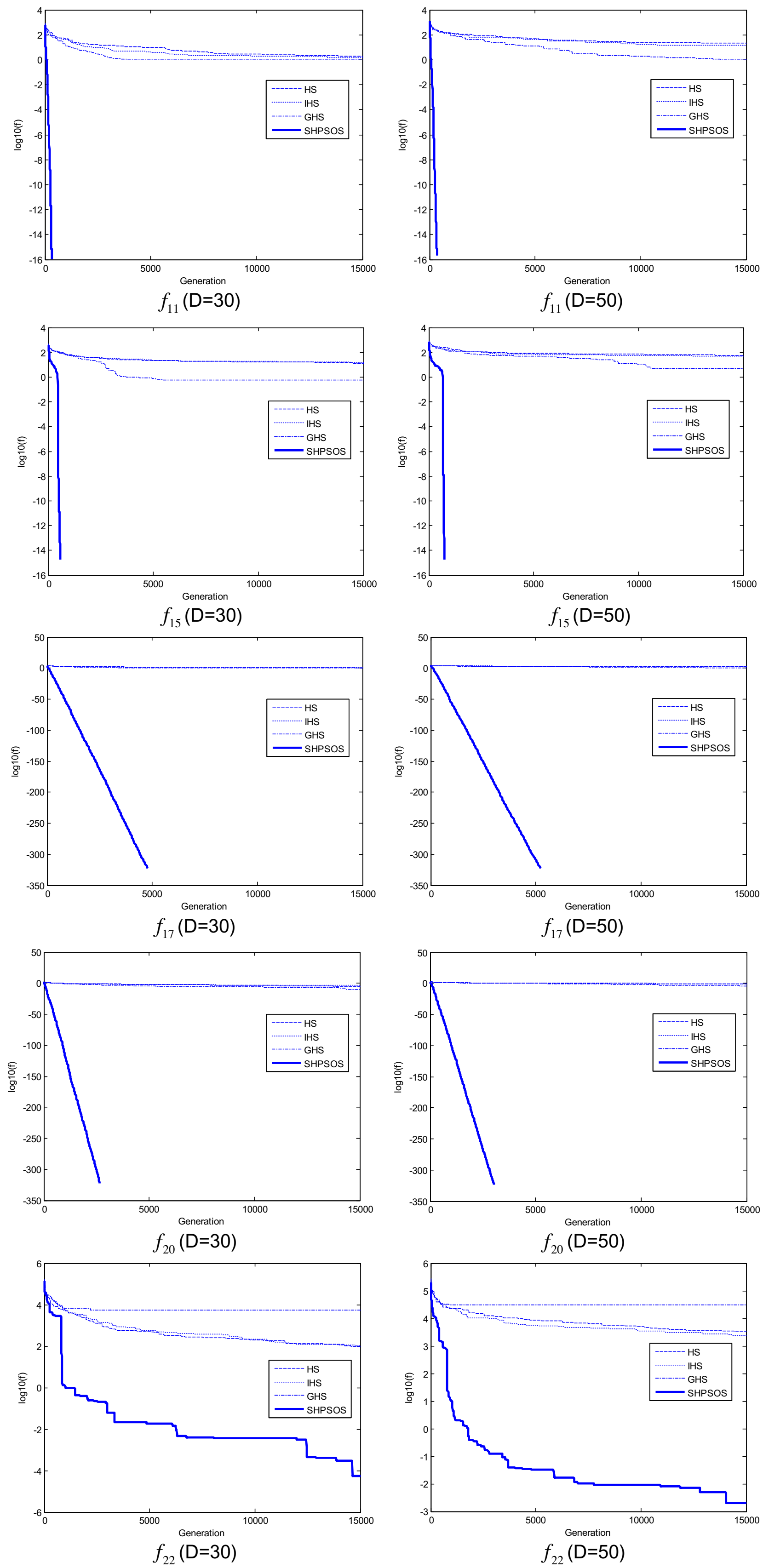


Fig. 2 (*continued*)

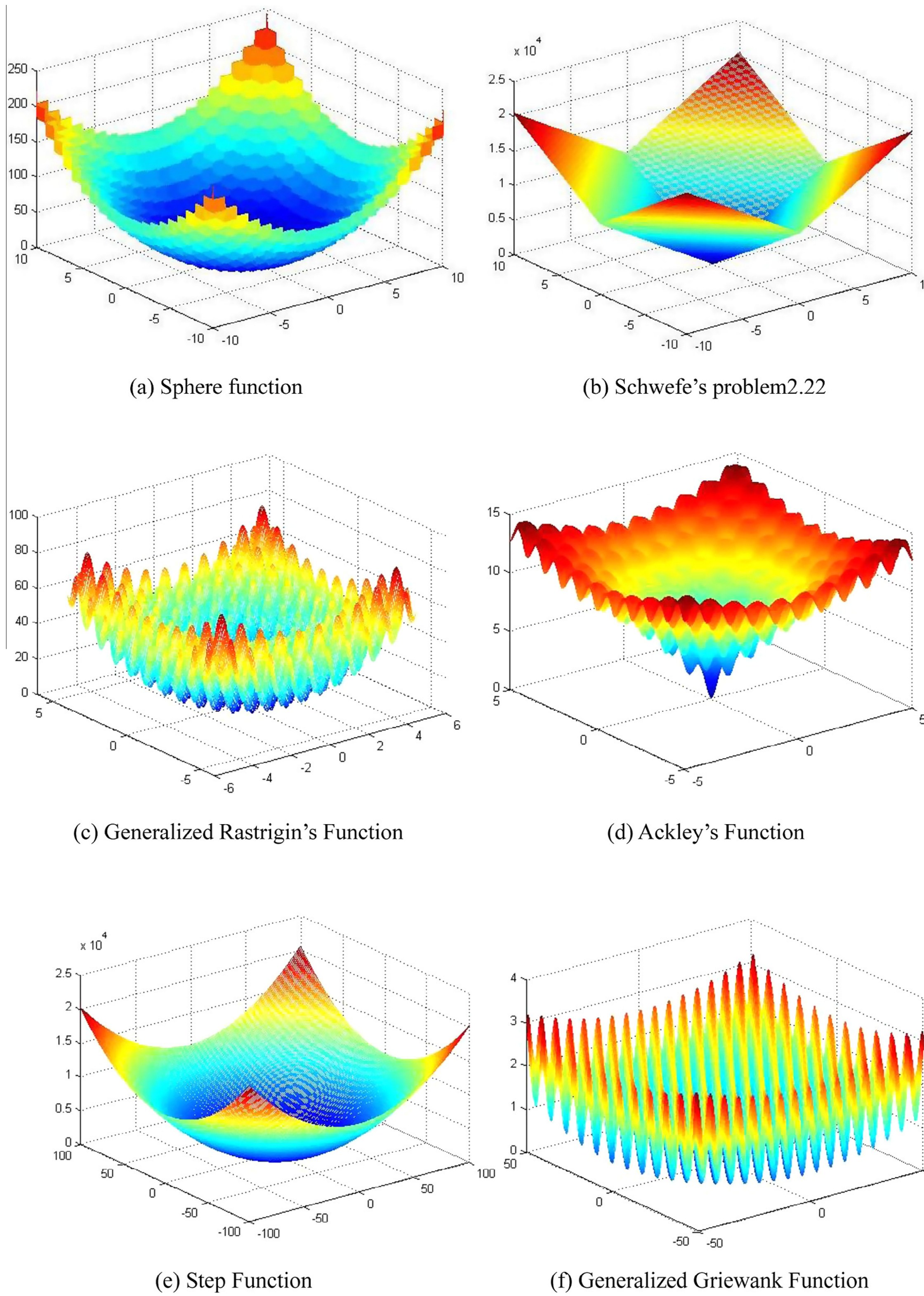
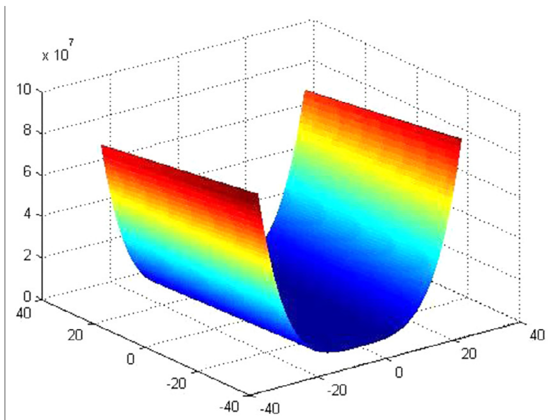


Fig. 3. 3D graphs of some typical benchmark functions.

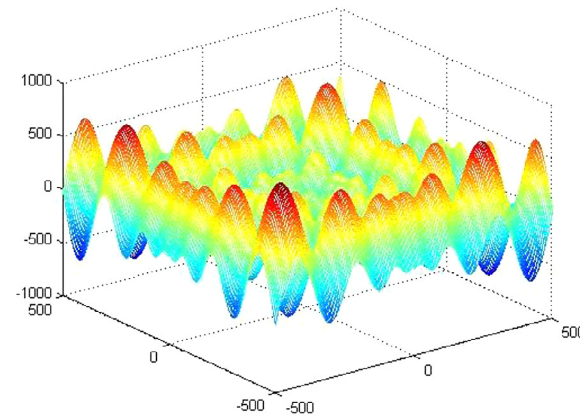
minima, and the number of the minima increases exponentially with the dimension size, so it is easy to sink into local optimum. Functions $f_{22} - f_{27}$ in the third group are shifted functions. Functions $f_{28} - f_{32}$ in the fourth group are either shifted rotated or scalable functions, which are too difficult to achieve better global solutions.

The experiments were implemented on an Intel 2.53 GHZ Core (TM) i3 processor with 2 GB of RAM where SHPSOS algorithm were programmed with MATLAB (2010b) under Win7 (64X).

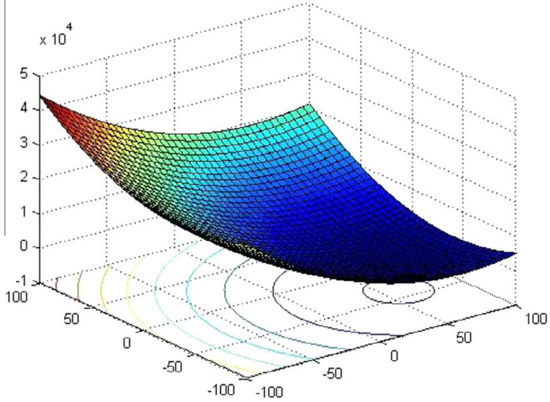
In our experimental study, all the functions are tested in thirty dimensions and fifty dimensions. For SHPSOS algorithm, the maximum number of function evaluations is set to 15,000, the other



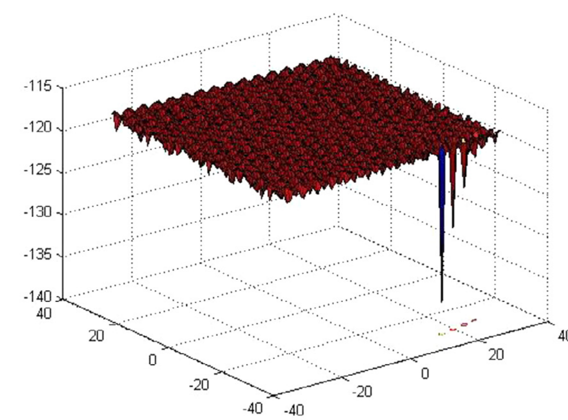
(g) Rosenbrock Function



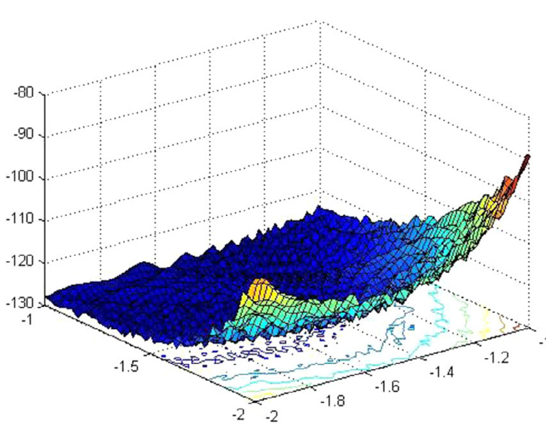
(h) Generalized Schwefel's 2.26 Problem



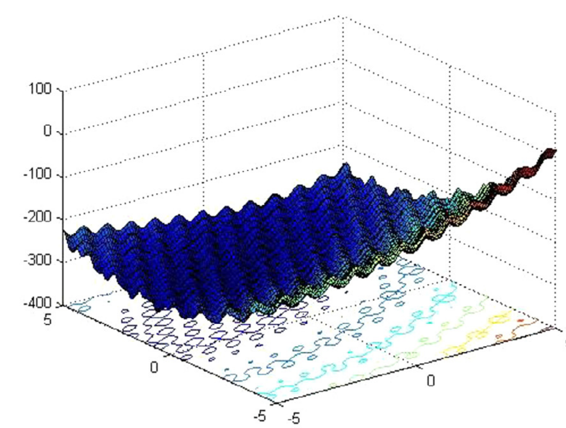
(i) Shifted Sphere Function



(j) Shifted Rotated Ackley's Function



(k) Griewank and Rosenbrock's Function



(l) Shifted Rotated Rastrigin's Function

Fig. 3 (*continued*)

compared algorithms are all set to 50000. In order to ensure the reliability of experimental results, SHPSOS is run thirty times independently for each case and takes best value, the worst values, the mean and standard deviation. This version of MATLAB considers the numbers smaller than 4.9407e−324 as zero. The best solutions (smallest is best) are highlighted in bold.

The 3D graphs of some typical benchmark functions, which provide a visualization of the shapes of the corresponding surfaces, are shown in Fig. 3 and indicated by a letter (a through l). For the meta-heuristics, these typical functions remain a great challenge. With the increasing number of the dimensions, the optimization problems will become more and more

Table 2
The optimization results of HS, IHS, GHS and SHPSOS (D = 30).

Functions	Algorithms	Iter	Best	Worst	Mean	SD
f_1	HS	5e4	3.24e−000	2.21e+001	8.92e−000	4.58e−000
	IHS	5e4	1.94e−000	2.06e+001	6.69e−000	3.43e−000
	GHS	5e4	8.85e−009	4.23e−001	3.24e−002	7.90e−002
	SHPSOS	1.5e4	0	0	**0**	**0**
f_2	HS	5e4	5.00e−002	2.49e−001	1.14e−001	5.39e−002
	IHS	5e4	5.47e−001	6.86e−000	3.31e−000	1.45e−000
	GHS	5e4	2.14e−003	2.07e−001	5.50e−002	5.91e−002
	SHPSOS	1.5e4	0	0	**0**	**0**
f_3	HS	5e4	1.92e+003	8.04e+003	4.05e+003	1.27e+003
	IHS	5e4	2.25e+003	7.79e+003	4.17e+003	1.31e+003
	GHS	5e4	9.66e+000	2.13e+004	5.25e+003	6.63e+003
	SHPSOS	1.5e4	3.27e−001	9.04e+001	**1.80e+001**	**2.11e+001**
f_4	HS	5e4	5.33e−000	9.57e−000	7.08e−000	1.07e−000
	IHS	5e4	5.42e−000	9.68e−000	7.06e−000	1.08e−000
	GHS	5e4	1.67e−001	1.06e+001	4.90e−000	3.42e−000
	SHPSOS	1.5e4	1.26e−015	5.30e−014	**1.26e−014**	**2.26e−014**
f_5	HS	5e4	6.34e+001	6.06e+002	2.12e+002	1.12e+002
	IHS	5e4	9.93e+002	1.30e+004	4.70e+003	2.98e+003
	GHS	5e4	4.12e−001	2.45e+003	1.71e+002	4.75e+002
	SHPSOS	1.5e4	1.34e−001	9.24e+001	**4.15e−001**	**2.76e+001**
f_6	HS	5e4	8.00e−000	2.70e+001	1.98e+001	5.52e−000
	IHS	5e4	6.00e−000	2.10e+001	1.25e+001	3.51e+001
	GHS	5e4	0	0	**0**	**0**
	SHPSOS	1.5e4	0	0	**0**	**0**
f_7	HS	5e4	4.28e−002	1.34e−001	7.66e−002	2.31e−002
	IHS	5e4	3.51e−002	1.80e−001	7.25e−002	3.17e−002
	GHS	5e4	1.68e−004	4.53e−002	1.21e−002	1.04e−002
	SHPSOS	1.5e4	4.58e−004	2.79e−003	**1.51e−003**	**8.62e−004**
f_8	HS	5e4	1.10e+001	4.28e+001	2.29e+001	9.28e−000
	IHS	5e4	1.15e+001	6.26e+001	3.08e+001	1.40e+001
	GHS	5e4	9.04e−007	1.46e−001	3.29e−002	4.29e−002
	SHPSOS	1.5e4	2.52e−026	2.24e−004	**1.26e−024**	**5.06e−025**
f_9	HS	5e4	2.68e−002	3.06e−000	1.08e−000	9.28e−001
	IHS	5e4	2.39e−000	1.29e+001	7.03e−000	2.27e−000
	GHS	5e4	2.70e−005	9.22e−002	7.29e−003	1.80e−002
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{10}	HS	5e4	1.10e−001	2.01e−000	1.06e−000	4.92e−001
	IHS	5e4	4.94e−001	2.01e−000	1.40e−000	3.91e−001
	GHS	5e4	7.17e−004	1.40e−001	2.31e−002	2.91e−002
	SHPSOS	1.5e4	4.44e−015	1.50e−014	**5.87e−015**	**2.18e−015**
f_{11}	HS	5e4	1.04e−000	1.14e−000	1.08e−000	2.53e−002
	IHS	5e4	1.04e−000	1.13e−000	1.08e−000	2.21e−002
	GHS	5e4	9.05e−006	6.10e−001	9.87e−002	1.60e−002
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{12}	HS	5e4	9.46e−004	1.09e−001	1.67e−002	2.69e−002
	IHS	5e4	8.42e−003	1.41e−001	5.03e−002	4.39e−002
	GHS	5e4	1.82e−008	5.22e−003	6.26e−004	1.18e−003
	SHPSOS	1.5e4	2.64e−009	4.03e−008	**1.13e−008**	**1.62e−008**
f_{13}	HS	5e4	1.28e−001	7.94e−001	3.06e−001	1.33e−001
	IHS	5e4	2.04e−001	6.92e−001	3.91e−001	1.20e−001
	GHS	5e4	1.67e−009	6.65e−002	5.38e−003	1.34e−002
	SHPSOS	1.5e4	4.28e−008	4.94e−001	**1.38e−004**	**2.05e−004**
f_{14}	HS	5e4	3.12e−001	4.14e−001	3.85e−001	2.82e−002
	IHS	5e4	2.72e−001	4.41e−001	3.73e−001	4.63e−002
	GHS	5e4	9.71e−003	2.27e−001	5.73e−002	4.50e−002
	SHPSOS	1.5e4	4.88e−003	6.81e−002	**1.03e−002**	**1.32e−003**
f_{15}	HS	5e4	4.54e−002	4.01e−000	9.11e−001	1.05e−000
	IHS	5e4	3.40e−001	4.10e−000	1.81e−000	8.88e−001
	GHS	5e4	5.08e−006	1.60e−001	1.31e−002	3.04e−002
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{16}	HS	5e4	2.36e+001	4.42e+001	3.46e+001	4.53e+000
	IHS	5e4	2.63e+001	4.61e+001	3.66e+001	4.61e+000
	GHS	5e4	2.16e+000	5.73e+001	1.44e+001	1.25e+001
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{17}	HS	5e4	3.22e−003	3.74e−002	8.22e−003	6.46e−003
	IHS	5e4	3.90e−001	2.25e+000	1.29e+000	4.61e−001
	GHS	5e4	3.91e−006	1.87e−002	2.38e−003	4.58e−003

(continued on next page)

Table 2 (*continued*)

Functions	Algorithms	Iter	Best	Worst	Mean	SD
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{18}	HS	5e4	1.33e+006	1.02e+007	4.73e+006	2.17e+006
	IHS	5e4	2.93e+006	1.63e+007	8.38e+006	3.49e+006
	GHS	5e4	9.40e+000	3.36e+005	2.24e+004	6.61e+004
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{19}	HS	5e4	9.43e+002	1.88e+004	5.11e+003	3.89e+003
	IHS	5e4	1.22e+004	4.59e+005	1.24e+005	8.87e+004
	GHS	5e4	8.44e+000	2.07e+004	2.11e+003	4.02e+003
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{20}	HS	5e4	3.22e−008	9.84e−008	5.66e−008	1.62e−008
	IHS	5e4	4.40e−014	7.59e−010	2.54e−011	1.38e−010
	GHS	5e4	1.01e−018	2.13e−010	1.73e−011	4.56e−011
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{21}	HS	5e4	3.57e+000	2.89e+001	1.57e+001	5.85e+000
	IHS	5e4	6.43e+000	4.87e+003	1.12e+003	1.34e+003
	GHS	5e4	2.64e−001	3.50e+003	3.04e+002	6.63e+002
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{22}	HS	5e4	2.56e−000	1.33e+001	7.20e−000	3.05e−000
	IHS	5e4	3.53e−000	1.88e+001	8.04e−000	3.90e−000
	GHS	5e4	1.19e+003	3.45e+003	2.13e+003	5.83e+002
	SHPSOS	1.5e4	6.73e−004	1.18e−003	**1.86e−004**	**2.66e−004**
f_{23}	HS	5e4	3.74e−002	3.03e−000	7.92e−001	7.84e−001
	IHS	5e4	5.87e−002	6.82e−000	2.47e−000	1.58e−000
	GHS	5e4	1.19e+003	3.45e+003	2.13e+003	5.83e+002
	SHPSOS	1.5e4	4.37e−002	1.45e+001	**5.39e−001**	**3.26e−001**
f_{24}	HS	5e4	5.25e−002	1.80e−000	1.01e−000	7.84e−001
	IHS	5e4	2.79e−001	2.25e−000	1.47e−000	4.39e−001
	GHS	5e4	7.15e−000	1.00e+000	8.53e−000	7.13e−001
	SHPSOS	1.5e4	3.37e−004	4.35e−000	**3.91e−002**	**4.55e−002**
f_{25}	HS	5e4	1.04e−000	1.15e−000	1.08e−000	2.75e−002
	IHS	5e4	1.03e−000	1.17e−000	1.08e−000	3.48e−002
	GHS	5e4	7.19e−000	1.45e+001	1.10e+001	2.06e−000
	SHPSOS	1.5e4	5.84e−004	6.87e−002	**1.84e−002**	**2.89e−002**
f_{26}	HS	5e4	3.26e+003	8.66e+003	4.55e+003	1.20e+003
	IHS	5e4	2.45e+003	6.99e+003	4.02e+003	1.15e+003
	GHS	5e4	1.18e+004	3.57e+004	2.19e+004	4.90e+003
	SHPSOS	1.5e4	1.19e+003	1.70e+003	**1.40e−003**	**2.39e+002**
f_{27}	HS	5e4	1.08e+003	1.72e+004	3.94e+003	3.46e+003
	IHS	5e4	1.13e+003	1.43e+004	4.65e+003	3.65e+003
	GHS	5e4	1.26e+007	1.01e+008	3.70e+007	2.16e+007
	SHPSOS	1.5e4	9.63e+000	2.44e+002	**8.80e+001**	**6.60e+001**
f_{28}	HS	5e4	4.98e+001	1.71e+002	1.04e+002	3.15e+001
	IHS	5e4	5.32e+001	1.34e+002	**8.62e+001**	**2.09e+001**
	GHS	5e4	2.34e+002	3.27e+002	2.82e+002	1.99e+001
	SHPSOS	1.5e4	3.46e+002	5.95e+002	4.46e+002	1.02e+002
f_{29}	HS	5e4	2.06e+001	2.11e+001	2.10e+001	8.17e−002
	IHS	5e4	1.03e−001	1.89e+001	5.98e+000	5.74e+000
	GHS	5e4	2.09e+001	2.11e+001	2.10e+001	5.03e−002
	SHPSOS	1.5e4	2.05e+001	2.09e+001	**2.07e+001**	**1.59e−002**
f_{30}	HS	5e4	1.77e+000	5.70e+000	2.95e+000	9.43e−001
	IHS	5e4	1.08e+000	6.46e+000	1.08e+000	3.48e−002
	GHS	5e4	3.36e+001	1.39e+002	6.32e+001	2.42e+001
	SHPSOS	1.5e4	6.05e−003	4.18e+000	**1.50e−002**	**1.32e−002**
f_{31}	HS	5e4	2.32e+000	7.88e+000	3.91e+000	1.26e+000
	IHS	5e4	1.66e+000	4.03e+000	2.34e+000	5.41e−001
	GHS	5e4	7.16e+000	1.39e+001	1.08e+001	1.78e+000
	SHPSOS	1.5e4	7.20e−001	3.49e+000	**1.28e−tO00**	**2.87e−001**
f_{32}	HS	5e4	1.15e+001	1.37e+001	1.31e+001	4.60e−001
	IHS	5e4	1.20e+001	1.37e+001	1.30e+001	4.20e−001
	GHS	5e4	1.32e+001	1.38e+001	1.35e+001	1.43e−001
	SHPSOS	1.5e4	1.05e+001	1.35e+001	**1.10e−001**	**5.31e−002**

The bold values are the optimization values obtained by SHPSOS.

Table 3
The optimization results of HS, IHS, GHS and SHPSOS ($D = 50$).

Functions	Algorithms	Iter	Best	Worst	Mean	SD
f_1	HS	5e4	3.59e+002	8.13e+002	5.63e+002	1.08e+002
	IHS	5e4	3.20e+002	7.88e+002	5.59e+002	1.27e+002
	GHS	5e4	2.26e−003	1.76e+001	2.56e−000	4.08e−000
	SHPSOS	1.5e4	0	0	**0**	**0**
f_2	HS	5e4	7.29e−000	1.14e+001	9.20e−000	1.22e−000
	IHS	5e4	4.25e+001	8.13e+001	6.57e+001	9.63e−000
	GHS	5e4	9.29e−003	1.27e−000	3.81e−001	3.33e−001
	SHPSOS	1.5e4	0	0	**0**	**0**
f_3	HS	5e4	2.05e+004	3.66e+004	2.85e+004	4.26e+003
	IHS	5e4	1.96e+004	4.33e+004	3.07e+004	6.69e+003
	GHS	5e4	1.42e+004	8.94e+004	5.99e+004	2.30e+004
	SHPSOS	1.5e4	1.08e+003	4.60e+003	**2.36e+003**	**1.01e+003**
f_4	HS	5e4	1.83e+001	2.42e+001	2.08e+001	1.31e+001
	IHS	5e4	1.67e+001	2.50e+001	2.15e+001	2.13e+000
	GHS	5e4	2.21e−000	2.10e+001	1.39e+001	4.96e−000
	SHPSOS	1.5e4	6.00e−008	5.16e−006	**1.16e−006**	**2.24e−006**
f_5	HS	5e4	7.13e+003	4.79e+004	2.41e+004	1.04e+004
	IHS	5e4	9.52e+005	4.75e+006	2.58e+006	1.05e+006
	GHS	5e4	2.11e+001	7.86e+003	6.79e+002	1.79e+003
	SHPSOS	1.5e4	4.09e+001	5.08e+002	**l.lle+002**	**1.39e+002**
f_6	HS	5e4	3.37e+002	8.56e+002	5.42e+002	1.42e+002
	IHS	5e4	3.74e+002	7.76e+002	5.56e+002	1.14e+002
	GHS	5e4	0	0	**0**	**0**
	SHPSOS	1.5e4	0	0	**0**	**0**
f_7	HS	5e4	2.35e−001	6.51e−001	3.96e−001	9.84e−002
	IHS	5e4	2.21e−001	8.16e−001	4.73e−001	1.37e−000
	GHS	5e4	1.49e−002	2.13e−001	5.62e−002	3.92e−002
	SHPSOS	1.5e4	8.37e−004	1.23e−002	**5.29e−003**	**4.68e−003**
f_8	HS	5e4	6.21e+002	1.22e+003	8.72e+002	1.56e+002
	IHS	5e4	4.64e+002	1.26e+003	8.96e+002	1.94e+002
	GHS	5e4	4.59e−004	4.70e−000	1.40e−000	1.38e−000
	SHPSOS	1.5e4	4.32e−018	3.35e−004	**1.47e−010**	**7.86e−010**
f_9	HS	5e4	3.04e+001	5.31e+001	3.95e+001	5.32e−000
	IHS	5e4	5.71e+001	9.24e+001	7.58e+001	1.07e+001
	GHS	5e4	2.64e−003	3.88e−000	3.86e−001	7.94e−001
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{10}	HS	5e4	4.35e−000	6.01e−000	5.33e−000	4.50e−001
	IHS	5e4	4.25e−000	6.13e−000	5.18e−000	3.94e−001
	GHS	5e4	1.08e−002	1.30e−000	3.17e−001	3.76e−001
	SHPSOS	1.5e4	5.06e−014	1.15e−014	**3.23e−014**	**8.58e−015**
f_{11}	HS	5e4	3.89e−000	7.50e−000	5.80e−000	9.81e−001
	IHS	5e4	3.80e−000	8.02e−000	5.50e−000	9.88e−001
	GHS	5e4	8.63e−003	1.05e−000	7.17e−001	3.68e−001
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{12}	HS	5e4	5.71e−001	2.48e−000	1.17e−000	4.71e−001
	IHS	5e4	5.16e−001	2.75e−000	1.07e−000	4.97e−001
	GHS	5e4	2.25e−004	6.13e−001	1.57e−001	1.57e−001
	SHPSOS	1.5e4	3.29e−009	6.61e−002	**1.32e−002**	**2.95e−002**
f_{13}	HS	5e4	1.91e+001	7.48e+002	8.69e+001	1.38e+002
	IHS	5e4	2.30e+001	4.31e+002	8.58e+001	1.03e+002
	GHS	5e4	5.09e−004	1.20e−000	2.18e−001	2.68e−001
	SHPSOS	1.5e4	2.57e−008	9.88e−002	**1.97e−002**	**4.42e−002**
f_{14}	HS	5e4	4.53e−001	4.89e−001	4.79e+001	7.69e−003
	IHS	5e4	4.59e−001	4.91e−001	4.78e−001	7.24e−003
	GHS	5e4	3.73e−002	3.12e−001	1.66e−001	7.41e−002
	SHPSOS	1.5e4	4.88e−003	4.07e−002	**2.74e−002**	**1.05e−002**
f_{15}	HS	5e4	2.43e+001	3.51e+001	2.89e+001	2.95e−000
	IHS	5e4	2.68e+001	3.63e+001	3.20e+001	2.14e−000
	GHS	5e4	2.98e−004	5.43e−000	4.44e−001	1.02e−000
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{16}	HS	5e4	1.08e+002	1.48e+002	1.30e+002	9.07e+000
	IHS	5e4	1.04e+002	1.62e+002	1.33e+002	9.02e+000
	GHS	5e4	7.41e+000	9.33e+001	3.61e+001	2.58e+001
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{17}	HS	5e4	6.47e+001	1.55e+002	9.51e+001	2.29e+001
	IHS	5e4	7.45e+001	1.54e+002	1.09e+002	2.06e+001
	GHS	5e4	2.52e−003	2.95e−000	6.26e−001	8.50e−001

(*continued on next page*)

Table 3 (*continued*)

Functions	Algorithms	Iter	Best	Worst	Mean	SD
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{18}	HS	5e4	2.98e+008	5.94e+008	4.50e+008	7.41e+007
	IHS	5e4	2.32e+008	6.21e+008	4.20e+008	1.10e+008
	GHS	5e4	1.34e+003	7.13e+006	1.57e+006	1.99e+006
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{19}	HS	5e4	3.09e+005	2.12e+006	9.27e+005	4.16e+005
	IHS	5e4	5.15e+005	2.38e+006	1.31e+006	5.31e+005
	GHS	5e4	7.92e+000	1.94e+006	2.55e+005	4.11e+005
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{20}	HS	5e4	3.97e−003	2.27e−002	9.79e−003	4.33e−003
	IHS	5e4	7.20e−003	3.23e−002	1.75e−002	7.06e−003
	GHS	5e4	2.56e−013	5.69e−006	8.08e−007	1.39e−006
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{21}	HS	5e4	4.29e+002	1.10e+003	6.77e+002	1.34e+002
	IHS	5e4	6.01e+002	2.48e+004	4.27e+003	4.90e+003
	GHS	5e4	1.51e−002	1.46e+004	2.42e+003	3.75e+003
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{22}	HS	5e4	3.10e+002	7.08e+002	4.98e+002	9.48e+001
	IHS	5e4	3.43e+002	8.34e+002	5.59e+002	1.23e+002
	GHS	5e4	8.84e+003	2.06e+004	1.58e+004	2.61e+003
	SHPSOS	1.5e4	1.47e−003	8.00e−003	**4.15e−003**	**2.37e−003**
f_{23}	HS	5e4	2.85e+001	4.73e+001	3.89e+001	4.81e−000
	IHS	5e4	3.10e+001	5.35e+001	4.28e+001	4.96e−000
	GHS	5e4	1.90e+002	2.73e+002	2.29e+002	2.13e+001
	SHPSOS	1.5e4	1.41e+001	1.91e−002	**2.79e+001**	**1.73e+001**
f_{24}	HS	5e4	4.54e−000	5.94e−000	5.23e−000	3.24e−001
	IHS	5e4	4.36e−000	6.11e−000	5.27e−000	4.16e−001
	GHS	5e4	1.20e+001	1.50e+001	1.39e+001	6.31e−001
	SHPSOS	1.5e4	2.32e−000	5.58e−000	**3.42e−000**	**1.37e−001**
f_{25}	HS	5e4	3.97e−000	8.91e−000	6.61e−000	1.13e−001
	IHS	5e4	4.56e−000	9.19e−000	6.54e−000	1.35e−000
	GHS	5e4	5.41e+001	1.52e+001	1.05e+002	2.22e+001
	SHPSOS	1.5e4	3.05e−003	1.17e−001	**1.65e−002**	**1.75e−002**
f_{26}	HS	5e4	1.71e+004	4.67e+004	3.38e+004	7.35e+003
	IHS	5e4	1.72e+004	4.71e+004	3.28e+004	8.07e+003
	GHS	5e4	5.97e+004	1.35e+005	1.00e+005	1.77e+004
	SHPSOS	1.5e4	4.32e+002	5.93e+003	**1.48e+003**	**1.43e+002**
f_{27}	HS	5e4	7.37e+005	4.51e+006	2.24e+006	8.82e+005
	IHS	5e4	1.27e+006	3.77e+006	2.31e+006	7.32e+005
	GHS	5e4	8.86e+008	2.34e+009	1.56e+009	3.47e+008
	SHPSOS	1.5e4	1.36e+002	3.08e+002	**1.97e+002**	**5.70e−001**
f_{28}	HS	5e4	3.01e+002	5.25e+002	**4.36e+002**	**6.25e+001**
	IHS	5e4	3.73e+002	5.20e+002	4.70e+002	3.29e+001
	GHS	5e4	5.23e+002	6.67e+002	5.93e+002	3.40e+001
	SHPSOS	1.5e4	9.37e+002	1.46e+003	1.24e+003	2.01e+002
f_{29}	HS	5e4	2.10e+001	2.12e+001	2.11e+001	5.42e−002
	IHS	5e4	7.00e+000	1.30e+001	**9.08e+000**	**1.92e+000**
	GHS	5e4	2.11e+001	2.12e+001	2.12e+001	3.10e−002
	SHPSOS	1.5e4	2.06e+001	2.09e+001	2.08e+001	1.39e−002
f_{30}	HS	5e4	2.45e+001	8.17e+001	4.83e+001	1.30e+001
	IHS	5e4	2.51e+001	8.41e+001	4.72e+001	1.58e+001
	GHS	5e4	2.65e+002	8.08e+002	4.99e+002	1.13e+002
	SHPSOS	1.5e4	6.51e−001	1.82e+002	**1.47e−001**	**1.86e+001**
f_{31}	HS	5e4	1.07e+001	2.38e+001	1.42e+001	2.78e+000
	IHS	5e4	7.18e+000	1.45e+001	1.43e+001	1.73e+000
	GHS	5e4	3.36e+001	5.94e+001	4.36e+001	5.68e+000
	SHPSOS	1.5e4	2.63e+000	2.10e+001	**1.43e−001**	**3.57e+000**
f_{32}	HS	5e4	2.24e+001	2.36e+001	2.33e+001	2.23e−001
	IHS	5e4	2.28e+001	2.35e+001	2.32e+001	1.92e−001
	GHS	5e4	2.28e+001	2.37e+001	2.34e+001	2.03e−001
	SHPSOS	1.5e4	2.03e+001	2.34e+001	**2.08e+001**	**1.25e−001**

The bold values are the optimization values obtained by SHPSOS.

Table 4
The optimization results of SGHS, NGHS, AIP_MS and SHPSOS ($D = 30$).

Functions	Algorithms	Iter	Best	Worst	Mean	SD
f_1	SGHS	5e4	3.0687e−10	3.6440e−09	1.6297e−09	8.6970e−10
	NGHS	5e4	5.1518e−16	5.8343e−13	3.6910e−14	1.0503e−13
	AIP_MS	5e4	5.7263e−130	3.0308e−122	3.6204e−123	7.6392e−123
	SHPSOS	1.5e4	0	0	**0**	**0**
f_2	SGHS	5e4	8.5163e−05	5.0352e−01	4.1831e−02	1.0598e−01
	NGHS	5e4	3.4158e−08	2.5321e−06	2.5538e−07	5.0247e−07
	AIP_MS	5e4	8.3871e−82	9.8618e−77	9.0420e−78	2.1160e−77
	SHPSOS	1.5e4	0	0	**0**	**0**
f_5	SGHS	5e4	1.4127e−01	8.7849e+03	5.8959e+02	1.8626e+03
	NGHS	5e4	7.3642e−04	7.9918e+03	7.2139e+02	1.7810e+03
	AIP_MS	5e4	3.3723e−03	2.1935e+01	**2.1130e−00**	**4.7047e−00**
	SHPSOS	1.5e4	1.3430e−01	9.2412e+01	4.1578e+01	2.7644e+01
f_9	SGHS	5e4	9.1138e−08	9.9511e−01	1.6897e−01	3.7611e−01
	NGHS	5e4	2.3093e−14	9.9492e+01	1.3266e+01	3.4399e+01
	AIP_MS	5e4	0	0	**0**	**0**
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{10}	SGHS	5e4	1.5493e−05	3.7779e−05	2.3888e−05	5.5631e−06
	NGHS	5e4	7.6989e−09	1.6170e−01	2.3760e−00	5.4184e−00
	AIP_MS	5e4	0	7.1054e−15	**2.4343e−15**	**1.1363e−15**
	SHPSOS	1.5e4	4.4409e−15	1.5027e−14	5.8700e−15	2.1844e−15
f_{11}	SGHS	5e4	4.3710e−04	1.6278e−01	6.5408e−02	4.0092e−02
	NGHS	5e4	0	2.6130e−01	6.8302e−02	5.8552e−02
	AIP_MS	5e4	0	0	**0**	**0**
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{14}	SGHS	5e4	1.2699e−01	2.7274e−01	2.1850e−01	3.7970e−02
	NGHS	5e4	2.2769e−01	4.9954e−01	3.7484e−01	7.7901e−02
	AIP_MS	5e4	5.5960e−06	1.7224e−02	1.1553e−02	3.8825e−03
	SHPSOS	1.5e4	4.8810e−03	6.8132e−02	**1.0317e−02**	**1.3251e−03**
f_{22}	SGHS	5e4	4.0355e−10	9.9651e−03	3.3217e−04	1.8194e−03
	NGHS	5e4	0	6.3665e−13	**3.1699e−14**	**1.2183e−13**
	AIP_MS	5e4	6.016 1e−03	6.9236e−02	2.1185e−02	1.1560e−02
	SHPSOS	1.5e4	6.7362e−04	1.1821e−03	1.8607e−04	2.6673e−04
f_{23}	SGHS	5e4	1.3059e−07	5.9698e−00	1.4096e−00	1.4940e−00
	NGHS	5e4	0	1.1781e+02	2.4762e+01	2.7917e+01
	AIP_MS	5e4	2.7147e−04	3.0547e−00	**3.8445e−01**	**7.2196e−01**
	SHPSOS	1.5e4	4.3781e−02	1.4501e+01	5.3942e−01	3.2615e−01
f_{24}	SGHS	5e4	1.3818e−05	3.4308e−05	**2.2360e−05**	**4.8064e−06**
	NGHS	5e4	8.2548e−08	1.9367e+01	8.5089e−00	7.1262e−00
	AIP_MS	5e4	5.8265e−02	1.0999e−00	3.0369e−01	2.7880e−01
	SHPSOS	1.5e4	3.3710e−04	4.3589e−00	3.9111e−02	4.5532e−02
f_{25}	SGHS	5e4	8.0206e−03	1.8122e−01	6.2682e−02	4.4721e−02
	NGHS	5e4	7.3960e−03	5.9945e−01	9.6056e−02	1.0605e−01
	AIP_MS	5e4	9.3695e−03	1.4413e−01	4.1634e−02	3.1300e−02
	SHPSOS	1.5e4	5.8412e−04	6.8721e−02	**1.8439e−02**	**2.8910e−02**
f_{26}	SGHS	5e4	1.9538e+01	1.6802e+02	**7.2317e+01**	**3.8188e+01**
	NGHS	5e4	1.1609e+02	7.0737e+02	3.4382e+02	1.8134e+02
	AIP_MS	5e4	1.091 1e+03	5.1254e+03	3.1672e+03	1.1453e+03
	SHPSOS	1.5e4	1.1900e+03	1.7052e+03	1.4041e+03	2.3955e+02
f_{27}	SGHS	5e4	1.3486e+01	1.2081e+04	6.6414e+02	2.2298e+03
	NGHS	5e4	1.9870e−02	1.1956e+04	8.9862e+02	2.9882e+03
	AIP_MS	5e4	4.2259e+01	5.3001e+02	1.0355e+02	9.1995e+01
	SHPSOS	1.5e4	9.6339e+00	2.4407e+02	**8.8037e+01**	**6.6052e+01**
f_{28}	SGHS	5e4	5.9697e+01	2.0795e+02	**1.2012e+02**	**3.5172e+01**
	NGHS	5e4	3.0048e+02	9.4828e+02	5.7227e+02	1.2256e+02
	AIP_MS	5e4	1.6537e+02	2.8972e+02	2.1284e+02	3.8259e+01
	SHPSOS	1.5e4	3.4632e+02	5.9598e+02	4.4628e+02	1.0215e+02
f_{29}	SGHS	5e4	2.6701e−04	2.0925e+01	7.4583e+00	9.2410e+00
	NGHS	5e4	1.9400e+01	1.9961e+01	1.9792e+01	1.1524e−01
	AIP_MS	5e4	2.0178e+00	9.0717e+00	**5.5007e+00**	**2.0312e+00**
	SHPSOS	1.5e4	2.0587e+01	2.0932e+01	2.0710e+01	1.5909e−02
f_{30}	SGHS	5e4	2.3514e−01	9.6579e−01	4.9828e−01	2.2441e−01
	NGHS	5e4	4.0215e−03	2.0101e−01	4.8668e−02	3.8342e−02
	AIP_MS	5e4	1.7366e−01	5.6021e−01	3.2360e−01	8.7042e−02
	SHPSOS	1.5e4	6.0525e−03	4.1806e+00	**1.5089e−02**	**1.3203e−02**
f_{31}	SGHS	5e4	1.6139e+00	2.9676e+00	2.1256e+00	3.4831e−01
	NGHS	5e4	8.5180e−01	2.5556e+00	1.4934e+00	3.9305e−01
	AIP_MS	5e4	9.5564e−01	2.2366e+00	1.4253e+00	3.3879e−01
	SHPSOS	1.5e4	7.2020e−01	3.4964e+00	**1.2866e+00**	**2.8712e−01**

(*continued on next page*)

Table 4 (*continued*)

Functions	Algorithms	Iter	Best	Worst	Mean	SD
f_{32}	SGHS	5e4	1.1436e+01	1.3993e+01	1.3044e+01	5.9219e−01
	NGHS	5e4	1.2947e+01	1.4039e+01	1.3648e+01	3.0831e−01
	AIP_MS	5e4	1.0029e+01	1.3432e+01	1.2604e+01	7.0435e−01
	SHPSOS	1.5e4	1.0532e+01	1.3501e+01	**1.1083e+01**	**5.3108e−02**

The bold values are the optimization values obtained by SHPSOS.

Table 5
The optimization results of SGHS, NGHS, AIP_MS and SHPSOS (D = 50).

Functions	Algorithms	Iter	Best	Worst	Mean	SD
f_1	SGHS	5e4	7.5862e−09	6.0400e−07	5.0688e−08	1.1309e−07
	NGHS	5e4	9.8383e−08	4.8281e−06	5.2782e−07	8.5435e−07
	AIP_MS	5e4	2.1517e−130	2.1715e−123	1.7270e−124	5.2616e−124
	SHPSOS	1.5e4	0	0	**0**	**0**
f_2	SGHS	5e4	8.9269e−02	8.2942e−01	2.9727e−01	1.7518e−01
	NGHS	5e4	8.9361e−04	6.1073e−03	2.6752e−03	1.4763e−03
	AIP_MS	5e4	1.4990e−78	1.8877e−73	8.1485e−75	3.4518e−74
	SHPSOS	1.5e4	0	0	**0**	**0**
f_5	SGHS	5e4	8.8120e+01	8.3891e+03	1.1098e+03	2.0902e+03
	NGHS	5e4	2.8900e−00	5.8845e+03	4.809 1e+02	1.4647e+03
	AIP_MS	5e4	7.6312e−03	1.0092e+02	**1.3659e+01**	**2.5424e+01**
	SHPSOS	1.5e4	4.0897e+01	5.0818e+02	1.1133e+02	1.3906e+02
f_9	SGHS	5e4	9.9496e−01	9.7178e−00	4.4783e−00	2.2448e−00
	NGHS	5e4	3.8720e−05	1.1048e+02	1.7518e+01	3.7689e+01
	AIP_MS	5e4	0	0	**0**	**0**
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{10}	SGHS	5e4	3.6857e−05	8.0018e−05	5.0353e−05	9.4438e−06
	NGHS	5e4	7.6252e−05	1.4449e+01	2.0704e−05	4.7272e−04
	AIP_MS	5e4	0	7.1054e−15	**5.3291e−15**	**2.0334e−15**
	SHPSOS	1.5e4	1.1546e−14	5.0626e−14	3.2389e−14	8.5882e−15
f_{11}	SGHS	5e4	3.0415e−02	1.0462e−00	1.1601e−01	1.8058e−01
	NGHS	5e4	9.8671e−08	1.7505e−01	3.9735e−02	4.7654e−02
	AIP_MS	5e4	0	0	**0**	**0**
	SHPSOS	1.5e4	0	0	**0**	**0**
f_{14}	SGHS	5e4	3.1210e−01	4.4191e−01	3.9111e−01	3.6797e−02
	NGHS	5e4	4.1467e−01	4.9965e−01	4.8181e−01	2.3408e−02
	AIP_MS	5e4	9.7159e−03	7.8189e−02	4.0852e−02	1.6433e−02
	SHPSOS	1.5e4	4.8842e−03	4.0781e−02	**2.7442e−02**	**1.0506e−02**
f_{22}	SGHS	5e4	1.1553e−08	9.4503e−03	3.2044e−04	1.7246e−03
	NGHS	5e4	3.9479e−08	7.5387e−07	**2.0119e−07**	**1.7393e−07**
	AIP_MS	5e4	4.4971e−02	2.9019e−01	1.3584e−01	6.7357e−02
	SHPSOS	1.5e4	1.4792e−03	8.0011e−03	4.1542e−03	2.3706e−03
f_{23}	SGHS	5e4	9.9499e−01	8.9547e−00	4.2868e−00	1.6567e−00
	NGHS	5e4	3.1169e−00	1.4948e+02	6.0165e+01	4.0244e+01
	AIP_MS	5e4	3.3034e−01	6.8566e−00	**2.2488e−00**	**1.4399e−00**
	SHPSOS	1.5e4	1.4128e+02	1.9103e+02	2.7900e+01	1.7345e+01
f_{24}	SGHS	5e4	3.2486e−05	6.5672e−05	**4.7326e−05**	**8.8534e−06**
	NGHS	5e4	3.9579e−05	1.9068e+01	1.4651e+01	5.8037e−00
	AIP_MS	5e4	2.9076e−01	2.4901e−00	1.1696e−00	3.9617e−01
	SHPSOS	1.5e4	3.3290e−00	3.5822e−00	3.4241e−00	1.3789e−01
f_{25}	SGHS	5e4	3.1732e−02	2.0239e−01	8.4691e−02	4.1155e−02
	NGHS	5e4	2.0650e−08	3.3219e−01	3.7817e−02	6.6703e−02
	AIP_MS	5e4	1.0065e−02	1.3788e−01	4.8811e−02	3.5757e−02
	SHPSOS	1.5e4	3.0502e−03	1.1711e−01	**1.6543e−02**	**1.7510e−02**
f_{26}	SGHS	5e4	8.9339e+02	3.1171e+03	1.8990e+03	5.4127e+02
	NGHS	5e4	2.2651e+03	7.9632e+03	4.7277e+03	1.6261e+03
	AIP_MS	5e4	1.0773e+03	2.8183e+03	1.8208e+03	5.9332e+02
	SHPSOS	1.5e4	4.3210e+02	5.932 1e+03	**1.4811e+03**	**1.4301e+02**
f_{27}	SGHS	5e4	8.1019e+01	1.4357e+04	1.1340e+03	2.8072e+03
	NGHS	5e4	1.057 1e−00	1.6311e+04	1.0619e+03	3.6149e+03
	AIP_MS	5e4	1.8808e+02	5.9655e+02	3.0795e+02	9.9375e+01
	SHPSOS	1.5e4	1.3689e+02	3.0841e+02	**1.9704e+02**	**5.7007e+01**
f_{28}	SGHS	5e4	1.4924e+02	3.3331e+02	2.4491e+02	5.8355e+01
	NGHS	5e4	7.3725e+02	1.4586e+03	1.0726e+03	2.0238e+02
	AIP_MS	5e4	8.0244e+01	4.7670e+02	**2.2825e+02**	**1.0678e+01**
	SHPSOS	1.5e4	9.3741e+02	1.4616e+03	1.2450e+03	2.0110e+02

Table 5 (*continued*)

Functions	Algorithms	Iter	Best	Worst	Mean	SD
f_{29}	SGHS	5e4	5.5883e−04	2.0603e+01	1.3270e+01	9.0301e+00
	NGHS	5e4	1.9503e+01	1.9986e+01	1.9821e+01	1.0101e−01
	AIP_MS	5e4	6.7951e+00	1.0741e+01	**8.8373e+00**	**1.0340e+00**
	SHPSOS	1.5e4	2.0644e+01	2.0983e+01	2.0881e+01	1.3951e−02
f_{30}	SGHS	5e4	7.9771e−01	1.7149e+00	1.0508e+00	1.4829e−01
	NGHS	5e4	1.0011e+00	1.0724e+00	1.0428e+00	2.1147e−02
	AIP_MS	5e4	5.1441e−01	1.3652e+00	**1.0341e+00**	**1.6974e−01**
	SHPSOS	1.5e4	6.5138e−01	1.8233e+02	1.4717e+01	1.8657e+01
f_{31}	SGHS	5e4	3.0784e+00	5.7562e+00	4.2185e+00	6.8054e−01
	NGHS	5e4	2.0714e+00	3.6325e+00	2.7260e+00	3.9343e−01
	AIP_MS	5e4	1.0507e+00	4.4003e+00	**2.5200e+00**	**8.3096e−01**
	SHPSOS	1.5e4	2.6327e+00	2.1053e+01	1.4339e+01	3.5736e+00
f_{32}	SGHS	5e4	2.0830e+01	2.3540e+01	2.2563e+01	5.6482e−01
	NGHS	5e4	2.2543e+01	2.3977e+01	2.3340e+01	3.6090e−01
	AIP_MS	5e4	2.1785e+01	2.3222e+01	2.2465e+01	3.1577e−01
	SHPSOS	1.5e4	2.0319e+01	2.3426e+01	**2.0886e+01**	**1.2583e−01**

complicated, which can well test the overall performance of the algorithm.

5.1. Analysis and discussion

To verify the effectiveness of SHPSOS, this paper selects classical variants of HS and other variants which have achieved better results than state-of-the-art HS variants in most of the cases. The associated results reported here are taken directly from the literature (Ashrafi & Dariane, 2013) except for SHPSOS and several benchmark function results. Meanwhile, some representative cases of convergence graphs of HS, IHS, GHS and SHPSOS are also shown in Fig. 2 so as to demonstrate the convergence performance of SHPSOS more clearly. In this paper, four grouping experiments are performed. The results are shown in Tables 2–5. Seen from Table 2, it can be observed that SHSPSO is significantly better than variants of HS on almost all the benchmark functions except for functions f_{28}. For the function f_{28} with $D = 30$, the solution accuracy obtained by SHPSOS is one order smaller than IHS. The optimization results achieved by SHPSOS on benchmark functions $f_{14}, f_{29}, f_{31}, f_{32}$ are a bit better than HS, IHS and GHS. What deserves to be mentioned the most is that SHPSOS can find global optimal values on test functions $f_1, f_2, f_6, f_9, f_{11}, f_{15} - f_{21}$ with $D = 30$, which also have fast convergence speed as shown in Fig. 1. Generally, SHPSOS is much better than compared algorithms on most of the cases.

In order to test the stability and performance of SHPSOS, all the benchmark functions with $D = 50$ are also executed. From Table 3, SHPSOS still performs better than variants of HS on most of the test function. To our surprise, SHPSOS can obtain global optimal values all the same on benchmark functions $f_1, f_2, f_6, f_9, f_{11}, f_{15} - f_{21}$. However, the results acquired by SHPSOS are one or two orders smaller than that of other comparison algorithms. It can be observed that SHPSOS can work efficiently with fewer generations (15,000 times) on large scale, which indicates the stability of SHPSOS and verifies the effectiveness of the mutation strategy. Meanwhile, the convergence speed is greatly accelerated as described in Fig. 1.

A comparison between SHPSOS and other three state-of-the-art variants of HS including SGHS, NGHS and AIP_MS is provided to further verify the performance of SHPSOS. Eighteen representative benchmark functions are used in the experiment. Half of them are either shifted rotated functions or scalable functions, which are very challenging and too difficult to obtain excellent solutions. The experimental results are presented in Tables 4 and 5.

From Table 4, the solutions achieved by SHPSOS on test functions $(f_1, f_2, f_9, f_{11}, f_{14}, f_{25}, f_{27}, f_{31}, f_{30} f_{32})$ are superior to compared algorithms. Among test functions, SHPSOS can find the global optimal values on four functions (f_1, f_2, f_9, f_{11}). For the function f_{10}, the objective values obtained by AIP_MS and SHPSOS are the same order and are far better than that of other two approaches. For the functions f_{24}, f_{26}, f_{28}, SGHS outperforms a bit better than other algorithms. For the function f_{22}, NGHS is the best among them. And the result achieved by SHPSOS is second only to NGHS. In general, numerical results show that SHPSOS is better or similar to other three algorithms on the majority of benchmark functions.

The solutions acquired by SHPSOS with $D = 50$ are reported in Table 5. It can be found that comparison results with $D = 50$ are almost the same as that of the results with $D = 30$ except for test functions f_{30}, f_{31}. For functions f_{30}, f_{31}, the results acquired by AIP_MS are one order smaller than SGHS, NGHS and SHPSOS, which implicitly suggests that there is little obvious difference between the solutions obtained by the other three algorithms. In any case, the SHPSOS exhibits the extremely convergence performance. And the experimental results show that it outperforms conspicuously than the well-known variants of HS in different situations.

6. Conclusions

This paper presents a self-adaptive harmony PSO search (SHPSOS) algorithm to further improve the performance of HS algorithm. In the SHPSOS, by means of using PSO algorithm, an effective initialization scheme is employed with the aim of improving solution quality of the initial harmony memory. Meanwhile, a new self-adaptive adjusting scheme for two control parameters PAR and BW is designed after fully considering the process of the evolution. Then, to further accelerate convergence rate and solution accuracy, a new efficient improvisation scheme based on differential evolution and the best individual in HM is introduced. In addition, Gaussian mutation strategy is embedded in SHPSOS algorithm to enhance the robustness and prevent SHPSOS from trapping into local minima and the global convergence of the SHPSOS algorithm has been also proved theoretically through the Markov model. In order to verify the performance of SHPSOS, thirty-two benchmark functions are used in the experiments. The results show that SHPSOS is more effective and the performance of SHPSOS has been greatly improved. From Tables 2 and 3, it can be observed that SHPSOS with fewer generations is far better than HS, IHS and GHS on almost all the benchmark functions. At the same time, to further testify the performance of SHPSOS, SHPSOS is also compared with other three state-of-the-art variants including SGHS, NGHS and AIP_MS. From Tables 4 and 5, it can be seen that SHPSOS can still work efficiently in most of the cases. So it can be concluded that SHPSOS is an efficient algorithm and developing the SHPSOS algorithm for large-scale scheduling problems is our objective in near future. Meanwhile, it is also a promising direction of applying SHPSOS to other varieties of combination optimization problem in the real world.

However, the superiority of SHPSOS is not very obvious while optimizing the shifted or shifted rotated and scalable functions. The type of functions has a great number of local optima, which make it difficult to obtain better solutions, especially for functions with high dimensions.

Further research will be conducted in following directions. Firstly, we will consist in improving the performance of SHPSOS and testing its efficiency through a comparison with other types of EAs, and then applying the improved SHPSOS to other varieties of combination optimization problems in the application domain. Secondly, we will try to combine SHPSOS with other meta-heuristics for the multi-objective scheduling problems. Thirdly, a novel and efficient neighborhood structures based on SHPSOS in the local search will be performed for the job shop and flow shop scheduling problems with more flexible processing plan under fuzzy constraints involved in application problems. Furthermore, SHPSOS will be also used in the classification such as machine learning, data mining, and other research fields.

Acknowledgement

This work was financially supported by the National Natural Science Foundation of China under Grant No. 51365030. It was also supported by the General and Special Program of the Postdoctoral Science Foundation of China, the Science Foundation for Distinguished Youth Scholars of Lanzhou University of Technology, Lanzhou Science Bureau project under Grant Nos. 2012M521802, 2013T60889, J201405, and 2013–4-64, respectively.

References

Arce, T., Román, P. E., Velásquez, J., & Parada, V. (2014). Identifying web sessions with simulated annealing. *Expert Systems with Applications, 41*, 1593–1600.
Ashrafi, S., & Dariane, A. (2013). Performance evaluation of an improved harmony search algorithm for numerical optimization: Melody Search (MS). *Engineering Applications of Artificial Intelligence, 26*, 1301–1321.
Castelli, M., Silva, S., Manzoni, L., & Vanneschi, L. (2014). Geometric selective harmony search. *Information Sciences, 279*, 468–482.
Chen, J., Pan, Q.-k., & Li, J.-q. (2012). Harmony search algorithm with dynamic control parameters. *Applied Mathematics and Computation, 219*, 592–604.
Chung, S. H., Chan, H. K., & Chan, F. T. S. (2013). A modified genetic algorithm for maximizing handling reliability and recyclability of distribution centers. *Expert Systems with Applications, 40*, 7588–7595.
Costantino, F., Di Gravio, G., Shaban, A., & Tronci, M. (2014). A real-time SPC inventory replenishment system to improve supply chain performances. *Expert Systems with Applications*.
Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (Vol. 1, pp. 39–43). New York, NY.
El-Abd, M. (2013). An improved global-best harmony search algorithm. *Applied Mathematics and Computation, 222*, 94–106.
Fan, Q., & Yan, X. (2014). Self-adaptive differential evolution algorithm with discrete mutation control parameters. *Expert Systems with Applications*.
Gao, W.-f., & Liu, S.-y. (2012). A modified artificial bee colony algorithm. *Computers & Operations Research, 39*, 687–697.
Geem, Z. W. (2007). Optimal scheduling of multiple dam system using harmony search algorithm. In *Computational and Ambient Intelligence* (pp. 316–323). Springer.
Geem, Z. W., Kim, J. H., & Loganathan, G. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*, 60–68.
Geem, Z. W., & Sim, K.-B. (2010). Parameter-setting-free harmony search algorithm. *Applied Mathematics and Computation, 217*, 3881–3889.
Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning, 3*, 95–99.
Hasan, B. H. F., Abu Doush, I., Al Maghayreh, E., Alkhateeb, F., & Hamdan, M. (2014). Hybridizing Harmony Search algorithm with different mutation operators for continuous problems. *Applied Mathematics and Computation, 232*, 1166–1182.
Hosseini, S. D., Akbarpour Shirazi, M., & Karimi, B. (2014). Cross-docking and milk run logistics in a consolidation network: A hybrid of harmony search and simulated annealing approach. *Journal of Manufacturing Systems, 33*, 567–577.
Kadambur, R., & Kotecha, P. (2015). Multi-level production planning in a petrochemical industry using elitist Teaching–Learning-Based-Optimization. *Expert Systems with Applications, 42*, 628–641.
Khalili, M., Kharrat, R., Salahshoor, K., & Sefat, M. H. (2014). Global dynamic harmony search algorithm: GDHS. *Applied Mathematics and Computation, 228*, 195–219.
Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics, 34*, 975–986.
Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. *Journal of Computational Science, 5*, 144–155.
Leung, S. Y. S., Tang, Y., & Wong, W. K. (2012). A hybrid particle swarm optimization and its application in neural networks. *Expert Systems with Applications, 39*, 395–405.
Li, Y., Li, X., & Gupta, J. N. (2014). Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search. *Expert Systems with Applications*.
Liang, J., Qu, B., & Suganthan, P. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory*.
Mahdavi, M., & Abolhassani, H. (2009). Harmony K-means algorithm for document clustering. *Data Mining and Knowledge Discovery, 18*, 370–391.
Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*, 1567–1579.
Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M. N., Salcedo-Sanz, S., et al. (2013). A survey on applications of the harmony search algorithm. *Engineering Applications of Artificial Intelligence, 26*, 1818–1831.
Moh'd Alia, O., & Mandava, R. (2011). The variants of the harmony search algorithm: An overview. *Artificial Intelligence Review, 36*, 49–68.
Moslehi, G., & Khorasanian, D. (2014). A hybrid variable neighborhood search algorithm for solving the limited-buffer permutation flow shop scheduling problem with the makespan criterion. *Computers & Operations Research, 52, Part B*, 260–268.
Mun, S., & Cho, Y.-H. (2012). Modified harmony search optimization for constrained design problems. *Expert Systems with Applications, 39*, 419–423.
Omran, M. G., & Mahdavi, M. (2008). Global-best harmony search. *Applied Mathematics and Computation, 198*, 643–656.
Pan, Q.-K., Suganthan, P. N., Liang, J. J., & Tasgetiren, M. F. (2011). A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem. *Expert Systems with Applications, 38*, 3252–3259.
Pan, Q.-K., Suganthan, P. N., Tasgetiren, M. F., & Liang, J. J. (2010). A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation, 216*, 830–848.
Salcedo-Sanz, S., Manjarres, D., Pastor-Sánchez, Á., Del Ser, J., Portilla-Figueras, J. A., & Gil-Lopez, S. (2013). One-way urban traffic reconfiguration using a multi-objective harmony search approach. *Expert Systems with Applications, 40*, 3341–3350.
Solis, F. J., & Wets, R. J.-B. (1981). Minimization by random search techniques. *Mathematics of Operations Research, 6*, 19–30.
Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. -P., Auger, A., & Tiwari, S. (2005). *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Report, 2005005*.
Valian, E., Tavakoli, S., & Mohanna, S. (2014). An intelligent global harmony search approach to continuous optimization problems. *Applied Mathematics and Computation, 232*, 670–684.
Wang, C.-M., & Huang, Y.-F. (2010). Self-adaptive harmony search algorithm for optimization. *Expert Systems with Applications, 37*, 2826–2837.
Wang, G. G., & Guo, L. H. (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics, 2013*, 1–21.
Wang, L., & Li, L.-p. (2012). A coevolutionary differential evolution with harmony search for reliability–redundancy optimization. *Expert Systems with Applications, 39*, 5271–5278.
Wang, L., Yang, R., Xu, Y., Niu, Q., Pardalos, P. M., & Fei, M. (2013). An improved adaptive binary harmony search algorithm. *Information Sciences, 232*, 58–87.
Wang, X., & Yan, X. (2013). Global best harmony search algorithm with control parameters co-evolution based on PSO and its application to constrained optimal problems. *Applied Mathematics and Computation, 219*, 10059–10072.
Xiang, W.-l., An, M.-q., Li, Y.-z., He, R.-c., & Zhang, J.-f. (2014). An improved global-best harmony search algorithm for faster optimization. *Expert Systems with Applications, 41*, 5788–5803.
Xiao, J., Ao, X.-T., & Tang, Y. (2013). Solving software project scheduling problems with ant colony optimization. *Computers & Operations Research, 40*, 33–46.
Xiao, M., Shao, X., Gao, L., & Luo, Z. (2014). A new methodology for multi-objective multidisciplinary design optimization problems based on game theory. *Expert Systems with Applications*.
Yadav, P., Kumar, R., Panda, S. K., & Chang, C. (2012). An intelligent tuned harmony search algorithm for optimisation. *Information Sciences, 196*, 47–72.
Yoo, D. G., Kim, J. H., & Geem, Z. W. (2014). Overview of Harmony Search algorithm and its applications in civil engineering. *Evolutionary Intelligence, 7*, 3–16.
Zhao, F., Jiang, X., Zhang, C., & Wang, J. (2014a). A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem. *International Journal of Computer Integrated Manufacturing*, 1–16.

Zhao, F., Tang, J., Wang, J., & Jonrinaldi (2014b). An improved particle swarm optimization with decline disturbance index (DDPSO) for multi-objective job-shop scheduling problem. *Computers & Operations Research, 45*, 38–50.

Zhao, F., Tang, J., Wang, J., & Jonrinaldi, J. (2014c). An improved particle swarm optimisation with a linearly decreasing disturbance term for flow shop scheduling with limited buffers. *International Journal of Computer Integrated Manufacturing, 27*, 488–499.

Zhao, F., Zhang, J., Wang, J., & Zhang, C. (2014d). A shuffled complex evolution algorithm with opposition-based learning for a permutation flow shop scheduling problem. *International Journal of Computer Integrated Manufacturing*, 1–16.

Zhao, F., Zhang, J., Zhang, C., & Wang, J. (2015). An improved shuffled complex evolution algorithm with sequence mapping mechanism for job shop scheduling problems. *Expert Systems with Applications, 42*, 3953–3966.

Zou, D., Gao, L., Li, S., & Wu, J. (2011). An effective global harmony search algorithm for reliability problems. *Expert Systems with Applications, 38*, 4642–4648.

Zou, D., Gao, L., Wu, J., Li, S., & Li, Y. (2010). A novel global harmony search algorithm for reliability problems. *Computers & Industrial Engineering, 58*, 307–316.