

## Research Article

# A Novel Harmony Search Algorithm Based on Teaching-Learning Strategies for 0-1 Knapsack Problems

Shouheng Tuo, Longquan Yong, and Fang'an Deng

*School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong 723001, China*

Correspondence should be addressed to Shouheng Tuo; [tuo\\_sh@126.com](mailto:tuo_sh@126.com)

Received 17 August 2013; Accepted 17 September 2013; Published 8 January 2014

Academic Editors: R. Alex and Z. Cui

Copyright © 2014 Shouheng Tuo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To enhance the performance of harmony search (HS) algorithm on solving the discrete optimization problems, this paper proposes a novel harmony search algorithm based on teaching-learning (HSTL) strategies to solve 0-1 knapsack problems. In the HSTL algorithm, firstly, a method is presented to adjust dimension dynamically for selected harmony vector in optimization procedure. In addition, four strategies (harmony memory consideration, teaching-learning strategy, local pitch adjusting, and random mutation) are employed to improve the performance of HS algorithm. Another improvement in HSTL method is that the dynamic strategies are adopted to change the parameters, which maintains the proper balance effectively between global exploration power and local exploitation power. Finally, simulation experiments with 13 knapsack problems show that the HSTL algorithm can be an efficient alternative for solving 0-1 knapsack problems.

## 1. Introduction

Harmony search (HS) [1, 2] is a new population-based meta-heuristic optimization algorithm. It has received much attention regarding its application potential as continuous and discrete optimal problem. Inspired by the process of the musicians' improvisation of the harmony, the HS algorithm improvises its instruments' pitches searching for a perfect state of harmony. The effort to find a new harmony in music is analogous to finding a better solution in an optimization process. HS has been applied to optimization problems in different areas [3–11]. The HS algorithm has powerful exploration ability in a reasonable time but is not good at performing a local search. In order to improve the performance of the harmony search method, several variants of HS have been proposed [12–20]. These variants have some improvement on continuous optimization problems. However, their effectiveness in dealing with discrete problems is still unsatisfactory.

The knapsack problem is one of the classical combinatorial optimization problems. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items. The knapsack problem often applies to resource allocation

where there are financial constraints and is studied in fields such as combinatorics, computer science, complexity theory, cryptography, and applied mathematics.

The 0-1 knapsack problem is as follows. Given a set of  $D$  items and a knapsack, with

$$\begin{aligned} p_j &= \text{profit of item } j, \\ w_j &= \text{weight of item } j, \\ c_j &= \text{volume of item } j, \\ W &= \text{weight capacity of knapsack,} \\ V &= \text{volume capacity of knapsack,} \end{aligned} \quad (1)$$

select some items so that the total profit of the selected items is maximum, and the total weight and the total volumes of selected items are not more than the weight capacity and the volume capacity of the knapsack. Formally,

$$\text{maximize } z = \sum_{j=1}^D p_j x_j, \quad (2)$$

subject to

$$\sum_{j=1}^D w_j x_j \leq W, \quad (3)$$

$$\sum_{j=1}^D c_j x_j \leq V,$$

where

$$x_j = \begin{cases} 1, & \text{if item } j \text{ is assigned to knapsack} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Many methods have been employed to solve 0-1 knapsack problems. Zou et al. proposed a novel global harmony search algorithm (NGHS) for 0-1 knapsack problems [21]. Y. Liu and C. Liu presented an evolutionary algorithm to solve 0-1 knapsack problems [22]. Shi used an improved ant colony algorithm to solve 0-1 knapsack problems [23]. Lin solved the knapsack problems with imprecise weight coefficients by using genetic algorithm [24]. Boyer et al. solved knapsack problems on GPU [25]. Hill et al. proposed a heuristic method for 0-1 multidimensional knapsack problems [26]. Gherboudj et al. propose a discrete binary cuckoo search (BCS) algorithm in order to deal with binary optimisation problems [27]. A novel quantum inspired cuckoo search for knapsack problems is present in the literature [28].

In recent years, more and more discrete optimization problems are solved by HS method. To some extent, this is due to the memory consideration rule that is appropriate to be employed to resolve the discrete optimization problems. However, for a high-dimensional discrete optimization problem, classical HS algorithm can be easy to cause premature convergence and stagnation behavior. Therefore, we present a dynamic parameters-adjustment mechanism for solving the high-dimensional 0-1 knapsack problems. To enhance the performance of dealing with discrete problems by HS method, this paper proposed an improved HS algorithm based on teaching-learning (HSTL) strategies.

The rest of the paper is organized as follows. Section 2 introduces the classical HS algorithm and three state-of-the-art variants of HS. The teaching-learning-based optimization (TLBO) algorithm and the proposed approach (HSTL) are introduced in Section 3. Section 4 presents related constraint-handling technique and integer processing method. Experimental results are reported in Section 5. Finally, Section 6 concludes this paper.

## 2. HS Algorithm and Other Variants

In this section, we introduce the classical HS algorithm and three state-of-the-art variants of HS algorithms: NGHS algorithm [17], intelligent tuned harmony search algorithm (ITHS) [18], and exploratory power of harmony search algorithm (EHS) [19].

*2.1. Classical Harmony Search Algorithm (HS).* Classical harmony search (HS) is derivative-free meta-heuristic algorithm. It mimics the improvisation process of music players

and uses three rules (memory consideration, pitch adjustments, and randomization) to optimize the harmony memories. The steps in the procedure of classical harmony search algorithm are as follows.

*Step 1* (initialize the harmony memory). The harmony memory (HM) consists of HMS harmony. Each harmony is generated from a uniform distribution in the feasible space, as

$$x_i^j = x_i^L + \text{rand}() \cdot (x_i^U - x_i^L), \quad (5)$$

$$i = 1, 2, \dots, D; \quad j = 1, 2, \dots, \text{HMS},$$

where  $\text{rand}()$  is a uniform distribution random number between 0 and 1.

Consider the following:

$$\text{HM} = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^{\text{HMS}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_D^1 \\ x_1^2 & x_2^2 & \dots & x_D^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \dots & x_D^{\text{HMS}} \end{bmatrix}. \quad (6)$$

*Step 2* (improvise a new harmony via three rules). Improvise a new harmony  $X^{\text{new}}$  via three rules: memory consideration, pitch adjustment, and random generation.

(a) *Memory Consideration.* Decision variable value of the new harmony will be generated by choosing from the harmony memory with probability HMCR.

(b) *Pitch Adjustment.* Get a component randomly from an adjacent value of one decision variable of a harmony vector with probability PAR.

(c) *Random Generation.* Generate a component randomly in the feasible region with probability 1-HMCR.

The improvisation procedure of a new harmony works as Algorithm 1.

A new potential variation (or an offspring) is generated in Step 2, which is equivalent to mutation and crossover operator in standard Evolution Algorithms (EAs).

*Step 3* (update the worst harmony). Consider the following:

If  $X^{\text{new}}$  is better than  $X^{\text{worst}}$

$$X^{\text{new}} = X^{\text{worst}}, \quad (7)$$

Endif,

where  $X^{\text{worst}} = (x_1^{\text{worst}}, x_2^{\text{worst}}, \dots, x_D^{\text{worst}})$  denotes the worst harmony in HM.

*Step 4* (check stopping criterion). If the stopping criterion (maximum function evaluation times: MaxFEs) is satisfied, computation is terminated. Otherwise, Step 2 and Step 3 are repeated.

```

For  $i = 1$  to  $D$ 
  If  $\text{rand}() < \text{HMCR}$ 
     $x_i^{\text{new}} = x_i^j, j \in U\{1, 2, \dots, \text{HMS}\}$ 
    If  $\text{rand}() < \text{PAR}$ 
       $x_i^{\text{new}} = x_i^{\text{new}} \pm \text{rand}() \times \text{BW}(i)$ 
       $x_i^{\text{new}} = \min(\max(x_i^{\text{new}}, x_i^L), x_i^U)$ 
    Endif
  Else
     $x_i^{\text{new}} = x_i^L + (x_i^U - x_i^L) \times \text{rand}()$ 
  Endif
EndFor

```

ALGORITHM 1: The improvisation procedure of new harmony by classical HS.

```

For  $i = 1$  to  $D$ 
   $x_i^r = 2 \times x_i^{\text{best}} - x_i^{\text{worst}}$ 
   $x_i^r = \min(\max(x_i^r, x_i^L), x_i^U)$ 
   $x_i^{\text{new}} = x_i^{\text{worst}} + \text{rand}() \times (x_i^r - x_i^{\text{worst}})$ 
  If  $\text{rand}() \leq p_m$  %random mutation
     $x_i^{\text{new}} = x_i^L + (x_i^U - x_i^L) \times \text{rand}()$ 
  Endif
EndFor

```

ALGORITHM 2: The improvisation procedure of new harmony by NGHS.

2.2. *The NGHS Algorithm.* In NGHS algorithm, three significant parameters, harmony memory considering rate (HMCR), bandwidth (BW), and pitch adjusting rate (PAR), are excluded from NGHS, and a random select rate ( $p_m$ ) is included in the NGHS. In Step 3, NGHS works as Algorithm 2, where  $X^{\text{best}} = (x_1^{\text{best}}, x_2^{\text{best}}, \dots, x_D^{\text{best}})$  and  $X^{\text{worst}} = (x_1^{\text{worst}}, x_2^{\text{worst}}, \dots, x_D^{\text{worst}})$  denote the best harmony and the worst harmony in HM, respectively. We set parameter  $p_m = 2/D$  for the 0-1 knapsack problem and  $p_m \in [0.005, 0.1]$  for continuous optimal problem.

2.3. *The EHS Algorithm.* The EHS algorithm uses the same structure with the classical HS algorithm and does not introduce any complex operations. The only difference between HS and EHS is that the EHS algorithm proposed a new scheme of tuning BW, and it works as follows (proportional to the current population variance):

$$\text{BW} = k \sqrt{\frac{1}{\text{HMS}} \sum_{i=1}^{\text{HMS}} (x_i - \bar{x})^2}, \quad (8)$$

where  $k$  is the proportionality constant. If the value of  $k$  is high, the population variance will maintain or increase, and thus the global exploration power of algorithm will enhance. While if the value of  $k$  is low, the population variance will decrease and the local exploitation performance will increase. The experimental results tested by Das et al. [19] have shown that keeping  $k$  around 1.2 provided reasonably accurate results.

The EHS algorithm can enhance the exploratory power, and can provide a better balance between diversification and intensification. However, the exploratory power of EHS leads to the slower convergence [18], and the computational time of BW is greatly increased.

2.4. *The ITHS Algorithm.* The ITHS algorithm [18] proposed a subpopulation formation technique. The population is divided into two groups: Group A and Group B, which is carried out based on  $\bar{\text{fit}}$  (the mean objective function value of all harmony vectors in HM). The harmony vectors, whose objective function value is less than  $\bar{\text{fit}}$ , belong to Group A, and the rest belong to Group B. The improvisation procedure of new harmony by ITHS is shown in Algorithm 3. In ITHS algorithm, the parameter PAR is updated with the number of iterations:

$$\text{PAR} = \text{PAR}_{\text{max}} - \frac{(\text{PAR}_{\text{max}} - \text{PAR}_{\text{min}}) \times t}{T_{\text{max}}}, \quad (9)$$

where  $t$  and  $T_{\text{max}}$  are the current iterations and the maximum iterations.

### 3. HSTL Algorithm

In this section, we proposed a novel harmony search with teaching-learning strategy which derived from Teaching-Learning-Based Optimization (TLBO) algorithm. Above all, the TLBO algorithm is introduced and analyzed, and then we

```

For  $i = 1$  to  $D$  do
  If  $\text{rand}() < \text{HMCR}$ 
     $x_i^{\text{new}} = x_i^j, j \in U \{1, 2, \dots, \text{HMS}\}$ 
    If  $\text{rand}() < \text{PAR}$ 
      If  $\text{fit}(x^j) \leq \overline{\text{fit}}$  //Group A
        If  $\text{rand}() < 0.5$ 
           $y_i = x_i^{\text{best}} - \text{rand}() \times (x_i^{\text{best}} - x_i^{\text{new}})$ 
        Else
           $y_i = x_i^{\text{best}} - \text{rand}() \times (x_i^{\text{worst}} + x_i^{\text{new}})$ 
        Endif
      Else //Group B
         $x_m^{\text{best}} = x_m^{\text{best}} \times (x_i^U / x_m^U), m \in U \{1, 2, \dots, \text{HMS}\}$ 
         $y_i = x_i^{\text{new}} + \text{rand}() \times (x_m^{\text{best}} - x_i^{\text{new}})$ 
      Endif
       $x_i^{\text{new}} = \min(\max(y_i, x_i^L), x_i^U)$ 
    Endif
  Else
     $x_i^{\text{new}} = x_i^L + (x_i^U - x_i^L) \times \text{rand}()$ 
  Endif
EndFor

```

ALGORITHM 3: The improvisation procedure of new harmony by ITHS.

focus on the details of HSTL algorithm and the strategies of dynamically adjusting the parameters.

Since its origination, HS algorithm has been applied to many practical optimization problems. However, for large scale optimization problems, HS has slow convergence and low precision, which is because a new decision variable value in HM can be generated only by pitch adjustment and randomization strategies during the search process, the memory consideration rule is only used to adjust the decision variable values according to the current HM. HS can maintain a strong exploration power in the early stage, but it does not have a good exploitation in the later stage, and thus it is characterized by earlier mature and slow convergence. Therefore, for solving large scale optimization problem, the key is how to balance between global exploration performance and local exploitation ability.

**3.1. Dimension Reduction Adjustment Strategy.** As we know, for a complex optimization problem, its optimization may experience a process from extensive exploration in a large range to fine adjustment in a small range. For a  $D$ -dimensional optimization problem, we assume that its optimal solution is  $X^* = (x_1^*, x_2^*, \dots, x_D^*)$ . Let initial HM be as follows:

$$\text{HM} = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^{\text{HMS}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \cdots & x_D^{\text{HMS}} \end{bmatrix}. \quad (10)$$

After several iterations, the HM turns into HM2. It can be seen from HM2 that each solution  $X^j$  ( $j = 1, 2, \dots, \text{HMS}$ )

in HM2 has nearly achieved the best solution except for one dimension ( $y_i^j, j = 1, 2, \dots, \text{HMS}; i = 1, 2, \dots, D$ ):

$$\text{HM2} = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^{\text{HMS}} \end{bmatrix} = \begin{bmatrix} y_1^1 & x_2^* & \cdots & x_D^* \\ x_1^* & y_2^2 & \cdots & x_D^* \\ \vdots & \vdots & \vdots & \vdots \\ x_1^* & x_2^* & \cdots & y_D^{\text{HMS}} \end{bmatrix}. \quad (11)$$

Then we assume that only the harmony memory consideration rule of HS algorithm is employed to optimize the HM2. In the following, we employ the two methods to generate two new harmonies:  $X^{\text{new1}} = (x_1^{\text{new1}}, x_2^{\text{new1}}, \dots, x_D^{\text{new1}})$  and  $X^{\text{new2}} = (x_1^{\text{new2}}, x_2^{\text{new2}}, \dots, x_D^{\text{new2}})$ , respectively, and then analyze which method is better.

**Method 1.** Generate the solution  $X^{\text{new1}}$  on each variable  $x_i^{\text{new1}}$  ( $i = 1, 2, \dots, D$ ) by using the harmony memory consideration rule, as in Algorithm 4.

**Method 2.** Let  $X^{\text{new2}} = X^{\text{worst}}$  and then adjust one of the variables of the new solution  $X^{\text{new2}}$  by using the harmony memory consideration rule, as in Algorithm 5.

In the following, we analyze the two methods.

In Method 1, all of the decision variables of harmony  $X^{\text{new1}}$  are chosen from HM2. If  $X^{\text{new1}}$  hopes to be the optimal solution  $X^* = (x_1^*, x_2^*, \dots, x_D^*)$ , we should choose  $x_i^{\text{new1}} = x_i^*, i = 1, 2, \dots, D$ . In other words,  $y_i^j$  ( $j = 1, 2, \dots, \text{HMS}; i = 1, 2, \dots, D$ ) in HM2 cannot be chosen as

```

For  $i = 1$  to  $D$ 
  If  $\text{rand}() < \text{HMCR}$ 
     $x_i^{\text{new1}} = x_i^j, j \in U \{1, 2, \dots, \text{HMS}\}$ 
  EndIf
EndFor
    
```

ALGORITHM 4

```

 $X^{\text{new2}} = X^{\text{worst}}$ 
 $i = \text{round}(\text{rand}() * D)$  //Generating a random integer between 1 and D.
 $x_i^{\text{new2}} = x_i^j, j \in U \{1, 2, \dots, \text{HMS}\}$ 
    
```

ALGORITHM 5

$x_i^{\text{new1}}$  in the latter iteration. We can see that the probability  $x_i^{\text{new1}}$  that turns into  $x_i^*$  is  $\text{HMS} - 1/\text{HMS}$  in HM2, so the probability  $X^{\text{new1}}$  turns into  $X^* = (x_1^*, x_2^*, \dots, x_D^*)$  is  $((\text{HMS} - 1)/\text{HMS})^D$ .

In Method 2, assume that the  $X^{\text{new2}} = X^{\text{worst}} \triangleq (y_1^1, x_2^*, \dots, x_D^*)$ , during each iteration, and only one decision variable of  $X^{\text{new2}}$  needs to be adjusted with harmony memory consideration rule. So the probability that decision variable  $y_1^1$  is chosen to adjust is  $1/D$ , and then the probability that  $y_1^1$  will be replaced with  $x_1^*$  is  $\text{HMS} - 1/\text{HMS}$ . Therefore, in one iteration, the probability that  $X^{\text{new2}}$  turns into  $X^* = (x_1^*, x_2^*, \dots, x_D^*)$  by Method 2 is  $1/D \cdot \text{HMS} - 1/\text{HMS}$ .

Next, we compare the success rate between Method 1 and Method 2 at different dimensions.

When  $\text{HMS} = 10, D = 10$ , the success rate of Method 1 is

$$P \{X^{\text{new1}} \rightarrow X^*\} = \left(\frac{\text{HMS} - 1}{\text{HMS}}\right)^D = \left(\frac{10 - 1}{10}\right)^{10} = 0.34867844, \tag{12}$$

and the success rate of Method 2 is

$$P \{X^{\text{new2}} \rightarrow X^*\} = \frac{1}{D} \cdot \frac{\text{HMS} - 1}{\text{HMS}} = \frac{1}{10} \cdot \frac{10 - 1}{10} = 0.09. \tag{13}$$

Apparently, under this condition, Method 1 is superior to Method 2. However, if we set  $\text{HMS} = 10, D = 100$ , the success rate of Method 1 is

$$P \{X^{\text{new1}} \rightarrow X^*\} = \left(\frac{\text{HMS} - 1}{\text{HMS}}\right)^D = \left(\frac{100 - 1}{100}\right)^{100} = 2.65614E - 05, \tag{14}$$

while the success rate of Method 2 is

$$P \{X^{\text{new2}} \rightarrow X^*\} = \frac{1}{D} \cdot \frac{\text{HMS} - 1}{\text{HMS}} = \frac{1}{100} \cdot \frac{100 - 1}{100} = 9.0E - 03. \tag{15}$$

For different  $D$ , a more detailed success rate of Method 1 and Method 2 is listed in Table 1.

From Table 1, it can be seen that, for low dimensional problem ( $D < 50$ ), success rate of Method 1 is greater than that of Method 2; however, Method 2 has higher success rate than Method 1 when  $D \geq 50$ , and with the increasing of dimensionality, the success rate of Method 1 will drop dramatically, but Method 2 can maintain a higher success rate.

By above idea, in the beginning stages, exploring is on a wider domain so as to search fast, and in the later stages, the search focus on a small space to improve the accuracy of solution. Aiming at the HS algorithm, we design a dynamic dimension selection strategy to adjust some selected decision variables. A simple process is shown in Figure 1.

In Figure 1, the parameter TP represents the tune probability of each decision variable. TP changes from  $\text{TP}_{\text{max}}$  to  $\text{TP}_{\text{min}}$ . In other words, each decision variable of the objective harmony vector  $X^{\text{new}}$  will be tuned with probability TP which decreases from  $\text{TP}_{\text{min}}$  to  $\text{TP}_{\text{min}}$  with the increase of iterations.

**3.2. The TLBO Algorithm.** Teaching-Learning-Based Optimization (TLBO) algorithm [29–38] is a new nature-inspired algorithm; it mimics the teaching process of teacher and learning process among learners in a class. TLBO shows a better performance with less computational effort for large scale problems [31]. In addition, TLBO needs very few parameters.

In the TLBO method, the task of a teacher is to try to increase mean knowledge of all learners of the class in the subject taught by him or her depending on his or her capability. Learners make efforts to increase their knowledge by interaction among themselves. A learner is considered as a solution or a vector, different design variables of a vector will be analogous to different subjects offered to learners, and the learners' result is analogous to the "fitness" as in other population-based optimization techniques. The teacher is considered as the best solution obtained so far. The process of TLBO is divided into two phases, "Teacher Phase" and "Learner Phase."

**3.2.1. Teacher Phase.** Assume that there are  $D$  number of subjects (i.e., design variables) and  $NP$  number of learners (i.e., population size);  $X^{\text{teacher}} = (x_1^{\text{teacher}}, x_2^{\text{teacher}}, \dots, x_D^{\text{teacher}})$

TABLE 1: Success rate of Methods 1 and 2.

HMS	D	The success rate	
		Method 1	Method 2
10	10	3.48678E-01	9.00000E-02
10	20	1.21577E-01	4.50000E-02
10	30	4.23912E-02	3.00000E-02
10	40	1.47809E-02	2.25000E-02
10	50	5.15378E-03	1.80000E-02
10	60	1.79701E-03	1.50000E-02
10	70	6.26579E-04	1.28571E-02
10	80	2.18475E-04	1.12500E-02
10	90	7.61773E-05	1.00000E-02
10	100	2.65614E-05	9.00000E-03
10	110	9.26139E-06	8.18182E-03
10	120	3.22925E-06	7.50000E-03
10	130	1.12597E-06	6.92308E-03
10	140	3.92601E-07	6.42857E-03
10	150	1.36891E-07	6.00000E-03
10	160	4.77311E-08	5.62500E-03
10	170	1.66428E-08	5.29412E-03
10	180	5.80299E-09	5.00000E-03
10	190	2.02338E-09	4.73684E-03
10	200	7.05508E-10	4.50000E-03
10	300	1.87393E-14	3.00000E-03
10	400	4.97741E-19	2.25000E-03
10	500	1.32207E-23	1.80000E-03
10	600	3.51161E-28	1.50000E-03
10	700	9.32731E-33	1.28571E-03
10	800	2.47747E-37	1.12500E-03
10	900	6.58049E-42	1.00000E-03
10	1000	1.74787E-46	9.00000E-04

is the best learner (i.e., teacher). For each learner  $X^j = (x_1^j, x_2^j, \dots, x_D^j)$ , the works of teaching are as follows:

$$x_i^{j,\text{new}} = x_i^{j,\text{old}} + \text{rand}() \times (x_i^{\text{teacher}} - T_F \times \text{Mean}_i),$$

$$\text{Mean}_i = \frac{1}{NP} \sum_{j=1}^{NP} x_i^j, \quad (16)$$

$$j = 1, 2, \dots, NP, \quad i = 1, 2, \dots, D,$$

where  $x_i^{j,\text{old}}$  and  $x_i^{j,\text{new}}$  denote the knowledge of the  $j$ th learner ( $X^j$ ) before and after learning the  $i$ th subject, respectively.  $T_F$  is the teaching factor which decides the value of mean  $\text{Mean}_i$  to be changed.  $T_F$  is decided by  $T_F = \text{round}[1 + \text{rand}()]$ .

**3.2.2. Learner Phase.** Another important approach to increase knowledge for a learner is to interact with other learners. Learning method is expressed as in Algorithm 6.

Even since the TLBO algorithm proposed by Rao et al. [29], it has been applied to the fields of engineering optimization, such as mechanical design optimization [29, 32, 37], heat exchangers [33], thermoelectric cooler [34], and

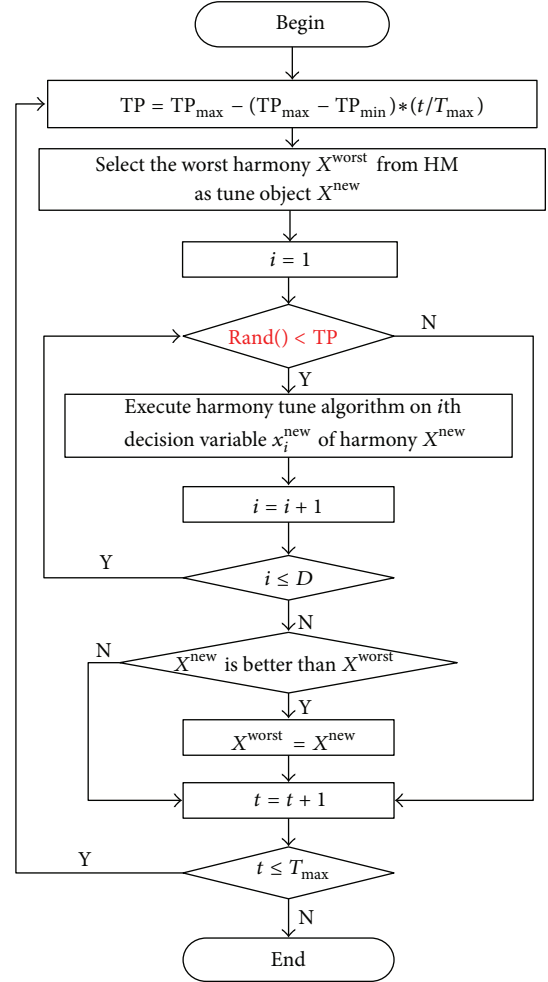


FIGURE 1: Dynamic dimension selection strategy for HS algorithm.

unconstrained and constrained real parameter optimization problems [35, 36]. Some improved TLBO algorithm was present in last two years. An elitist TLBO algorithm for solving unconstrained optimization problems [38] by Rao and Patel and an improved harmony search based on teaching-learning strategy for unconstrained optimization problems by Tuo et al. [39] are proposed.

In the TLBO method, the teacher phase relying on the best solution found so far usually has the fast convergence speed and the well ability of exploitation; it is more suitable for improving the accuracy of the global optimal solution. Learner phase relying on other learners usually has the slow convergence speed; however, it bears stronger exploration capability for solving multimodal problems.

**3.3. The HSTL Algorithm.** In order to achieve satisfactory optimization performance by applying the HS algorithm to a given problem, we develop a novel harmony search algorithm combined teaching-learning strategy, in which both new harmony generation strategies and associated control parameter values can be dynamically changed according to the process of evolution.

```

For each learner  $X^j$ ,  $j = 1, 2, \dots, NP$ 
  Randomly select another learner  $X^k$  ( $j \neq k$ )
  If  $X^j$  is superior to  $X^k$ 
     $X^{j,new} = X^{j,old} + \text{rand}(1, D) \cdot (X^j - X^k)$ 
  Else
     $X^{j,new} = X^{j,old} + \text{rand}(1, D) \cdot (X^k - X^j)$ 
  Endif
Endfor
If  $X^{j,new}$  is superior to  $X^{j,old}$ 
   $X^j = X^{j,new}$ 
Endif
    
```

ALGORITHM 6: The procedure of learner phase.

It is of great importance to realize the balance between the convergence and the diversity. In the classical HS algorithm, a new harmony is generated in Step 3. After the selecting operation in Step 4, the population diversity may increase or decrease. With high population diversity, the algorithm will have strong exploration power, and at the same time the convergence and the exploitation power will decrease accordingly. Conversely, with a low population variance, the convergence and the exploitation power will increase [18]; the diversity and the exploration power will decrease. So it is significant how to keep balance between the convergence and the diversity. Classical HS algorithm loses exploitation ability easily at later evolution process [19], because of improvising new harmony from HM with a high HMCR and local adjusting with PAR. Diversity of HM decreases gradually from the early iteration to the last. Moreover, in HS algorithm, a low HMCR employed will increase the probability (1-HMCR) of random selection in search space; the exploration power will enhance, but the local search ability and the exploitation accuracy cannot be improved by single pitch adjusting strategy.

To overcome the inherent weaknesses of HS, in this section, we propose an HSTL method. In the HSTL method, an improved teaching-learning strategy is employed to improve the search ability. The HSTL algorithm works as follows.

- (1) Optimization target vector preparation is as follows:  $X^{new} = X^{worst}$ , where  $X^{worst}$  is the worst harmony in the current HM.
- (2) Improve the target vector  $X^{new}$  with the following 4 strategies.

(a) *Harmony Memory Consideration.* The values of the target vector  $x_i^{new}$  ( $i = 1, 2, \dots, D$ ) are randomly from HM with a probability of HMCR:

$$x_i^{new} = x_i^j, \quad j \in U\{1, 2, \dots, HMS\}, \quad i = 1, 2, \dots, D. \quad (17)$$

(b) *Teaching-Learning Strategy.* If the  $i$ th ( $i = 1, 2, \dots, D$ ) design variable of the target vector  $x_j^{new}$  has not been considered in (a), it will learn from the best harmony (i.e., teacher) with probability TLP in the teacher phase or from the other harmony (i.e., learner) in the learner phase. The TLP is

the rate of performing teaching-learning operator on design variables that have not been carried out in (a): harmony memory consideration. It works as follows.

*Teacher Phase.* In this phase, the learner will learn from the best learner (i.e., teacher) in the class. Learner modification is expressed as

$$x_i^{new} = x_i^{new} + \text{rand}() \times [x_i^{best} - T_F \times M_i], \quad (18)$$

$$M_i = \frac{x_i^{worst} + x_i^{new}}{2}, \quad i = 1, 2, \dots, D,$$

where  $X^{best}$  is the best harmony in HM and  $X^{worst}$  is the worst harmony in HM.

The contribution of this paper is that  $M_i$  is replaced by  $(x_i^{worst} + x_i^{new})/2$ , instead of the mean value of population. This replacement will enhance diversity of population more than standard TLBO algorithm.

*Learner Phase.* Randomly select  $r_1$  and  $r_2$  from  $\{1, 2, \dots, HMS\}$ , and  $r_1 \neq r_2$ :

```

If  $x^{r_1}$  is better than  $x^{r_2}$ 
   $x_i^{new} = x_i^{new} + \text{rand}() \times (x_i^{r_1} - x_i^{r_2})$ 
Else
   $x_i^{new} = x_i^{new} + \text{rand}() \times (x_i^{r_2} - x_i^{r_1})$ 
Endif.
    
```

(c) *Local Pitch Adjusting Strategy.* To achieve better solutions in search space, it will carry out the local pitch adjusting strategy with probability PAR if design variables have not been selected to perform harmony memory consideration and teaching-learning strategy:

$$x_i^{new} = x_i^{new} \pm \text{rand}() \times BW(i). \quad (20)$$

(d) *Random Mutation Operator.* HSTL carries out random mutation in feasible space with probability  $P_m$  as follows:

$$x_i^{new} = x_i^L + (x_i^U - x_i^L) \times \text{rand}(). \quad (21)$$

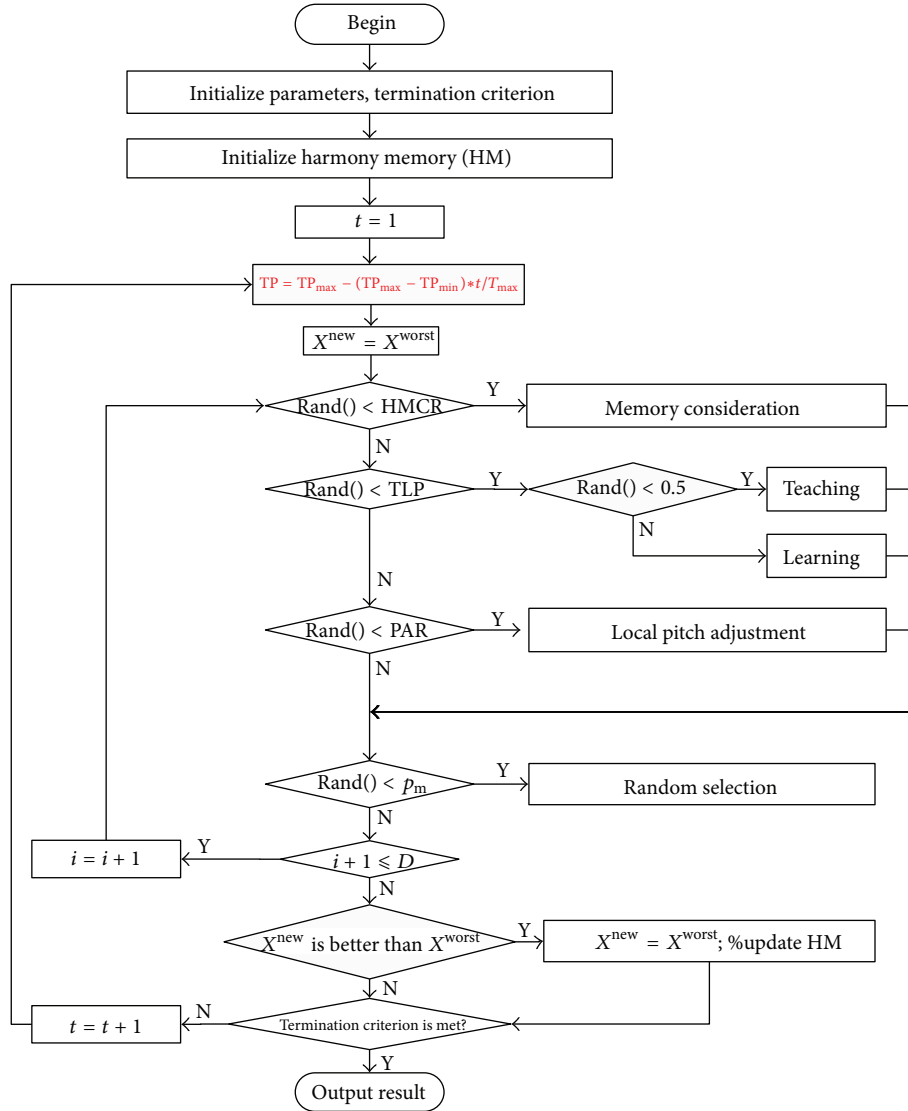


FIGURE 2: The flowchart of HSTL algorithm.

The improvisation of new target harmony in HSTL algorithm is given in Algorithm 7.

The flow chart of HSTL algorithm is shown in Figure 2.

(i) *Update Operation.* In HSTL algorithm, update operation has some changes, as follows.

Get the best harmony  $X^{best}$  and the worst harmony  $X^{worst}$  from the HM:

**If**  $X^{new}$  is better than  $X^{best}$   
 $X^{worst} = X^{best}$ ,  
 $X^{best} = X^{new}$ ,  
**ElseIf**  $X^{new}$  is better than  $X^{worst}$   
 $X^{worst} = X^{new}$ ,  
**EndIf.** (22)

(ii) *Parameters Changed Dynamically.* To efficiently balance the exploration and exploitation power of the HSTL algorithm, HMCR, PAR, BW, and TLP are dynamically adapted to a suitable range with the increase of generations:

$$HMCR = HMCR_{min} + (HMCR_{max} - HMCR_{min}) \times \left(\frac{t}{T_{max}}\right), \quad (23)$$

$$TLP = TLP_{min} + (TLP_{max} - TLP_{min}) \times \left(\frac{t}{T_{max}}\right)^k, \quad k = 5, \quad (24)$$

$$PAR = PAR_{max} - \frac{(PAR_{max} - PAR_{min}) \times t}{T_{max}}, \quad (25)$$

$$BW = BW_{max} + \exp \left[ \ln \left( \frac{BW_{min}}{BW_{max}} \right) \times \sqrt{\frac{t}{T_{max}}} \right], \quad (26)$$



```

 $X_i^{new} = X_i^{worst};$       % select  $X_i^{worst}$  as Optimization target vector
For  $i = 1$  to  $D$ 
    If  $\text{rand}() \leq \text{HMCR}$       % (a) Harmony memory consideration
         $x_i^{new} = x_i^j$  ( $j = 1, 2, \dots, \text{HMS}$ )
    Elseif  $\text{rand}() \leq \text{TLP}$       % (b) Teaching-Learning strategy
        If  $\text{rand}() \leq 0.5$       % Teaching
             $x_i^{new} = x_i^{new} + \text{rand}() \times [x_i^{best} - 0.5\text{TF} \times (x_i^{worst} + x_i^{new})]$ 
        Else      % Learning
            Randomly select  $r_1$  and  $r_2$  from  $\{1, 2, \dots, \text{HMS}\}$ 
            If  $x^{r_1}$  is better than  $x^{r_2}$ 
                 $x_i^{new} = x_i^{new} + \text{rand}() \times (x_i^{r_1} - x_i^{r_2})$ 
            Else
                 $x_i^{new} = x_i^{new} + \text{rand}() \times (x_i^{r_2} - x_i^{r_1})$ 
            end
        Endif
         $x_i^{new} = \min(\max(x_i^{new}, x_i^L), x_i^U)$ 
    Elseif  $\text{rand}(0,1) \leq \text{PAR}$       % (c) Local pitch adjusting strategy
         $x_i^{new} = x_i^{new} \pm \text{rand}() \times \text{BW}(i)$ 
         $x_i^{new} = \min(\max(x_i^{new}, x_i^L), x_i^U)$ 
    Endif
    If  $\text{rand}(0,1) \leq P_m$       % (d) Random mutation operator in feasible space
         $x_i^{new} = x_i^L + (x_i^U - x_i^L) \times \text{rand}()$ 
    Endif
Endfor

```

ALGORITHM 7: Improvisation of new harmony in HSTL algorithm.

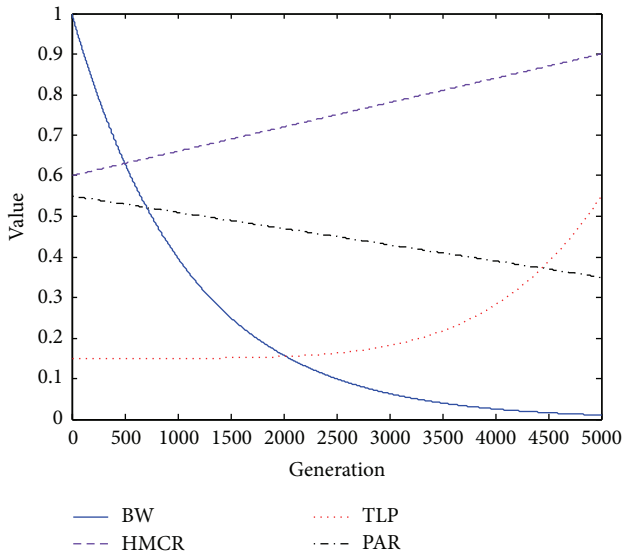


FIGURE 3: The evolution curves of parameters (HMCR, TLP, PAR, and BW) of HSTL algorithm.

where (25) and (26) are quoted from the literature studies [18] and [16], respectively.

Let  $\text{HMCR}_{\max} = 0.9$ ,  $\text{HMCR}_{\min} = 0.6$ ,  $\text{TLP}_{\max} = 0.55$ ,  $\text{TLP}_{\min} = 0.15$ ,  $\text{PAR}_{\max} = 0.5$ ,  $\text{PAR}_{\min} = 0.33$ ,  $\text{BW}_{\max} = 1$ , and  $\text{BW}_{\min} = 0.001$ . The changing curves of parameters (HMCR, TLP, PAR, and BW) are shown in Figure 3.

It can be seen that the parameter HMCR increases gradually from 0.6 to 0.9 linearly. TLP increases with low

velocity in the early stage and rises sharply in the final stage. That is to say, in the beginning, the harmony consideration rule and teaching-learning strategy are carried out with a smaller probability; in the later stage, HSTL algorithm begins to focus on local exploitation with harmony consideration and teaching-learning strategy. The benefits of doing so can get more opportunities to reinforce the global exploration by strengthening disturbance in the early stage and in the final stage to intensify local search step by step, thereby acquiring high-precision solution. For the same reason, BW decreases gradually in order to reduce perturbation gradually, and the PAR's variation from 0.5 to 0.3 is to reduce the probability of pitch adjustment.

Equation (27) shows that the random mutation operation is changed dynamically from  $5/D$  to  $3/D$  with iterations. It can make the random disturbance change from strong to weak, and thus HSTL algorithm has strong global exploration ability in the early stage and has effective local exploitation ability in the latter stage:

$$P_m = \frac{5}{D} - \frac{2t/T_{\max}}{D}. \tag{27}$$

#### 4. Other Related Techniques for Solving 0-1 Knapsack Problem

The 0-1 knapsack problem is a large scale, multiconstraint, and nonlinear integer programming problem. So solving 0-1 knapsack problems with HSTL algorithm needs to employ constraint handling technique and integer processing method.

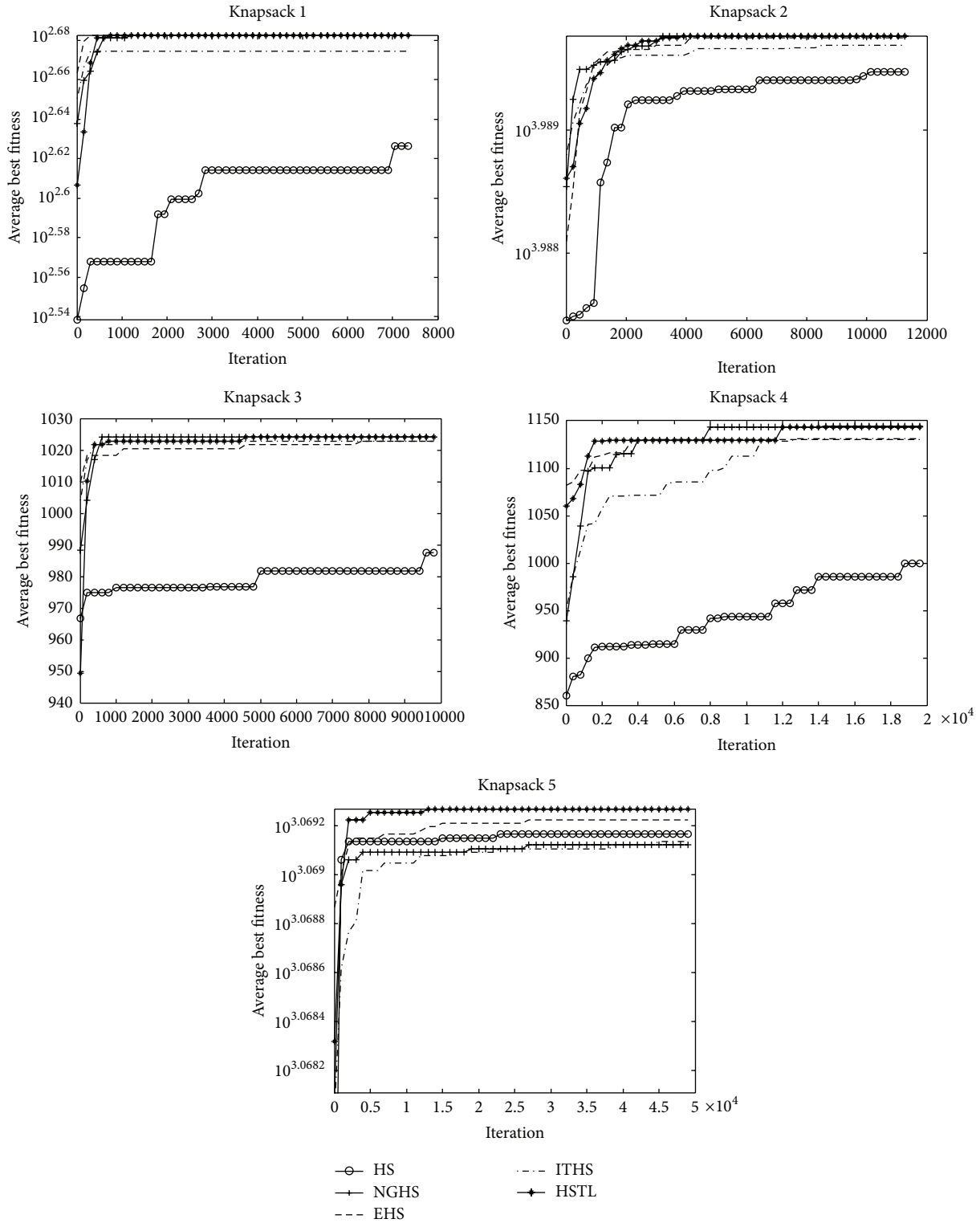


FIGURE 4: The convergence graphs of  $KP_1$ – $KP_5$ .

4.1. *Constraint-Handling Technique.* For constrained optimization problems, there have existed many successful constraint-handling techniques on solving constrained optimization problems, such as penalty function method, special representations and operators, repair method, and

multiobjective method [40–53]. In this paper, because we mainly do some research on discrete optimization problems by using the HS algorithm, so special handling technique and the revision methods [23–28] for adjusting the 0-1 knapsack problems have not been adopted in this paper.

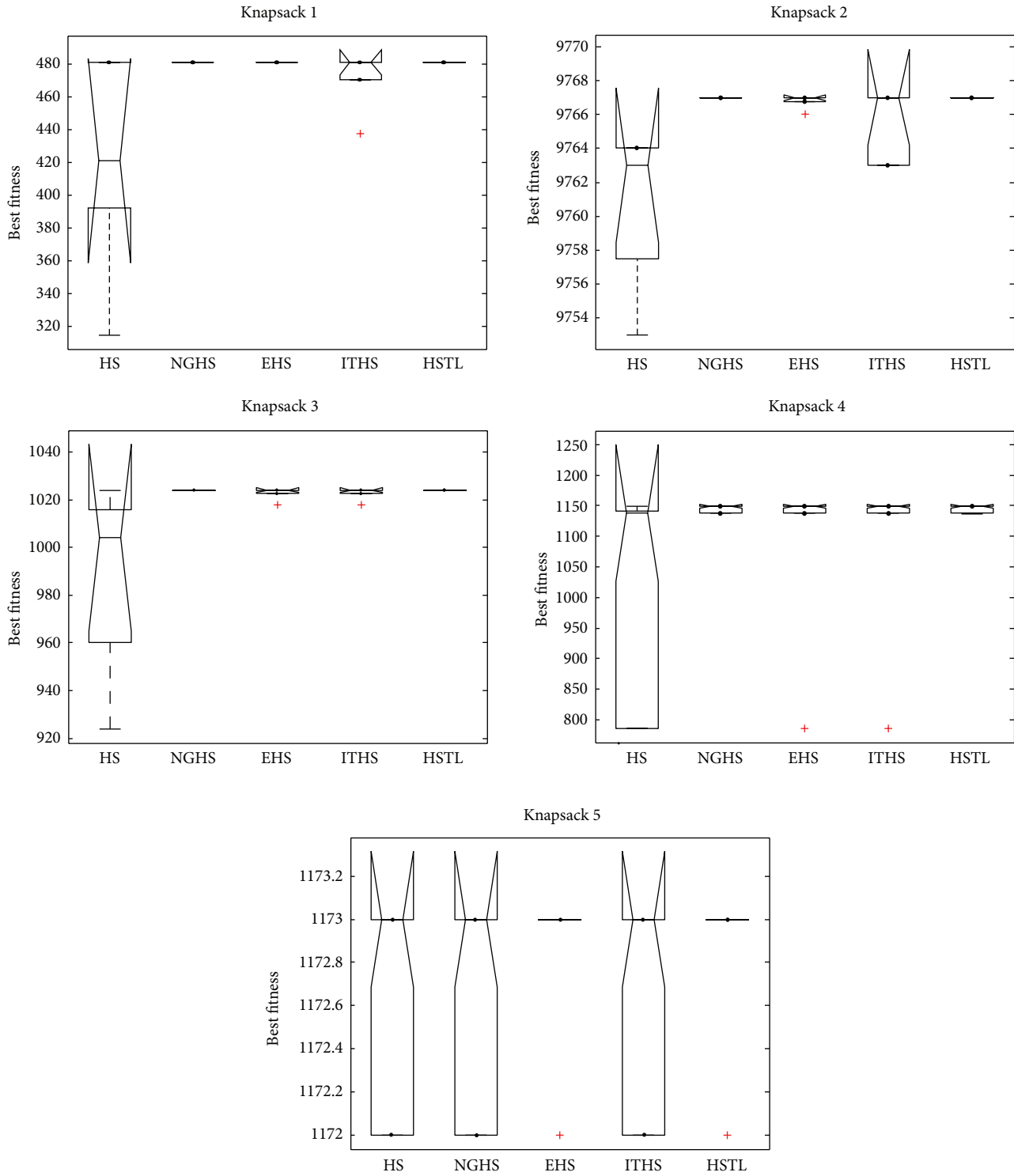


FIGURE 5: The box plots of  $KP_1 - KP_5$ .

In this paper, the multiobjective method has been used for handling constrained 0-1 knapsack problems. The multiobjective method [42] is as follows.

(1) Any feasible solution is better than any infeasible solution.

(2) Among two feasible solutions, the one that has a better objective function value is preferred.

(3) Among two infeasible solutions, the one that has a smaller degree of constraint violation is preferred. The purpose is to render the infeasible solution with large

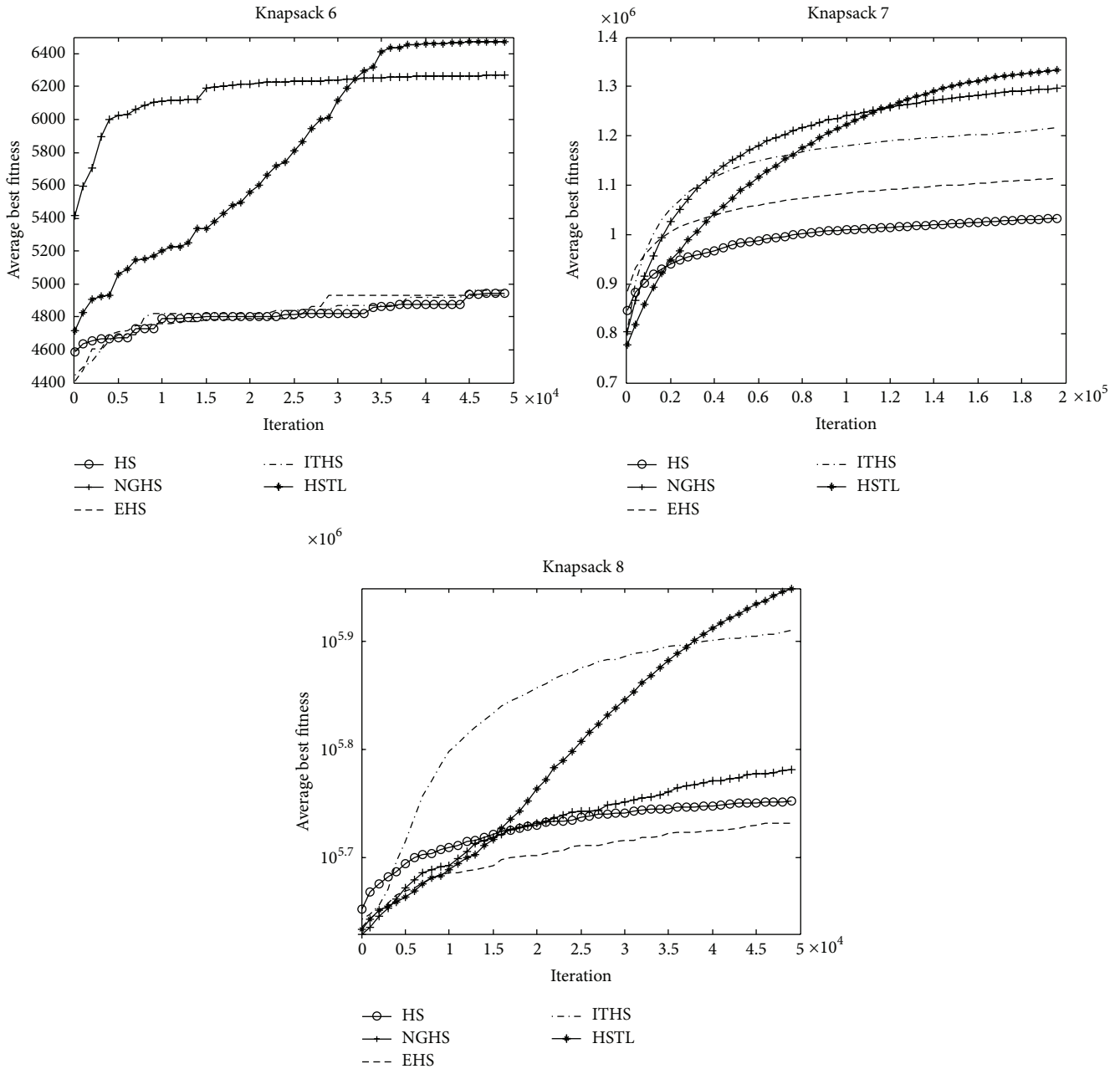


FIGURE 6: The convergence graphs of  $KP_6$ – $KP_8$ .

degree of constraint violation from moving gradually towards the solution with no constraint violation or with smaller degree of constraint violation.

However, for a knapsack problem with many items ( $D > 10000$ ) and the knapsack with small capacity, if we only execute HS algorithm without any revise method to this small-capacity knapsack, all of the harmony vectors in HM are probably infeasible. In this case, even in the very good methods it is hard to obtain the feasible solutions. So, for an infeasible solution, we will do simple revise through randomly removing some item from this knapsack. In this

way, the infeasible solution can gradually turn into the feasible solution.

**4.2. Integer Processing Method.** In HSTL algorithm, because the variables are adjusted by the teaching-learning strategy, local pitch adjusting strategy and random mutation operator are real numbers, so every variable is replaced by the nearest integer, that is, as follows:

$$x' = \text{round}(x). \tag{28}$$

Let  $x = (0.8, 0.3, 0.9, 0.6, 1.2, 0.4)$ . Then  $x' = \text{round}(x) = (1, 0, 1, 1, 1, 0)$ .

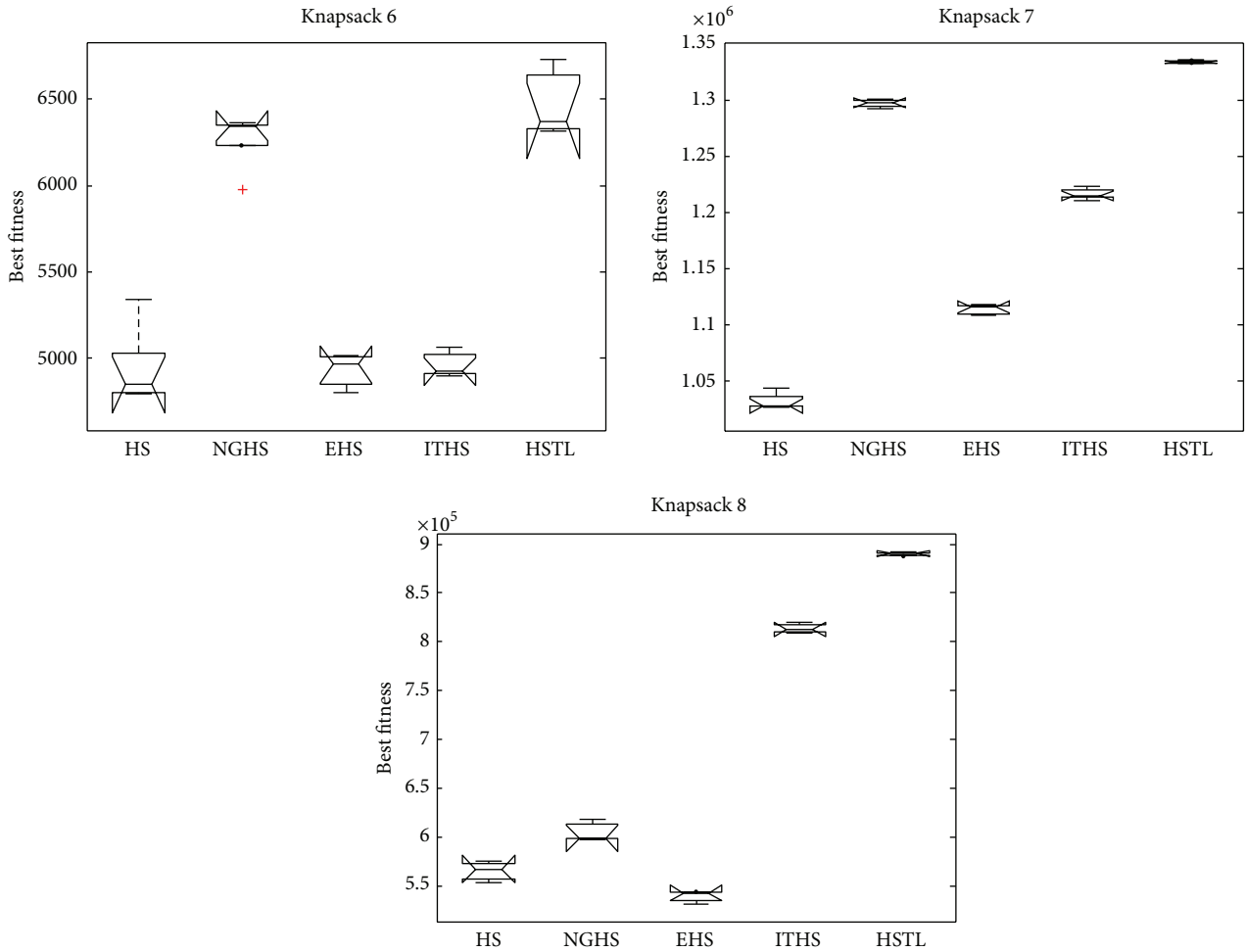


FIGURE 7: The box plots of  $KP_6$ - $KP_8$ .

### 5. Solving 0-1 Knapsack Problems with HSTL Algorithm

5.1. *Experimental Setup and Parameters Setting.* In order to evaluate the performance of the HSTL algorithm, we used a set of 13 knapsack problems ( $KP_1$ - $KP_{13}$ ).  $KP_1$ - $KP_3$  is quoted, respectively, from  $f_5$ ,  $f_8$  and  $f_{10}$  in the literature [21].  $KP_4$ - $KP_{13}$  is quoted from website <http://homepage.ntlworld.com/walter.barker2/Knapsack%20Problem.htm>. In all the knapsack problems ( $KP_1$ - $KP_{13}$ ),  $KP_1$ - $KP_8$  are called one-dimensional problems that only include weight constraint and  $KP_9$ - $KP_{13}$  are called two-dimensional problems that include both weight constraint and volume constraint.

All simulation experiments are carried out to compare the optimization performance of the presented method (HSTL) with respect to (a) classical HS, (b) NGHS, (c) EHS, and (d) ITHS. In the experiments, the parameters setting for the compared HS algorithms is shown in Table 2. To make the comparison fair, the populations for all the competitor algorithms were initialized using the same random seeds. The variants of HS algorithm were set at the same termination

criteria: the number of improvisations (function evaluation times: FEs)  $FEs = 500 \times D$ , respectively. However, if the  $FEs > 5E + 05$ , then set  $FEs = 5E + 05$ .

The best and worst fitness value of each test problem are recorded for 30 independent runs; the mean fitness, standard deviation (Std), and mean runtime of each knapsack problem are calculated for 30 independent runs.

5.2. *The Experimental Results and Analysis.* Table 3 reports the worst, mean, best, and Std of problem results by applying the five algorithms (HS, NGHS, EHS, ITHS, and HSTL) to optimize the knapsack problems  $KP_1$ - $KP_8$ , respectively. The best results are emphasized in boldface. Figures 4 and 6 illustrate the convergence characteristics in terms of the best values of the median run of each algorithm for knapsack problems  $KP_1$ - $KP_8$ . Figures 5 and 7 demonstrate the performance and stability characteristics according to the distributions of the best values of 30 runs of each algorithm for knapsack problems  $KP_1$ - $KP_8$ .

Based on the resulting data in Table 3, the optimal objective values (best, mean, worst, and Std) can be easily

TABLE 2: Parameter settings for the compared HS algorithms (HS, NGHS, EHS, ITHS, and HSTL).

Algorithm	HMS	HMCR	PAR	BW	others
HS	5	0.99	0.33	$BW = (X^U - X^L)/1000$	/
NGHS	5	/	/	/	/
EHS	50	0.99	0.33	$BW = 1.17\sqrt{(1/HMS) \sum_{i=1}^{HMS} (x_i - \bar{x})^2}$	/
ITHS	10	0.99	0.33	$BW_{max} = (X^U - X^L)/2$ $BW_{min} = (X^U - X^L)/10$	/
HSTL	10	$HMCR_{max} = 0.95$ $HMCR_{min} = 0.6$	$PAR_{max} = 0.5$ $PAR_{min} = 0.2$	$BW_{max} = (X^U - X^L)/2$ $BW_{min} = (X^U - X^L)/10$	$TLP_{max} = 0.55,$ $TLP_{min} = 0.15$

TABLE 3: The result of 1-dimensional (weight versus value) knapsack problems.

Problem	D	Target weight	Total weight	Total values	Optimal result	Algorithm	Outcomes				Runtime
							Worst	Mean	Best	Std	
KP <sub>1</sub>	15	375	741.9172	62.9963	<b>481.069</b>	HS	314.9297	423.191	<b>481.069</b>	67.92161	0.169532
						NGHS	<b>481.069</b>	<b>481.069</b>	<b>481.069</b>	<b>6.4E-14</b>	0.173909
						EHS	<b>481.069</b>	<b>481.069</b>	<b>481.069</b>	<b>6.4E-14</b>	0.814028
						ITHS	437.9345	472.4424	<b>481.069</b>	19.2905	0.447476
						HSTL	<b>481.069</b>	<b>481.069</b>	<b>481.069</b>	<b>6.4E-14</b>	0.330757
KP <sub>2</sub>	23	10000	19428	19309	<b>9767</b>	HS	9747	9760	<b>9767</b>	5.533596	0.313597
						NGHS	<b>9767</b>	<b>9767</b>	<b>9767</b>	<b>0</b>	0.305331
						EHS	9643	9751.6	<b>9767</b>	22.64509	1.595463
						ITHS	9756	9765.7	<b>9767</b>	2.641186	0.790986
						HSTL	<b>9767</b>	<b>9767</b>	<b>9767</b>	<b>0</b>	0.585312
KP <sub>3</sub>	20	878	1098	1085	<b>1024</b>	HS	924	987.4	<b>1024</b>	40.395544	0.254834
						NGHS	<b>1024</b>	<b>1024</b>	<b>1024</b>	<b>0</b>	0.245683
						EHS	1018	1022.8	<b>1024</b>	2.6832816	1.274226
						ITHS	1018	1022.8	<b>1024</b>	2.6832816	0.654352
						HSTL	<b>1024</b>	<b>1024</b>	<b>1024</b>	<b>0</b>	0.485799
KP <sub>4</sub>	40	15	374	14049	<b>1149</b>	HS	786	1109.4	<b>1149</b>	113.7494	12.51123
						NGHS	<b>1138</b>	<b>1147.9</b>	<b>1149</b>	<b>3.478505</b>	12.72128
						EHS	786	1109.4	<b>1149</b>	113.7494	15.64561
						ITHS	786	1112.7	<b>1149</b>	114.7907	13.6885
						HSTL	<b>1138</b>	<b>1147.9</b>	<b>1149</b>	<b>3.478505</b>	13.29233
KP <sub>5</sub>	100	27	1360	34965	<b>1173</b>	HS	1172	1172.5	<b>1173</b>	0.508548	37.14283
						NGHS	1172	1172.633	<b>1173</b>	0.490133	37.22416
						EHS	1172	1172.567	<b>1173</b>	0.504007	53.20237
						ITHS	1172	1172.567	<b>1173</b>	0.504007	41.50243
						HSTL	1172	<b>1172.9</b>	<b>1173</b>	<b>0.305129</b>	37.11053
KP <sub>6</sub>	10000	431	349354	3011792	<b>unknown</b>	HS	4795	4941	5338	228.20495	2902.844
						NGHS	5976	6270.4	6363	165.43821	1881.969
						EHS	4797	4929.8	5014	94.099416	16979.43
						ITHS	4899	4960.6	5064	72.081898	5047.957
						HSTL	<b>6318</b>	<b>6474.4</b>	<b>6730</b>	<b>185.7156</b>	<b>1506.648</b>
KP <sub>7</sub>	10000	1765326	5033006	2063406	<b>unknown</b>	HS	1068628	1070926.5	1073225	3250.5699	4526.0894
						NGHS	1126584	1127050.5	1127517	659.73063	5060.0062
						EHS	1041435	1043733.5	1046032	3250.5699	5435.1693
						ITHS	1124261	1129389	1134517	7252.0871	4555.54
						HSTL	<b>1193743</b>	<b>1194524</b>	<b>1195304</b>	<b>1103.794</b>	<b>3326.1473</b>
KP <sub>8</sub>	11000	1000000	5526981	2263955	<b>unknown</b>	HS	529153	532801.8	537353	3001.1646	10633.825
						NGHS	738654	762855.4	798991	26865.404	11912.341
						EHS	480038	486268.8	490627	4481.0616	13084.292
						ITHS	795202	800908	807164	4864.5424	10785.623
						HSTL	<b>940656</b>	<b>944780</b>	<b>947850</b>	<b>2630.108</b>	<b>9606.735</b>

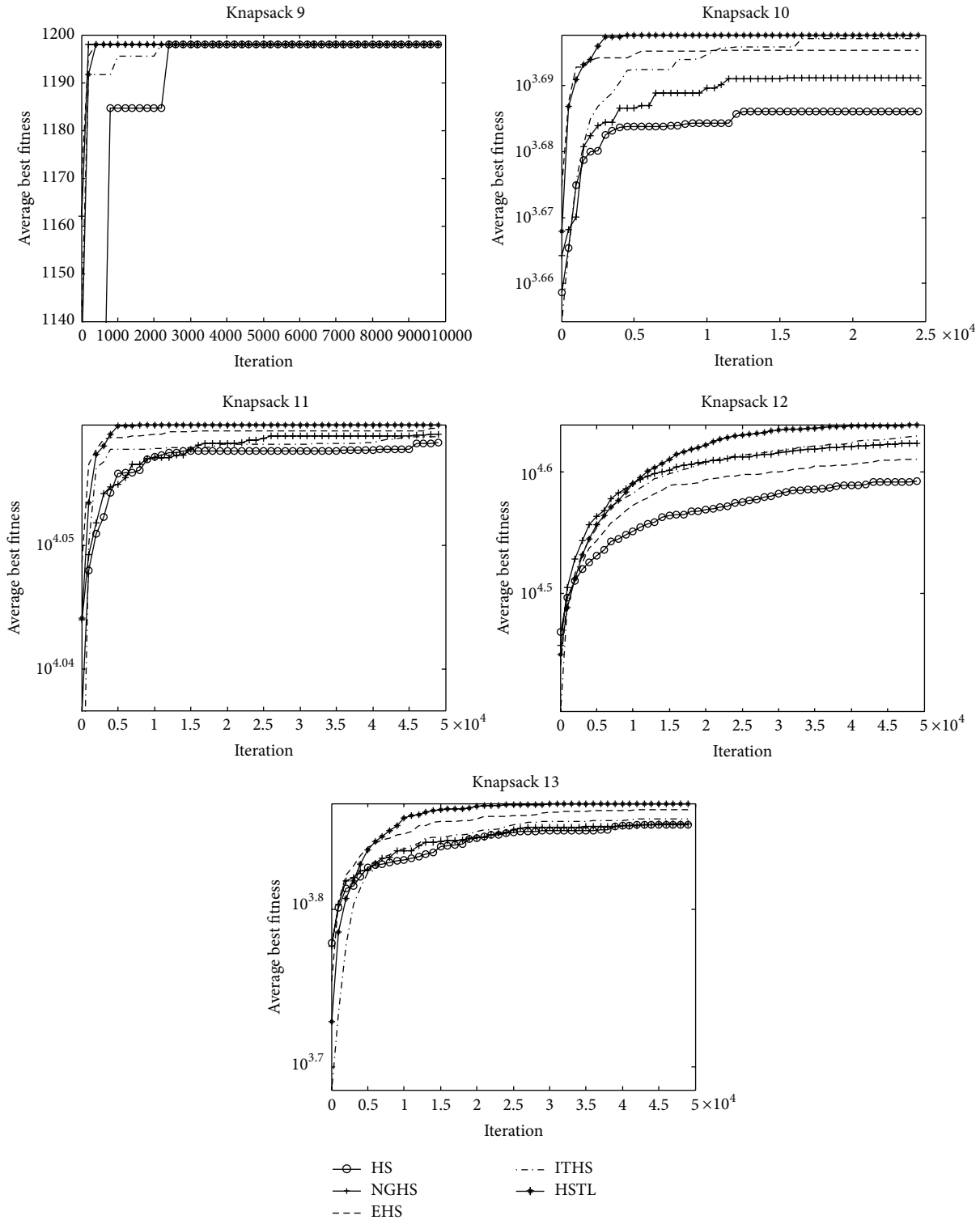


FIGURE 8: The convergence graphs of KP<sub>9</sub>-KP<sub>13</sub>.

obtained by NGHS, EHS, ITHS, and the HSTL algorithms with high accuracy on the knapsack problems KP<sub>1</sub>-KP<sub>5</sub>. However, comparatively speaking, NGHS and the HSTL algorithm have better optimal results on worst, mean, best,

and Std, and thus NGHS method and the HSTL algorithm are more effective and stabilized to solve the problems KP<sub>1</sub>-KP<sub>5</sub>.

For the high-dimensional knapsack problems KP<sub>6</sub>-KP<sub>8</sub>, Table 3 shows that the HSTL algorithm has obvious

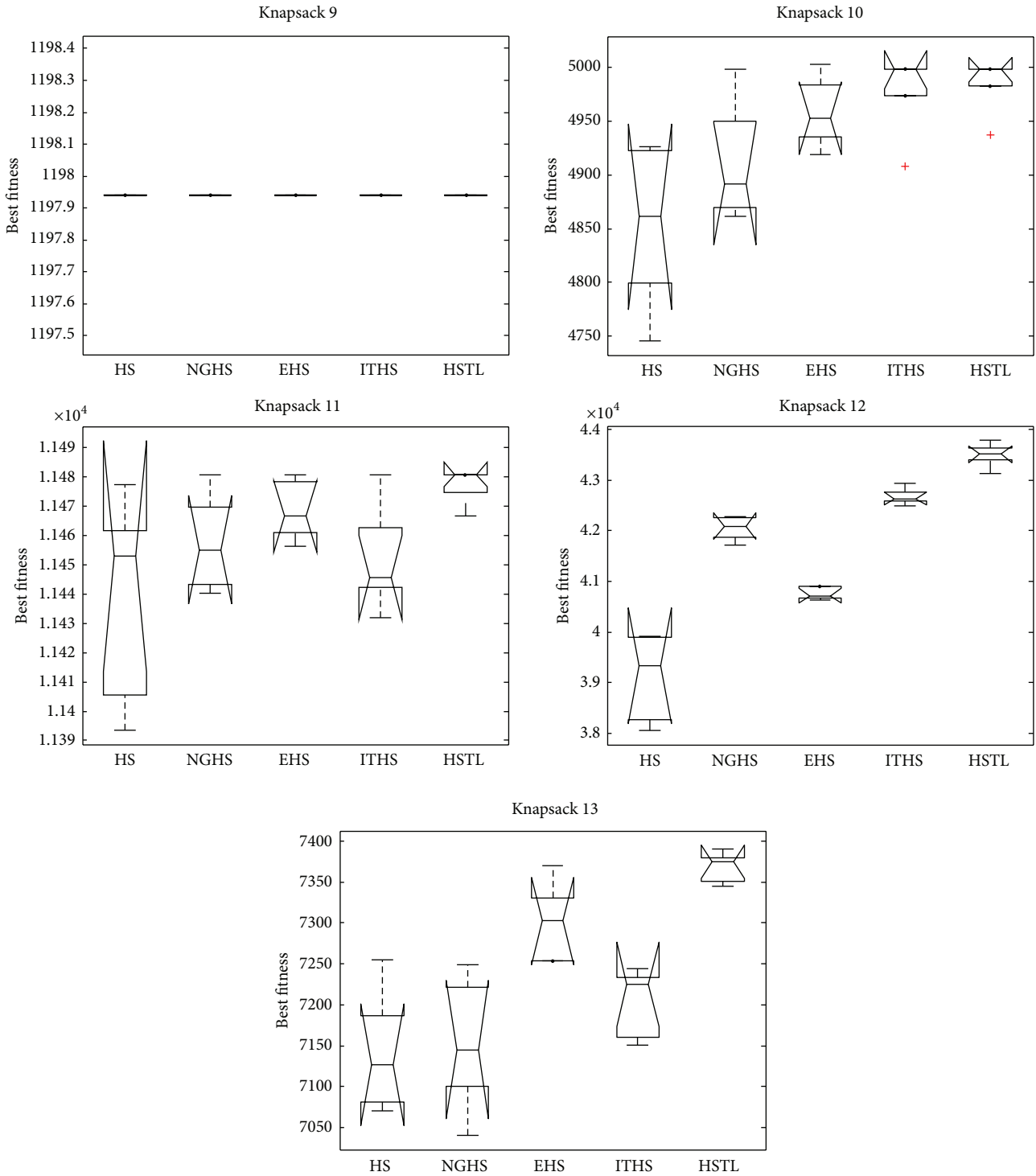


FIGURE 9: The box plots of KP<sub>9</sub>-KP<sub>13</sub>.

advantages over other variants of HS algorithms. Comparing with other HS algorithms, although HSTL algorithm has slow convergence speed in the early stage, it can be constant to optimize the solutions and obtain high precision solutions in the latter stage, which can be seen from Figure 6.

It is evident from Figure 7 that HSTL algorithm has better convergence, stability, and robustness in most cases than HS, NGHS, EHS, and ITHS algorithms.

Table 4 shows the experimental results on algorithms HS, NGHS, EHS, ITHS, and HSTL for two-dimensional knapsack problems: KP<sub>9</sub>-KP<sub>13</sub>. Figure 8 shows the convergence graphs, and Figure 9 is the box plots of independent 30 runs of knapsack problems KP<sub>9</sub>-KP<sub>13</sub>.

It can be seen evidently from Table 4 that the HSTL algorithm attained the optimal best, mean, worst, and Std results among all two-dimensional knapsack problems.



TABLE 4: The result of 2-dimensional (weight and volume versus value) knapsack problems.

Problem	D	Target weight   volume	Total weight	Total volume	Total values	Optimal result	Algorithm	Outcomes				Runtime
								Worst	Mean	Best	Std	
KP <sub>9</sub>	20	20   25	111.63	170.27	3813.6	<b>1197.94</b>	HS	<b>1197.94</b>	<b>1197.94</b>	<b>1197.94</b>	0	<b>7.90831</b>
							NGHS	<b>1197.94</b>	<b>1197.94</b>	<b>1197.94</b>	0	8.092175
							EHS	<b>1197.94</b>	<b>1197.94</b>	<b>1197.94</b>	0	8.978277
							ITHS	<b>1197.94</b>	<b>1197.94</b>	<b>1197.94</b>	0	8.588568
							HSTL	<b>1197.94</b>	<b>1197.94</b>	<b>1197.94</b>	0	8.31578
KP <sub>10</sub>	50	114   133	400.56	408.57	10211	<b>5002.81</b>	HS	4849.19	4935.062	<b>5002.81</b>	45.16856	19.03619
							NGHS	4917.42	4963.421	<b>5002.81</b>	32.60289	19.15758
							EHS	4933.24	4983.169	4998.27	24.18182	23.48809
							ITHS	4926.07	4975.32	<b>5002.81</b>	29.72861	20.82369
							HSTL	<b>4995.48</b>	<b>5000.063</b>	<b>5002.81</b>	<b>2.981432</b>	<b>18.9528</b>
KP <sub>11</sub>	100	1500   3000	3090.5	4599.2	15278	<b>11480.84</b>	HS	11393.59	11438.024	11477.52	35.010842	51.07192
							NGHS	11440.36	11457.378	<b>11480.84</b>	16.483244	51.2287
							EHS	11456.44	11468.79	<b>11480.84</b>	10.221502	65.89753
							ITHS	11431.96	11452.068	<b>11480.84</b>	18.277327	54.87239
							HSTL	<b>11466.63</b>	<b>11477.33</b>	<b>11480.84</b>	<b>6.153989</b>	<b>45.31429</b>
KP <sub>12</sub>	1000	30000   40000	250320	400090	157890	<b>unknown</b>	HS	38064.94	39108.266	39924.89	866.36169	414.50957
							NGHS	41709.51	42048.386	42280.62	235.6772	424.71994
							EHS	40627.36	40766.678	40903.3	123.45544	454.46526
							ITHS	42485.83	42671.15	42928.94	163.78948	439.51866
							HSTL	<b>43953.97</b>	<b>44119.987</b>	<b>44409.66</b>	<b>119.80076</b>	<b>393.5962</b>
KP <sub>13</sub>	300	800   700	13605	14157	48464	<b>unknown</b>	HS	7070.4	7140.222	7254.76	73.854458	36.347165
							NGHS	7040.92	7153.61	7249.1	81.266043	36.41226
							EHS	7253.67	7299.32	7369.58	48.490527	39.440654
							ITHS	7150.24	7202.6	7244.01	42.540662	39.450648
							HSTL	<b>7345.46</b>	<b>7367.95</b>	<b>7390.52</b>	<b>18.42674</b>	<b>29.889155</b>

Bold indicates best results.

From the convergence graphs (Figure 8), it can be seen that the HSTL algorithm has a strong search ability and convergence throughout the search process for the two-dimensional knapsack problems. As can be seen from the box plots (Figure 9), the HSTL has demonstrated some advantage over the other four algorithms on solving two-dimensional 0-1 knapsack problems.

## 6. Conclusion

In this paper, a novel harmony search with teaching-learning strategies is presented to solve 0-1 knapsack problems. The HSTL algorithm employs the idea of teaching and learning. Four strategies are used to maintain the proper balance between exploration power and exploitation power. With the process of evolution, the dynamic strategy is adopted to change the parameters HMCR, TLP, BW, and PAR. Experimental results showed that the HSTL algorithm has stronger ability to resolve the high-dimensional 0-1 knapsack problems. However, the HSTL algorithm has more parameters. In the future, we should focus on improving the structure of HSTL algorithm, decreasing the running cost and enhancing efficiency for solving complex optimization problems.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 81160183, Scientific Research Program funded by Shaanxi Provincial Education Department under Grant no. 12JK0863, Ministry of Education "Chunhui Plan" project (no. Z2011051), Natural Science Foundation of Ningxia Hui Autonomous Region (no. NZ12179), and Scientific Research Fund of Ningxia Education Department (no. NGY2011042).

## References

- [1] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [2] Z. W. Geem, "Novel derivative of harmony search algorithm for discrete design variables," *Applied Mathematics and Computation*, vol. 199, no. 1, pp. 223–230, 2008.
- [3] L. D. S. Coelho and V. C. Mariani, "An improved harmony search algorithm for power economic load dispatch," *Energy Conversion and Management*, vol. 50, no. 10, pp. 2522–2526, 2009.
- [4] A. Askarzadeh and A. Rezaeizadeh, "A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell," *International Journal of Hydrogen Energy*, vol. 36, no. 8, pp. 5047–5053, 2011.
- [5] S. Ghosh, D. Kundu, K. Suresh, S. Das, and A. Abraham, "Design of optimal digital IIR filters by using a bandwidth adaptive harmony search algorithm," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 481–486, Coimbatore, India, December 2009.
- [6] D. C. Hoang, P. Yadav, R. Kumar, and S. K. Panda, "A robust harmony search algorithm based clustering protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC '10)*, Cape Town, South Africa, May 2010.
- [7] M. Jaberipour and E. Khorram, "Solving the sum-of-ratios problems by a harmony search algorithm," *Journal of Computational and Applied Mathematics*, vol. 234, no. 3, pp. 733–742, 2010.
- [8] M. Poursha, F. Khoshnoudian, and A. S. Moghadam, "Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls," *International Journal of Civil Engineering*, vol. 9, no. 1, pp. 1–8, 2011.
- [9] A. H. Khazali and M. Kalantar, "Optimal reactive power dispatch based on harmony search algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 3, pp. 684–692, 2011.
- [10] A. H. Khazali, A. Parizad, and M. Kalantar, "Optimal voltage/reactive control by an improve harmony search algorithm," *International Review of Electrical Engineering*, vol. 5, no. 1, pp. 217–224, 2010.
- [11] E. Khorram and M. Jaberipour, "Harmony search algorithm for solving combined heat and power economic dispatch problems," *Energy Conversion and Management*, vol. 52, no. 2, pp. 1550–1554, 2011.
- [12] D. Zou, L. Gao, J. Wu, and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 16–18, pp. 3308–3318, 2010.
- [13] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830–848, 2010.
- [14] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic subpopulations," *Engineering Optimization*, vol. 42, no. 2, pp. 101–117, 2010.
- [15] S. Tuo and L. Yong, "An improved harmony search algorithm with chaos," *Journal of Computational Information Systems*, vol. 8, no. 10, pp. 4269–4276, 2012.
- [16] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.
- [17] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 307–316, 2010.
- [18] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, "An intelligent tuned harmony search algorithm for optimisation," *Information Sciences*, 2012.
- [19] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 1, pp. 89–106, 2011.
- [20] H. Sarvari and K. Zamanifar, "Improvement of harmony search algorithm by using statistical analysis," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 181–215, 2012.
- [21] D. Zou, L. Gao, S. Li, and J. Wu, "Solving 0-1 knapsack problem by a novel global harmony search algorithm," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1556–1564, 2011.

- [22] Y. Liu and C. Liu, "A schema-guiding evolutionary algorithm for 0-1 knapsack problem," in *Proceedings of the International Association of Computer Science and Information Technology—Spring Conference (IACSIT-SC '09)*, pp. 160–164, April 2009.
- [23] H. Shi, "Solution to 0/1 knapsack problem based on improved ant colony algorithm," in *Proceedings of the IEEE International Conference on Information Acquisition (ICIA '06)*, pp. 1062–1066, August 2006.
- [24] F.-T. Lin, "Solving the knapsack problem with imprecise weight coefficients using genetic algorithms," *European Journal of Operational Research*, vol. 185, no. 1, pp. 133–145, 2008.
- [25] V. Boyer, D. El Baz, and M. Elkihel, "Solving knapsack problems on GPU," *Computers and Operations Research*, vol. 39, no. 1, pp. 42–47, 2012.
- [26] R. R. Hill, Y. K. Cho, and J. T. Moore, "Problem reduction heuristic for the 0-1 multidimensional knapsack problem," *Computers and Operations Research*, vol. 39, no. 1, pp. 19–26, 2012.
- [27] A. Gherboudj, A. Layeb, and S. Chikhi, "Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm," *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, pp. 229–236, 2012.
- [28] A. Layeb, "A novel quantum inspired cuckoo search for knapsack problems," *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 297–305, 2011.
- [29] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [30] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, no. 1, pp. 1–15, 2012.
- [31] R. V. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 3, pp. 535–560, 2012.
- [32] R. V. Venkata, R. Savsani, and V. J. Rao, *Mechanical Design Optimization Using Advanced Optimization Techniques*, Springer, London, UK, 2012.
- [33] R. V. Rao and V. Patel, "Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1147–1162, 2013.
- [34] R. Venkata Rao and V. Patel, "Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 430–445, 2013.
- [35] R. V. Rao, V. J. Savsani, and J. Balic, "Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems," *Engineering Optimization*, vol. 44, no. 12, pp. 1447–1462, 2012.
- [36] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, no. 3, pp. 710–720, 2013.
- [37] P. J. Pawar and R. V. Rao, "Parameter optimization of machining processes using teaching-learning-based optimization algorithm," *The International Journal of Advanced Manufacturing Technology*, pp. 1–12, 2012.
- [38] R. Rao and V. Patel, "Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 4, no. 1, 2013.
- [39] S. Tuo, L. Yong, and T. Zhou, "An improved harmony search based on teaching-learning strategy for unconstrained optimization problems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 413565, p. 29, 2013.
- [40] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.
- [41] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
- [42] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658–675, 2006.
- [43] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
- [44] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658–675, 2006.
- [45] T.-C. Chiang and Y.-P. Lai, "MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 1473–1480, June 2011.
- [46] N. Bacanin and M. Tuba, "Artificial bee colony (ABC) algorithm for constrained optimization improved with genetic operators," *Studies in Informatics and Control*, vol. 21, no. 2, pp. 137–146, 2012.
- [47] D. Bianchi, S. Genovesi, and A. Monorchio, "Constrained Pareto Optimization of Wide Band and Steerable Concentric Ring Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3195–3204, 2012.
- [48] P.-T. Chang and J.-H. Lee, "A fuzzy DEA and knapsack formulation integrated model for project selection," *Computers and Operations Research*, vol. 39, no. 1, pp. 112–125, 2012.
- [49] M. Daneshyari and G. G. Yen, "Constrained multiple-swarm particle swarm optimization within a cultural framework," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 42, no. 2, pp. 475–490, 2012.
- [50] R. Datta and K. Deb, "A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, Barcelona, Spain, July 2012.
- [51] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers and Operations Research*, vol. 38, no. 12, pp. 1877–1896, 2011.
- [52] A. H. Gandomi, X.-S. Yang, S. Talatahari, and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization," *Computers and Mathematics with Applications*, vol. 63, no. 1, pp. 191–200, 2012.
- [53] A. W. Mohamed and H. Z. Sabry, "Constrained optimization based on modified differential evolution algorithm," *Information Sciences*, vol. 194, pp. 171–208, 2012.