

Research Article

Construction Example for Algebra System Using Harmony Search Algorithm

FangAn Deng,¹ Shouheng Tuo,¹ Longquan Yong,¹ and Tao Zhou²

¹School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong 723000, China

²School of Science, Ningxia Medical University, Yinchuan, Ningxia 750004, China

Correspondence should be addressed to Shouheng Tuo; tuo_sh@126.com

Received 24 June 2014; Accepted 15 September 2014

Academic Editor: Shifei Ding

Copyright © 2015 FangAn Deng 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.

The construction example of algebra system is to verify the existence of a complex algebra system, and it is a NP-hard problem. In this paper, to solve this kind of problems, firstly, a mathematical optimization model for construction example of algebra system is established. Secondly, an improved harmony search algorithm based on NGHS algorithm (INGHS) is proposed to find as more solutions as possible for the optimization model; in the proposed INGHs algorithm, to achieve the balance between exploration power and exploitation power in the search process, a global best strategy and parameters dynamic adjustment method are present. Finally, nine construction examples of algebra system are used to evaluate the optimization model and performance of INGHs. The experimental results show that the proposed algorithm has strong performance for solving complex construction example problems of algebra system.

1. Introduction

Algebra system is one of the broad parts of mathematics, together with number theory, geometry, and analysis. It has wide applications in astronomy, biology, construction, computer science, and so on. Many scientific researchers focus on its studies. However, with the rapid development of science and technology, more and more complex algebra systems emerge in large numbers, so the algebra system researchers meet a lot of difficult problems. One of them is to construct appropriate examples for the algebra system so as to prove the existence of the algebra system. This is because there is the tremendous computing workload to construct appropriate examples that are satisfied with the operations of the algebra system. In the current study of algebra system, the construction examples of algebra system are often manually achieved. However, for a multielement complex algebra system, the construction examples are very difficult. In order to understand the construction example of algebra system better, we introduce its construction by an example of $N(2, 2, 0)$ algebra system.

Above all, we begin this section by introducing a simplified definition of $N(2, 2, 0)$ algebra system [1, 2].

Definition 1. An $N(2, 2, 0)$ algebra is a system $(S, *, \Delta, 0)$ where S is nonempty set, 0 is a constant element of S , and $*$ and Δ are two binary operations on S , obeying the following axioms. For all $x, y, z \in S$,

$$(F_1) \quad x * (y\Delta z) = z * (x * y),$$

$$(F_2) \quad (x\Delta y) * z = y * (x * z),$$

$$(F_3) \quad 0 * x = x.$$

For Definition 1, there are two works for researchers to do.

- (1) Give some examples to prove the existence of this algebra.
- (2) How many solutions are satisfying the three conditions (F_1) , (F_2) , and (F_3) simultaneously?

Example 2. Let $S = \{a_1, a_2, \dots, a_n\}$ be a finite set of distinct elements. We define operations $*$ and Δ by the following equations, respectively:

$$\begin{array}{c|cccc} * & a_1 & a_2 & \cdots & a_n \\ \hline a_1 & x_{11} & x_{12} & \cdots & x_{1n} \\ a_2 & x_{21} & x_{22} & \cdots & x_{2n} \\ & & & \cdots & \\ a_n & x_{n1} & x_{n2} & \cdots & x_{nn} \end{array} \quad (1)$$

$$\begin{array}{c|cccc} \Delta & a_1 & a_2 & \cdots & a_n \\ \hline a_1 & y_{11} & y_{12} & \cdots & y_{1n} \\ a_2 & y_{21} & y_{22} & \cdots & y_{2n} \\ & & & \cdots & \\ a_n & y_{n1} & y_{n2} & \cdots & y_{nn} \end{array} \quad (2)$$

In (1) and (2), $a_i * a_j = x_{ij} \in S$, $a_i \Delta a_j = y_{ij} \in S$ ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$).

Next we consider

$$(F_1) \quad a_i * (a_j \Delta a_k) = a_k * (a_i * a_j) \Leftrightarrow a_i * (y_{jk}) = a_k * (x_{ik}),$$

$$(F_2) \quad (a_i \Delta a_j) * a_k = a_j * (a_i * a_k) \Leftrightarrow (y_{ij}) * a_k = a_j * (x_{ik}) \Leftrightarrow a_i * (y_{jk}) = a_k * (x_{ik}),$$

$$(F_3) \quad a_1 * a_i = a_i \Leftrightarrow x_{1i} = a_i,$$

where $i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$; $k = 1, 2, \dots, n$.

For this algebra system, researchers hope to construct some examples that satisfy the conditions (F₁), (F₂), and (F₃) and to prove the existence of the algebra. However, the calculated amount of this work is so large that computing cost is typically high. Like Example 2, each variable x_{ij} or y_{ij} ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$) can take n different values (a_1, a_2, \dots, a_n); the combinatorial number with the repetition for all x_{ij} and y_{ij} is n^{2n^2} . Obviously it is a NP-hard problem. It will result in the combination blast when the dimension increases. So the enumeration method can hardly be used to construct an example for the algebra system $(S, *, \Delta, 0)$.

Therefore, for this kind of NP-hard problem, this paper proposes an intelligent metaheuristic algorithm to get a set of $(x_{11}, x_{12}, \dots, x_{1n}; \dots; x_{n1}, x_{n2}, \dots, x_{nn})$ and $(y_{11}, y_{12}, \dots, y_{1n}; \dots; y_{n1}, y_{n2}, \dots, y_{nn})$ quickly that satisfy conditions (F₁), (F₂), and (F₃).

2. The Optimization Model of $N(2, 2, 0)$ Algebra System

For the construction example of an $N(2, 2, 0)$ algebra system, we can consider it as an optimized constraint satisfaction problem (OCSP) that satisfies conditions (F₁), (F₂), and (F₃). Then it can be solved by the intelligent optimization algorithm.

To predigest the solving process and promote its practicality, we turn this problem into nonconstrained optimization problem. The optimization model is as follows:

$$\begin{aligned} \min F(\mathbf{X}, \mathbf{Y}) \\ = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n [f_1(a_i, a_j, a_k) + f_2(a_i, a_j, a_k) + f_3(a_i)], \end{aligned} \quad (3)$$

where

$$\begin{aligned} \mathbf{X} &= (x_{11}, x_{12}, \dots, x_{1n}; \dots; x_{n1}, x_{n2}, \dots, x_{nn}) \\ &\text{is algebra } (S, *, 0); \\ \mathbf{Y} &= (y_{11}, y_{12}, \dots, y_{1n}; \dots; y_{n1}, y_{n2}, \dots, y_{nn}) \\ &\text{is algebra } (S, \Delta, 0); \end{aligned} \quad (4)$$

for any a_i, a_j , and a_k ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$; $k = 1, 2, \dots, n$) in $S = \{a_1, a_2, \dots, a_n\}$,

$$a_i * a_j = x_{ij} \in S, \quad a_i \Delta a_j = y_{ij} \in S,$$

$$\begin{aligned} f_1(a_i, a_j, a_k) &= \begin{cases} 0, & a_i * (a_j \Delta a_k) = a_k * (a_i * a_j) \\ 1, & a_i * (a_j \Delta a_k) \neq a_k * (a_i * a_j) \end{cases} \\ f_2(a_i, a_j, a_k) &= \begin{cases} 0, & (a_i \Delta a_j) * a_k = a_j * (a_i * a_k) \\ 1, & (a_i \Delta a_j) * a_k \neq a_j * (a_i * a_k) \end{cases} \\ f_3(a_i) &= \begin{cases} 0, & 0 * a_i = a_i \\ 1, & 0 * a_i \neq a_i. \end{cases} \end{aligned} \quad (5)$$

The functions f_1 , f_2 , and f_3 are to transform the constraint satisfaction problem to a nonconstraint optimization problem by adding the penalty factors. When f_1 , f_2 , and f_3 meet conditions (F₁), (F₂), and (F₃), respectively, the objective values of each functions are 0; otherwise, a penalty factor 1 is as the objective value.

For this problem, there are multiple solutions that meet conditions (F₁), (F₂), and (F₃). Therefore, sometimes researchers hope to get as many solutions as possible. To resolve the multisolution problem, we introduce tube table (tube table: TT) to store the obtained solutions and to avoid duplication of solutions. The definite means are as follows:

$$f_4(\mathbf{X}) = \begin{cases} 0 & \mathbf{X} \notin \text{TT} \\ n & \mathbf{X} \in \text{TT}, \end{cases} \quad (6)$$

where TT is the tube table. f_4 is defined as a penalty function. If \mathbf{X} is not in tube table TT, f_4 is equal to 0. Otherwise, f_4 is assigned a penalty value n .

Under this condition, the objective function can be expressed as

$$\begin{aligned} \min F(\mathbf{X}, \mathbf{Y}) \\ = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n [f_1(a_i, a_j, a_k) \\ + f_2(a_i, a_j, a_k) + f_3(a_i)] + f_4(\mathbf{X}). \end{aligned} \quad (7)$$

It is clear that, for this optimization problem, the objective value of optimal solution is $\min F^* = 0$.

In order to solve the nonconstrained optimization problem, we propose a new harmony search (HS) algorithm which is a big improvement for classical HS algorithm [1] and NGHS [3] method. We called it improved NGHS (INGHS).

3. Classical Harmony Search (HS) and NGHS

Classical harmony search (HS) is derivative-free metaheuristic algorithm [4, 5]. It mimics the improvisation process of music players and attains the ideal harmony by adjusting the HM. HS algorithm has the same feature as genetic algorithm (GA). The HS algorithm is good at identifying the new regions of the searching space in a reasonable time; however, it has difficulties performing a local search for numerical applications [3, 6–8].

So, several variants of HS have been proposed to improve the performance of the HS algorithm [3, 6–14], such as local-best harmony search algorithm with dynamic subpopulations (DLHS) [6], self-adaptive global best harmony search algorithm (SGHS) [7], intelligent tuned harmony search (ITHS) [8] algorithm, and exploratory power of the harmony search (EHS) [3] algorithm. In addition, Zou et al. presented a novel global harmony search algorithm for unconstrained problems (NGHS) [9, 10]. The literature [11] presents the recent advances in HS algorithm. Tuo and Yong proposed HSTL methods for large scale optimization problem and presented an improved harmony search with chaos (HSCH) [12], as well as other variants [13–15] of HS and some applications [16–19] of HS.

Consider an optimization model as follows:

$$\begin{aligned} \min \quad & f(\mathbf{X}), \quad \mathbf{X} = (x_1, x_2, \dots, x_D) \\ \text{subject to} \quad & x_i \in [x_i^L, x_i^U], \quad i = 1, 2, \dots, D, \end{aligned} \quad (8)$$

where $f(\mathbf{X})$ is the optimization objective function, \mathbf{X} is the solution vector that consists of D decision variables x_i ($i = 1, 2, \dots, D$), and $[x_i^L, x_i^U]$ is feasible range of values for decision variable x_i .

In the HS algorithm, \mathbf{X} represents the harmony and $f(\mathbf{X})$ denotes the melody of harmony \mathbf{X} . HS uses three rules (memory consideration, pitch adjustments, and randomization) to optimize the harmony memories (HM) composed of HMS harmony vectors.

3.1. The Classical Harmony Search (HS) Algorithm. 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 vectors. Each harmony vector is generated from a uniform distribution in the feasible space, as

$$\begin{aligned} x_i^j &= x_i^L + r \cdot (x_i^U - x_i^L), \\ i &= 1, 2, \dots, D; \quad j = 1, 2, \dots, \text{HMS}, \end{aligned} \quad (9)$$

where D and HMS represent the number of decision variables and the size of harmony memory, respectively, and r denotes a uniform distribution random number between 0 and 1, as

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

Step 2 (improvise a new harmony via three rules). There are three rules that can be used to improvise a new harmony vector \mathbf{X}^{new} .

(a) *Memory Consideration.* A decision variable value of the harmony vector will be adopted 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 $1 - \text{HMCR}$.

The improvisation procedure of a new harmony vector works as Algorithm 2.

A trial harmony vector \mathbf{X}^{new} is generated in Step 2. Next, in Step 3, it will be decided whether to survive.

Step 3 (select operator). Get the worst harmony vector $\mathbf{X}^{\text{worst}}$ from the HM (see Algorithm 1).

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

3.2. The NGHS Algorithm. In the novel global harmony search (NGHS) algorithm [9], three significant parameters, harmony memory considering rate (HMCR), fret width (FW), and pitch adjusting rate (PAR), are excluded from NGHS and a random selection rate (p_m) is included in the NGHS. In Step 3, NGHS works as Algorithm 3, where x_i^{best} and x_i^{worst} , respectively, represent indexes of the best harmony and the worst harmony in HM. $\text{rand}(0, 1)$ is a uniformly generated random number in $[0, 1]$, and it should set parameter $p_m = 2/D$ for the 0-1 knapsack problem and set parameter $p_m = 0.01 \sim 0.1$ for continuous optimal problem.

3.3. The Proposed HS Method. In this section, we proposed a new harmony search algorithm which is improved based on classical HS algorithm and NGHS method, so we call the proposed algorithm INGHS.

Since the HS algorithm origination, it has been applied to many practical optimization problems. However, for large scale optimization problems, classical HS has slow convergence and low precision, which is due to the fact that because a new decision variable value in harmony memory (HM) can

```

If fitness ( $\mathbf{X}^{\text{new}}$ ) is better than fitness ( $\mathbf{X}^{\text{worst}}$ ) then
     $\mathbf{X}^{\text{worst}} = \mathbf{X}^{\text{new}}$ ; ( $\mathbf{X}^{\text{worst}}$  is the worst harmony vector in HM).
EndIf

```

ALGORITHM 1

```

For  $i = 1$  to  $D$  do
    If  $\text{rand}(0, 1) < \text{HMCR}$  then
         $x_i^{\text{new}} = x_i^j, j \in U\{1, 2, \dots, \text{HMS}\}$ 
        If  $\text{rand}(0, 1) < \text{PAR}$  then
             $x_i^{\text{new}} = x_i^{\text{new}} \pm \text{rand}() \times \text{FW}(i)$ ; //FW: fretwidth
             $x_i^{\text{new}} = \min(\max(x_i^{\text{new}}, x_i^L), x_i^U)$ 
        Endif
    Else
         $x_i^{\text{new}} = x_i^L + (x_i^U - x_i^L) \times \text{rand}()$ 
    Endif
End For

```

ALGORITHM 2: The improvisation procedure of new harmony vector by HS.

```

For  $i = 1$  to  $D$  do
     $x_i^r = 2 \times x_i^{\text{best}} - x_i^{\text{worst}}$ 
     $x_i^r = \min(\max(x_i^r, x_i^L), x_i^U)$ 
     $x_i^{\text{new}} = x_i^{\text{worst}} + \text{rand}(0, 1) \times (x_i^r - x_i^{\text{worst}})$ 
    If  $\text{rand}(0, 1) \leq p_m$  %random mutation
         $x_i^{\text{new}} = x_i^L + (x_i^U - x_i^L) \times \text{rand}(0, 1)$ 
    Endif
EndFor

```

ALGORITHM 3: The NGHS modifies improvisation step.

be generated only by using pitch adjustment and randomization strategies during the search procedure, the memory consideration rule is only used to adjust the existing decision variable values according to the harmony memory (HM). Thus HS can maintain a strong performance of exploration, but not a good performance of exploitation, and, in the later stage of search, it is characterized by slow convergent. Therefore, for solving the large scale optimization problem, the key is to balance the global exploration performance and the local exploitation ability.

Because the construction example of the algebra is a large scale high-dimensional optimization problem, to achieve the most satisfactory optimization performance by applying the HS algorithm to a given problem, we adopt four optimization strategies and dynamical parameter control method to balance the global exploration power and the local exploitation ability.

It is of very importance between the convergence and the diversity in order to improve the efficiency of the search. In the classical HS algorithm, a new harmony is generated in Step 2. After the selecting operation in Step 3, the population

variance may increase or decrease. With a high population variance, the diversity and exploration power will increase, and in 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 [8]; the diversity and the exploration power will decrease. So it is significant how to keep the balance between the convergence and the diversity. Classical HS algorithm loses its ability easily at the later evolution process [3], because of improvising a new harmony from HM with a high HMCR and local adjusting with PAR. And HM diversity decreases gradually from the early iteration to the last. However, in HS algorithm, a low HMCR employed will increase the probability (1-HMCR) of random select in search space; the exploration power will improve, but the local search ability and the exploitation accuracy cannot be improved by the single pitch adjusting strategy.

To overcome the inherent weaknesses of HS, in this section, we present INGHS method to construct example for algebra. The INGHS algorithm works as Algorithm 4 and Figure 1.

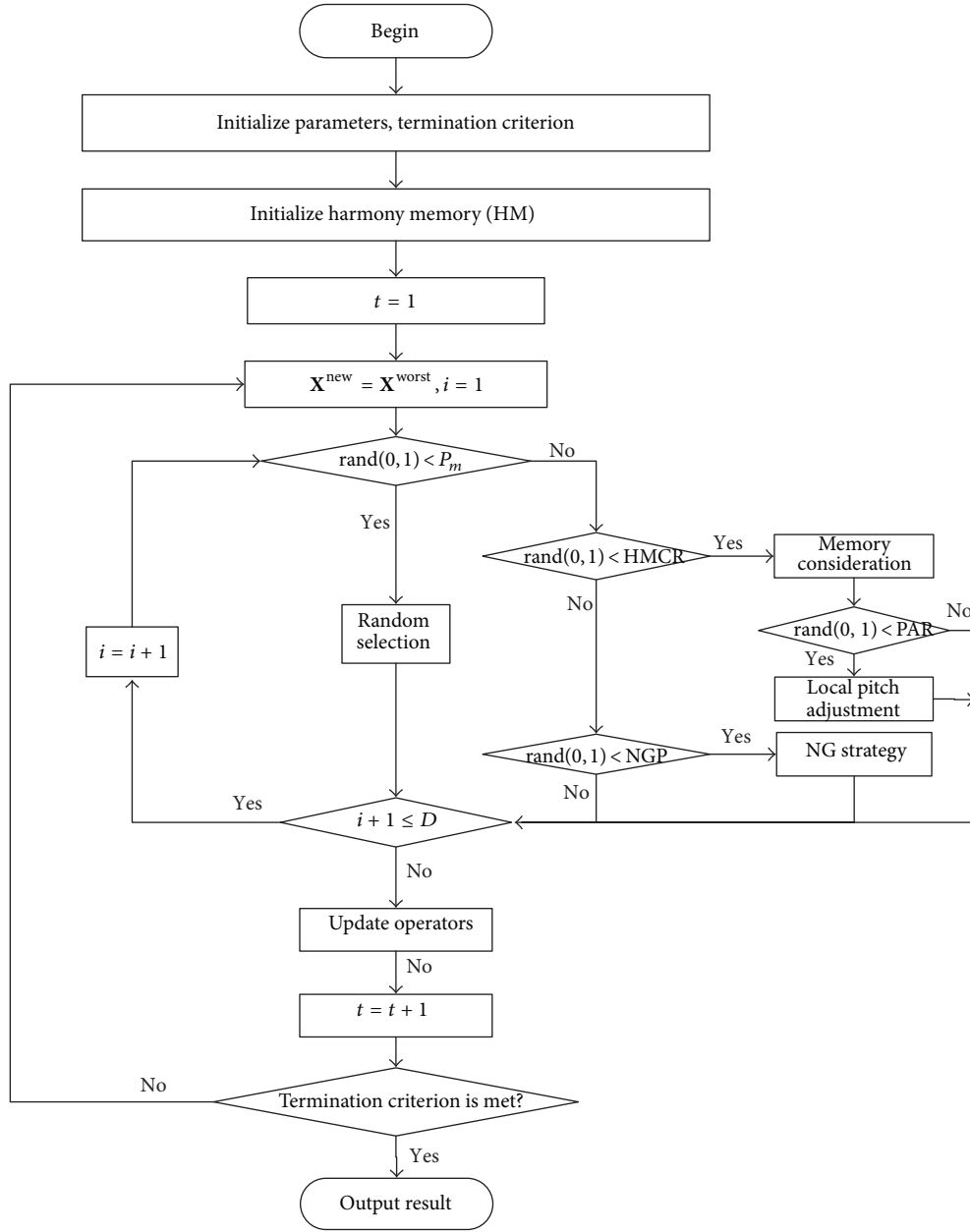


FIGURE 1: The flow chart of the proposed algorithm.

In Algorithm 4, to resolve the discrete optimization problems, the function $\text{Round}(x)$ is used to round each element of x to the nearest integer.

(1) *Novel Global (NG) Best Strategy.* In NGHS algorithm, the novel global best strategy is very effective to explore the global best solution. So, in the proposed method, the novel global best strategy is adopted (see Algorithm 5), where the best and worst, respectively, represent indexes of the best harmony and the worst harmony in HM.

(2) *Parameters Dynamically Changed.* To balance the exploration and exploitation power of the INGHS algorithm efficiently, HMCR, PAR, FW, and NGP parameters are

dynamically adapted to a suitable range with the increase of generations. Equation (11) shows the dynamic change of HMCR, PAR, and NGP, respectively,

$$\text{HMCR} = \text{HMCR}_{\min} + (\text{HMCR}_{\max} - \text{HMCR}_{\min}) \times \left(\frac{t}{T_{\max}} \right),$$

$$\text{NGP} = \text{NGP}_{\min} + (\text{NGP}_{\max} - \text{NGP}_{\min}) \times \left(\frac{t}{T_{\max}} \right)^3, \quad (11)$$

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

(see [8]).

```

Xnew = Xworst   %%% worst is the index of the worst harmony in HM
For i = 1 to D
  If rand(0,1) < Pm
    % Random playing: randomly select any pitch within bounds
     $x_i^{\text{new}} = \text{Round}(x_i^L + (x_i^U - x_i^L) \times \text{rand}(0, 1))$ 
  Else
    If rand(0,1) < HMCR
      % Memory considering: randomly select a note stored in HM
       $x_i^{\text{new}} = x_i^a; a \in U(1, 2, \dots, \text{HMS})$ 
      If rand(0,1) < PAR
        % Pitch adjusting: randomly adjust the pitch slightly
         $x_i^{\text{new}} = \text{Round}(x_i^{\text{new}} \pm \text{rand}(0, 1) \times \text{FW}(i))$ 
      EndIf
    ElseIf rand(0,1) < NGP %
       $x_r = \text{Round}(2x_i^{\text{best}} - x_i^{\text{worst}})$ 
       $x_i^{\text{new}} = \text{Round}(x_i^{\text{worst}} + \text{rand}(0, 1) \times (x_r - x_i^{\text{worst}}))$ 
    EndIf
  EndIf
EndFor % Finished improvising a new harmony

```

ALGORITHM 4: The pseudocode of the proposed algorithm INGHS.

```

If rand(0, 1) < NGP (NGP is the rate of choosing the novel global best strategy)
   $x_r = \text{Round}(2x_i^{\text{best}} - x_i^{\text{worst}})$ 
   $x_i^{\text{new}} = \text{Round}(x_i^{\text{worst}} + \text{rand}(0, 1) \times (x_r - x_i^{\text{worst}}))$ 
End

```

ALGORITHM 5

It can be seen that the parameter HMCR gradually increased from HMCR_{\min} to HMCR_{\max} linearly and the parameter NGP increased with low velocity in the early stage and it increased sharply in the final stage. That is because, in the beginning, in order to explore the global optimal solution, the harmony consideration rules and NG strategies are carried out with a smaller probability, and, in later stage, INGHS methods begin to focus on the local exploitation that needs a high probability to employ the NG strategy and harmony consideration rules. The benefits of doing so can get more opportunities to reinforce the global exploration by strengthening disturbance in the early stage and can acquire high precision solution by carrying local intensification search in the later stage. For the same reason, FW is decreased gradually in order to reduce perturbation step size step by step, and the variation of PAR from 0.55 to 0.3 is to reduce the probability of pitch adjustment.

3.4. The Construction Example for $N(2, 2, 0)$ Algebra System with the Proposed HS Method. For an $N(2, 2, 0)$ algebra system, there are multiple solutions that meet the conditions. Sometimes we need to obtain multiple solutions. So we adopt the proposed HS algorithm and tube table technology to resolve the multisolution problem. Its flow chart is as shown in Figure 2 and in Pseudocode 1.

3.5. Effect of Parameters HMCR and NGP on INGHS Performance. In this section, we determine the effect of parameters HMCR_{\max} , HMCR_{\min} , NGP_{\max} , and NGP_{\min} on the performance of the proposed algorithm INGHS. The experiments are investigated for algebra system $(S, *, \Delta, 0)$, where $S = \{a, b, c, d, e, f, g\}$.

(1) *Investigation for NGP_{\max} and NGP_{\min}* (initialize $\text{HMCR}_{\max} = 0.9$, $\text{HMCR}_{\min} = 0.6$). Let NGP_{\min} change from 0.1 to 0.6; then, for each NGP_{\min} , let $\text{NGP}_{\max} = 0.5$ to 1. So there are $6 \times 6 = 36$ pairs of parameters $(\text{NGP}_{\min}, \text{NGP}_{\max})$. For each pair of parameters $(\text{NGP}_{\min}, \text{NGP}_{\max})$, we execute the proposed algorithm (INGHS) $|S|^4$ ($7^4 = 2401$) times and then record the number of solutions obtained. The result is shown in Figure 3.

(2) *Investigation for HMCR_{\max} and HMCR_{\min}* (initialize $\text{NGP}_{\max} = 0.5$, $\text{NGP}_{\min} = 0.15$). Let $\text{HMCR}_{\min} = 0.2$ to 0.7 and let $\text{HMCR}_{\max} = 0.5$ to 1, respectively. For each pair of parameters $(\text{HMCR}_{\min}, \text{HMCR}_{\max})$, the proposed algorithm INGHS is performed $|S|^4$ ($7^4 = 2401$) times and then we record the number of solutions obtained. The number of solutions obtained is shown in Figure 4.

It can be seen from Figure 3 that most solutions can be obtained when $\text{NGP}_{\min} = 0.2$ and $\text{NGP}_{\max} = 0.5$.

```

Set TT as empty
For  $L = 1$  to Max Loop times (ML)
    Run the proposed HS algorithm to get a best solution  $\mathbf{X}^{new}$ ;
    If the objective function value  $F(\mathbf{X}^{new}) == 0$ 
        Put  $\mathbf{X}^{new}$  into tube table TT.
    EndIf
EndFor
    
```

PSEUDOCODE 1

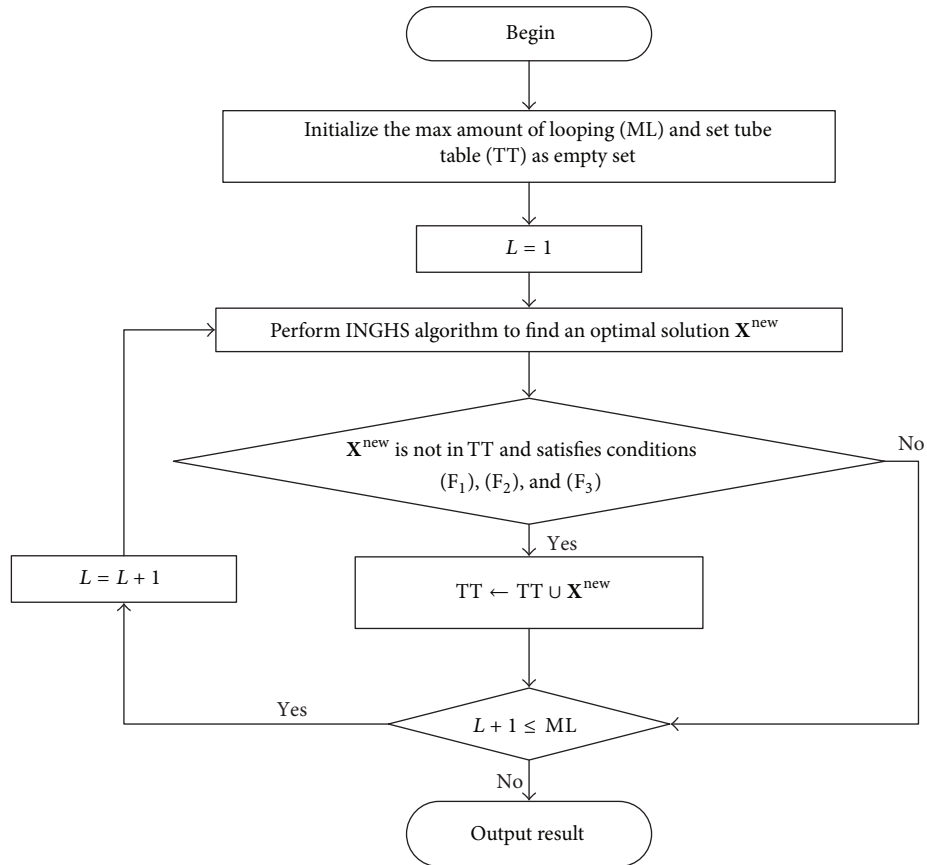


FIGURE 2: The flow chart for construction example of algebra system based on INGHS algorithm.

From Figure 4, we find that most solutions are found when $HMCR_{min} = 0.6$ and $HMCR_{max} = 0.9$.

From the above, the proposed algorithm has better performance when it is set as $HMCR_{min} = 0.6$, $HMCR_{max} = 0.9$, $NGP_{min} = 0.2$, and $NGP_{max} = 0.5$.

4. Computational Experiments and Results

In this section, we have tested the proposed algorithm over a set of $N(2, 2, 0)$ algebra system construction examples problems.

4.1. The Proposed Algorithm for Solving the Construction Examples of Algebra System. Definition 1 is chosen as the

optimization objective algebra system. For Definition 1, we, respectively, set the following:

- (1) $S = \{a, b\}$;
- (2) $S = \{a, b, c\}$;
- (3) $S = \{a, b, c, d\}$;
- (4) $S = \{a, b, c, d, e\}$;
- (5) $S = \{a, b, c, d, e, f\}$;
- (6) $S = \{a, b, c, d, e, f, g\}$;

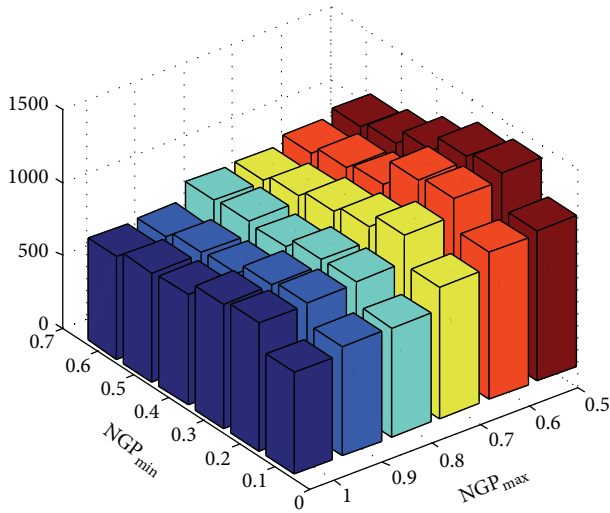


FIGURE 3: Variation of number of solutions with change in NGP_{max} and NGP_{min} .

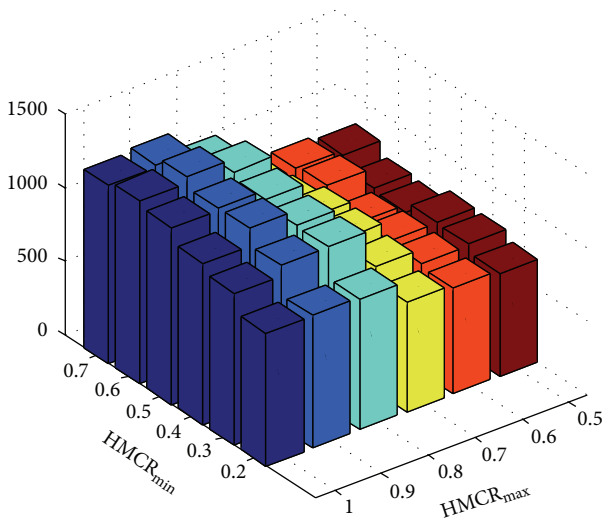


FIGURE 4: Variation of number of solutions with change in $HMCR_{max}$ and $HMCR_{min}$.

$$(7) S = \{a, b, c, d, e, f, g, h\};$$

$$(8) S = \{a, b, c, d, e, f, g, h, i\};$$

$$(9) S = \{a, b, c, d, e, f, g, h, i, j\}; n = |S| \text{ represents the number of set } S, D = 2n^2.$$

Then we use the proposed algorithm to construct the examples for the algebra system on S .

In order to deal with the optimization problems easily, we replace the elements $a, b, c, d, e, f, g, h, i$, and j with 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10, respectively. So, for each S , the upper limit of variables about optimization problems is set as $|S|$ and the lower limit is set as 1.

4.2. The Parameters Setting for Algorithms. In the experiment test, we set the following parameter values: $T_{max} = 500 \times D$; for NGHS algorithm, $HMS = 10$, $p_m = \min(0.01, 2/D)$; for the proposed algorithm (INGHS), $HMS = 10$, $HMCR_{max} = 0.9$, $HMCR_{min} = 0.6$, $NGP_{max} = 0.55$, $NGP_{min} = 0.15$, $PAR_{max} = 0.8$, $PAR_{min} = 0.25$, $FW_{max} = n/2$, $FW_{min} = n/10$, and $p_m = \min(0.01, 2/D)$.

When $n = 2$ and $n = 3$, there are 3 and 16 solutions for the $N(2, 2, 0)$ algebra system, respectively. So we set max looping times (ML) = 3 and 16, respectively. When $n > 3$ and $n < 9$, we do not know how many solutions are there for the algebra system. So we set $ML = n^4$. However, with the increasing of n , there are much more solutions, and computer's running time is very long. So we set $ML = 8000$ and 5000 when $n = 9$ and $n = 10$.

4.3. The Experiment Results and Analysis. In order to evaluate the performance of the proposed algorithm INGHS, we compared its success rate of finding the solutions with HS, HSTL, and NGHS algorithms in the same conditions. The results are shown in Table 1.

When $S = \{a, b\}$ ($n = 2$), the solutions, as shown in (12a)–(12c), are obtained by using the proposed algorithm. And when $S = \{a, b, c\}$ ($n = 3$), the solutions obtained by INGHS method are shown in (13a)–(13p). For other instances ($n > 3$), due to the existence of many solutions, we only present one solution for each instance. All is shown in (14)–(20).

The first solution on $n = 2$ is as follows:

$$\begin{array}{c|cc} * & a & b \\ \hline a & a & b \\ b & a & b \end{array} \quad \begin{array}{c|cc} \Delta & a & b \\ \hline a & a & a \\ b & b & b \end{array} \quad (12a)$$

The second solution on $n = 2$ is as follows:

$$\begin{array}{c|cc} * & a & b \\ \hline a & a & b \\ b & b & a \end{array} \quad \begin{array}{c|cc} \Delta & a & b \\ \hline a & a & b \\ b & b & a \end{array} \quad (12b)$$

The third solution on $n = 2$ is as follows:

$$\begin{array}{c|cc} * & a & b \\ \hline a & a & b \\ b & b & b \end{array} \quad \begin{array}{c|cc} \Delta & a & b \\ \hline a & a & b \\ b & b & b \end{array} \quad (12c)$$

The first solution on $n = 3$ is as follows:

$$\begin{array}{c|ccc} * & a & b & c \\ \hline a & a & b & c \\ b & b & b & b \\ c & c & b & c \end{array} \quad \begin{array}{c|ccc} \Delta & a & b & c \\ \hline a & a & b & c \\ b & b & b & b \\ c & c & b & c \end{array} \quad (13a)$$

The second solution on $n = 3$ is as follows:

$$\begin{array}{c|ccc} * & a & b & c \\ \hline a & a & b & c \\ b & b & b & b \\ c & a & b & c \end{array} \quad \begin{array}{c|ccc} \Delta & a & b & c \\ \hline a & a & b & a \\ b & b & b & b \\ c & c & b & c \end{array} \quad (13b)$$

TABLE 1: The success rate comparison of the proposed method with HS and NGHs algorithm.

n	The number of solutions	HS				HSTL				NGHS				INGHS			
		Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate	
2	3	3	3	100.00%	3	3	100.00%	3	100.00%	3	3	100.00%	3	3	100.00%		
3	16	16	15	93.75%	16	16	100.00%	16	100.00%	16	16	100.00%	16	16	100.00%		
4	Unknown	256	127	49.61%	256	148	57.81%	256	57.81%	256	127	49.61%	256	162	63.28%		
5	Unknown	625	485	77.60%	625	600	96.00%	625	96.00%	625	579	92.64%	625	601	96.16%		
6	Unknown	1296	1016	78.40%	1296	1258	97.07%	1296	97.07%	1296	1243	95.91%	1296	1271	98.07%		
7	Unknown	2401	1834	76.38%	2401	2212	92.13%	2401	92.13%	2401	2119	88.25%	2401	2217	92.34%		
8	Unknown	4096	79	1.93%	4096	3364	82.13%	4096	82.13%	4096	3174	77.49%	4096	3383	82.59%		
9	Unknown	8000	0	0.00%	8000	5763	72.04%	8000	72.04%	8000	4768	59.60%	8000	6151	76.89%		
10	Unknown	5000	0	0.00%	5000	3451	69.02%	5000	69.02%	5000	3251	65.02%	5000	3749	74.98%		

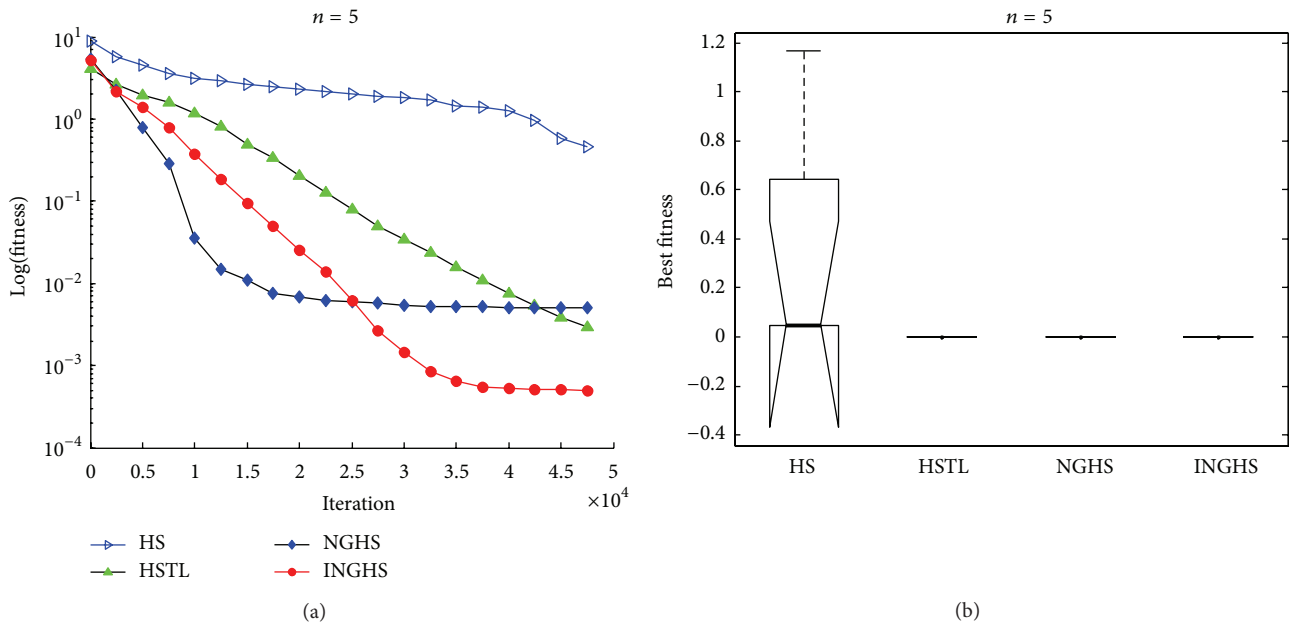


FIGURE 5: The convergence curve and the box plot when $n = 5$.

One of the solutions on $n = 6$ ($S = \{a, b, c, d, e, f\}$) is as follows:

*	a	b	c	d	e	f
a	a	b	c	d	e	f
b	a	b	c	d	e	f
c	c	c	c	d	c	c
d	d	d	d	d	d	d
e	e	e	c	d	e	e
f	a	b	c	d	e	f

(16)

Δ	a	b	c	d	e	f
a	a	a	c	d	e	a
b	b	b	c	d	e	b
c	c	c	c	d	c	c
d	d	d	d	d	d	d
e	e	e	c	d	e	e
f	f	f	c	d	e	f

One of the solutions on $n = 7$ ($S = \{a, b, c, d, e, f, g\}$) is as follows:

*	a	b	c	d	e	f	g
a	a	b	c	d	e	f	g
b	a	b	c	d	e	f	g
c	a	b	c	d	e	f	g
d	d	e	d	d	e	f	e
e	d	e	d	d	e	f	e
f	f	f	f	f	f	f	f
g	d	e	d	d	e	f	e

Δ	a	b	c	d	e	f	g
a	a	a	a	d	d	f	d
b	b	b	b	e	e	f	e
c	c	c	c	d	d	f	d
d	d	d	d	d	d	f	d
e	e	e	e	e	e	f	e
f	f	f	f	f	f	f	f
g	g	g	g	e	e	f	e

(17)

One of the solutions on $n = 8$ ($S = \{a, b, c, d, e, f, g, h\}$) is as follows:

*	a	b	c	d	e	f	g	h
a	a	b	c	d	e	f	g	h
b	a	b	c	d	e	f	g	h
c	c	c	c	c	c	c	c	c
d	d	d	c	c	c	c	d	d
e	e	e	c	c	e	f	e	e
f	c	c	c	c	c	c	c	c
g	a	b	c	d	e	f	g	h
h	a	b	c	d	e	f	g	h

Δ	a	b	c	d	e	f	g	h
a	a	a	c	d	e	c	a	a
b	b	b	c	d	e	c	b	b
c	c	c	c	c	c	c	c	c
d	d	d	c	c	c	c	d	d
e	e	e	c	c	e	c	e	e
f	f	f	c	c	f	c	f	f
g	g	g	c	d	e	c	g	g
h	h	h	c	d	e	c	h	h

(18)

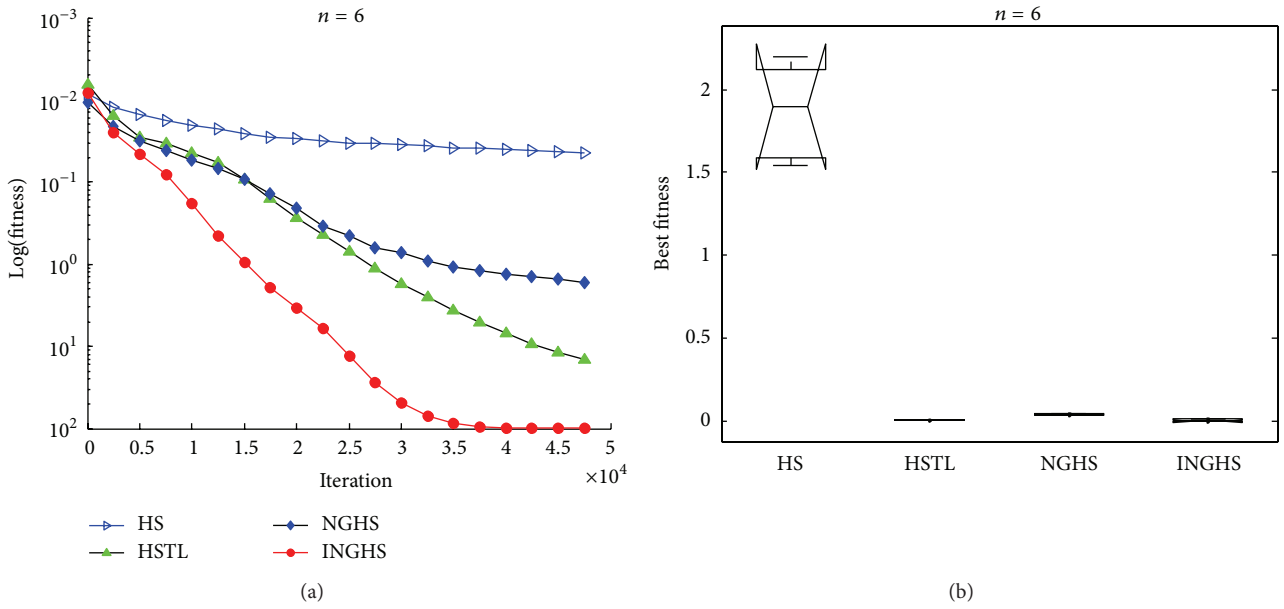


FIGURE 6: The convergence curve and the box plot when $n = 6$.

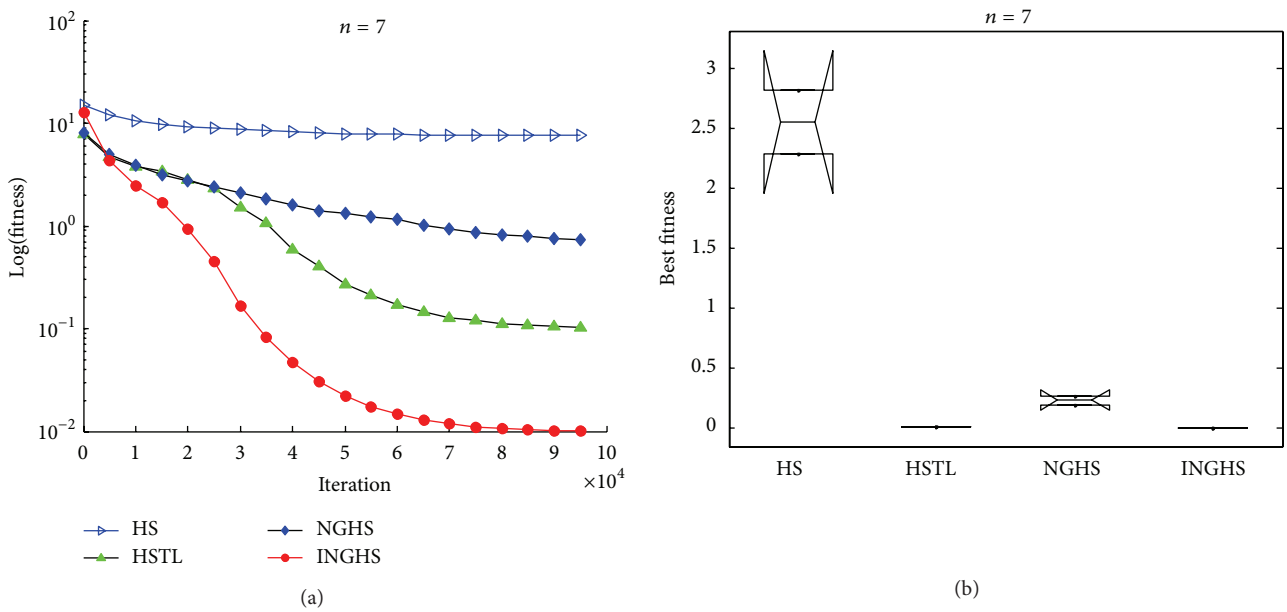


FIGURE 7: The convergence curve and the box plot when $n = 7$.

One of the solutions on $n = 9$ ($S = \{a, b, c, d, e, f, g, h, i\}$) is as follows:

*	a	b	c	d	e	f	g	h	i
a	a	b	c	d	e	f	g	h	i
b	b	b	b	b	b	b	b	b	b
c	c	b	c	b	e	e	g	b	c
d	d	b	b	b	b	d	b	b	d
e	c	b	c	b	e	e	g	b	c
f	a	b	c	d	e	f	g	h	i
g	b	b	b	b	b	b	b	b	b
h	b	b	b	b	b	b	b	b	b
i	a	b	c	d	e	f	g	h	i

Δ	a	b	c	d	e	f	g	h	i
a	a	b	c	d	c	a	b	b	a
b	b	b	b	b	b	b	b	b	b
c	c	b	c	b	c	c	b	b	c
d	d	b	b	b	b	d	b	b	d
e	e	b	e	b	e	e	b	b	e
f	f	b	e	d	e	f	b	b	f
g	g	b	g	b	g	g	b	b	g
h	h	b	b	b	b	h	b	b	h
i	i	b	c	d	c	i	b	b	i

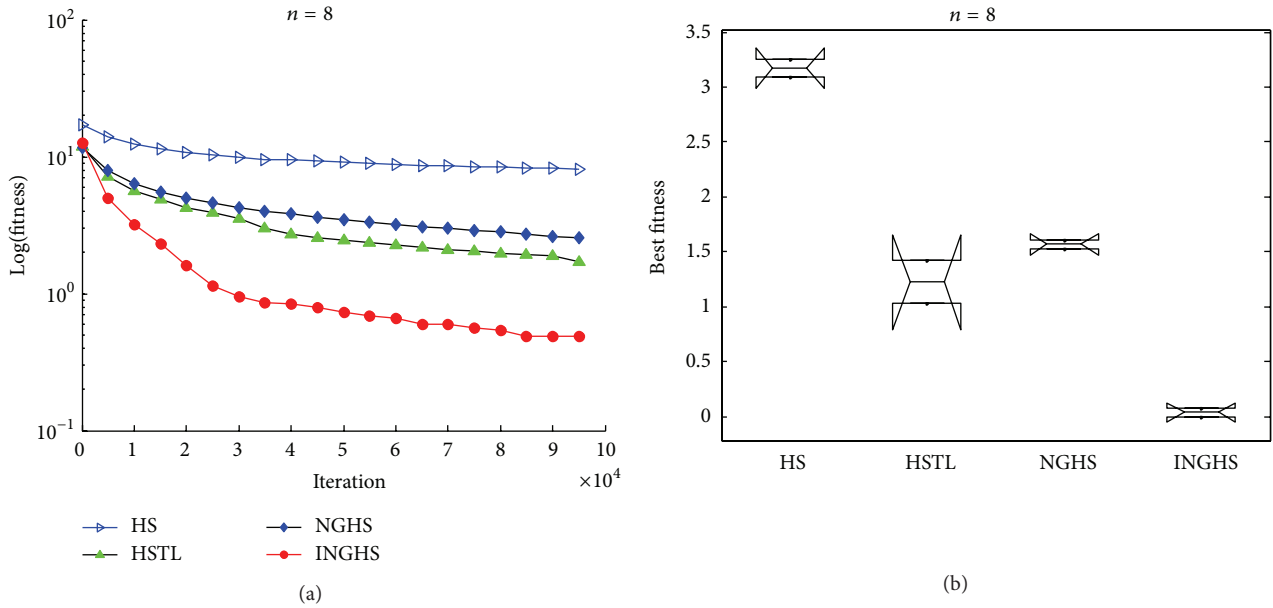


FIGURE 8: The convergence curve and the box plot when $n = 8$.

One of the solutions on $n = 10$ ($S = \{a, b, c, d, e, f, g, h, i, j\}$) is as follows:

*	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>a</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>b</i>	<i>e</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>c</i>	<i>e</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>d</i>	<i>d</i>	<i>e</i>	<i>c</i>	<i>f</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>e</i>	<i>e</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>f</i>	<i>f</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>g</i>	<i>e</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>h</i>	<i>h</i>	<i>b</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>h</i>	<i>e</i>	<i>e</i>
<i>i</i>	<i>i</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>j</i>	<i>e</i>	<i>e</i>	<i>c</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>g</i>	<i>e</i>	<i>e</i>	<i>e</i>
Δ	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>a</i>	<i>a</i>	<i>e</i>	<i>e</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>e</i>	<i>h</i>	<i>i</i>	<i>e</i>
<i>b</i>	<i>b</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>b</i>	<i>e</i>	<i>e</i>
<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>
<i>d</i>	<i>d</i>	<i>e</i>	<i>e</i>	<i>f</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>f</i>	<i>f</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>
<i>h</i>	<i>h</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>h</i>	<i>e</i>	<i>e</i>
<i>i</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>j</i>	<i>j</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>

From Table 1, it can be seen obviously that the success rate of all instances of the proposed method is higher than HS, HSTL, and NGHS.

For instance, $n = 6, 7, 8, 9$, and 10 , and Figures 5, 6, 7, 8, 9, and 10 show the convergence curves of three algorithms (HS, NGHS, and INGHS). It is evident from Figures 5–10 that the proposed algorithm is better than HS and NGHS for all

instances. The convergence curve of the proposed algorithm can maintain falling until it finds the best solution. When $n > 6$, the convergence curve of INGHS method falls faster than HS, HSTL, and NGHS algorithms, and it keeps declining actively until the best solution is found.

5. Conclusions

This paper has investigated nine construction example problems of algebra system and converted the construction example problems into optimization problems. A novel harmony search algorithm (INGHS) is proposed to solve the problems. Global best strategy and dynamic parameters adjustment are employed in INGHS. The experimental results on 9 instances of algebra system demonstrate that the proposed algorithm is more effective than HS, HSTL, and NGHS algorithms. Further research will investigate the INGHS algorithm to solve the combinatorial optimization problems and some practical optimization problems.

Conflict of Interests

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

Acknowledgments

The authors would like to thank all the editors and referees for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grant no. 11401357 and no. 81160183, the Scientific Research Program funded by Shaanxi Provincial Education Department

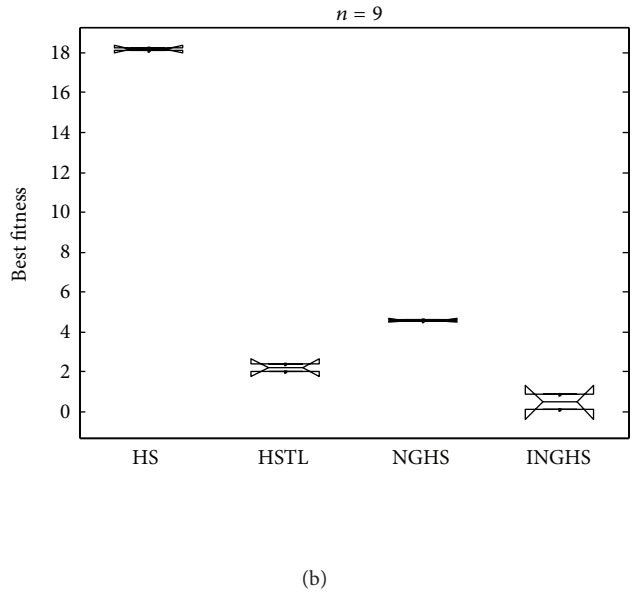
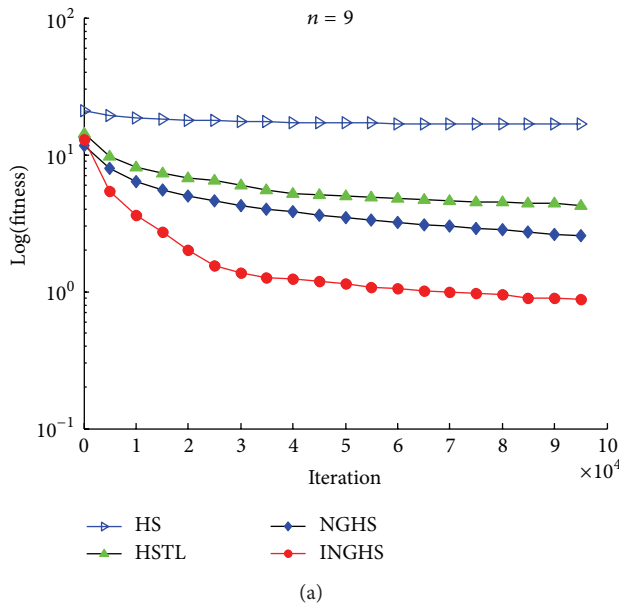


FIGURE 9: The convergence curve and the box plot when $n = 9$.

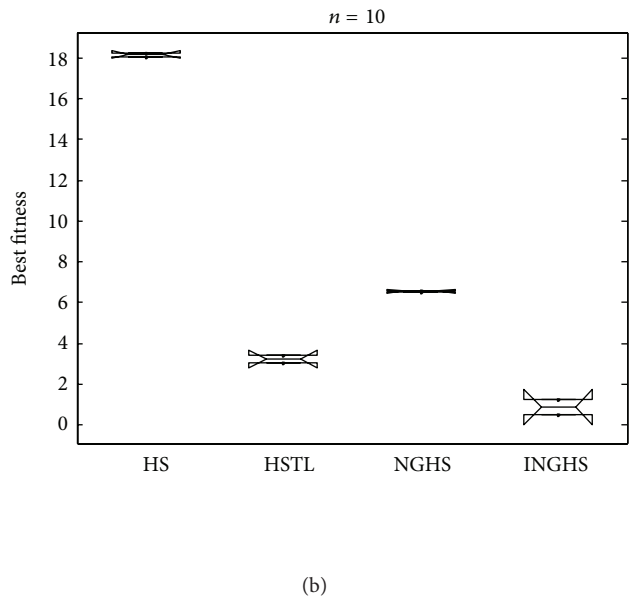
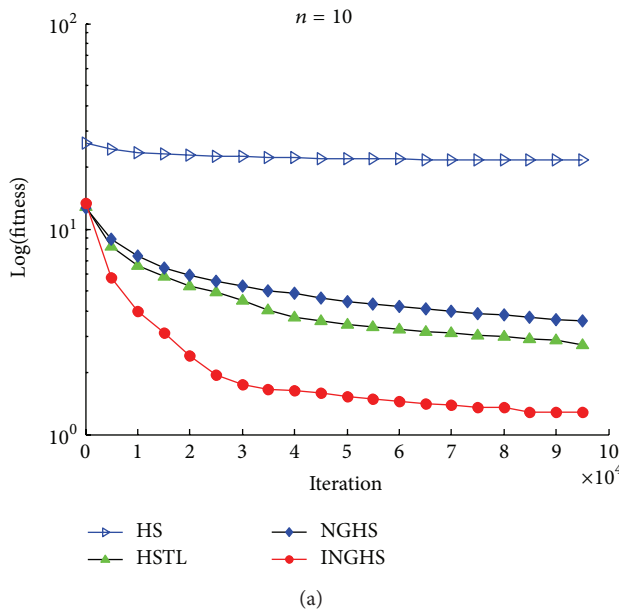


FIGURE 10: The convergence curve and the box plot when $n = 10$.

under Grant no. 14JK1141, and Hanzhong Administration of Science & Technology under Grant no. 2013hzzx-39.

References

[1] F. Deng and Y. Xu, "On the $N(2,2,0)$ algebra," *Journal of Southwest Jiao Tong University*, vol. 31, no. 4, pp. 457–653, 1996.

[2] F. Deng and L. Yong, "Regular semigroups of an $N(2,2,0)$ algebra," *Journal of Advances in Mathematics*, vol. 41, no. 6, pp. 665–671, 2012.

[3] 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, Part B: Cybernetics*, vol. 41, no. 1, pp. 89–106, 2011.

[4] 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.

[5] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

- [6] 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.
- [7] 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.
- [8] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, "An intelligent tuned harmony search algorithm for optimisation," *Information Sciences*, vol. 196, pp. 47–72, 2012.
- [9] 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.
- [10] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 307–316, 2010.
- [11] Z. W. Geem, *Recent Advances in Harmony Search Algorithm*, Springer, New York, NY, USA, 2010, <https://sites.google.com/a/hydrroteq.com/www/HS.Structure.pdf>.
- [12] S. Tuo and L. Yong, "Improved harmony search algorithm with chaos," *Journal of Computational Information Systems*, vol. 8, no. 10, pp. 4269–4276, 2012.
- [13] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.
- [14] A. Kattan and R. Abdullah, "A dynamic self-adaptive harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 16, pp. 8542–8567, 2013.
- [15] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, and M. Fei, "An improved adaptive binary harmony search algorithm," *Information Sciences*, vol. 232, pp. 58–87, 2013.
- [16] 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.
- [17] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, vol. 233, pp. 126–140, 2013.
- [18] N. Sinsuphan, U. Leeton, and T. Kulworawanichpong, "Optimal power flow solution using improved harmony search method," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2364–2374, 2013.
- [19] H. Wang, X. Yuan, Y. Wang, and Y. Yang, "Harmony search algorithm-based fuzzy-PID controller for electronic throttle valve," *Neural Computing and Applications*, vol. 22, no. 2, pp. 329–336, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

