Swarm and Evolutionary Computation 48 (2019) 31–43

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Review of harmony search with respect to algorithm structure

Tonghua Zhang [a], Zong Woo Geem [b,*]

[a] Department of Mathematics, Swinburne University of Technology, Hawthorn, Victoria, 3122, Australia
[b] Department of Energy IT, Gachon University, Seongnam, 13120, South Korea

ARTICLE INFO

Keywords:
Harmony search
Optimisation
Hybrid method
Algorithm structure
Metaheuristic optimisation algorithm

ABSTRACT

Harmony Search (HS) is a metaheuristic optimisation algorithm inspired by musical improvisation. So far it has been applied to various optimisation problems, and there are several application-oriented review papers. However, this review paper tries to focus on the historical development of algorithm structure instead of applications. This paper explains the original HS algorithm along with a selection of modified and hybrid HS methods: adaption of original operators of the basic harmony search, parameter adaption, hybrid methods, handling multi-objective optimisation problems and constraint handling.

1. Introduction

Harmony search is a population-based metaheuristic optimisation algorithm, which was initially proposed by Zong Woo Geem in his PhD thesis [1] even though it is now well known from the first journal publication [2]. Since then, it has obtained great success in various engineering applications, resulting numerous publications as suggested in Fig. 1. It shows a sustained and growing interest in harmony search method and its variants, particularly since 2011 in terms of number of publications on using of harmony search technique as shown in Fig. 1, but it is since 2006 in terms of the major modifications being made to change the algorithm structure as suggested in Fig. 2. Several monographs [3–6] and survey articles [7–17] have addressed the advances of harmony search technique and its variants, but as far as we know that there is no such review addressing harmony search technique from the aspect of algorithm structure thoroughly. In this review article, we aim to acknowledge the advances of harmony search from the point of view of the algorithm structure. We will try our best to include as many as such advances, see Fig. 1.

2. The algorithm structure of the basic harmony search

To clearly see the modifications made in the structure of harmony search, in this section, we will briefly recall the original technique, referring to the basic Harmony Search (HS) algorithm by Geem [1] for solving the optimisation problems with discrete variables and then adapted for cases with continuous decision variables and mixed variables by Lee & Geem [18].

Let $f(\cdot)$ be the objective function and $\mathbf{X} = (x_1, \ldots, x_N)$, where $x_i, i = 1, \ldots, N$ are the decision or design variables. For the convenience of discussion, we state the optimisation problem as follows

$$\text{Minimise } f(\mathbf{X}) \text{ by varying } \mathbf{X} \tag{1}$$

subject to constraints

$$\mathbf{g}(\mathbf{X}) \le \mathbf{0} \text{ and } \mathbf{h}(\mathbf{X}) = \mathbf{0}. \tag{2}$$

Here $x_i \in [x_i^L, x_i^U]$ for continuous variables, $x_i \in \mathbf{X}_i = \left\{ x_{i,j}, j = 1, \ldots, K_i \mid x_{i,1} < x_{i,2} < \cdots < x_{i,K_i} \right\}$ for discrete variables, and N is the number of decision variables and K_i is the number of possible values for discrete variable x_i. For the basic HS, the algorithm structure is summarised as a flowchart in Fig. 3.

The basic HS algorithm has three key parameters, being listed and explained in Table 1, and other parameters, such as the maximum number of improvisations (MaxImp) and bandwidth, which later was renamed as fret width, vector (bw) used in pitch adjustment for the implementation of the algorithm. The convergence capability of the basic HS in terms of these parameters was discussed by Geem and his collaborators [19].

As shown in Fig. 3 and pointed out by Geem et al. [2,20], Lee & Geem [18], Ingram & Zhang [10] and Wang et al. [6], there are five main steps in the basic algorithm structure of HS:

* Corresponding author.
E-mail addresses: tonghuazhang@swin.edu.au (T. Zhang), zwgeem@gmail.com (Z.W. Geem).

https://doi.org/10.1016/j.swevo.2019.03.012
Received 29 April 2018; Received in revised form 4 February 2019; Accepted 20 March 2019
Available online 26 March 2019

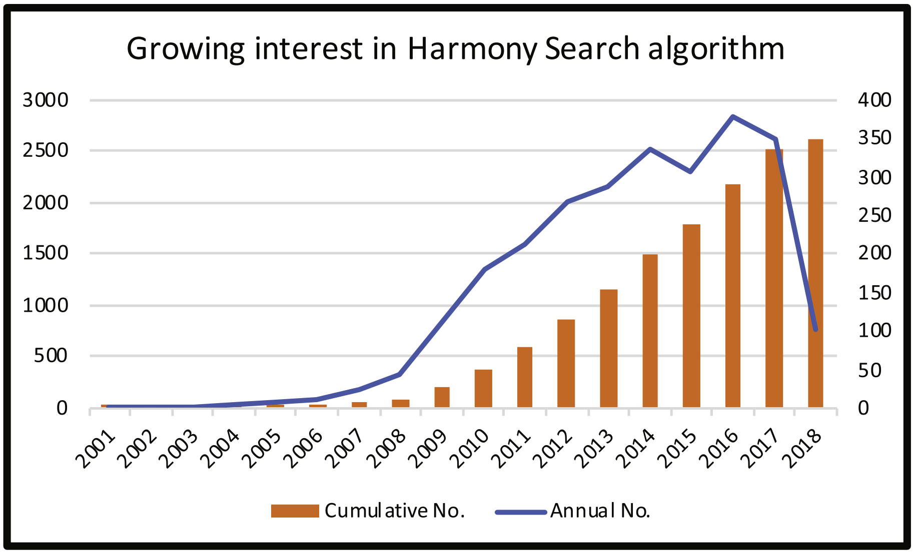

Fig. 1. Growing interest in the study and applications of Harmony Search since 2001. (Source: Scopus, topic: "harmony search" in fields of title + abstract + keywords, duplicates removed, at 11.30am 16 April 2018.)

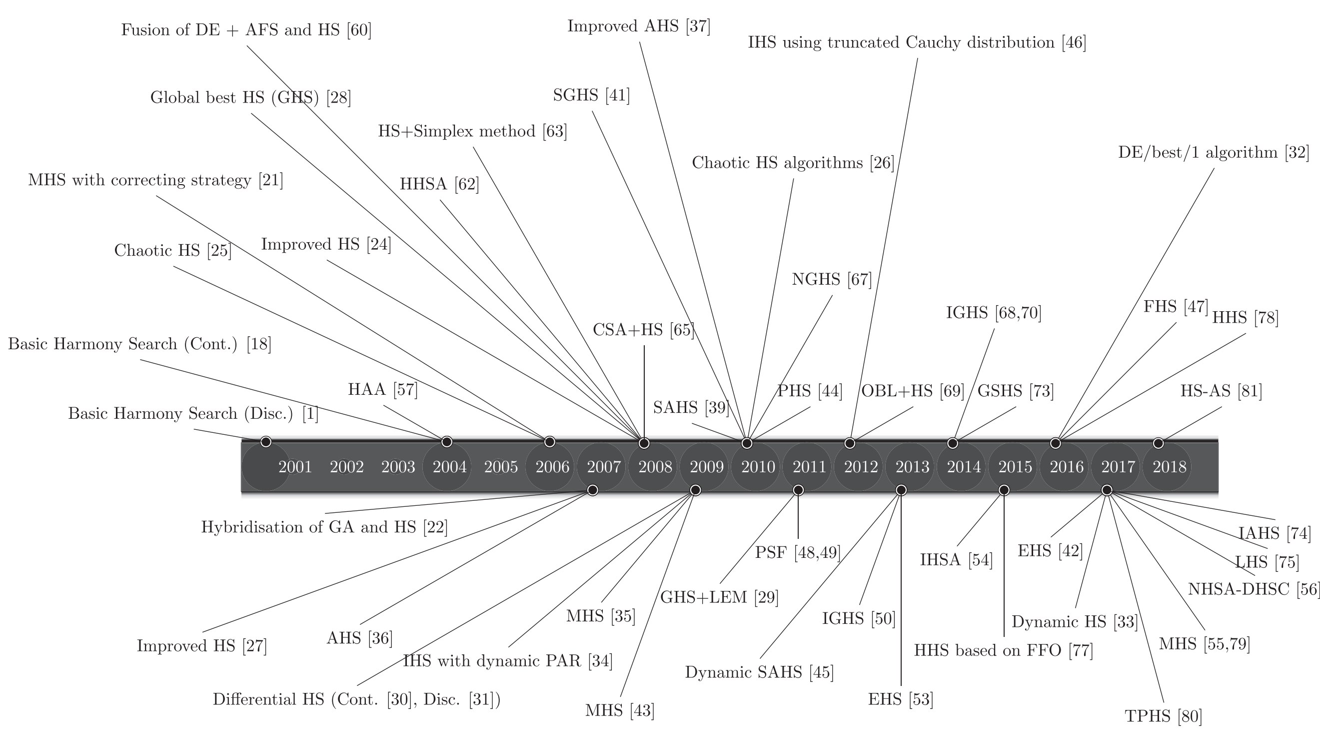

Fig. 2. Chronology of major variants of the basics harmony search since 2000.

Step 1: Initialise the algorithm by specifying parameters
Step 2: Initialise the Harmony Memory (HM) and sort to find the worst harmony, $\mathbf{X}_{worst}$
Step 3: Generate a new harmony, $\mathbf{X}_{new}$
Step 4: Update HM if $\mathbf{X}_{new}$ is better than $\mathbf{X}_{worst}$
Step 5: Repeat **Steps** 3 and 4 until the Termination Criterion is satisfied

The implementation of the basic harmony search algorithm is detailed one-by-one as follows.

2.1. Initialising the algorithm

In this step, we will not only supply the objective function, $f(\mathbf{X})$, the functions defined the constrained conditions, $\mathbf{g(X)} \leq \mathbf{0}$ and $\mathbf{h(X)} = \mathbf{0}$,

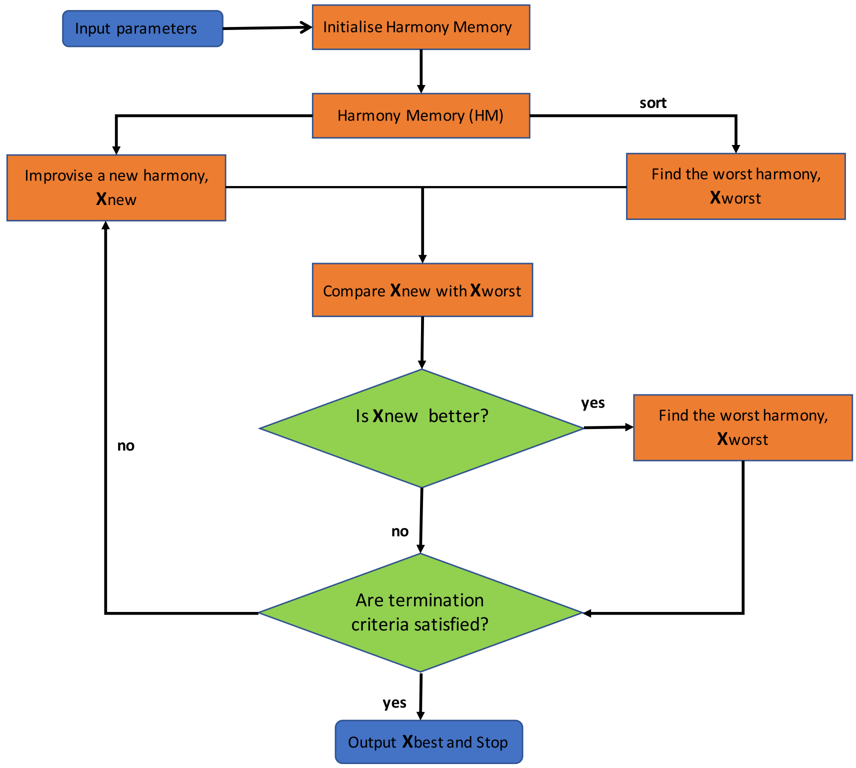

Fig. 3. Flowchart of the basic harmony search.

Table 1
Key parameters in the basic Harmony Search of [2].

HMS:	harmony memory size, namely the number of solution vectors
HMCR:	harmony memory considering rate
PAR:	pitch adjusting rate

but also predefine all parameters, such as lower and upper boundaries, x_i^L, x_i^U for continuous variables and $\mathbf{X}_i$ for discrete variables, HMS, HMCR, PAR, MaxImp and bandwidth vector, $\mathbf{b} \in \mathbf{R}^N$ for continuous variables and neighboring index vector $\mathbf{m} \in \mathbf{Z}^N > \mathbf{0}$ for discrete variables.

In general, the parameter values are to be selected arbitrarily, but helpful recommendations were made in previous publications, see Table 2 of [10] as an example, showing they could vary in a very large range, respectively. Lee et al. [21] also made the following recommendation:

$$10 \le \text{HMS} \le 50, 0.7 \le \text{HMCR} \le 0.95, 0.2$$
$$\le \text{PAR} \le 0.5 \text{ and } 30000 \le \text{MaxImp} \le 80000.$$

Selection of parameter values is sensitive and in general depends on the type of applications.

2.2. Initialising the harmony memory

The HM is initialised randomly and is a matrix of order HMS $\times$ N as in Refs. [6,8,18] or an augmented matrix of order HMS $\times$ $(N+1)$ as in Ref. [10]:

$$\text{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \dots x_N^1 & f(\mathbf{X}^1) \\ x_1^2 & x_2^2 & \dots x_N^2 & f(\mathbf{X}^2) \\ \vdots & \vdots & \ddots \dots & \vdots \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \dots x_N^{\text{HMS}} & f(\mathbf{X}^{\text{HMS}}) \end{bmatrix}. \quad (3)$$

Here, $\mathbf{X}^j = (x_1^j, \dots, x_N^j)$ is a solution vector, $f(\mathbf{X}^j)$ is the value of the objective function at $\mathbf{X}^j$ and x_i^j is the ith value in the jth solution vector for $j = 1, \dots, \text{HMS}$. The difference between references [6,8,18] and [10] is whether the values of the associated objective function are included in HM. In practice, the HM matrix (3) is a sorted matrix according to the fitness of the objective function, for example, Ingram & Zhang [10] suggested to store the best solution vector in the first row of HM matrix and the worst in the last row, namely $\mathbf{X}^1 = \mathbf{X}_{\text{best}}$ and $\mathbf{X}^{\text{HMS}} = \mathbf{X}_{\text{worst}}$.

2.3. Improvising a new harmony and if required update HM

Generating a new harmony is the key and determines the performance of the algorithm. Thus, variants of the Harmony Search method were mainly developed by providing new ways of generating a new harmony. It is randomly done for each variable x_i by firstly using parameter, HMCR for harmony memory consideration

$$x_{\text{new,i}} \leftarrow \begin{cases} x_i^j \in \left\{x_i^1, \dots, x_i^{\text{HMS}}\right\} \text{ with probability HMCR} \\ x_i^j \in \{\text{set of possible values for } x_i\} \text{ with probability } (1 - \text{HMCR}) \end{cases} \quad (4)$$

and then using PAR for pitch adjustment of $x_{\text{new,i}}$ according to (5)

$$x_{\text{new,i}} \leftarrow \begin{cases} \min(\max(x_{\text{new,i}}, x_i^L), x_i^U) \leftarrow x_{\text{new,i}} + \text{rnd}([-b_i, b_i]) \text{ for continuous decision variables} \\ x_i^{j+m_i}, m_i \text{ is the neighbouring index for discrete decision variables} \end{cases} \quad (5)$$

as suggested by Refs. [6,8,10,18]. The pseudo codes for continuous and discrete decision variables are summarised in Algorithms 1 and 2, respectively.

Algorithm 1 Pseudo code for improvising a new harmony, $\mathbf{X}_{\text{new}}$ in basic HS for continuous variables ([6,8,10,18]).

1: **For** each variable, $x_i, i = 1, 2, \ldots, N$
2: **if** rnd([0,1]) $\le$ HMCR **then**

$$x_{\text{new,i}} \leftarrow x_i^j, \quad j \in \text{rnd}(\{1, 2, \ldots, \text{HMS}\})$$

3: **if** rnd([0,1]) $\le$ PAR **then** randomly adjust $x_{\text{new,i}}$ within a small band, $\pm b_i$ by
4: $x_{\text{new,i}} \leftarrow x_{\text{new,i}} + \text{rnd}([-b_i, b_i])$
5: $x_{\text{new,i}} \leftarrow \min(\max(x_{\text{new,i}}, x_i^L), x_i^U)$
6: **else**

$$x_{\text{new,i}} \leftarrow \text{rnd}([x_i^L, x_i^U])$$

7: **Return** the new harmony, $\mathbf{X}_{\text{new}} = (x_{\text{new,1}}, \ldots, x_{\text{new,N}})$

Algorithm 2 Pseudo code for improvising a new harmony, $\mathbf{X}_{\text{new}}$ in basic HS for discrete variables ([6,8,10,18]).

1: **For** each variable, $x_i, i = 1, 2, \ldots, N$
2: **if** rnd([0,1]) $\le$ HMCR **then**

$$x_{\text{new,i}} \leftarrow x_i^j, \quad j \in \text{rnd}(\{1, 2, \ldots, \text{HMS}\})$$

3: **if** rnd([0,1]) $\le$ PAR **then** randomly adjust $x_{\text{new,i}}$ by m_i,
4: Find L such that $x_i^j = x_{i,L}$
5: Define $l = \text{rnd}(\{\text{L} - \text{m}_\text{i}, \text{L} - \text{m}_\text{i} + 1, \ldots, \text{L} + \text{m}_\text{i}\})$
6: $l \leftarrow \min(\max(l, 1), K_i)$
7: $x_{\text{new,i}} \leftarrow x_{i,l}$
8: **else**

$$x_{\text{new,i}} \leftarrow x_i^k, k = \text{rnd}(\{1, \ldots, K_i\})$$

9: **Return** the new harmony, $\mathbf{X}_{\text{new}} = (x_{\text{new,1}}, \ldots, x_{\text{new},N})$

Once a new harmony, $\mathbf{X}_{\text{new}}$ has been improvised, compare it with the worst harmony, $\mathbf{X}_{\text{worst}}$ in HM. If the former fits the objective function better than the latter, then update HM by replacing the latter with the former, which is then followed by outputting the new harmony. It is seen that this process is the key step of the basic harmony search algorithm and of its variants.

2.4. Check termination criterion and output best harmony

The searching of new harmony stops if the termination criterion is satisfied; otherwise goto **Step 3** and repeat it, performing Algorithm 1 for continuous decision variables and Algorithm 2 for discrete decision variables.

The variants either have been developed to modify the structures of the aforementioned five steps, by

(a) providing alternative initialisation procedures for harmony memory
(b) creating new improvisation procedures, including but not limited to
  (b1) the use of variable parameters, such as HMS, HMCR, PAR even bw
  (b2) options for handling constraints when generating new harmonies
  (b3) different criteria for filtering and selecting new harmonies for the HM
(c) setting distinct termination criteria or have been developed to hybridise HS with other heuristics.

Ingram & Zhang [10] and Wang et al. [6] did some introductory reviews; Ingram and Zhang [8] provided a chronology of selected developments of HS up to November 2008; reviews on the existing HS flavors can be found from Geem [9] and Alia & Mandava [12]. In addition, surveys on applications of HS can be found from Refs. [10,13] and the references therein. But, to the best of our knowledge, there are no any reviews or surveys having been done with respect to the algorithm structures. In the rest of this paper, we will use two sections to address the modifications towards the algorithm structures: one section will be focusing on the changes made within the original structures and the other will be focusing on the structure modifications due to hybridisation with other heuristic algorithms.

Due to its success in solving the single objective optimisation problems in engineering applications, the basic Harmony Search method has been adapted into different variants. For example the ones for solving multi-objective optimisation problems, the ones for mixed variables, and the ones with modified algorithm structures to improve the performance of the basic HS method. In following sections, we will provide an overview of all kinds of modifications of the basic HS, particularly focusing on the structural variations.

3. Algorithm structure changes due to the introduction of new operations or adaption of HS operators

Harmony search type of algorithms has been proven very successful as evidenced by numerous applications. In order to improve its efficiency or to overcome some shortcomings, the original HS operators have been adapted and/or new operators have been introduced. In this section, we summarised the algorithm structure changes that were induced by adaptions of operator or new operators.

To take full advantage of the accumulated information in the harmony memory, Li, Chi and Chu [22] used a two-point search strategy rather than random search strategy in HM consideration, resulting the following structure change summarised in Algorithm 3.

Li and his collaborators modified the pitch adjusting operator by using the non-uniform mutation operator of the genetic algorithm [23], namely when $\text{rnd}([0,1]) < \text{PAR}$, the pitch adjusting in Algorithm 1 is replaced with

$$x_{\text{new,i}} \leftarrow \begin{cases} \text{x}_{\text{new,i}} + \Delta\left(\text{Imp}, \text{x}_\text{i}^\text{U} - \text{x}_{\text{new,i}}\right) & \text{if rnd}(\{0,1\}) = 0 \\ \text{x}_{\text{new,i}} - \Delta\left(\text{Imp}, \text{x}_{\text{new,i}} - \text{x}_\text{i}^\text{L}\right) & \text{if rnd}(\{0,1\}) = 1 \end{cases}$$

where

$$\Delta(t, y) = y \times \text{rnd}([0, 1]) \times \left(1 - \frac{t}{\text{MaxImp}}\right)^b$$

and b is a parameter, in general $b = 2$.

In stead of a single new harmony, Li and Chi [24] and Cheng et al. [25] proposed to improvise $n > 1$ new harmonies, respectively. Particularly, Cheng [25] suggested that $n = 0.1 \times \text{HMS}$. Li et al. also proposed similar operators [26]: simple chaos harmony method, fixed chaos harmony method and dynamic chaos harmony method. The key difference between the basic HS and chaotic harmony methods lies in that HS generates one new harmony, $\mathbf{X}_{\text{new}}$ in the step of Improvisation

Algorithm 3 Pseudo code for improvising $\mathbf{X}_{\text{new}}$ in the modified HS with two-point search strategy ([22]).

1: **For** each decision variable, $x_i, i = 1, 2, \dots, N$

if rnd([0,1]) $\leq$ HMCR **then** randomly select $j \in \{1, 2, \dots, \text{HMS}\}$

$$x_{\text{new,i}} \leftarrow R_i^j + \lambda \left(R_i^j - x_i^j\right) \text{ with } \lambda = \text{rnd}([0,1]) \times (\lambda_{\max} - \lambda_{\min}),$$

$$\lambda_{\max} = \max\{\lambda_L, \lambda_U\}, \lambda_{\min} = \min\{\lambda_L, \lambda_U\}, R_i^j = \sum_{k=1, k\neq j}^{\text{HMS}} x_i^k,$$

$$\lambda_L = \frac{x_i^L - R_i^j}{R_i^j - x_i^j} \text{ and } \lambda_U = \frac{x_i^U - R_i^j}{R_i^j - x_i^j}$$

if rnd([0,1]) $\leq$ PAR **then** randomly adjust $x_{\text{new},i}$ as in Algorithm 1

else generate the ith decision variable of the new harmony using mutation strategy at the two endpoints of $[x_i^L, x_i^U]$

$$x_{\text{new,i}} \leftarrow y_{\max} + (x_i^U - y_{\max}) \times \text{rnd}([0,1]) \text{ or } x_i^L + (y_{\min} - x_i^L) \times \text{rnd}([0,1])$$

$$y_{\max} = \max_i\{R_i^j\}, y_{\min} = \min_i\{R_i^j\}, j = 1, \dots, \text{HMS}$$

2: **Return** the new harmony, $\mathbf{X}_{\text{new}} = (x_{\text{new},1}, \dots, x_{\text{new},N})$

while the latter not only generates $\mathbf{X}_{\text{new}}$ but also several new harmonies using the chaotic logistic map

$$\mathbf{X}^j_{\text{chaotic new}} = l_i + (u_i - l_i) \times \chi_i^{Imp} \text{ with } \chi_i^{Imp} = 4 \times \chi_i^{Imp-1}\left(1 - \chi_i^{Imp-1}\right) \tag{6}$$

and here l_i, u_i are modified bounds. Besides the logistic map (6), Alatas [27] introduced 6 other chaotic maps,

$$\mathbf{X}_{n+1} = \begin{cases} a\mathbf{X}_n(1-\mathbf{X}_n) & \text{for logistic map,} \\ \mathbf{X}_n \leftarrow \begin{cases} \mathbf{X}_n/0.7 & \text{if } \mathbf{X}_n < 0.7 \\ (10/3) \times \mathbf{X}_n(1-\mathbf{X}_n) & \text{otherwise} \end{cases} & \text{for tent map,} \\ a\mathbf{X}_n^2 \sin(\pi\mathbf{X}_n) \text{ or a simple version} \sin(\pi\mathbf{X}_n) & \text{for sinusoidal map,} \\ \dfrac{1}{\mathbf{X}_n} - \left\lfloor \dfrac{1}{\mathbf{X}_n} \right\rfloor, \mathbf{X}_n \in (0,1) & \text{for Gaussian map,} \\ \mathbf{X}_n + b - (a/2\pi) \times \sin(2\pi\mathbf{X}_n) \times \text{mod}(1) & \text{for circle map,} \\ 2.3 \times \mathbf{X}_n^{2\sin(\pi\mathbf{X}_n)} & \text{for sinus map,} \\ 1 - a\mathbf{X}_n^2 + b\mathbf{X}_{n-1} & \text{for Henon map.} \end{cases} \tag{7}$$

Based on the selected maps in (7), he proposed seven Chaotic Harmony Search algorithms summarised in Table 2, where ✓ indicates the parameter/operation being modified by chaotic maps.

Based on the basic HS and the work of [28], Omran and Mahdavi [29] proposed a so-called Global-best HS method (GHS), in which the structure of the pitch adjusting process of Algorithm 1 was modified as

$$\text{replace } x_{\text{new,i}} \leftarrow x_{\text{new,i}} + \text{rnd}([-b_i, b_i]) \text{ with } x_{\text{new,i}} \leftarrow x_{\text{best,i}} \in \text{HM}. \tag{8}$$

The GHS algorithm was further improved in an algorithm called Global-best Harmony Search using Learnable Evolution Models (GHS + LEM) [30], which introduces three new parameters, the Rate of Rule Update (RRU), the High and Low performance GroupS (HLGS) and the Rate of Consideration of Rules (RCR) to modify the algorithm structure of GHS in 3 ways:

(a) adding a rule inference procedure right before the step of improvisation of the GHS based on HLGS
(b) defining the dimension values of new improvise based on RCR in the Improvisation step
(c) adding a procedure of checking the rule update criteria based on RRU right after HM is updated

Based on GHS, most recently Ruano-Daza and his collaborators proposed a multi-objective bilevel approach for defining optimal routes and frequencies for bus rapid transit systems [31].

Chakraborty and his collaborators [32] proposed to modify the pitch adjustment operation of the classical HS for continuous decision variables by a mutation strategy

$$x_{\text{new,i}} \leftarrow x_{\text{new,i}} + \text{rnd}([0,1]) \times (\mathrm{x}_\mathrm{i}^\mathrm{j} - \mathrm{x}_\mathrm{i}^\mathrm{k}), i = 1, \dots, \mathrm{N} \text{ and}$$
$$\mathrm{j,k} = \text{rnd}(\{1, 2, \dots, \text{HMS}\}) \tag{9}$$

which is borrowed from differential evolution algorithm. The new algorithm is called Differential Harmony Search (DHS), which was later on adapted for discrete decision variables [33]. Abedinpourshotorban et al. [34] further adapted DHS algorithm by adding an operation to effectively initialise harmony memory, but keeped the mutation operation (9) in the pitch adjusting operation, resulting a new algorithm called DE/best/1, inspired by which Guo et al. recently proposed an Adaptive Harmony Search method with best-based search strategy (ABHS) [35].

Most recently, a novel pitch adjusting scheme according to (11) was proposed by Keshtegar et al. [36] based on the dynamic bw (10)

Table 2
Chaotic Harmony Search algorithms (CHS-i), $i = 1, \ldots, 7$.

	CHS-1	CHS-2	CHS-3	CHS-4	CHS-5	CHS-6	CHS-7
HM	✓				✓	✓	✓
bw			✓	✓		✓	✓
PAR		✓		✓	✓		✓

$$x_{\text{new,i}} \leftarrow \begin{cases} x_{\text{new,i}} + \text{rnd}([0,1]) \times \text{bw}_{\text{i}}(\text{Imp}) & \text{with probability} \leq 0.5 \\ x_i^{\text{best}} + \gamma(\text{Imp}) \times \text{rnd}([0,1]) \times \text{bw}_{\text{i}}(\text{Imp}) & \text{with probability} > 0.5 \end{cases} \tag{10}$$

where $\gamma(t) = \sqrt{1 - \frac{t}{\text{MaxImp}}}$ and

$$\text{bw}_{\text{i}}(\text{Imp}) = \frac{\text{x}_{\text{i}}^{\text{U}} - \text{x}_{\text{i}}^{\text{L}} + 0.001}{10} \times \exp\left(-10 \times \frac{\text{Imp}}{\text{MaxImp}}\right) \tag{11}$$

4. Algorithm structure changes due to parameter adaptions

The values of parameters in harmony search technique are essential to the success implementation of this algorithm. In general, researchers suggested to use the variable parameters instead of the fixed ones. To implement these variable parameters, relevant algorithm structures should be modified when improvising a new harmony, such as adding iteration loop or blocks to the original structure. We summarise the variations of parameter settings as follows.

Instead of using fixed parameters, Mahdavi et al. [28] made an attempt proposing that PAR and $\mathbf{b} = (b_1 \cdots, b_N)$ should be functions of the generation, namely for the Imp-th improvisation

$$\begin{aligned} \text{PAR}(\text{Imp}) &= \text{PAR}_{\min} + \frac{\text{Imp}}{\text{MaxImp}} \times (\text{PAR}_{\max} - \text{PAR}_{\min}), \\ b_i(\text{Imp}) &= \text{b}_{\text{i,max}} \times \exp\left[\frac{\text{Imp}}{\text{MaxImp}} \ln\left(\frac{\text{b}_{\text{i,min}}}{\text{b}_{\text{i,max}}}\right)\right] \\ &= \text{b}_{\text{i,max}} \times \left(\frac{\text{b}_{\text{i,min}}}{\text{b}_{\text{i,max}}}\right)^{\frac{\text{Imp}}{\text{MaxImp}}}, \end{aligned} \tag{12}$$

resulting an Improved Harmony Search algorithm (IHS). A similar work has also been proposed by Kumar et al. [37,38] where parameters are adapted automatically in a linear form or exponential form. Then, Coelho et al. [39] changed this structure further by proposing a new way to generate a dynamical PAR

$$\text{PAR}(\text{Imp}) = \text{PAR}_{\min} + \frac{\text{f}_{\text{Imp,max}} - \text{f}_{\text{mean}}}{\text{f}_{\text{Imp,max}} - \text{f}_{\text{Imp,min}}} \times (\text{PAR}_{\max} - \text{PAR}_{\min}) \tag{13}$$

where $f_{\text{Imp, max}}$ is the maximum of the objective function and $f_{\text{Imp, min}}$ the minimum at generation Imp respectively and f_{mean} is the mean in HM. Besides the dynamical PAR and bw, Lee and Yoon [40] proposed to use variable HMCR and HMS defined by

$$\begin{aligned} \text{HMS} &\leftarrow \left\lfloor \text{HMS} \times \left(1 + \frac{\left|\text{f}_{\text{required}} - \text{f}_{\text{predicted}}\right|}{\text{f}_{\text{required}}}\right) + 1 \right\rfloor, \\ \text{HMCR}_{\text{max or min}} &\leftarrow \text{HMCR}_{\text{max or min}} \times \left(1 \pm \frac{\left|\text{f}_{\text{required}} - \text{f}_{\text{predicted}}\right|}{\text{f}_{\text{required}}}\right), \\ \text{PAR}_{\text{max or min}} &\leftarrow \text{PAR}_{\text{max or min}} \times \left(1 \pm \frac{\left|\text{f}_{\text{required}} - \text{f}_{\text{predicted}}\right|}{\text{f}_{\text{required}}}\right). \end{aligned} \tag{14}$$

Dong et al. [41] proposed an improved HS method, called Adaptive Harmony Search (AHS), for solving optimisation problems with discrete decision variables by firstly dynamically changing HMCR

$$\text{HMCR}(\text{Imp}) = \text{HMCR}_0 + \omega \times \left(1 - \frac{\text{Imp}}{\text{MaxImp}}\right), \omega \in (0, 1 - \text{HMCR}_0) \tag{15}$$

and then searching the jth-nearest value as a next value with probability

$$\text{PAR}_{\text{j}}(\text{Imp}) = \text{PAR}_{0,\text{j}} \times (1 - \text{HMCR}(\text{Imp}) + \alpha), \text{j} = 1, 2, 3, \tag{16}$$

where $\text{PAR}_{0,\text{j}}$ is the initial pitch adjusting probability and α the inertia weight. For solving large-scale structural optimisation problems, a method also referred to as AHS was proposed by Hasancebi et al. [42], where for each generation Imp, it dynamically changes the HMCR and PAR as follows:

$$\begin{aligned} \text{HMCR}^{\text{k}} &= \left(1 + \frac{1 - \overline{\text{HMCR}}}{\overline{\text{HMCR}}} \times \exp\left[-\gamma \times \text{rnd}([0,1])\right]\right)^{-1} \\ \text{PAR}^{\text{k}} &= \left(1 + \frac{1 - \overline{\text{PAR}}}{\overline{\text{PAR}}} \times \exp\left[-\gamma \times \text{rnd}([0,1])\right]\right)^{-1} \end{aligned} \tag{17}$$

where $\gamma \in [0.25, 0.50]$ is the learning rate of control parameter and the averages of HMCR and PAR are defined by

$$\overline{\text{HMCR}} = \frac{1}{\text{HMS}} \sum_{i=1}^{\text{HMS}} \text{HMCR}^{\text{i}} \text{ and } \overline{\text{PAR}} = \frac{1}{\text{HMS}} \sum_{i=1}^{\text{HMS}} \text{PAR}^{\text{i}}.$$

To better to explore and exploit the problem space exhaustively, a dynamic adaptation procedure was proposed by Li and his collaborator [43] for the harmony memory consideration rate:

$$\text{HMCR} = \begin{cases} \dfrac{C_d}{M_d} & \text{if } C_d \leq M_d, \\ 1.0 & \text{otherwise,} \end{cases} \tag{18}$$

where C_d is known as the central distance defined by

$$C_d = \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{\text{HMS}} \left(x_i^j - \frac{1}{N}\sum_{k=1}^{N} x_k^j\right)^2}$$

and M_d is a threshold value specified by researchers via

$$M_d = \eta \times N \times \sqrt{\sum \left(x_i^U - x_i^L\right)^2}, \eta \in (0, 1).$$

Different to the works of [28,29,41] where harmony consideration rate and pitch adjust rate are dynamically changed, the Self-Adaptive Harmony Search algorithm (SAHS) proposed to change the pitch adjusting bandwidth, which was called a trial by Wang and Huang [44], dynamically in each iteration:

$$\mathrm{trial_i} \leftarrow \begin{cases} \mathrm{trial_i} + \mathrm{rnd} \times \left(\mathrm{x_i^U} - \mathrm{trial}_i\right) \\ \mathrm{trial_i} - \mathrm{rnd} \times \left(\mathrm{trial_i} - \mathrm{x_i^L}\right) \end{cases} \tag{19}$$

Taherinejad [45] proposed a different way to dynamically vary the pitch adjusting bandwidth

$$\mathrm{bw(Imp)} = \mathrm{bw_{max}} \times \exp\left[\frac{\mathrm{Imp}}{\mathrm{MaxImp}} \times \ln\left(\frac{\mathrm{bw_{min}}}{\mathrm{bw_{max}}}\right)\right] \tag{20}$$

A Self-adaptive Global best Harmony Search (SGHS) algorithm for solving continuous optimisation problems were introduced by Pan et al. [46], where authors claimed that the new improvisation scheme could capture all good information of the current global best solution. A learning mechanism was used to dynamically adapt HMCR and PAR, while the pitch adjusting bandwidth is dynamically updated according to

$$\mathrm{bw(Imp)} = \begin{cases} \mathrm{bw_{max}} - \dfrac{2\times \mathrm{Imp}}{\mathrm{MaxImp}} \times (\mathrm{bw_{max}} - \mathrm{bw_{min}}) & \text{if } t < \dfrac{\mathrm{MaxImp}}{2}, \\ \mathrm{bw_{max}} & \text{if } t \geq \dfrac{\mathrm{MaxImp}}{2}, \end{cases} \tag{21}$$

which is different to Eqs. (12), (19) and (20). A much simpler way of dynamically updating bw was used in Ref. [47] recently

$$\mathrm{bw(Imp)} = \frac{\mathrm{x_i^U} - \mathrm{x_i^L}}{\mathrm{Imp}}.$$

A similar way to (21) was used in a so-called EHS algorithm to dynamically change bw in Ref. [48]

$$\mathrm{bw(Imp)} = \begin{cases} \mathrm{bw_{max}} - \dfrac{2\times \mathrm{Imp}}{\mathrm{MaxImp}} \times (\mathrm{bw_{max}} - \mathrm{bw_{min}}) & \text{if } t < \dfrac{3}{4} \times \mathrm{MaxImp}, \\ \mathrm{bw_{min}} & \text{if } t \geq \dfrac{3}{4} \times \mathrm{MaxImp}, \end{cases} \tag{22}$$

Instead of using the dynamical bw or the one given in the basic HS, Yang [11] used a uniform adjusting bandwidth,

$$\mathrm{bw_i} = \frac{\mathrm{x_i^U} - \mathrm{x_i^L}}{1000},$$

which was further adapted by Jaberipour [49] into the one based on the approximation of the derivative of the objective function with respective to each decision variable

$$\mathrm{bw_i} = \frac{f(\mathbf{X} + 2\epsilon \mathbf{e}_\mathrm{i}) - f(\mathbf{X} - 2\epsilon \mathbf{e}_\mathrm{i})}{4\epsilon(\mathrm{x_i^U} - \mathrm{x_i^L})} \tag{23}$$

for Proposed Harmony Search (PHS) and

$$\mathrm{bw_i} = \begin{cases} \dfrac{x_i^U - x_i^L}{1000} & \text{if } \mathrm{rnd}([0,1]) < \mathrm{PAR} - \mathrm{A} \\ \dfrac{f(\mathbf{X} + 2\epsilon \mathbf{e}_i) - f(\mathbf{X} - 2\epsilon \mathbf{e}_i)}{4\epsilon(x_i^U - x_i^L)} & \text{if } \mathrm{rnd}([0,1]) \geq \mathrm{PAR} - \mathrm{A} \end{cases} \tag{24}$$

for the Improving Proposed Harmony Search (IPHS), where $A \in [0, \mathrm{PAR}]$. Clearly, Yang [11] used $A = 0$, the PHS used $A = \mathrm{PAR}$ and otherwise the best choice is $A = \mathrm{PAR}/2$ or PAR [49].

The Self-Adaptive Harmony Search algorithm proposed by Kattan and Abdullah [50] not only used the dynamical pitch adjustment rate, PAR, but also the bandwidth, bw.

$$\begin{aligned} &\mathrm{PAR} \leftarrow \mathrm{PAR_{max}} + \frac{f(\mathbf{X}_\mathrm{best})}{f(\mathbf{X}_\mathrm{worst})} \times (\mathrm{PAR_{min}} - \mathrm{PAR_{max}}) \\ &\mathrm{bw} \leftarrow \mathrm{rnd}([-a, a]), \text{ where } a = C \times \mathrm{StdDev}(\mathrm{x_{HM}^i}) \end{aligned} \tag{25}$$

The improved HS algorithm (IHS) proposed by Weihmann et al. [51] uses truncated Cauchy distribution in the range [0,1] to generate values for HMCR. The utilization of Cauchy distribution can be useful in IHS. Due to Cauchy be more expanded than Gaussian distribution, it allows, probabilistically speaking, large steps and in this way, generating more different values for the HMCR. Furthermore, the proposed IHS employs bw with decreasing linear during the generations with initial value equal to 0.5 and final value equal to 0.01.

Peraza et al. [52] designed a fuzzy operation to dynamically update HMCR and PAR in a so called Fuzzy Harmony Search algorithm (FHS). The operation works according to

$$\mathrm{HMCR} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{r_{hmr}}} \mu_\mathrm{i}^{\mathrm{hmr}}(\mathrm{hmr_{1i}})}{\sum_{\mathrm{i=1}}^{\mathrm{r_{hmr}}} \mu_\mathrm{i}^{\mathrm{hmr}}} \text{ and } \mathrm{PAR} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{r_{par}}} \mu_\mathrm{i}^{\mathrm{par}}(\mathrm{par_{1i}})}{\sum_{\mathrm{i=1}}^{\mathrm{r_{par}}} \mu_\mathrm{i}^{\mathrm{par}}} \tag{26}$$

where $r_{(\cdot)}$ is the number of rules of the fuzzy system and $\mu_i^{(\cdot)}$ is the membership of function of rule i.

To better utilise the variation of solutions present in Harmony Memory, Kumar et al. [53,54] proposed a Variance-based Harmony Search Algorithm (VHSA). It generates a new harmony according to the variations present in the current harmony and the ones in HM. The pitch adjustment is done by

$$x_{\mathrm{new},i} \leftarrow x_{\mathrm{new},i} \pm \left[W_{\mathrm{var_{min}}} - \mathrm{std}(\mathrm{HM}_i)\right] \times \mathrm{rnd}([0,1])$$

where $W_{\mathrm{var_{min}}} = \min(\mathrm{std}(\mathrm{HM}))$.

It is worth mentioning that Geem and Sim [55] and Geem and Cho [56], instead of using variable or dynamical parameters, introduced a Parameter-Setting-Free (PSF) harmony search algorithm, where two major parameters are dynamically changed by the PSF techniques

$$\begin{aligned} &\mathrm{HMCR^k} \leftarrow \frac{\mathrm{n(ot^k = memory\ consideration\ or\ pitch\ adjustment)}}{\mathrm{HMS}} \\ &\mathrm{PAR^k} =\leftarrow \frac{\mathrm{n(ot^k = pitch\ adjustment)}}{\mathrm{n(ot^k = memory\ consideration\ or\ pitch\ adjustment)}} \end{aligned} \tag{27}$$

where $n(\cdot)$ is a count function representing the number of specific operations in HM and $\mathrm{ot^k}$ is the operation type, namely random selection, memory consideration or pitch adjustment, in generation k.

Lastly, very different to the aforementioned, the Improved Global Harmony Search algorithm (IGHS) [57] replaces the uniform distribution with the Gaussian distribution in the process of pitch adjustment. Geem [58] added a new operation, ensemble consideration to the algorithm structure of the basic HS. The new operation was introduced to update $\mathbf{X}_\mathrm{new}$ generated by Algorithms 1 or 2 with probability of Ensemble Consideration Rate (ECR).

5. Structure modifications by hybridising with the existing heuristic algorithms

To overcome the drawbacks of the basic harmony search algorithm and improve the performance of HS, we could modify the algorithm structure by changing the way of setting parameters, or by fusing the HS techniques with other algorithms. We will provide an overview of this type of structure changes in this section.

Tian, Bo and Gao [59] proposed the Harmony-Annealing Algorithm (HAA), which is based on the basic HS and modified the improvisation operator as follow: when rnd $<$ HMCR, the new harmony is to be generated by HS; otherwise, it will be generated by the Very Fast Simulated Annealing (FSA) algorithm:

$$x_{\mathrm{new,i}} \leftarrow x_{\mathrm{new,i}} + y_i \times \left(x_i^U - x_i^L\right)$$

$$y_i = T \times \mathrm{sign}\,(\mathrm{rnd} - 0.5) \times \left(\left(1 + \frac{1}{\mathrm{T}}\right)^{|2\mathrm{rnd}-1|} - 1\right)$$

Then, the same authors discussed the selection of parameters when applying the HAA [60]. A similar hybrid algorithm to Refs. [59,60] was proposed by Wang and Gao [61], where in step of improvisation there are also two new harmonies generated with one from HS, $x^{\mathrm{HS}}_{\mathrm{new}}$ and the other from SA, $x^{\mathrm{SA}}_{\mathrm{new}}$; and the difference is that $x_{\mathrm{new}} = \max\{x^{\mathrm{HS}}_{\mathrm{new}}, x^{\mathrm{SA}}_{\mathrm{new}}\}$ in Ref. [61].

Gao et al. [62] proposed a method by fusing the Differential Evolution (DE) algorithm, the artificial fish swarm (AFS) technique and HS for multi-modal optimisation, in which the new control mechanism was made using AFS to determine whether $\mathbf{X}_{\mathrm{worst}}$ was to be replaced with $\mathbf{X}_{\mathrm{new}}$. It works as follows: after generating a new harmony, $\mathbf{X}_{\mathrm{new}}$ we first define a distance function

$$\rho_i \equiv \|\mathbf{X}_{\mathrm{new}} - \mathbf{X}_i\|, i = 1, \dots, \mathrm{HMS}$$

using the current HM and then define a vicinity of $\mathbf{X}_{\mathrm{new}}$,

$$V \equiv \{\mathbf{X}_i \in \mathrm{HM} \mid \rho_i < \rho_V, \text{ the pre}-\text{defined threshold}\}.$$

We then replace $\mathbf{X}_{\mathrm{worst}}$ with $\mathbf{X}_{\mathrm{new}}$ if

$$\mathrm{HMS} < \#(V) \text{ and } \bar{\mathrm{f}} < \mathrm{f}(\mathbf{X}_{\mathrm{new}}) < \mathrm{f}(\mathbf{X}_{\mathrm{worst}})$$

where $\bar{f} = \frac{1}{\mathrm{HMS}} \sum_{i=1}^{\mathrm{HMS}} f(\mathbf{X}_i)$. The drawbacks of this method are: the selection of ρ_V is case-dependent and there is no general rule; the calculation of ρ_i may be very time consuming if the memory size is large. The remarkable feature of HS + DE is that it provides a new way to update HM so that to overcome the premature problem of the basic HS: instead of directly using the $\mathbf{X}_{\mathrm{new}}$ to replace $\mathbf{X}_{\mathrm{worst}}$, the new method uses the DE method [63] to do a fine-tuning resulting a new HM, based on which produce a new harmony and then resume the regular HS, see the pseudo code in Algorithm 4.

Algorithm 4 Pseudo code for HS + Differential Evolution method ([62]).

1: **Start** the algorithm: input all parameters for regular HS and DE
  Generate a randomly initialised HM according to the regular HS
2: **Apply** DE method to fine-tune HM
  For each harmony, $\mathbf{X}_i, i \in \{1, 2, \dots, \mathrm{HMS}\}$ in HM, apply the evolutional strategy
  $\mathbf{X}_i \leftarrow \lambda\mathbf{X}_i + (1-\lambda) \times \left(\mathbf{X}_{j_i} - \mathbf{X}_{k_i}\right), \lambda \in [0, 1]$
  $i \neq j_i, k_i = \mathrm{rnd}(\{1, 2, \dots, \mathrm{HMS}\})$
  To fine-tune $\mathbf{X}_i$, resulting $\mathbf{X}_{\mathrm{tuned,i}} \leftarrow \mathbf{X}_i$
3: **Update** HM by using the tuned one
  $\mathrm{HM} \leftarrow \mathrm{HM}_{\mathrm{tuned}} = \{\mathbf{X}_{\mathrm{tuned},1}, \mathbf{X}_{\mathrm{tuned},2}, \cdots, \mathbf{X}_{\mathrm{tuned,HMS}}\}$
4: **Improvise** a new harmony using Algorithms 1 or 2
5: **Goto** Step 4 of the regular HS and **Resume** HS

Based on [28], Fesanghary et al. [64] proposed a method called Hybrid Harmony Search Algorithm (HHSA), which uses Sequential Quadratic Programming (SQP) to improve the performance of local search. Then the algorithm structure of the basic HS is changed after returning a new harmony, $\mathbf{X}_{\mathrm{new}}$ in Algorithms 1 and 2. Pseudo code for this algorithm in Algorithm 5. Comparing with the basic HS, which directly rejects the infeasible solution, another feature of HHSA is that infeasible solutions in the current HM that violate the constraints have a chance to be included in the HM by considering the penalty cost, which is calculated using the static penalty function

$$\mathrm{fitness}(\mathbf{X}) = f(\mathbf{X}) + \sum_{k=1}^{M} \alpha_k \times \min\{0, g_k^2(\mathbf{X})\} + \sum_{k=1}^{P} \beta_k \times \min\{0, h_k^2(\mathbf{X})\} \tag{28}$$

for optimisation problem (1) with constraints: $\mathbf{g} = (g_1, \dots, g_M) \geq 0$ and $\mathbf{h} = (h_1, \dots, P_M) = 0$, here α_k and β_k are penalty coefficients.

Algorithm 5 Pseudo code for HS + SQP method ([64])

1: **Input** parameters and $\mathbf{X}_{\mathrm{new}}$ obtained in Algorithm 1 or 2
2: **Calculate** fitness, $f(\mathbf{X}_{\mathrm{new}})$
  if rnd([0,1]) $<$ P_c **then** improve $\mathbf{X}_{\mathrm{new}}$ by SQP
  Update HM if required
  else $x_{\mathrm{new},i} \leftarrow \mathrm{rnd}([x_i^L, x_i^U])$
3: **Repeat** Step 2 **Utill** a pre-defined termination criterion is satisfied
4: **For** each $\mathbf{X}^j \in$ HM applying SQP to improve the harmony
5: **Output** $\mathbf{X}_{\mathrm{best}}$

Hybrid simplex-harmony search method by proposed Jang et al. [65]. The Simplex Method (SM) is incorporated into the improvisation step of the basic HS method to fine-tune the new harmony, $\mathbf{X}_{\mathrm{new}}$ by the 4 operations, reflection, expansion, contraction and shrinkage so that HS can find an optimisation solution more accurately and quickly. The pseudo code is as in Algorithm 6, from which or reference [65], we can see that this hybrid method requires that HMS $\geq N + 1$.

Algorithm 6 Pseudo code for generating a new harmony in the Hybrid Simplex-Harmony Search ([65]).

1: **Start** the algorithm: input all parameters for regular HS and SM
  Generate a randomly initialised HM according to the regular HS
2: **Sort** HM and **Copy** the best N harmonies the next step
3: **Apply** SM to generate a new harmony, $\mathbf{X}^{\mathrm{SM}}_{\mathrm{new}}$

$$\mathbf{X}_{N+1} \leftarrow \mathbf{X}^{\mathrm{SM}}_{\mathrm{new}}$$

4: **Apply** HS to update the whole HM
5: **Goto** Step 2

Lo [66] introduced a hybrid algorithm, HS-DLM by combing the HS with Discrete Lagrange Multiplier (DLM) with the main purpose of constraint handling using DLM.

The fusion of Clonal Selection Algorithm (CSA) and HS [67,68] was developed based on the basic HS, while using CSA to improve the members of HM, more precisely the cloning and mutation operators of CSA are embedded into the HS method as a separate fine-tuning approach to improvise a new harmony.

Instead of having the pitch adjusting operator, the approach, Novel Global Harmony Search algorithm (NGHS) based on the IHS of [28] and the PSO modified the improvisation step to have a genetic mutation operator fine-tuning the new harmony, see Ref. [69] and Algorithm 7. It was further modified by Valian et al. [70], where an algorithm called Intelligent Global Harmony Search (IGHS) was proposed by modifying the improvisation step of the NGHS in a way such that the new harmony imitates one dimension of the best harmony in the HM, namely when randomly playing pitch adjusting, it firstly improvise the new harmony as the best one in HM.

Algorithm 7 Pseudo code for fine-tuning a new harmony in NGHS [69].

For each $i \in \{1, \dots, N\}$
  If rnd $\leq$ PAR **do** genetic mutation:
  $x_{\mathrm{new,i}} \leftarrow x_i^L + \mathrm{rnd} \times \left(\mathrm{x_i^U} - \mathrm{x_i^L}\right)$
  Else do $x_R = 2 \times x_i^{\mathrm{best}} - x_i^{\mathrm{worst}}$
    If $x_R > x_i^U$
    $x_R = x_i^U$
    ElseIf $x_R < x_i^L$
    $x_R = x_i^L$
    $x_R = x_i^L$
    End
  $x_{\mathrm{new,i}} \leftarrow x_i^{\mathrm{worst}} + \mathrm{rnd} \times \left(\mathrm{x}_R - \mathrm{x_i^{worst}}\right)$
End

Based on Opposition-Based Learning (OBL), a new algorithm called Opposition-based HS algorithm was proposed by Chatterjee, Ghoshal and Mukherjee [71], in which OBL is employed to initialise and produce HM. Employing the OBL, Xiang et al. [72] proposed an Improved Global-best Harmony Search algorithm (IGHS). First, the IGHS uses the OBL the solution quality of initial HM; Second, the DE is used to replace the random play of the harmony memory consideration of the basic HS, namely

$$x_{\mathrm{new,i}} \leftarrow \mathrm{rnd}([x_i^L, x_i^U])$$

is replaced with

$$x_{\mathrm{new,i}} \leftarrow x_i^{\mathrm{best}} + \zeta \times (x_i^j - x_i^k),\ \{j, k, \mathrm{best}\}_{\mathrm{j,k \neq best}} \in \mathrm{rnd}(\{1, 2, \dots, \mathrm{HMS}\}) \tag{29}$$

ζ is a scale factor; it is then followed by the pitch adjusting done by the Artificial Bee Colony algorithm (ABC) [73]; Once reach a new harmony, $\mathbf{X}_{\mathrm{new}}$ and harmony memory is updated, the best harmony $\mathbf{X}_{\mathrm{best}}$ in the current HM is then to be fine-tuned firstly by ABC and then by OBL, returning an improved new harmony. The two major parameters, HMCR and PAR are also dynamically updated during the calculations

$$\begin{aligned} \mathrm{PAR(Imp)} &= \left(\mathrm{PAR_{min}} + \frac{\mathrm{Imp}}{\mathrm{MaxImp}} \times (\mathrm{PAR_{max}} - \mathrm{PAR_{min}})\right) \times \gamma, \\ \mathrm{HMCR(Imp)} &= \left(\mathrm{HMCR_{min}} + \frac{\mathrm{Imp}}{\mathrm{MaxImp}} \times (\mathrm{HMCR_{max}} - \mathrm{HMCR_{min}})\right) \times \gamma, \end{aligned} \tag{30}$$

here $\gamma = \max\{0, \mathrm{sgn}(\sin(\mathrm{Imp}))\}$ and PAR is based on the formula (12), the work of Mahdavi et al. [28].

Hasan et al. [74] proposed a hybrid harmony search algorithm by replacing all random consideration in HS with five different types of mutation operations.

The selection operator of the Evolutionary Algorithm (EA) was integrated into HS algorithm, resulting a new algorithm known as Geometric Selective Harmony Search (GSHS) [75], which introduces a selection procedure and recombination operator in the memory consideration process, and a mutation operation in the pitch adjustment.

The Improved Adaptive Harmony Search algorithm (IAHS) [76] added two main blocks in the algorithm structure: forward and backward schemes when initialising harmony memory and forward scheduling scheme when improvising a new harmony.

Integrated with OBL technique and the competition selection mechanism, Ouyang et al. [77] proposed an improved version of harmony search algorithm, Local opposition-based learning self-adaptation global harmony search (LHS), which from aspect of algorithm structure has one key change: it generates two harmonies each improvisation with first new harmony, $\mathbf{X}_{\mathrm{new}} = (x_{\mathrm{new,i}})$ is generated by the basic HS, and then the second $\widetilde{\mathbf{X}}_{\mathrm{new}} = (\widetilde{x}_{\mathrm{new,i}})$ by OBL and the first one as follows

$$\widetilde{x}_{\mathrm{new,i}} = \begin{cases} x_i^U + x_i^L - x_{\mathrm{new},i} & \text{with probability HMCR,} \\ x_i^L + \mathrm{rnd}([0,1]) \times (x_i^U - x_i^L) & \text{with probability } 1 - \text{HMCR.} \end{cases} \tag{31}$$

Banerjee and his collaborators also proposed a new algorithm based on the OBL and called the Opposition-based HS (OHS) [78], where they first defined an opposite number for each decision variable x_i:

$$\hat{x}_i = x_i^L + x_i^U - x_i \tag{32}$$

Based on (32), they added a block to the algorithm structure when initialising HM, resulting an opposition-based harmony memory. The second innovation is the block of opposition-based jumping added after HM is updated in the improvisation process.

It is known that the basic harmony search has difficulties in performing local search, to overcome this, Zhang et al. [79] proposed an effective Hybrid Harmony Search algorithm (HHS) based on the Fruit Fly Optimisation algorithm (FFO) and the harmony search. The incorporation of FFO changes the structure of the improvisation process, which is to be done by the new rules for memory consideration and pitch adjustment. Particularly, for multidimensional knapsack problem, the rules take the following forms

HM Consideration:

$$x_{\mathrm{new,i}} \leftarrow \begin{cases} x_i^j \in \mathrm{HM}, j \in \mathrm{rnd}\{1, 2, \dots, \mathrm{HMS}\} & \text{if rnd} < \text{HMCR} \\ \mathrm{rnd}(\{0, 1\}) & \text{else} \end{cases}$$

Pitch Adjustment:

$$x_{\mathrm{new,i}} \leftarrow \begin{cases} \left| x_{\mathrm{new,i}} - 1 \right| & \text{if rnd} < \text{HMCR and rnd} < \text{PAR} \\ x_{\mathrm{new,i}} & \text{else} \end{cases} \tag{33}$$

This Hybrid Harmony Search (HHS) algorithm proposed by Cheng et al. [80] developed a new operation to improvising new harmonies by replacing the randomly playing harmony memory with Global-best Particle Swarm Optimisation (PSO) search and neighbourhood search, namely when HMCR $\geq$ rnd, the new harmony will be generated by PSO and neighbourhood searches.

Mohamed et al. [81] proposed a new technique called Modified Harmony Search (MHS), which modifies Improvisation of the basic HS by adding two operations: first operation from PSO employed in the harmony memory consideration phase will generate two harmonies, $\mathbf{X}_{\mathrm{new}}^{\mathrm{random}}$ is generated randomly by the basic HS and $\mathbf{X}_{\mathrm{new}}^{\mathrm{PSO}}$ by PSO; the second borrowed from GA that will generate the new harmony, $\mathbf{X}_{\mathrm{new}}$ based on $\mathbf{X}_{\mathrm{new}}^{\mathrm{random}}$ and $\mathbf{X}_{\mathrm{new}}^{\mathrm{PSO}}$. Thus, the structure of Improvisation will be modified and is summarised in Algorithm 8.

Based on the HS and Genetic Algorithm (GA), in 2017, Assad & Deep [82] proposed a hybrid algorithm called Two-Phase Harmony Search (TPHS) algorithm, in which the phase I focuses on diversification by utilising the concept of catastrophic mutation from GA, and the second phase on local search.

To overcome the issue of premature convergence of the HS algorithm, most recently, Assad and Deep [83] proposed a novel hybrid algorithm called Harmony Search and Simulated Annealing (HS-SA) algorithm. As pointed out by the authors that the HS-SA algorithm works the same as the HS, except that the suboptimal harmonies are accepted as in SA.

6. Algorithm structure modifications made towards handling multi-objective optimisation problems and constraints

Geem and Hwangbo [84] generalised the basic HS for single objective optimisation to multiobjective optimisation. Ricart and his collaborators also proposed a Multiobjective Harmony Search algorithm (MHS) [85], Sivasubramani and Swarup also proposed a multi-objective harmony search algorithm [86], where authors used the proposed algorithm for solving an optimal power flow problem.

Algorithm 8 Pseudo code for improvising $\mathbf{X}_{\text{new}}$ in the Modified HS based on PSO and GA [81].
1: **For** each decision variable, $x_i, i = 1, 2, \ldots, N$
2: **if** rnd([0,1]) $\leq$ HMCR, **then**

$$x_{\text{new,i}} = x_i^j \text{ randomly select } j \in \{1, 2, \ldots, \text{HMS}\}$$

$$x_{\text{PSO,i}} = x_i^{\text{best}} \in \text{HM based on PSO}$$

3: **if** rnd([0,1]) $\leq$ PAR **then** randomly adjust $x_{\text{new,i}}$ and $x_{\text{PSO,i}}$ as in Algorithm 1
4: **else**

$$x_{\text{new,i}} = \text{rnd}([\text{x}_\text{i}^\text{L}, \text{x}_\text{i}^\text{U}])$$

$$x_{\text{PSO,i}} = \text{rnd}([\text{x}_\text{i}^\text{L}, \text{x}_\text{i}^\text{U}])$$

5: **Return** two new harmonies,

$$\mathbf{X}_{\text{new}}^{\text{random}} = (x_{\text{new},1}, \ldots, x_{\text{new,N}}) \text{ and } \mathbf{X}_{\text{new}}^{\text{PSO}} = (x_{\text{PSO},1}, \ldots, x_{\text{PSO,N}})$$

6: **Applying** the cross-over operation of GA generates

$$\mathbf{X}_{\text{update}}^{\text{random}} = \text{rnd}([0,1]) \times \mathbf{X}_{\text{new}}^{\text{random}} + (1 - \text{rnd}([0,1])) \times \mathbf{X}_{\text{new}}^{\text{PSO}}$$

$$\mathbf{X}_{\text{update}}^{\text{PSO}} = \text{rnd}([0,1]) \times \mathbf{X}_{\text{new}}^{\text{PSO}} + (1 - \text{rnd}([0,1])) \times \mathbf{X}_{\text{new}}^{\text{randow}}$$

7: **Return** a new harmony

$$\mathbf{X}_{\text{new}} \leftarrow \text{ the best of } \left\{\mathbf{X}_{\text{update}}^{\text{random}}, \mathbf{X}_{\text{update}}^{\text{PSO}}\right\}$$

To handle multiobjective optimisation problems, a parallel algorithm, MR-DHS algorithm based on HS was developed by Li et al. [87]. The parallel algorithm is different to the basic HS in two aspects:

(a) instead of single harmony memory, MR-DHS algorithm divides HM into several sub-HMs
(b) HMCR is updated dynamically according to roulette, and PAR is adjusted according to Ref. [28] or

$$\begin{aligned}\text{PAR(Imp)} = \text{PAR}_{\min} \times \exp\left(-\frac{1}{\text{MaxImp}^2 - 1} \ln\left(\frac{\text{PAR}_{\max}}{\text{PAR}_{\min}}\right)\right) \\ \times \exp\left(\frac{\text{Imp}^2}{\text{MaxImp}^2 - 1} \times \ln\left(\frac{\text{PAR}_{\max}}{\text{PAR}_{\min}}\right)\right)\end{aligned} \tag{34}$$

7. Changes in structure to adapt alternative initialisation procedures for harmony memory

The original HS assumed that every row in HM has the same chance to supply a new value for $x_{\text{new,i}}$. To improve the performance of the basic HS, Li and Chi [24] and Cheng et al. [25] used biased selection by assuming the j-th row has probability

$$pr(j) = \frac{\delta(1-\delta)^{j-1}}{\sum_{k=1}^{\text{HMS}} \delta(1-\delta)^{k-1}}, \quad \delta \in (0,1] \text{ is the bias parameter,} \tag{35}$$

which ensures the harmonies with better fitness had a greater chance of selection. When analysing the slope stability, Cheng et al. [25,88] found that the basic HS worked extremely well if the number of control variables is less than 25; otherwise it could be trapped by the local minima easily. Thus, they modified the basic HS in the following ways:

(a) In the Step of generating a new harmony, not use a uniform probability to select a harmony from HM, instead each harmony in HM is assigned a selecting probability

$$pr(i) = \delta \times (1-\delta)^{i-1}, \text{ a simpler version of (35)}$$

(b) Instead of generating one new harmony at an iteration, the modified generates Nhm new harmonies, and when updating HM still keep the best HMS harmonies
(c) Proposed a new termination criterion: first run HMS $\times$ $\text{N}_\text{i}, \text{i} = 1, 2$ iterations generating a best solution $\mathbf{X}_{\text{best,i}}$. Then, it stops if $|f(\mathbf{X}_{\text{best},2}) - f(\mathbf{X}_{\text{best},1})| \leq \epsilon$ for pre-set threshold ϵ

Please notice that they still used fixed parameters, but recommended that HMS = 2 $\times$ number of decision variables, HMCR = 0.98, PAR = 0.1, $\text{N}_1 = 500, \text{N}_2 = 200$ and $\text{Hhm} = \max\{\text{HMS} \times 0.1, 10\}$.

Ayob et al. [89] proposed an Enhanced Harmony Search Algorithm (EHSA). It uses semi-cyclic shift patterns to generate the initial HM. They also employed the algorithm of [42] for dynamically updating HMCR and PAR, showing the EHSA outperformed AHS and the basic HS.

Jiang and Zhang [90] proposed an Improved Harmony Search Algorithm (IHSA) using a dynamical bw

$$\text{bw(Imp)} = \text{bw}_0 \times \frac{\text{Imp}}{\text{MaxImp}}, \tag{36}$$

but the key innovation in IHSA is that the introduction of weight for each harmony in HM in a way such that the best harmony takes a weight of 0.5 and the rest share the rest weight of 0.5. Improved harmony search algorithm was recently used to solve non-linear non-convex short-term hydrothermal scheduling problem [91].

Based on the IHS of [28], recently, Medeiros and Kripka [92] proposed a Modified Harmony Search algorithm. In addition to the modifications made in IHS, authors proposed to modify the step of initialising the harmony memory as follows: (a) instead of randomly initialising all harmonies, the HM may include at least one harmony predefined by the designer based on their knowledge; (b) if all harmonies perform the same, then reinitialise HM.

The algorithm called Niche harmony search (NHSA-DHSC), which was proposed by Tuo et al. [93], generates three harmony memories, $\text{HM}_i, i = 1, 2, 3$ at a time, instead of one and then improvises a new harmony, $\mathbf{X}_{\text{new}}$ by playing the harmony memory consideration as follows:

$$\mathbf{X}_{\text{new}} \leftarrow \begin{cases} \mathbf{X}_{\text{new}} \in \text{HM}_1 & \text{if rnd}([0,1]) < 1/3; \\ \mathbf{X}_{\text{new}} \in \text{HM}_2 & \text{else if } 1/3 \leq \text{rnd}([0,1]) < 2/3; \\ \mathbf{X}_{\text{new}} \in \text{HM}_3 & \text{otherwise.} \end{cases} \tag{37}$$

8. HS used to improve the performance of other algorithms

Li et al. [94,95] proposed a Heuristic Particle Swarm Optimizer (HPSO), in which the concept of HM of the HS was used in particle swarm optimizer to avoid trapping in the local solution. Please notice that the case of continuous decision variables was discussed in Ref. [94] while the discrete variables in Ref. [95]. Li and Li [96] proposed a similar algorithm combining the PSO with the HS.

Li et al. [97] also used HS to improve the performance of existing algorithm, Genetic Algorithm, where authors took advantage of the way of improvising a new harmony of HS.

Moeinzadeh developed a new optimisation algorithm, Harmony search-Class-Independent Linear Discriminate Analysis (HCI-LDA) [98], where the basic HS was used to improve the performance of LDA by finding an optimal transformation.

Most recently, Alberti and Nagar [99] proposed a hybrid algorithm consisting of HS and SA search algorithm, where the HS algorithm was used to provide an alternative approximated best response (HSBR) so that improvement can be made when calculating the maximum regret function. It results in the following change:

$$\overline{\epsilon} = \max_i \left\{u_i(SABR(d_{k,i}), d_{k,-i}) - u_i(d_k)\right\} \leftarrow \max_i \left\{u_i(HSBR(d_{k,i}), d_{k,-i}) - u_i(d_k)\right\}$$

9. Theoretical analysis and criticisms about HS

The harmony search algorithm is an emerging metaheuristic algorithm in optimisation [100], where Salem and Khelfi carried out a statistical comparison among some major variants of harmony search algorithm: Improved Harmony Search, Global-Best Harmony Search, Self-Adaptive Harmony Search and Adaptive Harmony Search. Based on the 10 benchmark functions, their analysis showed that harmony search algorithm outperformance other metaheuristic algorithms in terms of average error and Friedman test. The advantage of harmony search algorithm can also be seen from its scalability, for example, a water network design problem of 10^{454} candidate solutions was discussed by Ref. [101] and an ecological conservation problem of 2^{441} candidate solutions was studied by Ref. [102].

Even it has been successful in applications, theoretical analysis about HS techniques is limited. The first attempt was made by Geem [1,2] where they calculated the probability of finding an optimal solution when PAR = 0. Then, Geem [103] introduced a novel stochastic derivative for harmony search algorithm for discrete decision variables, which was then given in a different format by Ingram & Zhang [10]

$$\left.\frac{\partial f}{\partial x_i}\right|_{x_i = X_{i,k}} = \underbrace{\frac{1}{K_i} \times (1 - \mathrm{HMCR})}_{\text{random selection}} + \underbrace{\frac{n_i(X_{i,k})}{\mathrm{HMS}} \times \mathrm{HMCR} \times (1 - \mathrm{PAR})}_{\text{harmony memory consideration}} + \underbrace{\frac{\frac{1}{|K_p|}\sum_{l \in K_p} n_i(X_{i,l})}{\mathrm{HMS}} \times \mathrm{HMCR} \times \mathrm{PAR}}_{\text{pitch adjustment}} \tag{38}$$

which is the probability that $X_{i,k}$ will be selected as $\mathbf{X}_{\text{new}}$ rather than a "real derivative" with respect to x_i [10], and Geem and his collaborator pointed out in Ref. [104] that HS searches for an optimal solution with a probability (38), suggesting that if a certain value frequently appears in multiple vectors, the value has higher chance to be selected than other candidate values. Das et al. analysed the evolution of the population variance, $\mathrm{Var}(\mathbf{X}^j)$ over successive generations. Then, the expectation of the population variance can be calculated by equation (5) of [105], which indicates the explorative power of HS. A most recent attempt was done by Saka et al. [106], which also indicated the criticisms [107] about the HS due to limited work done mathematically about the harmony search techniques.

10. Conclusions and potential directions

In this paper, we have done a through review about the Harmony Search algorithm from the aspect of algorithm structure with the major developments or variants summarised in Fig. 2. It is seen that the major developments happened in 2006 and onwards even though the basic harmony search algorithm was proposed in 2000 [1]. From Fig. 1, we can see that there is a boost on the study of harmony search in terms of number of publications, but if we compare Fig. 1 with Fig. 2, we notice the majority of these studies are about the applications, particularly engineering applications of the HS, which actually can be seen from the several review articles and book/book chapters as well. In the way of changing the algorithm structure, the majority of these modifications have been made by parameter adaptation, modifying the HS operator and hybridising with other algorithms. Nevertheless, the aforementioned literature have hardly addressed the algorithm or its variants from theoretical point of view even though they have great success in engineering applications. So, there is an urgent call for carrying out theoretical study of this algorithm and the variants, and also it is calling for other potential applications, such as applications in business management and finance.

Regarding the future research direction, apart from the above mentioned, researchers may make contributions towards improving algorithm structures. So far, researchers have proposed various algorithm structures of HS, and each variant enhanced its performance when compared with previous ones. However, most times, these tweaks did not provide any theoretical background. Thus, the future direction of HS can be how to theoretically improve the algorithm's structure. As we reviewed in Ref. [7], HS possesses a human-experience-based stochastic derivative instead of calculus-based derivative, which explains how this algorithm searches for an optimal solution by accumulating computational intelligence which is composed of memory consideration, pitch adjustment, and random selection. Is there any idea to theoretically find solution-searching direction and its step size at certain improvisation based on this stochastic derivative? We think that this can be a good future research direction of HS, and it will become another major leap if it is accomplished.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.swevo.2019.03.012.

References

[1] Z.W. Geem, Optimal Design of Water Distribution Networks Using Harmony Search. (Ph.D. thesis), Korea University, 2000.
[2] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[3] Z.W. Geem (Ed.), Music-Inspired Harmony Search Algorithm Theory and Applications, Vol. 191 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2009.
[4] Z.W. Geem, Optimal Design of Water Distribution Networks Using Harmony Search: Theory and Applications of Music-Inspired Harmony Search Algorithm, LAP Lambert Academic Publishing, 2009.
[5] Z.W. Geem (Ed.), Harmony Search Algorithms for Structural Design Optimization, 2010th Edition, Studies in Computational Intelligence, Springer Berlin Heidelberg, 2010.
[6] X. Wang, X.-Z. Gao, K. Zenger, An Introduction to Harmony Search Optimization Method, Springer, 2015.
[7] Z.W. Geem, M. Fesanghary, J.Y. Choi, M.P. Saka, J.C. Williams, M.T. Ayvaz, L. Li, S. Ryu, A. Vasebi, Recent Advances in Harmony Search, Advances in Evolutionary Algorithms, I-Tech Education and Publishing, Vienna, Austria, 2008, pp. 127–142. Ch. 7.
[8] G. Ingram, T. Zhang, Overview of Applications and Developments in the Harmony Search Algorithm, Springer-Verlag, Berlin Heidelberg, 2009, pp. 15–37. Ch. 2.
[9] Z.W. Geem, State-of-the-Art in the structure of harmony search algorithm, in: Vol. 270 of Recent Advances in Harmony Search Algorithm. Studies in Computational Intelligence, Springer Berlin Heidelberg, 2010, pp. 1–10. Ch. 1.
[10] G. Ingram, T. Zhang, An introduction to the harmony search algorithm, in: Vol. 2 of Advances in Process Systems Engineering, World Scientific, Singapore, 2010, pp. 301–335. Ch. 9.
[11] X.S. Yang, Nature-inspired Metaheuristic Algorithms, second ed., Luniver Press, Frome, UK, 2010.
[12] O.M. Alia, R. Mandava, The variants of the harmony search algorithm: an overview, Artif. Intell. Rev. 36 (1) (2011) 49–68.
[13] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J.D. Ser, M.N. Bilbao, S. Salcedo-Sanz, Z.W. Geem, A survey on application of the harmony search algorithm, Eng. Appl. Artif. Intell. 26 (2013) 1818–1831.
[14] S.A. Patil, D.A. Patel, An overview: improved harmony search algorithm and its applications in mechanical engineering, Int. J. Eng. Sci. Innovat. Technol. 2 (1) (2013) 433–444.
[15] N. Siddique, H. Adeli, Harmony search algorithm and its variants, Int. J. Pattern Recognit. 29 (8) (2015) 22 1539001.
[16] A. Assad, K. Deep, Applications of harmony search algorithm in data mining: a survey, in: Proceedings of Fifth International Conference on Soft Computing for Problem Solving, Vol. 437 of Advances in Intelligent Systems and Computing, Springer Science+Business Media Singapore, 2016, pp. 863–874.
[17] A. Askarzdeh, Solving electrical power system problems by harmony search: a review, Artif. Intell. Rev. 47 (2017) 217–251.
[18] K.S. Lee, Z.W. Geem, A new structural optimization method based on the harmony search algorithm, Comput. Struct. 82 (2004) 781–798.
[19] Z.W. Geem, J.H. Kim, G.V. Loganathan, Harmony search optimization: application to pipe network design, Int. J. Model. Simul. 22 (2) (2002) 125–133.

[20] Y.Y. Moon, Z.W. Geem, G.T. Han, Vanishing point detection for self-driving car using harmony search algorithm, Swarm Evolut. Comput. 41 (2018) 111–119.
[21] K.S. Lee, Z.W. Geem, S.-H. Lee, K.-W. Bae, The harmony search heuristic algorithm for discrete structural optimization, Eng. Optim. 37 (7) (2005) 663–684.
[22] L. Li, S.-C. Chi, X.-S. Chu, Location of non-circular slip surface using the modified harmony search method based on correcting strategy, Rock Soil Mech. 27 (10) (2006) 1714–1718 (in Chinese).
[23] L. Li, S.-C. Chi, G. Lin, X.-S. Chu, Slope stability analysis using extremum principle by pan jiazheng and harmony search method, Rock Soil Mech. 28 (1) (2007) 157–162 (in Chinese).
[24] L. Li, S.-C. Chi, Application of new version of harmony search algorithm in slope stability analysis, J. Water Resour. Architect. Eng. 5 (3) (2007) 1–6 (in Chinese).
[25] Y.M. Cheng, L. Li, T. Lansivaara, S.C. Chi, Y.J. Sun, An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis, Eng. Optim. 40 (2) (2008) 95–115.
[26] L. Li, S.-C. Chi, G. Lin, Chaos harmony search method and its application to local factor of safety method for soil slopes, Chin. J. Rock Mech. Eng. 25 (Supp. 1) (2006) 2763–2769 (in Chinese).
[27] B. Alatas, Chaotic harmony search algorithms, Appl. Math. Comput. 216 (2010) 2687–2699.
[28] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.
[29] M.G.H. Omran, M. Mahdavi, Global-best harmony search, Appl. Math. Comput. 198 (2008) 643–656.
[30] C. Cobos, D. Estupinan, J. Perez, GHS+LEM: global-best harmony search using learnable evolution models, Appl. Math. Comput. 218 (2011) 2558–2578.
[31] E. Ruano-Daza, C. Cobos, J. Torres-Jimenez, M. Mendoza, A. Paz, A multiobjective bilevel approach based on global-best harmony search for defining optimal routes and frequencies for bus rapid transit systems, Appl. Soft Comput. 67 (2018) 567–583.
[32] P. Chakraborty, G.G. Roy, S. Das, D. Jain, A. Abraham, An improved harmony search algorithm with differential mutation operator, Fundam. Inf. 95 (2009) 1–26.
[33] G.G. Roy, B.K. Panigrahi, P. Chakraborty, M.K. Mallick, On optimal feature selection using modified harmony search for power quality disturbance classification, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), 2009, pp. 1355–1360.
[34] H. Abedinpourshotorban, S. Hasan, S.M. Shamsuddin, N.F. As'Sahra, A differential-based harmony search algorithm for the optimization of continuous problems, Expert Syst. Appl. 62 (2016) 317–322.
[35] Z. Guo, S. Wang, X. Yue, H. Yang, Global harmony search with generalized opposition-based learning, Soft Comput. 21 (8) (2018) 2129–2137.
[36] B. Keshtegar, T. Ozbakkaloglu, A. Gholampour, Modeling the behavior of frpcon ned concrete using dynamic harmony search algorithm, Eng. Comput. 33 (2017) 415–430.
[37] V. Kumar, J.K. Chhabra, D. Kumar, Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems, J. Comput. Sci. 5 (2014) 144–155.
[38] V. Kumar, J.K. Chhabra, D. Kumar, Automatic data clustering using parameter adaptive harmony search algorithm and its application to image segmentation, J. Intell. Syst. 25 (4) (2016) 595–610.
[39] L. dos Santos Coelho, D.L. de Andrade Bernert, An improved harmony search algorithm for synchronization of discrete-time chaotic systems, Chaos, Solit. Fractals 41 (2009) 2526–2532.
[40] J.-H. Lee, Y.-S. Yoon, Modified harmony search algorithm and neural networks for concrete mix proportion design, J. Comput. Civ. Eng. 23 (1) (2009) 57–61.
[41] H. Dong, Y. Bo, M. Gao, T. Zhu, Improved harmony search for the detection with photon density wave, in: L. Zhou (Ed.), International Symposium on Photoelectronic Detection and Imaging 2007: Related Technologies and Applications, Proc. of SPIE, vol. 6625, 2008, p. 662523. Beijing, China.
[42] O. Hasancebi, F. Erdal, M.P. Saka, Adaptive harmony search method for structure optimization, J. Struct. Eng. 136 (4) (2010) 419–431.
[43] L. Li, G.-M. Yu, S.-B. Lu, Y. Hong, X.-S. Chu, Harmony search algorithm with dynamic adaptation for the determination of minimum factor of safety of soil slopes, in: 2009 Second International Symposium on Computational Intelligence and Design, IEEE Computer Society, 2009, pp. 3–5.
[44] C.-M. Wang, Y.-F. Huang, Self-adaptive harmony search algorithm for optimization, Expert Syst. Appl. 37 (2010) 2826–2837.
[45] N. Taherinejad, Highly reliable harmony search algorithm, in: Circuit Theory and Design, 2009. ECCTD 2009. European Conference on, IEEE, 2009, pp. 818–822.
[46] Q.-K. Pan, P.N. Suganthan, M.F. Tasgetiren, J.J. Liang, A self-adaptive global best harmony search algorithm for continuous optimization problems, Appl. Math. Comput. 216 (2010) 830–848.
[47] E.A. Portilla-Flores, Á. Sánchez-Márquez, et al., Enhancing the harmony search algorithm performance on constrained numerical optimization, IEEE Access 5 (2017) 25759–25780.
[48] Q.-K. Pan, P.N. Suganthan, M.F. Tasgetiren, A harmony search algorithm with ensemble of parameter sets, in: 2009 IEEE Congress on Evolutionary Computation (CEC 2009), 2009, pp. 1815–1820.
[49] M. Jaberipour, E. Khorram, Two improved harmony search algorithms for solving engineering optimization problems, Commun. Nonlinear Sci. Numer. Simul. 15 (11) (2010) 3316–3331.
[50] A. Kattan, R. Abdullah, A dynamic self-adaptive harmony search algorithm for continuous optimization problems, Appl. Math. Comput. 219 (2013) 8542–8567.
[51] L. Weihmann, D. Martins, D.L. de Andrade Bernert, L. dos Santos Coelho, Optimization of kinematically redundant manipulators force capabilities using improved harmony search, ABCM Symp. Ser. Mechatron. 5 (2012) 988–994.
[52] C. Peraza, F. Valdez, M. Garcia, P. Melin, O. Castillo, A new fuzzy harmony search algorithm using fuzzy logic for dynamic parameter adaptation, Algorithms 9 (69) (2016) 19.
[53] V. Kumar, J.K. Chhabra, D. Kumar, Variance-based harmony search algorithm for unimodal and multimodal optimization problems with applications to clustering, Cybernetics and Systems, Int. J. 45 (6) (2014) 486–511.
[54] V. Kumar, D. Kumar, Gene expression data clustering using variance-based harmony search algorithm, IETE J. Res. (2018), https://doi.org/10.1080/03772063.2018.1452641.
[55] Z.W. Geem, K.-B. Sim, Parameter-setting-free harmony search algorithm, Appl. Math. Comput. 217 (2010) 3881–3889.
[56] Z.W. Geem, Y.-H. Cho, Optimal design of water distribution networks using parameter-setting-free harmony search for two major parameters, J. Water Resour. Plan. Manag. 137 (4) (2011) 377–380.
[57] M. El-Abd, An improved global-best harmony search algorithm, Appl. Math. Comput. 222 (2013) 94–106.
[58] Z.W. Geem, Improved harmony search from ensemble of music players, in: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Springer Berlin Heidelberg, 2006, pp. 86–93.
[59] Y.-H. Tian, Y. Bo, M. Gao, Harmony annealing algorithm for multi-dimensional function optimization, Comput. Simulat. 21 (10) (2004) 79–82 (in Chinese).
[60] Y.-H. Tian, Y. Bo, M. Gao, Parameters choice criteria in harmony annealing for function optimization, Comput. Simulat. 22 (4) (2005) 70–74.
[61] J.-S. Wang, X.-W. Gao, PCNN edge detection of sintering pellets image based on hybrid harmony search algorithm, in: 2009 International Conference on Artificial Intelligence and Computational Intellige, IEEE Computer Society, 2009, pp. 535–539.
[62] X.Z. Gao, X. Wang, S.J. Ovaska, Modified harmony search methods for uni-modal and multi-modal optimization, in: 2008 Eighth International Conference on Hybrid Intelligent Systems, IEEE, Barcelona, Spain, 2008, pp. 65–72.
[63] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[64] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, Y. Alizadeh, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems, Comput. Methods Appl. Mech. Eng. 197 (2008) 3080–3091.
[65] W.S. Jang, H.I. Kang, B.H. Lee, Hybrid simplex-harmony search method for optimization problems, in: 2008 IEEE Congress on Evolutionary Computation, 2008, pp. 4158–4165.
[66] K.-C. Lo, HS-DLM Hybrid Searching Algorithm for Structural Optimization. (Master's thesis), National Central University, Taiwan, 2008 (in Chinese).
[67] X. Wang, X.Z. Gao, S.J. Ovaska, A hybrid optimization method for fuzzy classification systems, in: 8th International Conference on Hybrid Intelligent Systems, IEEE Computer Society, 2008, pp. 264–271.
[68] X. Wang, X.-Z. Gao, S.J. Ovaska, Fusion of clonal selection algorithm and harmony search method in optimisation of fuzzy classification systems, Int. J. Bio-Inspired Comput. 1 (1–2) (2009) 80–88.
[69] D. Zou, L. Gao, J. Wu, S. Li, Y. Li, A novel global harmony search algorithm for reliability problems, Comput. Ind. Eng. 58 (2010) 307–316.
[70] E. Valian, S. Tavakoli, S. Mohanna, An intelligent global harmony search approach to continuous optimization problems, Appl. Math. Comput. 232 (2014) 670–684.
[71] A. Chatterjee, S.P. Ghoshal, V. Mukherjee, Solution of combined economic and emission dispatch problems of power systems by an opposition-based harmony search algorithm, Electr. Power Energy Syst. 39 (2012) 9–20.
[72] W.-L. Xiang, M.-Q. An, Y.-Z. Li, R.-C. He, J.-F. Zhang, An improved global-best harmony search algorithm for faster optimization, Expert Syst. Appl. 41 (2014) 5788–5803.
[73] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Tech. Rep. TR06, Erciyes University, Kayseri, Turkey, 2005.
[74] B.H.F. Hasan, I.A. Doush, E.A. Maghayreh, F. Alkhateeb, M. Hamdan, Hybridizing harmony search algorithm with different mutation operators for continuous problems, Appl. Math. Comput. 232 (2014) 1166–1182.
[75] M. Castelli, S. Silva, L. Manzoni, L. Vanneschi, Geometric selective harmony search, Inf. Sci. 279 (2014) 468–482.
[76] J.L. Ponz-Tienda, A. Salcedo-Bernal, E. Pellicer, J. Benlloch-Marco, Improved adaptive harmony search algorithm for the resource leveling problem with minimal lags, Autom. ConStruct. 77 (2017) 82–92.
[77] H.-B. Ouyang, L.-Q. Gao, S. Li, X.-Y. Kong, Q. Wang, Improved harmony search algorithm: LHS, Appl. Soft Comput. 53 (2017) 133–167.
[78] A. Banerjee, V. Mukherjee, S.P. Ghoshal, An opposition-based harmony search algorithm for engineering optimization problems, Ain Shams Eng. J. 5 (2014) 85–101.
[79] B. Zhang, Q.-K. Pan, X.-L. Zhang, P.-Y. Duan, An effective hybrid harmony search-based algorithm for solving multidimensional knapsack problems, Appl. Soft Comput. 29 (2015) 288–297.
[80] M.-Y. Cheng, D. Prayogo, Y.-W. Wu, M.M. Lukito, A hybird harmony search algorithm for discrete sizing optimization of truss structure, Autom. ConStruct. 69 (2016) 21–33.
[81] N. Mohamed, A.L.A. Ramli, A.A. Majid, A.R.M. Piah, Modifed harmony search, J. Phys. Conf. Ser. 890 (2017) 7 012097.
[82] A. Assad, K. Deep, A two-phase harmony search algorithm for continuous optimization, Comput. Intell. 33 (2017) 1038–1075.

[83] A. Assad, K. Deep, A hybrid harmony search and simulated annealing algorithm for continuous optimization, Inf. Sci. 450 (2018) 246–266.
[84] Z.W. Geem, H. Hwangbo, Application of harmony search to multi-objective optimization for satellite heat pipe design, in: Proceedings of US-Korea Conference on Science, Technology & Entrepreneurship, 2006, pp. 1–3.
[85] J. Ricart, G. Huttemann, J. Lima, B. Baran, Multiobjective harmony search algorithm proposals, Electron. Notes Theor. Comput. Sci. 281 (2011) 51–67.
[86] S. Sivasubramani, K.S. Swarup, Multi-objective harmony search algorithm for optimal power flow problem, Electr. Power Energy Syst. 33 (2011) 745–752.
[87] W. Li, W. Du, W. Tang, Y. Pan, J. Zhou, Z.-M. Lin, Parallel algorithm of multiobjective optimization harmony search based on cloud computing, J. Algorithms Comput. Technol. 11 (3) (2017) 301–313.
[88] Y.M. Cheng, L. Li, S.C. Chi, Performance studies on six heuristic global optimization methods in the location of critical slip surface, Comput. Geotech. 34 (2007) 462–484.
[89] M. Ayob, M. hadwan, M.Z.A. Nazri, Z. Ahmad, Enhanced harmony search algorithm for nurse rostering problems, J. Appl. Sci. 16 (6) (2013) 846–853.
[90] Z.-W. Jiang, H.-X. Zhang, The application of improved harmony search algorithm for solving shortest path problems, in: International Conference on Computational Science and Engineering (ICCSE 2015), Atlantis Press, 2015, pp. 38–43.
[91] M. Nazari-Heris, A.F. Babaei, B. Mohammadi-Ivatloo, S. Asadi, Improved harmony search algorithm for the solution of non-linear non-convex short-term hydrothermal scheduling, Energy 151 (2018) 226–237.
[92] G.F. Medeiros, M. Kripka, Modified harmony search and its application to cost minimization of RC columns, Adv. Comput. Design 2 (1) (2017) 1–13.
[93] S. Tuo, J. Zhang, X. Yuan, Z. He, Y. Liu, Z. Liu, Niche harmony search algorithm for detecting complex disease associated high-order SNP combinations, Sci. Rep. 7 (1) (2017) 18 11529.
[94] L.J. Li, Z.B. Huang, F. Liu, Q.H. Wu, A heuristic particle swarm optimizer for optimization of pin connected structures, Comput. Struct. 85 (2007) 340–349.
[95] L.J. Li, Z.B. Huang, F. Liu, A heuristic particle swarm optimization method for truss structures with discrete variables, Comput. Struct. 87 (2009) 435–443.
[96] H.-Q. Li, L. Li, A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems, in: 2007 International Conference on Intelligence Pervasive Computing, IEEE Computer Society, 2007, pp. 94–97.
[97] L. Li, S.-C. Chi, Y.-M. Cheng, G. Lin, Improved genetic algorithm and its application to determination of critical slip surface with arbitrary shape in soil slope, Shuili Xuebao 38 (2) (2007) 157–162 (in Chinese).
[98] H. Moeinzadeh, E. Asgarian, M. Zanjani, A. Rezaee, M. Seidi, Combination of harmony search and linear discriminate analysis to improve classification, in: 2009 Third Asia International Conference on Modelling & Simulation, IEEE Computer Society, 2009, pp. 131–135.
[99] R. Alberti, A.K. Nagar, Harmony search based algorithm for complete information equilibrium in infinite game, in: Proceedings of the Third International Conference on Soft Computing for Problem Solving, Vol. 258 of Advances in Intelligent Systems and Computing, Springer India, 2014, pp. 503–511.
[100] M. Salem, M.F. Khelfi, The variants of harmony search algorithm: statistical comparison, in: Proceedings of Third World Conference on Complex Systems (WCCS), IEEE, 2015, pp. 1–5.
[101] Z.W. Geem, Particle-swarm harmony search for water network design, Eng. Optim. 41 (4) (2009) 297–311.
[102] Z.W. Geem, Can music supplant math in environmental planning? Leonardo 48 (2) (2015) 147–150.
[103] Z.W. Geem, Novel derivative of harmony search algorithm for discrete design variables, Appl. Math. Comput. 199 (2008) 223–230.
[104] Z.W. Geem, J.C. Williams, Harmony search and ecological optimization, Int. J. Energy Environ. 1 (2) (2007) 150–154.
[105] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, B.K. Panigrahi, Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization, IEEE Trans. Syst. Man Cybern. B Cybern. 41 (1) (2011) 89–106.
[106] M.P. Saka, O. Hasancebi, Z.W. Geem, Metaheuristics instructural optimization and discussions on harmony search algorithm, Swarm Evolut. Comput. 28 (2016) 88–97.
[107] D. Weyland, A rigorous analysis of the harmony search algorithm: how the research community can be misled by a “novel” methodology, Int. J. Appl. Metaheuristic Comput. 1 (2) (2010) 50–60.