Artif Intell Rev (2011) 36:49–68
DOI 10.1007/s10462-010-9201-y

The variants of the harmony search algorithm: an overview

Osama Moh'd Alia · Rajeswari Mandava

Published online: 7 January 2011

Abstract The harmony search (HS) algorithm is a relatively new population-based metaheuristic optimization algorithm. It imitates the music improvisation process where musicians improvise their instruments' pitch by searching for a perfect state of harmony. Since the emergence of this algorithm in 2001, it attracted many researchers from various fields especially those working on solving optimization problems. Consequently, this algorithm guided researchers to improve on its performance to be in line with the requirements of the applications being developed. These improvements primarily cover two aspects: (1) improvements in terms of parameters setting, and (2) improvements in terms of hybridizing HS components with other metaheuristic algorithms. This paper presents an overview of these aspects, with a goal of providing useful references to fundamental concepts accessible to the broad community of optimization practitioners.

Keywords Harmony search · Metaheuristic optimization

1 Introduction

Optimization is the process of selecting the best element from some sets of available alternatives under certain constraints (if any). This process can be solved by minimizing or maximizing the objective or cost function of the problem. In each iteration of the optimization process, choosing the values (e.g. real or integer variables) from within an allowed set is done systematically until the minimum or maximum result is reached or when the stopping criterion is met. Optimization techniques are used on a daily basis for industrial planning, resource allocation, econometrics problems, scheduling, decision making,

O. M. Alia (✉) · R. Mandava
Computer Vision Research Group, School of Computer Sciences, University Sains Malaysia, 11800 USM, Penang, Malaysia
e-mail: sm_alia@yahoo.com

R. Mandava
e-mail: mandava@cs.usm.my

engineering, computer science applications. Research in the optimization field is very active and new optimization methods are being developed regularly.

Formally, optimization problem can be formulated as follows (Blum and Roli 2003):

$$\text{minimize\textbackslash maximize} \quad f(s), \; s \in S \tag{1}$$

f is considered as an objective function, $S \mapsto \Re^n$ is called a search (or a solution) space, as each solution in this set $s \in S$ satisfy all the given constraints and can be a candidate solution. $s_i \; of \; s = (s_1, s_2, \ldots, s_n)$ are called decision variables, and they can be either continous, or discrete or both. n here is the number of decision variables. Where each decision variable has it's own domain $D_1, \ldots, D_n$.

Based on the solution set S, the optimization problems can be categorized into several groups such as continuous and discrete or combinatorial and variational (Velho et al. 2008). Whereas, another categorization can be given based on the properties of the objective function such as linear, quadratic, convex (or convex), sparse, or separable. Another categorization can be also given based on the existence or absent of the problem constraints. These classifications are very important since they can specify the natural of the proposed optimization algorithm (Velho et al. 2008).

As such, various techniques have been come into sight for tackling different kinds of optimization problems. In the broadest sense, these techniques can be classified into exact and approximate algorithms (Stutzle 1998). Exact algorithms such as branch & bound and dynamic programming are guaranteed to find an optimal solution in bounded time (Weise 2009; Stutzle 1998). However, for the optimization problems that fall under the NP-hard category, exact methods require exponential computational time, which is impractical for practical purposes/applications (Stutzle 1998; Garey and Johnson 1979). Due to this, attention in the past four-decades has been given to approximate methods. Although this did not guarantee optimal solutions, such methods allow significant reduction in computational time (Stutzle 1998).

Metaheuristic algorithms are well known approximate algorithms which can solve optimization problems with satisfying results (Blum and Roli 2003, 2008). Metaheuristic came forth to overcome the major drawback of the well known approximate algorithms, local search algorithms (also known as iterative improved local search) and its improved version iterative local search algorithms, that may stop at a very poor quality local optima. As well as the other drawback of iterative local search algorithm which is the increase of algorithm's computational complexity when the problem's dimensionality increases (i.e. the number of local minima may increase exponentially).

Metaheuristics are general heuristic methods which are applicable to a wide range of different optimization problems. Metaheuristic algorithms can be defined as "high level strategies for exploring search spaces by using different methods" Blum and Roli (2003) or "the collection of ideas of how to use the search history to generate new solutions and how to extract the necessary information from the generated solutions." Yagiura and Ibaraki (2001).

The main goal of metaheuristic algorithms is to avoid the disadvantages of iterative improvement and, in particular, the local optima problem. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more intelligent way than just providing random initial solutions.

Metaheuristic algorithms have many features such as its simplicity, robustness and flexibility that make them very attractive research area (Yagiura and Ibaraki 2001). Many of them are inspired by natural phenomena. Examples are particle swarm optimization, simulated annealing, genetic algorithms and harmony search. These algorithms are intelligently inspired by natural phenomena to provide efficient solution techniques to yield high quality solutions in a reasonable time.

Furthermore, many classifications of metaheuristic algorithms can be found in the literature such as Nature-inspired versus non-nature inspired, Population-based versus local search-based (i.e. trajectory methods), Dynamic versus static objective function, One versus various neighborhood structures, and Memory usage versus memory-less methods (Blum and Roli 2003). Among them, population-based versus local search-based is considered the most used and can describe the metaheuristic algorithms very well (Tsai 2009; Blum and Roli 2003).

The metaheuristic population-based algorithms deal in every iteration of the algorithm with a set (i.e., a population) of solutions rather than with a single solution as in local search-based algorithms. As they deal with a population of solutions, population-based algorithms provide a natural, intrinsic way for the exploration of the search space. Yet, the final performance depends strongly on the way the population is manipulated.

During the last decades a lot of population-based metaheuristic algorithms were proposed. One population-based category is the evolutionary based algorithms including Genetic Programming, Evolutionary Programming, Evolutionary Strategies, Genetic Algorithms, Differential Evolution, Harmony Search algorithm, etc. Other category is the swarm based algorithms including Ant Colony Optimization, Particle Swarm Optimization, Bees Algorithms, Honey Bee Mating Optimization, etc.

The harmony search algorithm (Geem et al. 2001) is one of the most recently developed optimization algorithm and at a same time, it is one the most efficient algorithm in the field of combinatorial optimization (Geem 2009c). Since the emergence of this algorithm in 2001 by Geem et al., it attracted many researchers from various fields especially those working on solving optimization problems (Ingram and Zhang 2009). Consequently, this algorithm guided researchers to improve on its performance to be in line with the requirements of the applications being developed. These improvements primarily cover two aspects: (1) improvements in terms of parameters setting, and (2) improvements in terms of hybridizing HS components with other metaheuristic algorithms. This paper presents an overview of these aspects, with a goal of providing useful references to fundamental concepts accessible to the broad community of optimization practitioners.

This paper is organized as follows: Sect. 2 overview the harmony search algorithm with its basic concepts and Sect. 3 describe the harmony search characteristics. Section 4 provides the reader with most relative HS's modifications and improvements. In Sect. 5 we conclude this work.

2 Harmony search algorithm

Harmony search (HS) Geem et al. (2001) is a relatively new population-based metaheuristic optimization algorithm, that imitates the music improvisation process where the musicians improvise their instruments' pitch by searching for a perfect state of harmony. It was able to attract many researchers to develop HS-based solutions for many optimization problems such as music composition (Geem and Choi 2007), Sudoku puzzle solving (Geem 2007b), tour planning (Geem et al. 2005a), web page clustering (Forsati et al. 2008; Mahdavi and Abolhassani 2009), structural design (Lee et al. 2004; Geem 2009d), water network design (Geem 2009a), vehicle routing (Geem et al. 2005b), dam scheduling (Geem 2007b), ground water modeling (Ayvaz 2007, 2009), soil stability analysis (Cheng et al. 2008), ecological conservation (Geem and Williams 2008), energy system dispatch (Vasebi et al. 2007),

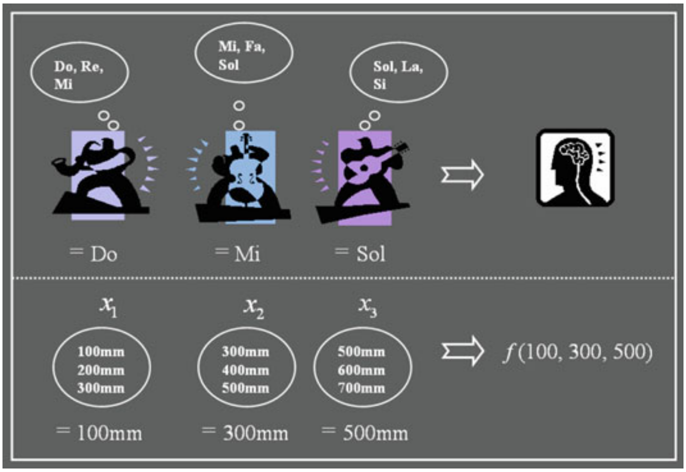

Fig. 1 Analogy between improvisation and optimization, obtained from Geem (2010)

heat exchanger design (Fesanghary et al. 2009), transportation energy modeling (Ceylan et al. 2008), satellite heat pipe design (Geem and Hwangbo 2006), medical physics (Panchal 2009), medical image (Alia et al. 2009b, 2010), timetabling (Al-Betar et al. 2008, 2010a), RNA structure prediction (Mohsen et al. 2010), image segmentation (Alia et al. 2009a,c), etc. HS imitates the natural phenomenon of musicians' behavior when they cooperate the pitches of their instruments together to achieve a fantastic harmony as measured by aesthetic standards. This musicians' prolonged and intense process led them to the perfect state. It is a very successful metaheuristic algorithm that can explore the search space of a given data in parallel optimization environment, where each solution (harmony) vector is generated by intelligently exploring and exploiting a search space (Geem 2009a). It has many features that make it as a preferable technique not only as standalone algorithm but also to be combined with other metaheuristic algorithms.

Harmony search as mentioned mimic the improvisation process of musicians' with an intelligent way as can be seen in Fig. 1. The analogy between improvisation and optimization is likely as follows Geem (2010):

1. Each musician corresponds to each decision variable;
2. Musical instrument's pitch range corresponds to the decision variable's value range;
3. Musical harmony at a certain time corresponds to the solution vector at a certain iteration;
4. Audience's aesthetics corresponds to the objective function.

Just like musical harmony is improved time after time, solution vector is improved iteration by iteration. In general, HS has five steps and they are described as in Geem et al. (2005a) as follow:

The optimization problem is defined as follow:

$$\begin{array}{ll} \text{minimize\textbackslash maximize} & f(a), \\ \text{subject to} & a_i \in \mathbf{A}_i, \quad i = 1, 2, \ldots, N \end{array} \tag{2}$$

where $f(a)$ is an objective function; a is the set of each decision variable (a_i); $\mathbf{A}_i$ is the set of possible range of values for each decision variable, ${}_La_i \leq \mathbf{A}_i \geq {}_Ua_i$; and N is the number of decision variables.

Then, the parameters of the HS are initialized. These parameters are:

1. Harmony Memory Size (HMS) (i.e. number of solution vectors in harmony memory);
2. Harmony Memory Considering Rate (HMCR), where HMCR $\in [0, 1]$;
3. Pitch Adjusting Rate (PAR), where PAR $\in [0, 1]$;
4. Stopping Criteria (i.e. number of improvisation (NI));

More explanation of these parameters is in the next steps.

2.1 Initialize harmony memory

The harmony memory (HM) is a matrix of solutions with a size of HMS, where each harmony memory vector represents one solution as can be seen in Eq. 3. In this step, the solutions are randomly constructed and rearranged in a reversed order to HM, based on their objective function values such as $f(a^1) \leq f(a^2) \cdots \leq f(a^{\text{HMS}})$.

$$\text{HM} = \left[\begin{array}{cccc|c} a_1^1 & a_2^1 & \ldots & a_N^1 & f(a^1) \\ a_1^2 & a_2^2 & \ldots & a_N^2 & f(a^2) \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ a_1^{\text{HMS}} & a_2^{\text{HMS}} & \ldots & a_N^{\text{HMS}} & f(a^{\text{HMS}}) \end{array}\right] \tag{3}$$

2.2 Improvise new harmony

This step is the essence of the HS algorithm and the cornerstone that has been building this algorithm. In this step, the HS generates (improvises) a new harmony vector, $a' = (a'_1, a'_2, a'_3, \ldots, a'_N)$. It is based on three operators: memory consideration; pitch adjustment; or random consideration. In the memory consideration, the values of the new harmony vector are randomly inherited from the historical values stored in HM with a probability of HMCR. Therefore, the value of decision variable (a'_1) is chosen from $(a_1^1, a_1^2, a_1^3, \ldots, a_1^{\text{HMS}})$ that is stored in HM. The next decision variable (a'_2) is chosen from $(a_2^1, a_2^2, a_2^3, \ldots, a_2^{\text{HMS}})$, and the other decision variables, $(a'_3, a'_4, a'_5, \ldots)$, are chosen consecutively in the same manner with the probability of HMCR $\in [0, 1]$. The usage of HM is similar to the step where the musician uses his or her memory to generate an excellent tune. This cumulative step ensures that good harmonies are considered as the elements of new harmony vectors.

Out of that, where the other decision variable values are not chosen from HM, according to the HMCR probability test, they are randomly chosen according to their possible range, $a'_i \in \mathbf{A}_i$. This case is referred to as random consideration (with a probability of $(1 - \text{HMCR})$), which increases the diversity of the solutions and drives the system further to explore various diverse solutions so that global optimality can be attained.

The following equation summarized these two steps i.e. memory consideration and random consideration.

$$a'_i \leftarrow \begin{cases} a'_i \in \left\{a_i^1, a_i^2, a_i^3, \ldots, a_i^{\text{HMS}}\right\} & w.p.\ \text{HMCR} \\ a'_i \in A_i & w.p.\ (1 - \text{HMCR}) \end{cases} \tag{4}$$

Furthermore, the additional search for good solutions in the search space is achieved through tuning each decision variable in the new harmony vector, $a' = (a'_1, a'_2, a'_3, \ldots, a'_N)$, inherited from HM using PAR operator. These decision variables (a'_i) are examined and to be tuned with the probability of PAR $\in [0, 1]$ as in Eq. 5.

$$a'_i \leftarrow \begin{cases} \textit{Adjusting Pitch} & w.p.\ \text{PAR} \\ \textit{Doing Nothing} & w.p.\ (1 - \text{PAR}) \end{cases} \tag{5}$$

```
HS Algorithm
begin
Define fitness function f(a), a = (a_1, a_2, ..., a_N)^T
Define (HMCR),(PAR),(HMS),(EOR)
Define Maximum number of iterations (NI).
HM ← GenerateInitialPoplution()
min = minimum visible value.
max = maximum visible value.
while (iter ≤ NI) do
   while (a_i ≤ number of variables) do
      if (rand ∈ (0,1) ≤ HMCR) then
         choose a value from HM for i
         if (rand ∈ (0,1) ≤ PAR) then
            adjust the value of i by:
            a_inew = a_iold + rand ∈ (0,1) × bw
         end if
      else
         choose a random variable:
         a_i = min + rand ∈ (0,1) × (max − min)
      end if
   end while
   if (FitFun(new harmony solution) ≤ worst(FitFun(HM)) then
      accept the new harmony and replace the worst in HM with it.
   end if
end while
best=find the current best solution
end
```

Fig. 2 Pseudo code of the HS algorithm

If a generated random number $rnd \in [0, 1]$ within the probability of PAR then, the new decision variable (a_i') will be adjusted based on the following equation:

$$(a_i') = (a_i') \pm rand() * \text{bw} \tag{6}$$

Here, bw is an arbitrary distance bandwidth used to improve the performance of HS and $(rand())$ is a function that generates a random number $\in [0, 1]$. Actually, bw determines the amount of movement or changes that may have occurred to the components of the new vector. The value of bw is based on the optimization problem itself i.e. continuous or discrete. In general, the way that the parameter (PAR) modifies the components of the new harmony vector is an analogy to the musicians' behavior when they slightly change their tone frequencies in order to get much better harmonies. Consequently, it explores more solutions in the search space and improves the searching abilities.

All of these operators are well illustrated using pseudo code as in Fig. 2.

2.3 Update the harmony memory

In order to update HM with the new generated vector $a' = (a_1', a_2', a_3', \ldots, a_N')$, the objective function is calculated for each new harmony vector $f(a')$. If the objective function value for the new vector is better than the worst harmony vector stored in HM, then the worst harmony vector is replaced by the new vector. Otherwise, this new vector is ignored.

$$a' \in \text{HM} \ \wedge \ a^{worst} \notin \text{HM} \tag{7}$$

However, for the diversity of harmonies in HM, other harmonies (in terms of least-similarity) can be considered. Also, the maximum number of identical harmonies in HM can be considered in order to prevent premature HM.

2.4 Check the stopping criterion

The iteration process in steps 3&4 is terminated when the maximum number of improvisations (NI) is reached. Finally, the best harmony memory vector is selected and is considered to be the best solution to the problem under investigation.

3 Harmony search characteristics

Harmony search algorithm has several characteristics that make it one of the most important metaheuristic algorithms (Geem et al. 2001). HS possesses several characteristics that distinguish it from other metaheuristics such as (1) the generation of a new vector after considering all the existing vectors, rather than considering only two vectors as in GA (parents); (2) independent consideration each of decision variable in a vector; (3) the consideration of continuous decision variable values without any loss of precision; (4) it does not require decimal-binary conversions or a fixed number (2n) of decision variable values as in GA; and (5) it does not require any starting values of the decision variables nor does it require complex derivatives as in gradient-based methods.

The other important strengths of HS are their improvisation operators, memory consideration; pitch adjustment; and random consideration, that play a major rule in achieving the desired balance between the two major extremes for any optimization algorithm, Intensification and diversification (Yang 2009b). Essentially, both pitch adjustment and random consideration are the key components of achieving the desired diversification in HS. In random consideration, the new vector's components are generated at random mode, has the same level of efficiency as in other algorithms that handle randomization, where this property allows HS to explore new regions that may not have been visited in the search space. While, the pitch adjustment adds a new way for HS to enhance its diversification ability by tuning the new vector's component within a given bandwidth. A small random amount is added to or subtracted from an existing component stored in HM. This operator, pitch adjustment, is a fine-tuning process of local solutions that ensures that good local solutions are retained, while it adds a new room for exploring new solutions. Further to that, pitch adjustment operator can also be considered as a mechanism to support the intensification of HS through controlling the probability of PAR. The intensification in the HS algorithm is represented by the third HS operator, memory consideration. A high harmony acceptance rate means that good solutions from the history/memory are more likely to be selected or inherited. This is equivalent to a certain degree of elitism. Obviously, if the acceptance rate is too low, solutions will converge more slowly.

Finally, the structure of the HS algorithm is relatively easy. This advantage makes it very flexible to combine HS with other metaheuristic algorithms as can be seen in Sect. 4.

4 Variants of harmony search

Harmony search algorithm got the attention of many researchers to solve many optimization problems such as engineering and computer science problems. Consequently, the interest in this algorithm led the researchers to improve and develop its performance in line with the requirements of problems that are solved. These improvements primarily cover two aspects: (1) improvement of HS in term of parameters setting, and (2) improvements in term of hybridizing of HS components with other metaheuristic algorithms. This section will highlight these developments and improvements to this algorithm in the ten years of this algorithm's age. The first part introduces the improvement of HS in term of parameters setting, while the second part introduces the development of HS in term of hybridizing of HS with other metaheuristic algorithms.

4.1 Variants based on parameters setting

The proper selection of HS parameter values is considered as one of the challenging task not only for HS algorithm but also for other metaheuristic algorithms. This difficulty is a result of different reasons, and the most important one is the absence of general rules governing this aspect. Actually, setting these values is problem dependant and therefore the experimental trials are the only guide to the best values. However, this matter guides the research into new variants of HS. These variants are based on adding some extra components or concepts to make part of these parameters dynamically adapted.

In this context, a very well known improvement to HS is done by Mahdavi and his colleagues that is named under Improved Harmony Search (IHS) (Mahdavi et al. 2007). Their proposed algorithm includes dynamic adaptation for both pitch adjustment rate (PAR) and bandwidth (bw) values. The PAR value is linearly increased in each iteration of HS by using the following equation:

$$\mathrm{PAR}(\mathrm{gn}) = \mathrm{PAR}_{\min} + \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{\mathrm{NI}} \times \mathrm{gn} \tag{8}$$

where $\mathrm{PAR}(\mathrm{gn})$ is the PAR value for each generation, $\mathrm{PAR}_{\min}$ and $\mathrm{PAR}_{\max}$ are the minimum pitch adjusting rate and maximum pitch adjusting rate respectively. NI is the maximum number of iterations (improvisation) and gn is the generation number. The bandwidth (bw) value is exponentially decreased in each iteration of HS by using the following equation:

$$\mathrm{bw}(\mathrm{gn}) = \mathrm{bw}_{\min} + \frac{\mathrm{bw}_{\max} - \mathrm{bw}_{\min}}{\mathrm{NI}} \times \mathrm{gn}, \tag{9}$$

where $\mathrm{bw}(\mathrm{gn})$ is the bandwidth value for each generation, $\mathrm{bw}_{\max}$ is the maximum bandwidth, $\mathrm{bw}_{\min}$ is the minimum bandwidth and gn is the generation number.

Another important improvement done by Omran and Mahdavi Omran and Mahdavi (2008) and named as Global-best harmony search (GHS). Their work was inspired by PSO concepts, the global best particle that is the fittest particle in term of objective function among other particles in the swarm. The authors proposed GHS to overcome the expected limitation of IHS as they reported. The limitation is the difficulty of determining the lower and upper bound of automatic bandwidth (bw) proposed by Mahdavi et al. Mahdavi et al. (2007). Therefore, they incorporate the PSO concept, global best particle, by replacing the bw parameter altogether and adding a randomly selected decision variables from the best harmony vector in HM as illustrated in Fig. 3.

Wang and Huang (2010) proposed a new variation of HS algorithm that focuses on the dynamic selection of bw and PAR parameters. Actually, they totally replaced bw parameter with a new concept that is based on using the maximal and minimal values in HM. This can be done through using the following equations:

```
if (rand ∈ (0,1) ≤ PAR) then
   x_i' = x_k^best % k ∈ [0,1] from thebest harmony vector in HM
end if
```

Fig. 3 Pseudo code to illustrate (GHS)

$$trial^i + \left[\max\left(\mathrm{HM}^i\right) - trial^i\right] \times ran\,[0, 1) \tag{10}$$

$$trial^i - \left[trial^i - \min\left(\mathrm{HM}^i\right)\right] \times ran\,[0, 1) \tag{11}$$

where $ran[0, 1)$ is a generated random number ranged from 0 to less than 1, $trial^i$ is the ith variable selected from HM, and $\max(\mathrm{HM}^i)$, $\min(\mathrm{HM}^i)$ are the highest and lowest values of the ith variable in the HM respectively. Wang and Huang used the same modification proposed by Mahdavi et al. (2007) to dynamically adapt PAR values during the search process, but this time in the opposite manner (decreasing the values of PAR). Finally they also modified the HM initialization step by using low-discrepancy sequences (Lecot 1989).

Mukhopadhyay et al. (2008) suggested that bw will be the standard deviation of the current population when HMCR is close to 1.

$$\mathrm{bw(gn)} = \sigma(x_i) = \sqrt{var(x_i)} \tag{12}$$

Chakraborty et al. (2009) proposed a new improvement to HS through inspiring the Differential Evolution (DE) mutation operator. Their proposed algorithm, named (DHS), replaces the pitch adjustment operation in classical HS with a mutation strategy borrowed from the DE (DE/rand/1/bin class) algorithm. This step was accomplished through the mutation of the target vector with the difference of two randomly selected HM members as shown in Eq. 13. It is worth mentioning here that the target vector is the new generated vector considering the memory selection step (with the probability of HMCR), and randomness (with the probability of $1 - \mathrm{HMCR}$). And the PAR operation is totally removed, thus, all new generated vector will be mutated according to the following equation:

$$\mathbf{x}_i{}' = \mathbf{x}_i{}' + F(\mathbf{x}_{r1} - \mathbf{x}_{r2}) \tag{13}$$

Hasancebi et al. (2009) and Saka and Hasancebi (2009) proposed a new adaptation for HS by making both HMCR and PAR change dynamically during the improvisation process of HS. This step is to make the selection of these parameter values problem independent, therefore, improves the performance of HS in finding an optimal solutions. Initially HMCR, PAR values are respectively set to $\mathrm{HMCR}^{(0)}$ and $\mathrm{PAR}^{(0)}$ in the initialization step of HM, then the dynamic calculating of these parameters is adapted as follows:

$$(\mathrm{HMCR})^K = \left(1 + \frac{1 - (\mathrm{HMCR})'}{(\mathrm{HMCR})'} \cdot e^{-\gamma \cdot N(0,1)}\right)^{-1} \tag{14}$$

$$(\mathrm{PAR})^K = \left(1 + \frac{1 - (\mathrm{PAR})'}{(\mathrm{PAR})'} \cdot e^{-\gamma \cdot N(0,1)}\right)^{-1} \tag{15}$$

where $(\mathrm{HMCR})^K$ and $(\mathrm{PAR})^K$ are the sampled values of the adapted parameters for a new harmony vector. $N(0, 1)$ is a normally distributed random number, γ is the learning rate of adapted parameters, which is recommended to be selected within a range of $\in [0.25, 0.50]$. $(\mathrm{HMCR})'$, $(\mathrm{PAR})'$ which are the average values of improvisation parameters within the harmony memory matrix, obtained by averaging the corresponding values of all the solution vectors within the HM matrix, that are,

$$(\mathrm{HMCR})' = \frac{\sum_{i=1}^{\mathrm{HMS}} \mathrm{HMCR}^i}{\mathrm{HMS}} \tag{16}$$

$$(\mathrm{PAR})' = \frac{\sum_{i=1}^{\mathrm{HMS}} \mathrm{PAR}^i}{\mathrm{HMS}} \tag{17}$$

In general, these new set of values for $(\mathrm{HMCR})^K$ and $(\mathrm{PAR})^K$ are calculated based on probabilistic selecting from around average values of these parameters observed within the current harmony memory matrix.

Degertekin (2008) proposed a new HM initialization technique that generated two times of HMS initial harmonies but placed only the best HMS of these into the initial HM.

Kattan et al. (2010) used HS as a new training technique for feed-forward artificial neural networks (ANN). They modified the standard stopping criteria that is based on counting the number of improvisation steps to best-to-worst (BtW) harmony ratio in the current harmony memory. Therefore they modified the existing improved version of HS (Mahdavi et al. 2007) to suit the new stopping criterion. These Modification would be more suitable for ANN training since parameters and termination would depend on the quality of the attained solution as reported by the authors.

A multiple PAR strategy was also proposed in the literature. Geem et al. in 2005a proposed a Multi-pitch Adjusting Rate (multiple PAR) for Generalized Orienteering Problem. They proposed three PAR's that are the rates of moving to nearest, second nearest, and third nearest cities, respectively.

Al-Betar et al. (2010a) also proposed a Multi-pitch Adjusting Rate strategy for enhancing the performance of HS in solving course timetabling problem. They proposed eight procedures instead of using one PAR value, each of which is controlled by its PAR value range. Each pitch adjustment procedure is responsible for a particular local change in the new harmony. Furthermore, the acceptance rule for each pitch adjustment procedure is changed to accept the adjustment that leads to a better or equal objective function.

Geem (2006) used fixed parameter values, such as HMS, HMCR, PAR, and NI, while bw was set to a range from 1 to 10% of the total value data range.

Furthermore, some researchers have proposed adaptive parameter theories that enable HS to automatically acquire the best parameter values at each iteration (Geem 2009d).

A summarization of what has been mentioned is described in the following Table 1.

4.2 Variants based on hybridization of HS with other metaheuristic

In this section, the hybridization of HS with other Metaheuristic is introduced. This hybridization can be categorized into two approaches; the first approach is the integration of some components of other metaheuristic algorithms into HS structure, while the second approach is in the opposite direction, where the integration of some HS components is integrated into other metaheuristic algorithm structure (Ingram and Zhang 2009). In general, such hybridization process is introduced to improve the search abilities of these optimization algorithms (Blum and Roli 2008; Grosan and Abraham 2007). In both cases, the origin of the ability of HS algorithm to be integrated with other metaheuristic return to the relative ease and flexible structure of HS as reported in Yang (2009a).

Table 1 Variants of HS based on parameters setting improvements

Algorithm name	Modified parameters	Description	References
IHS	PAR, bw	Dynamic setting during the improvisation process, where the PAR value is linearly increased and the bandwidth (bw) value is exponentially decreased	Mahdavi et al. (2007)
GHS	bw	The PSO concept, global best particle, is incorporated by replacing the bw parameter altogether and adding a randomly selected decision variables from the best harmony vector in HM	Omran and Mahdavi (2008)
HS-variant	PAR, bw, HM initialization	Dynamic selection of bw and PAR parameters. bw is totally replaced by maximal and minimal values in HM. The PAR value is linearly decreased. The initialization of HM is performed using low-discrepancy sequences	Wang and Huang (2010)
HS-variant	bw	bw will be the standard deviation of the current population when HMCR is close to 1	Mukhopadhyay et al. (2008)
DHS	PAR	A replacement of the PAR operator with a mutation strategy borrowed from the DE is proposed	Chakraborty et al. (2009)
HS-variant	HMCR, PAR	Dynamic setting of PAR and HMCR is occurred during the improvisation process through using the sampling of control parameters strategy	Hasancebi et al. (2009), Saka and Hasancebi (2009)
HS-variant	HM	Generating two times of HMS initial harmonies but placed only the best HMS of these into the initial HM	Degertekin (2008)
HS-variant	Stopping criterion	The stopping criterion is replaced by best-to-worst (BtW) harmony ratio in the current harmony memory	Kattan et al. (2010)
HS-variant	PAR	A Multi-pitch Adjusting Rate strategy is proposed	Geem et al. (2005a), Al-Betar et al. (2010a)
HS-variant	bw	bw set to a range from 1 to 10% of the total value data range	Geem (2006)

4.2.1 Hybridizing HS with other metaheuristic components

In the first approach, where other metaheuristic components or concepts are integrated into HS, different approaches have been proposed during the last few years.

Taherinejad (2009) modified HS by inspiring the SA's way of cooling (accepting some bad solutions in the early stage of research). By having this concept, the author modified the dynamic version of PAR (Mahdavi et al. 2007) parameter as follows:

$$\mathrm{PAR(gn)} = \mathrm{PAR}_{\max} - \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{\mathrm{NI}} \times \mathrm{gn} \tag{18}$$

This modification changed the direction of linearly adaptation of PAR from an increasing manner as in Mahdavi et al. (2007) to a decreasing manner. So both PAR and bandwidth (bw) values are updated in the same direction (decreasing manner). This as reported by the author, helps the HS to explore a maximum search space, therefore improving the HS performance.

The integration of HS with concepts from PSO is reported in the previous section (Omran and Mahdavi 2008), where the authors proposed to use the PSO concept, global best particle, by replacing the bw parameter altogether and adding a randomly selected decision variables from the best harmony vector in HM. On the same hand, Geem (2009a) used the same concept of *pbest* in PSO to improve the selection process in harmony memory consideration operator (HMCR), where the new mechanism select the new decision variable from the best harmony vector stored in HM not randomly as in standard HS.

$$D_i^{New} \leftarrow D_i^{Best} \tag{19}$$

where D_i^{Best} is the value of the ith decision variable in the best solution vector (D_1^{Best}, $D_2^{Best}, \ldots, D_n^{Best}$) found.

In the same context, Al-Betar et al. (2010b) investigated the performance of HS algorithm in solving the examination timetabling problem with three selection mechanisms in memory consideration operator. They proposed to evaluate the random selection mechanism as in standard HS, global-best memory consideration as in Geem (2009a) and Roulette-Wheel memory consideration which uses the survival for the fittest principle.

Santos Coelho and de Andrade Bernert (2009) modified HS by integrating a component from Dispersed particle swarm optimization (DPSO) (Cai et al. 2008). Actually, a dynamic PAR is introduced in their work by modifying PAR equation through using a new performance differences index ($grade$) proposed in Cai et al. (2008). Therefore, the PAR equation became as follows:

$$\mathrm{PAR}(t) = \mathrm{PAR}_{\min} + (\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min}) \cdot grade, \tag{20}$$

The grade as in Cai et al. (2008) is modified according to the following equation:

$$grade = \frac{(F_{\max}(t) - mean(F))}{(F_{\max}(t) - F_{\min}(t))}, \tag{21}$$

where $F_{\max}(t)$ and $F_{\min}(t)$ are the maximum and minimum objective function values in generation t, respectively; $mean(F)$ is the mean of objective function value of all vectors in HM.

Wang et al. (2009) improved the performance (convergence speed) of HS by integrating the Clonal Selection Algorithm (CSA) (Dasgupta 2006; Wang et al. 2004) into HS. Actually, they update all harmony memory vectors by calling CSA, where they were considered as individual antibodies and they can evolve in the population of the CSA. This operation is considered as a fine tuning mechanism for HS. Even though this approach moderately increases the computational complexity of the original HS method, but it improved the convergence capability of HS to deal with the prematurity problem.

Lee and Zomaya proposed a parallel metaheuristic framework in which HS is considered as the key component of it Lee and Zomaya (2009). In this framework, they used three metaheuristics, GA, SA, and Artificial Immune System (AIS) (Dasgupta 2006) to enhance the solutions stored in HM as an extra step to speed up the convergence, and at the same time to prevent the HS from getting stuck in the local optimal problem.

```
for each i ∈ [1, N]% N is number of decision variables do
  x_R = 2 · x_i^best − x_i^worst
  if x_R > x_iU then
    x_R = x_iU % x_iU is the upper bounds for decision variables
  else if x_R < x_iL then
    x_R = x_iL % x_iL is the lower bounds for decision variables
  end if
  x_i' = x_i^worst + rand() × (x_R − x_i^worst) %position updating
  if (rand() ≤ P_m) then
    x_i' = x_iL + rand() × (x_iU − x_iL) %genetic mutation
  end if
end for
```

Fig. 4 Improvisation step in NGHS Zou et al. (2010)

Zou et al. (2010) modified HS by inspiring the swarm intelligence of particle swarm to make it as a global optimization algorithm (NGHS) that can solve complex reliability problems. Their proposed algorithm is based on adding new two important operations: position updating and genetic mutation with a small probability. The position update operator mimics the PSO concept of global best particle in swarm, therefore, the worst harmony of HM is moved to the global best harmony rapidly in each iteration. This step may affect the diversity of HS and consequently a premature convergence problem may appear. This encourages the authors to propose the second operator which is a genetic mutation with a small probability to overcome this problem. The NGHS and the HS are different in three aspects as follows:

1. Harmony memory considering rate (HMCR) and pitch adjusting rate (PAR) are excluded from the NGHS, and genetic mutation probability (P_m) is included in the NGHS;
2. The improvisation step of the HS is modified to be as in Fig. 4;
3. The worst HM vector is replaced with the new generated vector even if it was the worst.

Fesanghary et al. (2008) proposed a new framework that combined HS with nonlinear programming methods: Sequential Quadratic Programming (SQP) (Boggs and Tolle 2008) to solve engineering optimization problems. Their algorithm introduced SQP in HS as a new local search component. This step is to support the exploitation mechanism of HS. SQP is introduced (called) a few times to improve the quality of the new improvised vector through the improvisation process of HS. This is controlled with the probability P_c which is experimentally set to be (0.1). Also as a final step and after HS met the stopping criterion, SQP is introduced to the best vector, in term of objective function, stored in HM as a final improvement step.

Alia et al. (2009c) proposed a new dynamic fuzzy clustering algorithm for image segmentation problems, called Dynamic Clustering Harmony Search (DCHS). DCHS is able to automatically determine the appropriate number of clusters, as well as the appropriate locations of cluster centers. In order to do that, a variable length of a harmony memory vector is proposed, where each vector can encode different number of clusters. The same authors in Alia et al. (2010) improved the performance of DCHS by proposing a new operator called *'empty operator'* to support the selection mechanism of number of cluster. Also they hybridized their algorithm with fuzzy c-means algorithm (FCM) to improve the quality of the segmentation results.

Alia et al. (2009a) proposed a new image segmentation algorithm that is based on combining HS with FCM in one framework. A model of HS was proposed, where the decision variables of the harmony vector is the fuzzy memberships of image pixels to a predefined number of clusters rather than centroids of clusters. Their algorithm introduced FCM in HS as a new local search component. This step is to support the exploitation mechanism of HS and speedup the convergence property of HS. FCM is introduced (called) a few times to improve the quality of the new improvised vector through the improvisation process of HS.

In Mahdavi et al. (2008), Forsati et al. (2008), the authors improved the performance of HS for web documents clustering by the integration of k-means algorithm as a local search component. This is done by calling k-means algorithm a few times with the best vector stored in HM as initial cluster centers, then k-means perform the clustering and the returned vector is added to HM if it has a better fitness values than those in HM.

Malaki et al. (2008) developed a hybrid IHS-FCM clustering algorithms, which were tested on a ~58,000 element NASA radiator data set. In their proposed algorithm, they used the modified version of HS that was proposed by Mahdavi et al. (2007) (IHS), further to that; they integrated it with FCM algorithm to improve its performance. FCM actually is integrated in two ways, where in the first way, FCM is integrated into IHS as a local search component to increase the convergence speed same as what has been done in Mahdavi et al. (2008). This way named as (FHSClust). In the second way, FCM is used as a further final clustering step to enhance the partitioning results, where it is initialized by the best solution vector improvised in FHSClust.

Ayvaz et al. (2009) inspired the work of Fesanghary and his colleagues, HS-SQP (Fesanghary et al. 2008), explained earlier to propose the same framework but with spreadsheet 'Solver' instead of SQP to improve the results of the HS algorithm in solving continuous engineering optimization problems.

Jang et al. (2008) are the same like other researchers in this domain that introduced some local search based algorithms into the global search optimization algorithm to improve its performance and increase its convergence speed. In this context, the authors proposed a hybrid framework that combined HS with Nelder-Mead Simplex Algorithm (NM-SA) (Nelder and Mead 1965) as a local search component to improve the quality of stored harmony memory vectors in HM.

Yildiz (2008), Yildiz and Ozturk (2010) proposed a new framework that combine HS with Taguchi method to improve the performance of HS. His proposed method is based on two-stages which are (1) Taguchi's robust design approach (Taguchi 1990) to find appropriate interval levels of design parameters to be used as an initialization step for harmony memory, (2) HS to generate optimal multi-objective solutions using refined intervals from the previous stage. This hybridization step is also introduced to reduce the effects of noise factors in the optimization process.

Gao et al. (2008, 2009) proposed two modification to HS to deal with the uni-modal and multi-modal optimization problems. The first modification is directed to increase the convergence speed of HS through integrating it with DE technique. DE is used to fine tune the vectors stored in HM. Actually, HM vectors become as DE population, then the evolving process is performed as the usual DE procedure. The second modification of HS is proposed to handle the multi-modal problems. A new harmony memory updating strategy is proposed such that any new harmony vector must meet the following criteria to be stored in HM:

1. it is better than the worst harmony in HM, and
2. there are less than a critical number of similar harmonies already in HM, and
3. its fitness is better than the average fitness of the similar harmonies;

A summarization of what has been mentioned in hybridizing of HS with other metaheuristic components is described in the following Table 2.

Table 2 Variants of HS based on hybridizing improvements

Type of hybridization	Description	References
HS + SA	It is used to modify the PAR parameter using the cooling strategy of SA	Taherinejad (2009)
HS + PSO	The PSO concept, global best particle, is incorporated by replacing the bw parameter altogether and adding a randomly selected decision variables from the best harmony vector in HM	Omran and Mahdavi (2008),
HS + PSO	The PSO concept, global best particle, is used to improve the selection process in harmony memory consideration operator (HMCR)	Geem (2009a)
HS + DPSO	DPSO component is introduced to dynamically update the value of PAR parameter	Santos Coelho and de Andrade Bernert (2009)
HS + GA	Roulette-Wheel memory consideration which uses the survival for the fittest principle is used to improve the selection process in HMCR	Al-Betar et al. (2010b)
HS + CSA	The CSA is used to fine tune all HM vectors and improve the convergence capability of HS	Wang et al. (2009)
HS + GA + SA + AIS	It is used to enhance the solutions stored in HM, to speed up the convergence, and to prevent the HS from getting stuck in the local optimal problem	Lee and Zomaya (2009)
HS + PSO + GA	It is used to make HS as a global optimization algorithm by adding two operations: position updating and genetic mutation	Zou et al. (2010)
HS + SQP	SQP is used to support the exploitation mechanism of HS	Fesanghary et al. (2008)
HS + FCM	FCM is integrated in HS to improve its convergence speed and fine tune the clustering quality	Alia et al. (2009a,c, 2010)
HS + K-means	k-means is used as a local search component in HS	Mahdavi et al. (2008), Forsati et al. (2008)
IHS + FCM	FCM is integrated into IHS to improve its local search ability and fine tuning the clustering result as a final step	Malaki et al. (2008)
HS + Solver	Solver is used to support the exploitation mechanism of HS	Ayvaz et al. (2009)
HS + NM-SA	It is used to improve the local search ability of HS	Jang et al. (2008)
HS + Taguchi	It is used to improve the initialization step for harmony memory and to reduce the effects of noise factors	Yildiz (2008), Yildiz and Ozturk (2010)
HS + DE	DE is used to fine tune the HM vectors and for multi-modal problems they proposed a new harmony memory updating strategy	Gao et al. (2008, 2009)

4.2.2 Hybridizing HS as components in other metaheuristic

The second approach of HS hybridization as mentioned earlier is the integration of HS concepts or components into other metaheuristic algorithms to improve their performance.

In Li et al. (2007), the authors improved the performance of PSO which is used in the designing of optimal pin connected structures by handling the particles, which fly outside the variables' boundary. This improvement is based on the use of the HM concept, where one of the main characteristics of HM is storing the feasible vectors, which all are in the feasible space. This concept (HM) is integrated into PSO algorithm to overcome the drawback of using *pbest* concept of PSO, since the selecting technique used in generating *pbest* swarm allows the new generated vector to violate the variables' boundary, therefore, the new generated vector may fly into infeasible regions.

Kaveh and Talatahari (2009) proposed a new framework that is based on a modified version of PSO named as particle swarm optimization with passive congregation (PSOPC) (He et al. 2004), ant colony optimization (Dorigo et al. 2006) and harmony search scheme. This framework used PSOPC as a based framework and as a global search technique, also used the idea of the ant colony approach as a local search for updating the positions of the particles by applied pheromone-guided mechanism. Furthermore,in term of using HM concept, HS is used in this framework to control the variable constraints same as reported in Li et al. (2007). This combination of metaheuristic algorithms is to enhance the performance of PSO in its strategies, exploration and exploitation.

Qinghua et al. (2006) proposed a new hybrid optimization algorithm that combined three metaheuristic algorithms, a modified genetic algorithm, a simplex algorithm and a tabu search. The modified version of genetic algorithm is based on mimicking the musical process of searching for a perfect state of harmony, which increases the robustness of GA.

Another version of improving the performance of GA by using HS is found in Li et al. (2008). Their proposed modification mimics the HS improvisation way, where the new generated vector is selected from all vectors stored in the HM, which is contrary to GA way of generating new vectors (parents).

In order to improve the performance of the evolutionary algorithm, Nadi et al. (2010) proposed a new technique that maintains the right balance between the exploration and exploitation of the evolutionary algorithm in the search process. The proposed approach that named as an adaptive parameter controlling approach is based on controlling the parameter values of the algorithm through the search process using harmony search algorithm. During the search process, harmony search directs the search from the current state to a desired state by determining suitable parameter values such that the balance between exploration and exploitation is suitable for that state transition.

Moeinzadeh et al. (2009) used HS to improve the accuracy of Linear Discriminate Analysis (LDA) (Fisher 1936; Duda et al. 2000) classification algorithm. HS is used as a preprocessing technique to overcome the LDA's problem which is the distribution of each class, where the case that each class has Gaussian distribution and also all classes have the same within-class covariance, while having different means is rarely in real world problems. For that, HS is used to compute a transformation (projection) matrix with the aim of decreasing the within-class covariance and increasing the between-class covariance. HM is initialized by representation of the transformation matrix as a float numerical vector, where each component of this vector $\in [-1, 1]$. Furthermore, the quality of each solution vector is measured by $J_{CI}(W)$ the Class-Independent LDA (CI-LDA) transformation matrix that represents the fitness function in HS.

Finally, the application of HS into multi-objective optimization problem appears to have been tackled in 2006 Geem and Hwangbo (2006) for the design and operation of a heat pipe on a satellite, in 2009 Geem (2009b) for completing a project with minimal time as well as minimal cost is a critical factor for scheduling a project and in Geem (2009e) for the design of water distribution networks which include pumps.

A summarization of what has been mentioned in hybridizing of other metaheuristics with some HS components or concepts is described in the following Table 3.

Table 3 Hybridizing of HS components and concepts in other metaheuristics

Type of hybridization	Description	References
PSO + HS	The (HM) concept in HS is integrated into PSO algorithm to prevent the pbest concept of PSO to violate the variables' boundary	Li et al. (2007)
PSOPC + ACO + HS	HM concept is used to control the variable constraints in PSOPC	Kaveh and Talatahari (2009)
GA + Simplex + TS + HS	The HS concept of searching is used to improve the performance of GA	Qinghua et al. (2006)
GA + HS	The concept of selecting the decision variables from all vectors stored in the HM is mimicked to improve the GA selection mechanism	Li et al. (2008)
GA + HS	HS is used to maintain a balance between the exploration and exploitation concepts in GA	Nadi et al. (2010)
LDA + HS	HS is used as a preprocessing technique to overcome the LDA's problem	Moeinzadeh et al. (2009)

5 Conclusion

As an important tool for optimization domain, metaheuristic harmony search algorithm exploring the search space of the given data in both intensification and diversification parallel optimization environment and provides a near-optimal solution within a reasonable time. It has many features that make it as a preferable technique not only as standalone algorithm but also to be combined with other metaheuristic algorithms.

Even the standard HS has been successfully implemented in various applications, however, many modification and improvements to this algorithm have been also reported in the literature by many research in various domains. Each of them is tightly related to some aspects of this algorithm such as parameters setting, balancing of intensification and diversification of HS and finally hybridizing it with other metaheuristic components.

In this paper, we turn the attention to this algorithm and survey most of the modifications proposed in the literature. Though we have already seen many examples of successful applications of harmony search, there still remain many open problems due to the existence of many inherent uncertain factors. These problems have already attracted and will continue to attract intensive efforts from broad disciplines.

Acknowledgments Many thanks to the anonymous reviewers for their valuable comments that helped to improve this paper. This research is supported by Universiti Sains Malaysia, USM's fellowship scheme and 'Universiti Sains Malaysia Research University Grant' grant titled 'Delineation and visualization of Tumour and Risk Structures—DVTRS' under grant number 1001/*PKOMP*/817001.

References

Alia OM, Mandava R, Ramachandram D, Aziz ME (2009a) A novel image segmentation algorithm based on harmony fuzzy search algorithm. In: International conference of soft computing and pattern recognition, 2009. SOCPAR '09, pp 335–340

Alia OM, Mandava R, Ramachandram D, Aziz ME (2009b) Harmony search-based cluster initialization for fuzzy c-means segmentation of mr images. In: TENCON 2009–2009 IEEE region 10 conference. pp 1–6

Alia OM, Mandava R, Ramachandram D, Aziz ME (2009c) Dynamic fuzzy clustering using harmony search with application to image segmentation. In: IEEE international symposium on signal processing and information technology (ISSPIT09). pp 538–543
Alia OM, Mandava R, Aziz ME (2010) A hybrid harmony search algorithm to mri brain segmentation. In: The 9th IEEE international conference on cognitive informatics, ICCI2010., Tsinghua University, Beijing, China, IEEE, pp 712–719
Al-Betar M, Khader A, Liao I (2010a) A harmony search with multi-pitch adjusting rate for the university course timetabling. In: Geem Z (ed) Recent advances in Harmony search algorithm. Springer-Verlag, Berlin, Heidelberg, pp 147–161
Al-Betar MA, Khader AT, Gani TA (2008) A harmony search algorithm for university course timetabling. In: The proceedings of the 7th international conference on the practice and theory of automated timetabling. Montreal, Canada
Al-Betar MA, Khader AT, Nadi F (2010b) Selection mechanisms in memory consideration for examination timetabling with harmony search. In: Proceedings of the 12th annual conference on genetic and evolutionary computation. ACM, Portland, Oregon, USA, pp 1203–1210
Ayvaz MT (2007) Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm. Adv Water Resour 30(11):2326–2338
Ayvaz MT (2009) Application of harmony search algorithm to the solution of groundwater management models. Adv Water Resour 32(6):916–924
Ayvaz MT, Kayhan AH, Ceylan H, Gurarslan G (2009) Hybridizing the harmony search algorithm with a spreadsheet 'solver' for solving continuous engineering optimization problems. Eng Optim 41(12):1119–1144
Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv 35(3):268–308
Blum C, Roli A (eds) (2008) Hybrid metaheuristics: an introduction. In: Hybrid metaheuristics. Springer, Berlin, pp 1–30
Boggs PT, Tolle JW (2008) Sequential quadratic programming. Acta Numer 4:1–51
Cai X, Cui Z, Zeng J, Tan Y (2008) Dispersed particle swarm optimization. Inf Process Lett 105(6):231–235
Ceylan H, Ceylan H, Haldenbilen S, Baskan O (2008) Transport energy modeling with meta-heuristic harmony search algorithm, an application to turkey. Energy Policy 36(7):2527–2535
Chakraborty P, Roy GG, Das S, Jain D, Abraham A (2009) An improved harmony search algorithm with differential mutation operator. Fundam Inform 95:1–26
Cheng YM, Li L, Lansivaara T, Chi SC, Sun YJ (2008) An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis. Eng Optim 40:95–115
Dasgupta D (2006) Advances in artificial immune systems. IEEE Comput Intell Mag 1(4):40–49
Degertekin S (2008) Optimum design of steel frames using harmony search algorithm. Struct Multidiscip Optim 36(4):393–401
Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
Duda RO, Hart PE, Stork DG (2000) Pattern classification. Wiley, New York
Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y (2008) Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Comput Methods Appl Mech Eng 197(33–40):3080–3091
Fesanghary M, Damangir E, Soleimani I (2009) Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm. Appl Therm Eng 29(5–6):1026–1031
Fisher R (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7:179–188
Forsati R, Mahdavi M, Kangavari M, Safarkhani B (2008) Web page clustering using harmony search optimization. In: Canadian conference on electrical and computer engineering, CCECE. pp 1601–1604
Gao XZ, Wang X, Ovaska SJ (2008) Modified harmony search methods for uni-modal and multi-modal optimization. In: Proceedings of the 2008 8th international conference on hybrid intelligent systems. pp 65–72
Gao XZ, Wang X, Ovaska SJ (2009) Uni-modal and multi-modal optimization using modified harmony search methods. Int J Innov Comput Inf Control
Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. WH freeman, San Francisco
Geem Z (2006) Improved harmony search from ensemble of music players. In: Gabrys B, Howlett RJ, Jain L (eds) Knowledge-based intelligent information and engineering systems. Springer, Heidelberg, pp 86–93
Geem Z (2007a) Optimal scheduling of multiple dam system using harmony search algorithm. In: Computational and ambient intelligence. Springer, Berlin, pp 316–323

Geem Z (2007b) Harmony search algorithm for solving sudoku. In: Apolloni B, Howlett RJ, Jain L (eds) Knowledge-based intelligent information and engineering systems, Lecture Notes in Computer Science, vol 4692. Springer, Berlin/Heidelberg, pp 371–378

Geem Z (ed) (2010) State-of-the-art in the structure of harmony search algorithm. In: Recent advances in Harmony search algorithm. Springer, Berlin, pp 1–10

Geem Z, Choi JY (2007) Music composition using harmony search algorithm. In: Giacobini M (ed) Applications of evolutionary computing. Springer, Berlin, pp 593–600

Geem ZW (2009a) Particle-swarm harmony search for water network design. Eng Optim 41(4):297–311

Geem ZW (2009b) Multiobjective optimization of time-cost trade-off using harmony search. J Constr Eng Manage 136(6):711–716

Geem ZW (2009c) Music-inspired Harmony search algorithm theory and applications. Springer, Berlin

Geem ZW (2009d) Harmony search algorithms for structural design optimization. Springer, Berlin

Geem ZW (2009e) Harmony search optimisation to the pump-included water distribution network design. Civil Eng Environ Syst 26(3):211–221

Geem ZW, Hwangbo H (2006) Application of harmony search to multi-objective optimization for satellite heat pipe design. In: Proceedings of US-Korea conference on science, technology, & entrepreneurship (UKC 2006). Teaneck, NJ, USA, Citeseer, pp 1–3

Geem ZW, Williams JC (2008) Ecological optimization using harmony search. In: Proceedings of the American conference on applied mathematics. World Scientific and Engineering Academy and Society (WSEAS), Cambridge, Massachusetts

Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

Geem ZW, Tseng CL, Park Y (2005a) Harmony search for generalized orienteering problem: best touring in china. In: Wang L, Chen K, Ong Y (eds) Advances in natural computation. Springer, Berlin, pp 741–750

Geem ZW, Lee KS, Park Y (2005b) Application of harmony search to vehicle routing. Am J Appl Sci 2(12):1552–1557

Grosan C, Abraham A (eds) (2007) Hybrid evolutionary algorithms: methodologies, architectures, and reviews. In: Hybrid evolutionary algorithms. Springer-Verlag, Berlin, Heidelberg, pp 1–17

Hasancebi O, Erdal F, Saka MP (2009) An adaptive harmony search method for structural optimization. J Struct Eng 1:72

He S, Wu QH, Wen JY, Saunders JR, Paton RC (2004) A particle swarm optimizer with passive congregation. Biosystems 78(1–3):135–147

Ingram G, Zhang T (2009) Overview of applications and developments in the harmony search algorithm. In: Geem Z (ed) Music-inspired Harmony search algorithm. Springer Berlin, Heidelberg, pp 15–37

Jang WS, Kang HI, Lee BH (2008) Hybrid simplex-harmony search method for optimization problems. In: (IEEE world congress on computational intelligence). IEEE congress on evolutionary computation. CEC 2008, pp 4157–4164

Kaveh A, Talatahari S (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 87(5–6):267–283

Kattan A, Abdullah R, Salam RA (2010) Harmony search based supervised training of artificial neural networks. In: International conference on intelligent systems, modelling and simulation (ISMS). pp 105–110

Lecot C (1989) An algorithm for generating low discrepancy sequences on vector computers. Parallel Comput 11(1):113–116

Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82(9–10):781–798

Lee YC, Zomaya AY (2009) Interweaving heterogeneous metaheuristics using harmony search. In: IEEE international symposium on parallel & distributed processing, 2009. IPDPS 2009, pp 1–8

Li Q, Mitianoudis N, Stathaki T (2007) Spatial kernel k-harmonic means clustering for multi-spectral image segmentation. Image Process IET 1(2):156–167

Li MJ, Ng MK, Cheung YM, Huang JZ (2008) Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. IEEE Trans Knowl Data Eng 20(11):1519–1534

Malaki M, Pourbaghery JA, Abolhassani H (2008) A combinatory approach to fuzzy clustering with harmony search and its applications to space shuttle data. In: SCIS & ISIS 2008. Nagoya, Japan

Mahdavi M, Abolhassani H (2009) Harmony k-means algorithm for document clustering. Data Min Knowl Discov 18(3):370–391

Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188(2):1567–1579

Mahdavi M, Chehreghani MH, Abolhassani H, Forsati R (2008) Novel meta-heuristic algorithms for clustering web documents. Appl Math Comput 201(1–2):441–451

Moeinzadeh H, Asgarian E, Zanjani M, Rezaee A, Seidi M (2009) Combination of harmony search and linear discriminate analysis to improve classification. In: Third Asia international conference on modelling & simulation, 2009. AMS '09, pp 131–135
Mohsen A, Khader A, Ramachandram D (2010) An optimization algorithm based on harmony search for rna secondary structure prediction. In: Geem Z (ed) Recent advances in Harmony search algorithm. Springer, Berlin, pp 163–174
Mukhopadhyay A, Roy A, Das S, Abraham A (2008) Population-variance and explorative power of harmony search: An analysis. In: Second national conference on mathematical techniques emerging paradigms for electronics and IT industries (MATEIT 2008), New Delhi, India
Nadi F, Khader AT, Al-Betar MA (2010) Adaptive genetic algorithm using harmony search. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation. ACM, Portland, Oregon, USA, pp 819–820
Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7(4):308–313
Omran MGH, Mahdavi M (2008) Global-best harmony search. Appl Math Comput 198(2):643–656
Panchal A (2009) Harmony search in therapeutic medical physics. In: Music-inspired Harmony search algorithm. pp 189–203
Qinghua L, Shida Y, Youlin R (2006) A hybrid algorithm for optimizing multi-modal functions. Wuhan Univers J Nat Sci 11(3):551–554
Saka M, Hasancebi O (2009) Adaptive harmony search algorithm for design code optimization of steel structures. In: Geem Z (ed) Harmony search algorithms for structural design optimization. Springer-Verlag, Berlin, Heidelberg, pp 79–120
Santos Coelho LD, de Andrade Bernert DL (2009) An improved harmony search algorithm for synchronization of discrete-time chaotic systems. Chaos Solitons Fractals 41(5):2526–2532
Stutzle TG (1998) Local search algorithms for combinatorial problems: analysis, improvements, and new applications. PhD thesis, Technical University of Darmstadt department of computer science, Darmstadt, Germany
Taguchi G (1990) Introduction to quality engineering. Asian Productivity Organization, Tokyo
Taherinejad N (2009) Highly reliable harmony search algorithm. In: Circuit theory and design, 2009. ECCTD 2009. European conference on. pp 818–822
Tsai CW (2009) On the study of efficient metaheuristics via pattern reductionComputer science and engineering. PhD thesis
Vasebi A, Fesanghary M, Bathaee SMT (2007) Combined heat and power economic dispatch by harmony search algorithm. Int J Electr Power Energy Syst 29(10):713–719
Velho L, Carvalho P, Gomes J, Figueiredo LD (2008) Mathematical optimization in computer graphics and vision. Elsevier, Amsterdam
Wang CM, Huang YF (2010) Selfadaptive harmony search algorithm for optimization. Expert Syst Appl 37(4):2826–2837
Wang X, Gao XZ, Ovaska SJ (2004) Artificial immune optimization methods and applications-a survey. In: Proceedings of the IEEE international conference on systems, man and cybernetics, vol 4. The Hague, The Netherlands, pp 3415–3420
Wang X, Gao XZ, Ovaska SJ (2009) Fusion of clonal selection algorithm and harmony search method in optimisation of fuzzy classification systems. Int J Bioinspired Comput 1(1):80–88
Weise T (2009) Global optimization algorithms—theory and application
Yagiura M, Ibaraki T (2001) On metaheuristic algorithms for combinatorial optimization problems. Syst Comput Jpn 32(3):33–55
Yang Y (2009a) Image segmentation based on fuzzy clustering with neighborhood information. Opt Appl 1
Yang XS (2009b) Harmony search as a metaheuristic algorithm. In: Geem Z (ed) Music-inspired Harmony search algorithm. Springer-Verlag, pp 1–14
Yildiz AR (2008) Hybrid taguchi-harmony search algorithm for solving engineering optimization problems. Int J Ind Eng Theory Appl Pract 15(3):286–293
Yildiz A, Ozturk F (2010) Hybrid taguchi-harmony search approach for shape optimization. In: Geem Z (ed) Recent advances in Harmony search algorithm. Springer, Berlin, pp 89–98
Zou D, Gao L, Wu J, Li S, Li Y (2010) A novel global harmony search algorithm for reliability problems. Comput Ind Eng 58(2):307–316