

Communication

Empirical Convergence Theory of Harmony Search Algorithm for Box-Constrained Discrete Optimization of Convex Function

Jin Hee Yoon ¹  and Zong Woo Geem ^{2,*} 

¹ Department of Mathematics and Statistics, Sejong University, Seoul 05006, Korea; jin9135@sejong.ac.kr

² College of IT Convergence, Gachon University, Seongnam 13120, Korea

* Correspondence: geem@gachon.ac.kr

Abstract: The harmony search (HS) algorithm is an evolutionary computation technique, which was inspired by music improvisation. So far, it has been applied to various scientific and engineering optimization problems including project scheduling, structural design, energy system operation, car lane detection, ecological conservation, model parameter calibration, portfolio management, banking fraud detection, law enforcement, disease spread modeling, cancer detection, astronomical observation, music composition, fine art appreciation, and sudoku puzzle solving. While there are many application-oriented papers, only few papers exist on how HS performs for finding optimal solutions. Thus, this preliminary study proposes a new approach to show how HS converges on an optimal solution under specific conditions. Here, we introduce a distance concept and prove the convergence based on the empirical probability. Moreover, a numerical example is provided to easily explain the theorem.



Citation: Yoon, J.H.; Geem, Z.W. Empirical Convergence Theory of Harmony Search Algorithm for Box-Constrained Discrete Optimization of Convex Function. *Mathematics* **2021**, *9*, 545. <https://doi.org/10.3390/math9050545>

Academic Editor: Alfredo Milani

Received: 19 January 2021

Accepted: 26 February 2021

Published: 4 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, many researchers have utilized various nature-inspired metaheuristic algorithms for solving scientific and engineering optimization problems. One of the popular algorithms is harmony search (HS), which was inspired by jazz improvisation [1]. As a musician plays a musical note from its memory or randomly, HS generates a value from its memory or randomly. As a new harmony, which is composed of musical notes, is evaluated in each practice and memorized if it is good, a new solution vector, which is composed of values, is evaluated in each iteration and memorized if it performs well. This HS optimization process, which utilizes three basic operations (memory consideration, pitch adjustment, and random selection), continues until it finds an optimal solution vector [2].

When compared with other algorithms, there is a similarity between HS and them [1]. HS is similar to Tabu Search with respect to keeping the past vectors in a memory called harmony memory (HM). In addition, HS can use the adaptative parameters of HMCR (Harmony Memory Consideration Rate) and PAR (Pitch Adjustment Rate), which are similar to Simulated Annealing in varying temperature. Furthermore, both HS and genetic algorithm (GA) manage multiple vectors simultaneously. However, there is also a difference between HS and GA. While HS generates a new vector by considering all the existing vectors, GA generates the new vector by considering only two of the existing vectors (the parents). In addition, HS considers each variable in a vector independently while GA cannot consider in that way because its major operation is crossover, which keeps the gene sequence (multiple variables together).

The HS algorithm has been applied to various optimization problems including project scheduling [3], structural design [4], energy system operation [5], car lane detection [6], ecological conservation [7], model parameter calibration [8], portfolio management [9], banking fraud detection [10], law enforcement [11], disease spread modeling [12], cancer

detection [13], astronomical observation [14], music composition [15,16], fine art appreciation [17], and sudoku puzzle solving [18]. Furthermore, there are some application-oriented reviews of HS [19–25].

While there are many applications proposed so far, only a few studies have been dedicated to the theoretical background of the HS algorithm. Beyer [26] dealt with the expected population variance of several evolutionary algorithms, and Das et al. [27] proposed an approximated variance of the expectation of solution candidates and discussed the exploratory power of the HS algorithm. However, there has been no study discussing the convergence of the HS algorithm while the convergence of other optimization algorithms has been discussed in some studies [28–31].

As an evolutionary computation algorithm, HS considers an optimization problem:

$$\min_{x \in X} f(x), \quad (1)$$

where

$$X = \{x = (x_i) \in \mathbb{R}^n \mid l_i < x_i < u_i, i = 1, \dots, n\},$$

$$\text{and } f : \mathbb{R}^n \rightarrow \mathbb{R}.$$

If all the bounds are infinite, the above problem becomes an unconstrained optimization. However, most practical optimization problems adopt variables having certain prefixed range of values, and, therefore, become box-constrained problems [32].

In this paper, we propose the convergence theory based on the empirical probability for the HS, which is one of box-constrained optimization algorithms. In particular, we consider the convex optimization functions with single and multiple variable cases. For this, we define a discrete sequence to prove convergence using distance. This new approach can be applied to the algorithms, such as HS, which have candidate sets that store the improved solutions in the storage and iteratively update them.

2. Harmony Search Algorithm

2.1. Basic Structure of Harmony Search

The HS algorithm is an optimization method inspired by musical phenomena. This algorithm mimics the musical performing process that occurs when a musician searches for a better harmonic sound, such as jazz improvisation. Jazz improvisation finds a musically pleasing harmony (perfect state) determined by aesthetic standards, just as the optimization process finds a global solution (perfect state) determined by an objective function. While the pitch of each instrument determines its aesthetic quality, the value of each decision/design variable determines its solution quality of the objective function.

In order to optimize a problem using HS, we first define all possible candidate value set called the candidate set (universal set) Λ . Let us assume that Λ includes K candidate values. That is, $\Lambda = [\Lambda_1 \cdots \Lambda_n]$ and $\Lambda_i = \{x_i(1), \dots, x_i(K_i)\}$.

Here, the memory storage HM of the HS algorithm can be expressed as in Equation (2). To initialize HM, we randomly choose values from the universal set, and generate vectors as many as HMS (harmony memory size, that is, the number of vectors stored in HM). The value of the objective function is also kept next to each solution vector.

$$HM = \left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_n^1 & f(x^1) \\ x_1^2 & x_2^2 & \cdots & x_n^2 & f(x^2) \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_n^{HMS} & f(x^{HMS}) \end{array} \right] \quad (2)$$

Once HM is prepared, the HM is refreshed with better solution vectors iteration by iteration. If a newly generated vector x_{New} is better than the worst vector x_{Worst} stored in the HM in terms of an objective function value, the new vector is swapped with the worst one.

$$HM_{New} = HM \setminus \{x_{Worst}\} \cup \{x_{New}\}$$

The value of variable x_i ($i = 1, 2, \dots, n$) can be randomly selected from the set of all candidate discrete values $\Lambda_i = \{x_i(1), x_i(2), \dots, x_i(K_i)\}$ with a probability of P_R (random selection rate). Otherwise, the value of x_i can be selected from the set of stored values, named $HM_i = \{x_i^1, x_i^2, \dots, x_i^{HMS}\}$, which is the i^{th} column of HM, with a probability of P_M (the probability with which the value is selected solely from HM), or, once $x_i(k_p^i)$ is selected from HM_i , it can be slightly adjusted by moving it to neighboring values $x_i(k_p^i + m)$ with a probability of P_P (the probability with which the value is selected solely using pitch adjustment) as follows:

$$x_i^{New} \leftarrow \begin{cases} x_i(k_p^i) \in \Lambda_i, & \text{with probability } P_R, \\ x_i^h \in HM_i, & \text{with probability } P_M, \\ x_i(k_p^i + m) \in \Lambda_i, & \text{with probability } P_P, \end{cases} \quad (3)$$

where $i = 1, \dots, n$; $k_p^i \in \{1, \dots, K_i\}$; $h \in \{1, \dots, HMS\}$, $P_R + P_M + P_P = 1$ (100%), $0 \leq P_R, P_M, P_P \leq 1$, and m is some non-zero integer such that $x_i(k_p^i + m) \in \Lambda_i$. Here, m is a predetermined parameter for adjustment. That is, $m = \pm 1$ or ± 2 or $\pm 3 \dots$, etc. For example, if $m = \pm 1$, then we take

$$m = \begin{cases} 1, & \text{with probability } \frac{1}{2} \\ -1, & \text{with probability } \frac{1}{2} \end{cases}$$

For P_M and P_P , we first define HMCR by $HMCR = 1 - P_R$. Then, after we define PAR, P_M and P_P are defined as $P_M = HMCR(1 - PAR)$ and $P_P = HMCR \cdot PAR$.

Here, for the pitch adjustment, we first select x_i^h randomly from HM_i using a uniform distribution between 0 and 1, i.e., $h = \text{Integer}(Rand(0, 1) \times HMS + 0.5)$. Then, we identify k_p^i , which satisfies $x_i^h = x_i(k_p^i)$, where $k_p^i \in \{1, \dots, K_i\}$. Next, we further tweak $x_i(k_p^i)$ into $x_i(k_p^i + m)$, where $x_i(k_p^i + m) \in \Lambda_i$.

This is the basic structure of the HS algorithm [33]. Although there are many structural variants of the HS algorithm [34], most variants basically have the above-mentioned three operations: memory consideration, pitch adjustment, and random selection.

2.2. Solution Formula for Harmony Search without Pitch Adjustment

Let us formulate the solution for the HS algorithm without a pitch adjustment case.

At the first generation, x_i^1 ($i = 1, \dots, n$) will be chosen with given probabilities as follows:

$$x_i^1 = \begin{cases} x_i(k) \text{ from } \Lambda_i \text{ with prob. } P_R, \\ x_i^h \text{ from } HM_i^0 \text{ with prob. } P_M, \end{cases} \quad (4)$$

where $i = 1, \dots, n$; $k = k_p \in \{1, \dots, K_i\}$; $h \in \{1, \dots, HMS\}$, $P_R + P_M = 1$ and HM_i^0 is the initial harmony memory for x_i . And let x_i^g be the solution with given probabilities at the g^{th} generation stage, then

$$x_i^g = \begin{cases} x_i(k) \text{ from } \Lambda_i \text{ with prob. } P_R, \\ x_i^h \text{ from } HM_i^{g-1} \text{ with prob. } P_M, \end{cases} \quad (5)$$

where HM_i^{g-1} is the newly updated i^{th} column of HM after the $(g-1)^{th}$ generation. At the first generation, $x^1 = (x_1^1, x_2^1, \dots, x_n^1)$ is obtained by two operations (random selection or memory consideration). If the newly generated vector x^1 is better than x_{worst}^0 which is the worst vector in HM^0 in terms of an objective function value, then x_{worst}^0 will be replaced with x^1 . Otherwise, the worst vector x_{worst}^0 will stay in the memory.

The element that we get from this comparison after the first generation will be represented by x_{new}^1 . That is,

$$x_{new}^1 \leftarrow \begin{cases} x^1 & \text{if } x^1 \text{ is better than } x_{worst}^0 \\ x_{worst}^0 & \text{otherwise.} \end{cases} \quad (6)$$

Following a similar procedure, at the g^{th} generation, if x^g is better than x_{worst}^{g-1} , which is the worst solution vector in HM^{g-1} , then x_{worst}^{g-1} will be replaced with x^g . Otherwise, the worst element x_{worst}^{g-1} will stay in the memory.

The element that we get from this comparison after the g^{th} generation will be represented by x_{new}^g . That is,

$$x_{new}^g \leftarrow \begin{cases} x^g & \text{if } x^g \text{ is better than } x_{worst}^{g-1} \\ x_{worst}^{g-1} & \text{otherwise,} \end{cases} \quad (7)$$

Using an indicator function, the solution formula for x_{new}^g in Equation (7) can be equivalently represented by:

$$x_{new}^g = x^g \cdot I^g + x_{worst}^{g-1} \cdot (1 - I^g), \quad (8)$$

where

$$I^g = \begin{cases} 1 & \text{if } x^g \text{ is better than } x_{worst}^{g-1} \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

2.3. Solution Formula for Harmony Search with Pitch Adjustment

Next, let us formulate the solution for the HS algorithm with a pitch adjustment case.

At the first generation, $x_i^1 (i = 1, \dots, n)$ will be chosen with given probabilities as follows:

$$x_i^1 = \begin{cases} x_i(k_R^i) \text{ from } \Lambda_i \text{ with } P_R, \\ x_i^h \text{ from } HM_i^0 \text{ with } P_M, \\ x_i(k_P^i + m) \text{ from } \Lambda_i \text{ with } P_P, \end{cases} \quad (10)$$

where $i = 1, \dots, n$; $k_R^i, k_P^i \in \{1, \dots, K\}$; $h \in \{1, \dots, HMS\}$, $P_R + P_M + P_P = 1$ and HM_i^0 is the initial harmony memory. Then, let x_i^g be the solution with given probabilities at the g^{th} generation stage, in which:

$$x_i^g = \begin{cases} x_i(k_R^i) \text{ from } \Lambda_i \text{ with } P_R, \\ x_i^h \text{ from } HM_i^{g-1} \text{ with } P_M, \\ x_i(k_R^i + m) \text{ from } \Lambda_i \text{ with } P_P \end{cases} \quad (11)$$

where HM_i^{g-1} is the newly updated i^{th} column of HM after the $(g-1)^{th}$ generation. At the first generation, $x^1 = (x_1^1, x_2^1, \dots, x_n^1)$ is obtained by three operations (random selection, memory consideration or pitch adjustment). If x^1 is better than x_{worst}^0 which is the worst one in HM^0 in terms of an objective function value, then $x_{new}^1 = x^1$. Otherwise, $x_{new}^1 = x_{worst}^0$. That is,

$$x_{new}^1 \leftarrow \begin{cases} x^1 & \text{if } x^1 \text{ is better than } x_{worst}^0 \\ x_{worst}^0 & \text{otherwise.} \end{cases} \quad (12)$$

Therefore, after the first generation, HM^1 will be updated by substituting x_{worst}^0 with x_{new}^1 . Then, after the g^{th} generation,

$$x_{new}^g \leftarrow \begin{cases} x^g & \text{if } x^g \text{ performs better than } x_{worst}^{g-1} \\ x_{worst}^{g-1} & \text{otherwise,} \end{cases} \quad (13)$$

where x_{worst}^{g-1} is the solution vector that performs the worst in the HM^{g-1} . Therefore, after the g^{th} generation, HM^g will be updated by substituting x_{worst}^0 with x_{new}^1 .

Let $x_{new}^g = \hat{x}^g$. Then, using an indicator function, the solution formula for \hat{x}^g in Equation (13) can be represented by:

$$\hat{x}^g = x^g \cdot I^g + x_{worst}^{g-1} \cdot (1 - I^g), \quad (14)$$

where

$$I^g = \begin{cases} 1 & \text{if } x^g \text{ performs better than } x_{worst}^{g-1}, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

3. Empirical Convergence of Harmony Search

3.1. One Variable Case

In this section, we discuss the behavior of the solutions in HM. Let us first consider $f(x)$ is a function of one discrete variable x . Without loss of generality, we can assume that $\Lambda = \{x(k) \in R^1 | k = 1, \dots, K\}$ satisfies $x(1) < x(2) < \dots < x(k) < \dots < x(K)$ after rearrangement, and we have $K - 1$ subintervals that are separated by $x(1), x(2), \dots, x(K)$ (See Figure 1). In addition, $HM = \{x^h \in R^1 | h = 1, \dots, HMS\}$ satisfies $x^1 \leq x^2 \leq \dots \leq x^h \leq \dots \leq x^{HMS}$ after rearrangement. The h^{th} element of HM at the g^{th} generation will be written as $x^{h,g}$.

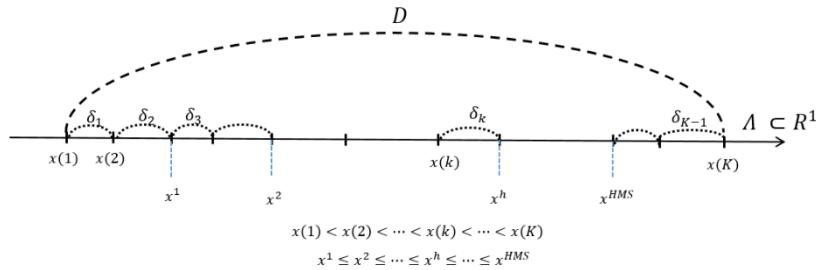


Figure 1. Elements of the candidate set and the subintervals.

Here, let us define the total ranges of the candidate set Λ and HM. In addition, we define the distances between neighboring endpoints, which are the lengths of subintervals as follows:

$$\begin{aligned} D &= |\Lambda| = |x(K) - x(1)| \\ D^g &= |HM^g| = |x^{HMS,g} - x^{1,g}| \end{aligned} \quad (16)$$

$$\delta_k = |x(k+1) - x(k)|, \quad k = 1, \dots, K - 1$$

$$\delta_{h_j}^g = |x^{j+1,g} - x^{j,g}|, \quad j = 1, \dots, HMS - 1 \quad (\text{at } g^{th} \text{ generation}) \quad (17)$$

and we define the minimum and maximum value of δ_k as follows:

$$\delta_m = \min_{1 \leq k \leq K-1} \delta_k \text{ and } \delta_M = \max_{1 \leq k \leq K-1} \delta_k. \quad (18)$$

Here, note that we have an important relation as follows:

$$(K - 1)\delta_m \leq D = \sum_{k=1}^{K-1} \delta_k \leq (K - 1)\delta_M. \quad (19)$$

As we mentioned, we deal with the convex function with discrete variables for its minimum optimal solution. However, note that it can be also applied to the concave function for the maximum optimal solution without loss of generality. If $f(x)$ has the minimum value at x^{h*} , then x^{h*} can be located in four different ways. Figure 2 shows these

four different cases. (a) and (b) show the convex objective functions. In this paper, we focus on the convex cases where x^{h*} is located in the middle of the range of HM, which is also applied to the concave cases. However, this theory can be also applied to the monotone increasing/decreasing objective functions such as (c) and (d) as trivial cases, where x^{h*} is located at the right or left end point.

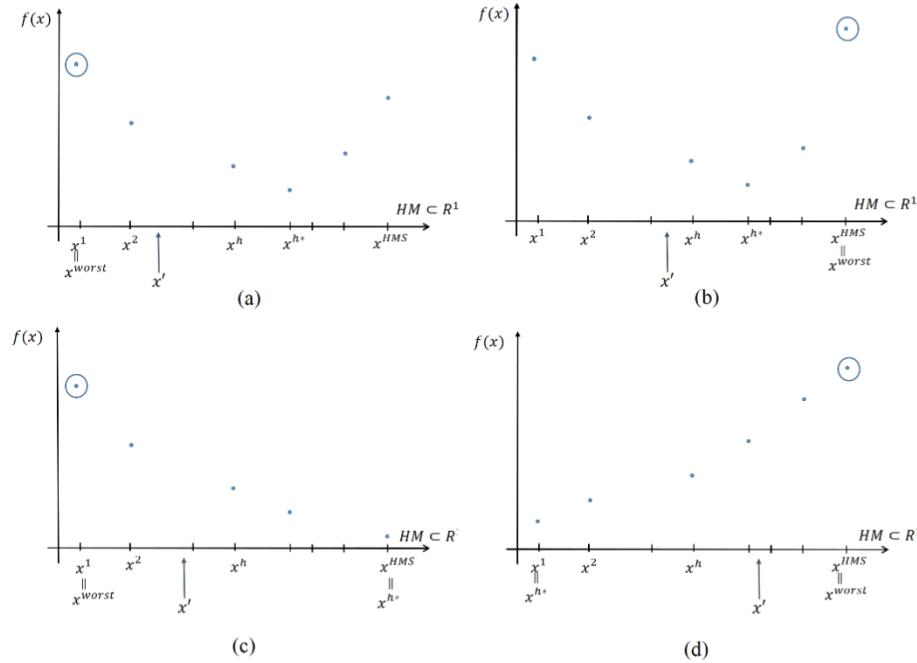


Figure 2. Candidate solutions and the worst element of one-variable functions.

By Equations (5) and (7), the worst element of HM is replaced with a better element as g is increased. Even though not every generation replaces the worst element of HM with a new one, it is true that the worst element of HM is replaced as g is increased. Now, let us consider the case when HM is updated at the g^{th} generation. From Figure 2, we can consider the following properties.

- (a) & (c): Because $x^1 = x^{worst}$, the left-end smallest-value point x^1 will be replaced by some value except in the case of $x' = x^1$. Here, note that $x^1 \leq x' \leq x^{HMS}$. The smallest-value element x^1 can be newly updated by x' or x^2 . Consider that we are at the g^{th} generation. Then, the previous x^1 is denoted as $x^{1,g-1}$, and the newly updated smallest one from HM will be denoted as $x^{1,g}$. Here,

$$x^{1,g} = \begin{cases} x^{2,g-1} & \text{if } x^{2,g-1} < x', \\ x' & \text{if } x^{2,g-1} \geq x'. \end{cases} \quad (20)$$

It is clear that

$$|x^{HMS,g} - x^{1,g}| \leq |x^{HMS,g-1} - x^{1,g-1}|. \quad (21)$$

- (b) & (d): Because $x^{HMS} = x^{worst}$, the right end point x^{HMS} will be replaced by some value except in the case of $x' = x^{HMS}$. x' , which shows better performance than x^{HMS} . Here, note that $x^1 \leq x' \leq x^{HMS}$. The largest-value element x^{HMS} can be updated by x' or x^{HMS-1} . Consider that we are at the g^{th} generation. Then, the previous x^{HMS} is denoted as $x^{HMS,g-1}$, and the newly updated largest one from HM will be denoted as $x^{HMS,g}$. Here,

$$x^{HMS,g} = \begin{cases} x^{(HMS-1),g-1} & \text{if } x^{(HMS-1),g-1} > x', \\ x' & \text{if } x^{(HMS-1),g-1} \leq x'. \end{cases} \quad (22)$$

It is clear that

$$|x^{HMS,g} - x^1,g| \leq |x^{HMS,g-1} - x^1,g-1|. \quad (23)$$

Now we prove the following theorem.

Theorem 1. Assume that there exists exactly one solution in the candidate set Λ for the objective function $f(x)$. If $D^g = |HM^g|$ is the length of HM at the g^{th} generation, then

- (i) $\{D^g\}$ is a monotone decreasing sequence as g is increased.
- (ii) Furthermore, the solutions in HM converge.

Proof. (i) As we have seen from Equations (21) and (23), we satisfy

$$D^g \leq D^{g-1}, \quad (24)$$

which means the sequence $\{D^g\}$ is a monotone decreasing sequence.

Note that not every generation makes HM updated. Therefore, we need to define a new notation for the generation that updates HM. If the worst element of HM is replaced by a new element x' at $g_1^{\text{th}}, g_2^{\text{th}}, \dots$ generation such that

$$D^{g_l} < D^{g_{l-1}}, \quad (25)$$

then $\{g_l | l = 1, 2, 3, \dots\}$ is the set of generation numbers that update HM when each update reduces $|x^{HMS} - x^1|$ in HM. It is clear that

$$\{g_l\} \subset \{g\}, \quad (26)$$

i.e., $\{g_l\}$ a subsequence of $\{g\}$.

Note that, if there are more than one worst solution in HM, then even if HM is updated, it does not satisfy Equation (25). Therefore, the generation number at that time, it is not an element of $\{g_l\}$.

Now, we consider that either $D^{g_l} > 0$ or $D^{g_l} = 0$ is true. Equivalently, if either $D^{g_l} \geq \delta_m$ or $D^{g_l} = 0$ is true, then, in the case of $D^{g_l} > 0$, we have $D^{g_l} \geq \delta_m$

Furthermore, for the smallest length of subinterval δ_m , it is easily seen that:

$$D^{g_l} \leq D^{g_{l-1}} - \delta_m. \quad (27)$$

Here, we need to check the existence of $\{g_l\}$ because we are not sure if HM is steadily updated as g is increased. That is, we are not sure if we can guarantee $x^* (= \text{argmin } f(x))$ is selected at certain generation. In addition, it is finally guaranteed based on the relation between empirical probability and theoretical probability. That is, if we repeat the generation, it is guaranteed that x^* is surely selected at certain generation. Therefore, there exist each generation number $g_l (l = 1, 2, \dots)$, which updates HM.

Meanwhile, we have the following property:

$$D^{g_1} \leq D^{g_1-1} - \delta_m \text{ and } D^{g_2} \leq D^{g_2-1} - \delta_m. \quad (28)$$

HM is updated only at g_l^{th} generation ($l = 1, 2, 3, \dots$). Therefore, $D^{g_1-1} = D^0$ and $D^{g_2-1} = D^{g_1}$ because g_1^{th} is the first generation which updates HM. Thus, $HM^{g_1-1} = HM^{g_1-2} = \dots = HM^0$, where HM^0 is the initial HM. Since g_2^{th} is the second generation that makes HM updated, $HM^{g_2-1} = HM^{g_2-2} = \dots = HM^{g_1}$.

If we repeat this procedure l times, we get:

$$D^{g_l} \leq D^0 - l\delta_m, \quad (29)$$

where $D^0 = |HM^0|$ is the initial range of HM. Here, $\{D^{g_l}\}$ is a subsequence of $\{D^g\}$. Here, $D^{g_l} = D^0 - l\delta_m$, is unreasonable form practical point of view. Therefore, we assume that

$D^{g_l} < D^0 - l\delta_m$. Since $\{D^g\}$ is decreasing via monotone, $\{D^{g_l}\}$ is decreasing via monotone as well. Furthermore, $\{D^g\}$ is bounded below by 0. For $\{D^g\}$, note that we have:

$$D^0 = \dots = D^{g_1-1} > D^{g_1} = D^{g_1+1} = \dots > D^{g_2} = \dots > D^{g_3} \dots \quad (30)$$

Now, let us define ϵ_l as follows:

$$\epsilon_l = D^0 - l\delta_m. \quad (31)$$

Since $\epsilon_l > D^{g_l} \geq 0$, as l increases, ϵ_l can be negative, but we consider only the case of $\epsilon_l > 0$. Therefore, let us have $\epsilon_l > 0$ for $l = 1, 2, 3, \dots, L$. Since ϵ_l ($l = 1, 2, 3, \dots, L$) have discrete values, we see that $\{\epsilon_l\}$ is a positive decreasing sequence (See Figure 3a). As g is increased, g_1, g_2, \dots exist empirically. Therefore, every time HM is updated, the sequence ϵ_l is decreased. That is, as l increases,

$$\epsilon_1 > \epsilon_2 > \dots > \epsilon_l > \dots > \epsilon_L > 0. \quad (32)$$

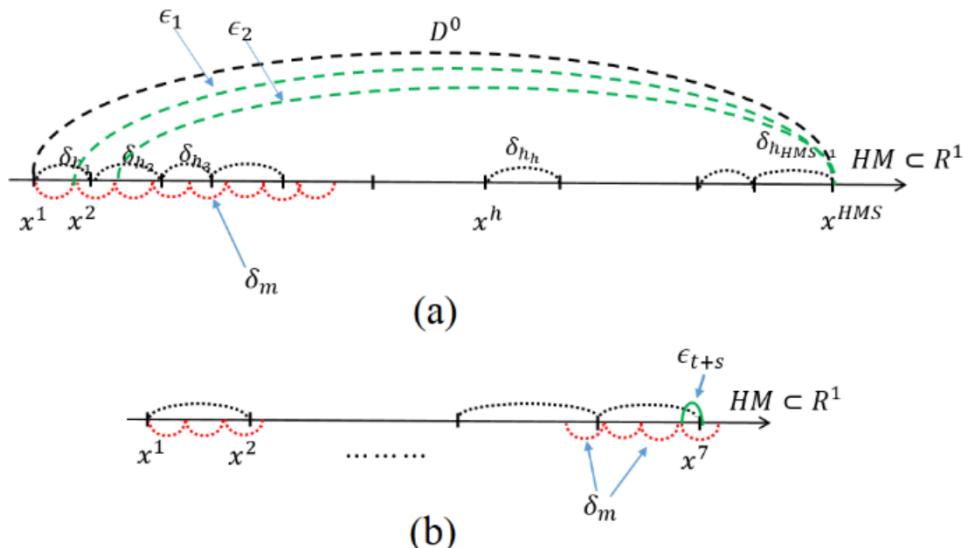


Figure 3. Convergence process when x^{worst} is located at the left end side.

Assume that the solutions in HM converge after the g_t^{th} generation. Then,

$$D^0 > D^{g_1} > D^{g_2} > \dots > D^{g_t} = D^{g_{t+1}} = D^{g_{t+2}} = \dots = D^{g_L} \quad (33)$$

And it is true that:

$$D^{g_t} < D^0 - t\delta_m = \epsilon_t, \text{ but } \epsilon_t \text{ is not very small,} \quad (34)$$

because δ_m is the minimum length of subintervals. Therefore, we continue the procedure even after the solutions in HM converge until ϵ_l is very small but positive. We repeat the generation s more times after the g_t^{th} generation until we satisfy:

$$D^{g_{t+s}} < D^0 - (t + s)\delta_m = \epsilon_{t+s}, \text{ and } \epsilon_{t+s} > 0, \quad (35)$$

$$D^{g_{t+s+1}} < D^0 - (t + s + 1)\delta_m = \epsilon_{t+s+1}, \text{ but } \epsilon_{t+s+1} < 0, \quad (36)$$

for some positive integer s . However, Equation (36) is not possible because we assume that $\epsilon_l > 0$, so we stop the procedure after s more generations. Therefore, $\epsilon_{t+s} = \epsilon_L$. Here, ϵ_L is still a positive constant as in Figure 3b. But,

$$\epsilon_L < \delta_m \leq \delta_{HMS-1}. \quad (37)$$

Since we have either $D^{g_l} \geq \delta_m$ or $D^{g_l} = 0$, Equations (35)–(37) guarantee that $D^{g_{t+s}} = D^{g_L}$ is now 0. It means that, at the g_L^{th} generation, it is guaranteed that:

$$x^{1,g_L} = x^{2,g_L} = \dots = x^{HMS,g_L}. \quad (38)$$

In fact, Equation (38) has been already satisfied from the g_t^{th} generation, but now the convergence is finally guaranteed by the inequality (34) mathematically. That is, the values in HM are convergent before the g_L^{th} generation. Therefore, we can conclude that there exists some $t < L$ for some positive integers t and L such that HM^{g_t} is convergent.

Therefore, the values in HM converge empirically, which proves the theorem. \square

Remark. In the general probability theory, for each event E of the sample space S , we define $n(E)$ to be the number of times in the first n repetitions of the experiment that the event E occurs. Then $P(E)$, which is the probability of the event E , is defined as:

$$P(E) = P_E = \lim_{n \rightarrow \infty} \frac{n(E)}{n}. \quad (39)$$

Here, $P(E)$ is defined as the (limiting) proportion of time that E occurs. It is, thus, the limiting frequency of E . $P(E) = P_E$ is called the “theoretical probability” and $\frac{n(E)}{n}$ is called the “empirical probability.” The empirical probability approaches the theoretical probability as the number of experiments is increased. It means when the repetition number n is small, the event E may not occur. However, if we repeat the experiment many times, i.e., if n is large enough, the ratio between $n(E)$ and n converges P_E . It guarantees that the E should occur (even $n(E)$ times) if we repeat the experiment many times. Therefore, even when HM does not include the solution, that minimizes $f(x)$, if Λ includes the unique solution, say x^* . Then it is absolutely selected as we repeat the iteration. That is:

$$P[x^* \text{ is selected}] = P_E \cdot \lim_{g \rightarrow \infty} \frac{n(E)}{g} = P_E \cdot \frac{1}{K} \quad (40)$$

where the event E is the event of selection of x^* . In addition, Equation (25) guarantees that x^* should be selected as $g \rightarrow \infty$.

Example 1. Let us consider the situation when the minimum value of the objective function $f(x)$ is at the right end side, as in Figure 2c. Define:

$$\Lambda = \left\{ \frac{k}{10} \mid k = 0, 1, 2, \dots, 120 \right\} \text{ and } HM^0 = \{0, 2, 3, 3.7, 6, 9, 11\}. \quad (41)$$

Then, as g increases, we get g_1, g_2, \dots . As we see from Figure 4, we have $D^0 = 11$, and

$$\delta_{h_1} = 2, \delta_{h_2} = 1, \delta_{h_3} = 0.7, \delta_{h_4} = 2.3, \delta_{h_5} = 3, \delta_{h_6} = 2, \text{ and } \delta_m = 0.7. \quad (42)$$

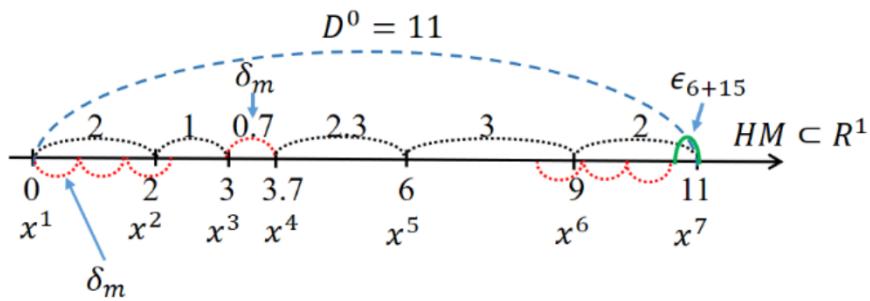


Figure 4. Convergence process in Example 1.

Furthermore, we get:

$$D^0 = 11, D^{g_1} = 9, D^{g_2} = 8, D^{g_3} = 7.3, D^{g_4} = 5, D^{g_5} = 2, D^{g_6} = 0. \quad (43)$$

Therefore, $t = 6$ from Equation (43). However, from $\delta_m = 0.7$,

$$\epsilon_0 = 11, \epsilon_1 = 10.3, \epsilon_2 = 9.6, \epsilon_3 = 8.9, \epsilon_4 = 8.2, \epsilon_5 = 7.5, \epsilon_6 = 6.8. \quad (44)$$

we get $\epsilon_6 = 6.8 > 0$ and it is not a very small number yet. Therefore, we repeat the procedure s more times until we get Equations (35) and (36). It can be easily calculated that $s = 15$. Hence,

$$D_{6+15} < \epsilon_{6+15} = 0.5 < \delta_m = 0.7 \quad (45)$$

We have either $D^{g_l} \geq \delta_m$ or $D^{g_l} = 0$ by the definition of D^{g_l} . Therefore, Equation (45) guarantees that $D^{g_{21}} = 0$. Figure 4 shows the convergence of HM in Example 1. In fact, we already have $D^{g_6} = 0$, but it is not proven mathematically yet. Therefore, it is finally shown after 15 more generations as follows.

$$D^{g_6} = D^{g_7} = \dots = D^{g_{21}} = 0 \quad (46)$$

Therefore, there exists a positive integer $t = 6$ that is less than $L = 21$ such that the solution in HM^{g_6} is convergent. Example 1 shows Theorem 1 is confirmed.

3.2. Multiple Variable Case

Let $f(x)$ be a function of n variables. Let $x = (x_1(k), \dots, x_n(k)) \in R^n$ be a candidate solution from the candidate set Λ . Let us consider the norm $\|\cdot\|$ of each solution vector as follows:

$$\|x(k)\| = \left[\sum_{i=1}^n x_i^2(k) \right]^{1/2}. \quad (47)$$

Without loss of generality, we can assume that $\Lambda = \{x(k) \in R^n | k = 1, \dots, K\}$ satisfies $\|x(1)\| < \|x(2)\| < \dots < \|x(k)\| < \dots < \|x(K)\|$ after rearrangement, and we have $K - 1$ distances between K vectors. The distance between two vectors $x(k_1)$ and $x(k_2)$ is defined by:

$$d(x(k_1), x(k_2)) = \|x(k_1) - x(k_2)\| = \left[\sum_{i=1}^n (x_i(k_1) - x_i(k_2))^2 \right]^{1/2}. \quad (48)$$

In addition, $HM = \{x^h \in R^n | h = 1, \dots, HMS\}$ satisfies $\|x^1\| \leq \|x^2\| \leq \dots \leq \|x^h\| \leq \dots \leq \|x^{HMS}\|$ after rearrangement. The h^{th} vector of HM at the g^{th} generation will be written as $x^{h,g}$.

Here, let us define the total ranges of the candidate set Λ and HM. Similarly, without loss of generality, we focus on the minimum case for this optimal solution problem in this section because we can easily apply this theory to the maximum case. In addition, we

define the distances between neighboring solution candidate vectors, which are the lengths of subintervals as follows.

$$D = |\Lambda| = \|x(K) - x(1)\|$$

$$D^g = |HM^g| = \left\| x^{HMS,g} - x^{1,g} \right\| \quad (49)$$

$$\delta_k = \|x(k+1) - x(k)\|, \quad k = 1, \dots, K-1$$

$$\delta_{h_j} = \left\| x^{j+1} - x^j \right\|, \quad j = 1, \dots, HMS-1 \quad (50)$$

Then we define the minimum and maximum value of δ_k as follows:

$$\delta_m = \min_{1 \leq k \leq K-1} \delta_k \text{ and } \delta_M = \max_{1 \leq k \leq K-1} \delta_k. \quad (51)$$

Here, note that we have an important relation as follows:

$$(K-1)\delta_m \leq D = \sum_{k=1}^{K-1} \delta_k \leq (K-1)\delta_M$$

$$HM = \left\{ x^h \in R^n \mid \left\| x^1 \right\| \leq \dots \leq \left\| x^h \right\| \leq \dots \left\| x^{HMS} \right\| \right\}$$

- (i) If $x^1 = x^{worst}$, then x^1 will be replaced by some x^t , which shows better performance than x^1 . Here, note that $\|x^1\| < \|x^t\| < \|x^{HMS}\|$. Therefore, x^1 , which has the smallest norm, will be newly replaced by x^t . Consider that we are at the g^{th} generation, then the previous x^1 is denoted as $x^{1,g-1}$. At that point, the newly updated smallest vector from HM will be denoted as $x^{1,g}$. Here,

$$x^{1,g} = \begin{cases} x^{2,g-1} & \text{if } \|x^{2,g-1}\| < \|x^t\|, \\ x^t & \text{o.w.} \end{cases} \quad (52)$$

It is clear that

$$\left\| x^{HMS,g} - x^{1,g} \right\| \leq \left\| x^{HMS,g-1} - x^{1,g-1} \right\|. \quad (53)$$

- (ii) If $x^{HMS} = x^{worst}$, then x^{HMS} will be replaced by another vector x^t , which shows better performance than x^{HMS} . Here, note that $\|x^1\| < \|x^t\| < \|x^{HMS}\|$. Therefore, x^{HMS} , which has the largest norm, will be newly replaced by x^t . Consider that we are at the g^{th} generation. Then the previous x^{HMS} is denoted as $x^{HMS,g-1}$. Then, the newly updated largest vector from HM will be denoted as $x^{HMS,g}$. Here,

$$x^{HMS,g} = \begin{cases} x^{(HMS-1),g-1} & \text{if } \|x^{(HMS-1),g-1}\| < \|x^t\|, \\ x^t & \text{o.w.} \end{cases} \quad (54)$$

It is clear that:

$$\left\| x^{HMS,g} - x^{1,g} \right\| \leq \left\| x^{HMS,g-1} - x^{1,g-1} \right\|. \quad (55)$$

Corollary 2. Assume that there exists exactly one solution vector in the candidate set Λ for the objective function $f(x) = f(x_1, x_2, \dots, x_n)$. In addition, if $D^g = |HM^g| = \|x^{HMS,g} - x^{1,g}\|$ is the length of HM at the g^{th} generation, then

- (i) $\{D^g\}$ is a monotone decreasing sequence as g is increased.
- (ii) Furthermore, the values in HM converge.

Proof. Based on the notation that are defined in Equations (47)–(51), the proof is clearly done by Theorem 1. \square

4. Conclusions

In this communication, we employed the distance concept and proved the convergence of the HS algorithm based on the empirical probability. The solution behavior of HS for one or more discrete variables was discussed and the given theorem was demonstrated with a numerical example.

For the future study, we will expand the theorem to include non-discrete variables, multi-modal functions, and adaptive parameters [35].

Author Contributions: J.H.Y. developed the conceptualization, proving the theorems, and drafted the manuscript. Moreover, supervising, reviewing, and editing were done by Z.W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C1A01011131).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
2. Lee, K.S.; Geem, Z.W. A new metaheuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **2004**, *194*, 3902–3933. [[CrossRef](#)]
3. Geem, Z.W. Multiobjective Optimization of Time-Cost Trade-Off Using Harmony Search. *J. Constr. Eng. Manag. ASCE* **2010**, *136*, 711–716. [[CrossRef](#)]
4. Geem, Z.W. *Harmony Search Algorithms for Structural Design Optimization*; Springer: Berlin, Germany, 2009.
5. Nazari-Heris, M.; Mohammadi-Ivatloo, B.; Asadi, S.; Kim, J.-H.; Geem, Z.W. Harmony search algorithm for energy system applications: An updated review and analysis. *J. Exp. Theor. Artif. Intell.* **2019**, *31*, 723–749. [[CrossRef](#)]
6. Moon, Y.Y.; Geem, Z.W.; Han, G.-T. Vanishing point detection for self-driving car using harmony search algorithm. *Swarm Evol. Comput.* **2018**, *41*, 111–119. [[CrossRef](#)]
7. Geem, Z.W. Can Music Supplant Math in Environmental Planning? *Leonardo* **2015**, *48*, 147–150. [[CrossRef](#)]
8. Lee, W.-Y.; Ko, K.-E.; Geem, Z.W.; Sim, K.-B. Method that determining the Hyperparameter of CNN using HS Algorithm. *J. Korean Inst. Intell. Syst.* **2017**, *27*, 22–28. [[CrossRef](#)]
9. Tuo, S.H. A Modified Harmony Search Algorithm for Portfolio Optimization Problems. *Econ. Comput. Econ. Cybern. Stud. Res.* **2016**, *50*, 311–326.
10. Daliri, S. Using Harmony Search Algorithm in Neural Networks to Improve Fraud Detection in Banking System. *Comput. Intell. Neurosci.* **2020**, *2020*, 6503459. [[CrossRef](#)] [[PubMed](#)]
11. Shih, P.-C.; Chiu, C.-Y.; Chou, C.-H. Using Dynamic Adjusting NGHS-ANN for Predicting the Recidivism Rate of Commuted Prisoners. *Mathematics* **2019**, *7*, 1187. [[CrossRef](#)]
12. Fairchild, G.; Hickmann, K.S.; Mniszewski, S.M.; Del Valle, S.Y.; Hyman, J.M. Optimizing human activity patterns using global sensitivity analysis. *Comput. Math. Organ Theory* **2014**, *20*, 394–416. [[CrossRef](#)] [[PubMed](#)]
13. Elyasigomari, V.; Lee, D.A.; Screen, H.R.C.; Shaheed, M.H. Development of a two-stage gene selection method that incorporates a novel hybrid approach using the cuckoo optimization algorithm and harmony search for cancer classification. *J. Biomed. Inform.* **2017**, *67*, 11–20. [[CrossRef](#)] [[PubMed](#)]
14. Deeg, H.J.; Moutou, C.; Erikson, A.; Csizmadia, S.; Tingley, B.; Barge, P.; Brunt, H.; Havel, M.; Aigrain, S.; Almenara, J.M.; et al. A transiting giant planet with a temperature between 250 K and 430 K. *Nature* **2010**, *464*, 384–387. [[CrossRef](#)] [[PubMed](#)]
15. Geem, Z.W.; Choi, J.Y. Music Composition Using Harmony Search Algorithm. *Lect. Notes Comput. Sci.* **2007**, *4448*, 593–600.
16. Navarro, M.; Corchado, J.M.; Demazeau, Y. MUSIC-MAS: Modeling a harmonic composition system with virtual organizations to assist novice composers. *Expert Syst. Appl.* **2016**, *57*, 345–355. [[CrossRef](#)]
17. Koenderink, J.; van Doorn, A.; Wagemans, J. Picasso in the mind’s eye of the beholder: Three-dimensional filling-in of ambiguous line drawings. *Cognition* **2012**, *125*, 394–412. [[CrossRef](#)] [[PubMed](#)]
18. Geem, Z.W. Harmony Search Algorithm for Solving Sudoku. *Lect. Notes Comput. Sci.* **2007**, *4692*, 371–378.
19. Geem, Z.W. *Music-Inspired Harmony Search Algorithm: Theory and Applications*; Springer: New York, NY, USA, 2009.
20. Manjarres, D.; Landa-Torres, I.; Gil-Lopez, S.; Del Ser, J.; Bilbao, M.N.; Salcedo-Sanz, S.; Geem, Z.W. A Survey on Applications of the Harmony Search Algorithm. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1818–1831. [[CrossRef](#)]

21. Askarzadeh, A. Solving electrical power system problems by harmony search: A review. *Artif. Intell. Rev.* **2017**, *47*, 217–251. [[CrossRef](#)]
22. Yi, J.; Lu, C.; Li, G. A literature review on latest developments of Harmony Search and its applications to intelligent manufacturing. *Math. Biosci. Eng.* **2019**, *16*, 2086–2117. [[CrossRef](#)]
23. Ala'a, A.; Alsewari, A.A.; Alamri, H.S.; Zamli, K.Z. Comprehensive Review of the Development of the Harmony Search Algorithm and Its Applications. *IEEE Access* **2019**, *7*, 14233–14245.
24. Alia, M.; Mandava, R. The variants of the harmony search algorithm: An overview. *Artif. Intell. Rev.* **2011**, *36*, 49–68. [[CrossRef](#)]
25. Gao, X.Z.; Govindasamy, V.; Xu, H.; Wang, X.; Zenger, K. Harmony Search Method: Theory and Applications. *Comput. Intell. Neurosci.* **2015**, *2015*, 258491. [[CrossRef](#)] [[PubMed](#)]
26. Beyer, H.-G. On the dynamics of EAs without selection. In *Foundations of Genetic Algorithms*; Banzhaf, W., Reeves, C., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1999; Volume 5, pp. 5–26.
27. Das, S.; Mukhopadhyay, A.; Roy, A.; Abraham, A. Exploratory Power of the Harmony Search Algorithm: Analysis and Improvements for Global Numerical Optimization. *IEEE Trans. Sys. Man Cybern. Part B Cybern.* **2001**, *41*, 89–106. [[CrossRef](#)]
28. Wu, C.F.J. On the convergence properties of the EM algorithm. *Ann. Stat.* **1983**, *11*, 95–103. [[CrossRef](#)]
29. Bull, A.D. Convergence Rates of Efficient Global Optimization Algorithms. *J. Mach. Learn. Res.* **2011**, *12*, 2879–2904.
30. Trelea, I.C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [[CrossRef](#)]
31. Zhang, X.; Zheng, X.; Cheng, R.; Qiu, J.; Jin, Y. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf. Sci.* **2018**, *427*, 63–76. [[CrossRef](#)]
32. Facchinei, F.; Júdice, J.; Soares, J. Generating Box-Constrained Optimization Problems. *ACM Trans. Math. Softw.* **1997**, *23*, 443–447. [[CrossRef](#)]
33. Geem, Z.W. Novel derivative of harmony search algorithm for discrete design variables. *Appl. Math. Comp.* **2008**, *199*, 223–230. [[CrossRef](#)]
34. Zhang, T.; Geem, Z.W. Review of Harmony Search with Respect to Algorithm Structure. *Swarm Evol. Comput.* **2019**, *48*, 31–43. [[CrossRef](#)]
35. Almeida, F.; Giménez, D.; López-Espín, J.J.; Pérez-Pérez, M. Parameterized Schemes of Metaheuristics: Basic Ideas and Applications with Genetic Algorithms, Scatter Search, and GRASP. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 570–586. [[CrossRef](#)]