

## HS 최적화 알고리즘 성능 향상에 관한 연구

# A Study on the Performance Improvement of Harmony Search Optimization Algorithm

이 태 봉\*

가천대학교 전자공학과

Tae-Bong Lee\*

Department of Electronic Engineering, Gachon University, Seoul, 13120, Korea

### [요 약]

Harmony Search(HS) 알고리즘은 음악 즉흥 연주 프로세스에서 영감을 받은 메타 휴리스틱 최적화 알고리즘으로 다양한 최적화 문제를 해결하는 데 성공적으로 적용되어 왔다. 본 논문에서는 HS의 성능을 더욱 향상시키기 위해 FSH(Fast Harmony Search) 알고리즘을 제안하였다. 이를 위해 본 논문에서는 HM을 이용하여 목적 변수의 경계값을 새롭게 정의하여 독립적인 두 개의 화음 개선과정을 하나로 통합하는 방법을 제안하였다. 그 결과 알고리즘의 처리 시간이 단축되고 대역폭의 명시적인 결정이 더 이상 필요하지 않게 되었다. 또한, 무작위 선택의 활용능력이 향상되었다. 수치적 예시 결과는 제안된 알고리즘이 기존의 HS에 비해 더 나은 해를 찾을 수 있으며 속도 또한 빠르다는 것을 보여준다.

### [Abstract]

Harmony Search(HS) algorithm is an emerging meta-heuristic optimization algorithm, which is inspired by the music improvisation process and has been successfully applied to solve different optimization problems. In order to further improve the performance of HS, this paper proposes a new method which is called Fast Harmony Search(FSH) algorithm. For the purpose, this paper suggest a method to unify two independent improvisation processes by newly defining the boundary value of a object variable using HM. As the result, the process time of the algorithm is shorten and explicit decision of bandwidth is no more needed. Furthermore, exploitative power of random selection is improved. The numerical results reveal that the proposed algorithm can find better solutions and is faster when compared to the conventional HS.

**Key word** : FHS, HS, Meta-heuristic, Optimization.

<https://doi.org/10.12673/jant.2021.25.5.403>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 9 August 2021; Revised 24 September 2021

Accepted (Publication) 23 October 2021 (30 October 2021)

\*Corresponding Author ; Tae-Bong Lee

Tel: +82-31-750-5816

E-mail: [tblee@gachon.ac.kr](mailto:tblee@gachon.ac.kr)

## I. 서론

최근 자연을 모방한 새로운 메타 휴리스틱 최적화 기법에 관한 연구에 대한 강한 요구가 있다. 자연현상을 모방한 알고리즘은 기존 고전적 수학적 알고리즘이 갖는 복잡함과 초깃값에 대한 민감성 등과 같은 단점을 극복하고자 1970년대 이후 많이 고안되어 연구되었다[1-5]. 그 중 대표적인 것이 유전자 알고리즘(GA: genetic algorithm), 개미군집최적화(ACO: ant colony optimization), 그리고 입자군집최적화(PSO: particle swarm optimization), 등이 있다[6-8]. 그 중 비교적 최근에 개발된 HS 알고리즘은 음악인이 즉흥적 연주 시 연주자들이 화음을 개선해 나가는 방식을 모방한 것이다[9]. 고전적 기법과 달리 HS와 같은 진화적인 기법(evolutionary algorithm)은 비수학적 알고리즘으로 목적변수에 대한 초깃값을 요구하지 않으며 경사법(gradient search) 대신 확률변수 기법(stochastic random search)을 기반으로 하고 있다.

진화적 기법의 전역 최적 해를 찾는 성능은 기존 정보를 활용하는 활용능력(exploitative power)과 새로운 영역을 탐색하는 탐색능력(exploration power)에 달려 있다. HS는 이들 둘 간의 균형이 좋고 비교적 구현이 쉬워 다양한 최적화 문제 해결에 성공적으로 적용되어 왔다[10]. 그러나 많은 장점에도 불구하고 알고리즘 매개변수를 고정된 값으로 설정한다거나 매개변수 값을 잘못 설정한 경우 극히 값으로 수렴하게 되는 단점이 있다. 이러한 단점을 극복하고 알고리즘 성능을 향상시키기 위한 많은 노력이 있었다. HS를 개량한 알고리즘으로는 Improved Harmony Search(IHS), Global-best Harmony Search(GHS), 및 Self-adaptive harmony search 등이 있다[11-13]. 또한 이태봉[14]은 HS의 핵심 알고리즘 매개변수 HMCR 값과 새로운 하모니 조성에 대한 확률적 분석을 통해 HS의 활용능력과 탐색능력을 정량적으로 결정하는 방안을 제시하였다.

HS에서 새로운 예비해를 구성할 때 HM을 이용하는 지역탐색과 무작위 선택을 하는 전역탐색은 수식적으로 독립적인 관계에 있다. 본 연구에서는 두 가지 탐색이 하나의 수식을 바탕으로 매개변수 값에 따라 이루어질 수 있도록 하였다. 두 과정이 수식적으로 통합되면 첫째는 알고리즘의 수행 속도가 빨라지며 둘째 HS 알고리즘에서 매우 중요하고 결정하기 어려운 대역폭을 명시적으로 결정하지 않아도 되며 셋째 전역탐색을 HM 값을 기반으로 함으로 전역탐색의 활용능력이 향상된다.

제시된 알고리즘의 효능과 성능을 확인하기 위해 저차원 함수와 고차원 함수를 대상으로 HS와 비교하여 알고리즘의 개선 효과를 확인하였다.

## II. HS 개요

HS는 Geem 등(2001)[9]이 음악 연주가들이 즉흥 연주를 할 때 좋은 하모니를 이루는 과정을 모방하여 제한한 발견적 탐색

방법이다. 이 과정은 구체적으로 다음과 같은 세 가지 방법을 통해 이루어진다고 할 수 있다.

- ① 기억 속 음을 연주.
- ② 기억 속 음을 기준으로 그에 이웃한 음을 연주.
- ③ 기억에 의존하지 않고 악기의 음역에서 임의로 선택하여 연주.

HS는 위 세 가지 방식을 각각 ‘기억회상’, ‘피치조정’ 및 ‘무작위 선택’으로 구분하여 목적변수 값을 확률적으로 선택해 가며 해벡터(solution vector)를 생성해 목적함수에 적용한다. 이 과정에서 더 좋은 해(화음)가 Harmony Memory(HM)에 저장된 안 좋은 해를 대체한다. HS의 매개변수는 Harmony Memory Size(HMS), Harmony Memory Considering Rate(HMCR), Pitch Adjusting Rate(PAR), 그리고 대역폭 bw가 있다. 알고리즘 구현은 단계별로 다음과 같이 이루어진다.

- Step 1. 최적화 문제와 알고리즘 매개변수 초기화.
- Step 2. HM 초기화.
- Step 3. 새로운 하모니 즉 해 벡터를 생성.
- Step 4. HM 최신화.
- Step 5. 중단 요건 검사.
- Step 6. 중단 또는 Step 3-4 반복

### 2-1 문제 및 알고리즘 초기화

제한 조건이 없는 최적화 문제는 일반적으로 다음과 같이 나타낼 수 있다.

$$\text{Min. (or Max.) } f(x), \quad x = [x_1 \ x_2 \ \dots \ x_n], \quad x_i \in B_i, \quad (i=1, \dots, n) \quad (1)$$

(1)식에서  $f(x)$ 는 목적함수이고  $x_i$ 는 목적변수이며 연속이거나 이산적이며  $n$ 은 목적변수의 개수이다.  $B_i$ 는 각 목적변수 범위를 나타내는 집합으로 다음과 같이 정의된다.

$$B_i = \{x_i | L_i \leq x_i \leq U_i\}, \quad (i=1, \dots, n) \quad (2)$$

알고리즘 초기화에는 해 벡터를 저장하기 위한 HM의 크기 HMS와 HM에서 목적변수를 선택할 확률값 HMCR 피치조정 여부를 결정할 확률값 PAR 그리고 피치조정을 할 때 적용되는 대역폭 bw 및 알고리즘 반복시수 또는 종료 조건이 필요하다.

### 2-2 HM 초기화

HM은 알고리즘 수행 과정에서 최신의 좋은 화음 즉, 현재까지 가장 좋은 해를 저장하는 장소로 그 크기 HMS를 정의한 후 균일분포 확률변수에 의해 결정되는 값을 이용하여 다음과 같

이 초기화한다.

$$x_i^j = L_i + \text{rand}(0,1) \cdot (U_i - L_i), \quad i = 1, \dots, n \quad j = 1, 2, \dots, \text{HMS} \quad (3)$$

rand(a,b)은 구간 (a,b)에서 정의되는 균일분포 확률변수에 의해 결정되는 무작위 값이다. HM에 저장된 초기 해 벡터는 다음과 같이 나타낼 수 있다.

$$\text{HM} = \begin{bmatrix} x^1 \\ \vdots \\ x^{\text{HMS}} \end{bmatrix}, \quad x^j = [x_1^j \quad x_2^j \quad \dots \quad x_n^j], \quad j = 1, \dots, \text{HMS} \quad (4)$$

**2-3 새로운 하모니 생성**

새로운 하모니, 즉 예비해  $x^N = \{x_1^N, x_2^N, \dots, x_n^N\}$ 는 각 원소  $x_i^N$ 가 ‘기억회상’, ‘피치조정’ 및 ‘무작위 선택’ 세 가지 방식 중 하나가 확률적으로 선택돼 결정된다. 기억회상과 피치조정은  $x_i^N$ 을 HM에 저장된 기존 값 중 하나를 기반으로 결정하는 HM 참조방식이고 무작위 선택은 기존 값이 아닌 변수 범위 내에서 임의의 값을 선택하는 방식이다. HM의 참조 여부를 결정하는 확률은 HMCR 값에 의해 다음 식과 같이 결정된다.

$$x_i^N \leftarrow \begin{cases} x_i^N \in \{x_1^1, x_2^1, \dots, x_i^{\text{HMS}}\}, & \text{w.p. HMCR} \\ x_i^N = L_i + \text{rand}(0,1) \cdot (U_i - L_i) & \text{w.p. (1-HMCR)} \end{cases} \quad (5)$$

(5)식에서 HM의 값이 선택된 경우 그 값을 그대로 사용할 것인지 아니면 피치조정을 할 것인지 다시 한번 확률적 결정을 하게 된다. 이때 사용되는 알고리즘 확률 매개변수가 확률값 PAR이며 피치조정 여부는 다음 식과 같이 결정된다.

$$x_i^N \leftarrow \begin{cases} x_i^N + \text{rand}(-1,1) \cdot bw_i, & \text{w.p. HMCR} \cdot \text{PAR} \\ x_i^N, & \text{w.p. HMCR} \cdot (1-\text{PAR}) \end{cases} \quad (6)$$

(6)식에서 bw<sub>i</sub>는 변수 x<sub>i</sub>의 피치 조정 대역폭이다. 결과적으로 새로운 하모니 조성을 위해 각 변수를 선택하는 방식은 ‘기억회상’, ‘피치조정’ 및 ‘무작위 선택’ 세 가지 중 하나이며 각 방식의 확률적 선택은 다음 식 (7)과 같다.

$$x_i^N \leftarrow \begin{cases} x_i^N \in \{x_1^1, x_2^1, \dots, x_i^{\text{HMS}}\}, & \text{w.p. HMCR} \cdot (1-\text{PAR}) \\ x_i^N = x_i^N + \text{rand}(-1,1) \cdot bw_i, & \text{w.p. HMCR} \cdot \text{PAR} \\ x_i^N = L_i + \text{rand}(0,1) \cdot (U_i - L_i), & \text{w.p. (1-HMCR)} \end{cases} \quad (7)$$

**2-4 HM 최신화**

단계 3에서 결정된 새로운 하모니는 목적함수에 적용되어

그 결과를 HM내 가장 나쁜 결과와 비교하여 더 좋은 경우 새로운 하모니는 해당 하모니를 대체한다. 이러한 해벡터 즉 하모니 개선과정을 그림으로 나타내면 그림 1과 같다[14].

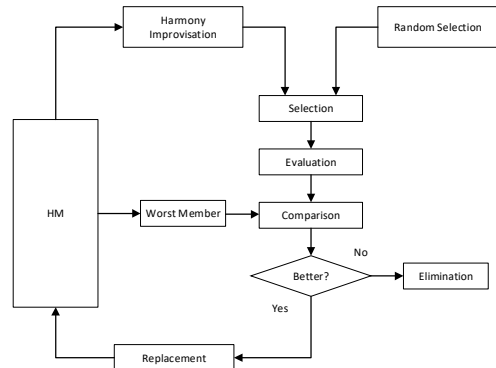


그림 1. HS의 하모니 개선과정  
Fig. 1. HS process of harmony improvisation

단계 3과 4의 새로운 하모니 생성과 최신화 과정은 미리 설정된 종료 기준을 만족할 때까지 반복된다. 지금까지 설명한 HS의 단계별 알고리즘 전개 과정을 그림으로 나타내면 다음 그림 2와 같다[14].

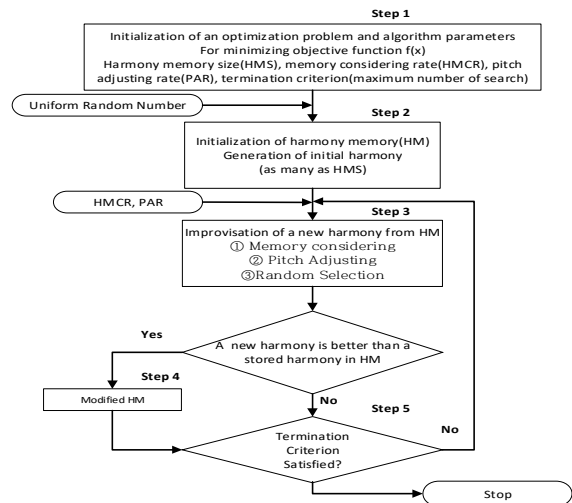


그림 2. HS 알고리즘의 단계별 최적화 절차  
Fig. 2. Optimization step procedure of the harmony search algorithm

**III. FHS(Fast Harmony Search)**

앞에서 살펴봤듯이 기존 HS는 피치조정에 필요한 대역폭의 명시적인 결정이 필요하고 HM을 기반으로 하는 지역탐색과 전역탐색은 서로 독립적인 구조를 갖는다.

본 연구에서는 지역탐색에 필요한 대역폭과 전역탐색을 HM을 기반으로 하나의 구조로 통합하였다. 지금부터 이에 대해 살

피보기로 한다.

(2)식으로 정의된 변수의 경계는 (4)식과 같이 HM에 저장된 값  $x_i^j$ 를 이용하여 다음과 같이 나타낼 수 있다.

$$\begin{cases} L_i = x_i^j + (L_i - x_i^j) \\ U_i = x_i^j + (U_i - x_i^j) \end{cases}, i \in 1 \sim n, j \in 1 \sim \text{HMS} \quad (8)$$

(8)식을 이용하면 변수  $x_i$ 의 임의 영역을 변숫값  $x_i^j$ 을 중심으로 다음과 같이 정의 할 수 있다.

$$x_i^j + p \cdot (L_i - x_i^j) \leq x_i \leq x_i^j + p \cdot (U_i - x_i^j), i \in 1 \sim n, j \in 1 \sim \text{HMS} \quad (9)$$

(9)식에서 매개변수  $p$ 는  $x_i^j$ 를 중심으로 탐색 영역을 결정하며  $0 \leq p \leq 1$  이다.  $p=1$ 이면  $x_i$ 의 영역은  $L_i \leq x_i \leq U_i$  인 전역이 되고  $p < 1$  인 경우  $x_i^j$ 를 중심으로 전역보다 작은 국지영역이 된다. 새로운 변숫값  $x_i^N$  결정을 위해  $V_i$ 를 다음과 같이 정의 하자.

$$V_i = \begin{cases} L_i, & \text{w.p. } 0.5 \\ U_i, & \text{w.p. } 0.5 \end{cases} \quad (10)$$

(10)식을 이용하여 (7)식의  $x_i^N$ 을 다음 식과 같이 정의하자.

$$x_i^N = x_i^j + \text{rand}(0,1) \cdot p \cdot (V_i - x_i^j), i \in 1 \sim n, j \in 1 \sim \text{HMS} \quad (11)$$

그러면  $x_i^N$ 는  $p$ 값에 따라 다음 식 (12)와 같이 나타낼 수 있다.

$$x_i^N = \begin{cases} x_i^j + \text{rand}(0,1) \cdot (V_i - x_i^j), & \text{if } p=1 \\ x_i^j, & \text{if } p=0 \\ x_i^j + \text{rand}(0,1) \cdot p \cdot (V_i - x_i^j), & \text{if } 0 < p < 1 \end{cases} \quad (12)$$

(12)식에 따라 결정된  $x_i^N$  값은 다음 식 (13)과 같다.

$$\begin{cases} L_i \leq x_i^N \leq U_i, & \text{if } p=1 \\ x_i^N = x_i^j, & \text{if } p=0 \\ x_i^j + p \cdot (L_i - x_i^j) \leq x_i^N \leq x_i^j + p \cdot (U_i - x_i^j), & \text{if } 0 < p < 1 \end{cases} \quad (13)$$

(13)식의 관계는 다음과 같이 그림 3.으로 나타낼 수 있다.

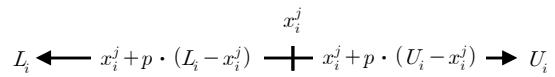


그림 3. 식 (13)에 대한 도식적 표현  
Fig. 3. Schematic figure for eq. (13)

따라서  $p=1$ 인 경우  $x_i^N$ 는  $x_i^j$  값을 중심으로 전역탐색에 의해 결정되고  $p=0$ 인 경우 피치조정 없는 기억회상이 된다.  $0 < p < 1$ 인 경우 피치조정이 이루어지며 대역폭은 다음과 같이 나타낼 수 있다.

$$bw_i = \begin{cases} \Delta_{iL} = p \cdot (L_i - x_i^j) \\ \Delta_{iU} = p \cdot (U_i - x_i^j) \end{cases}, p < 1 \quad (14)$$

이제 (11)식의  $p$ 를 다음식과 같이 결정하게 되면

$$p = \begin{cases} 1, & \text{w.p. } (1 - \text{HMCR}) \\ \delta, & \text{w.p. } \text{HMCR} \cdot \text{PAR} \\ 0, & \text{w.p. } \text{HMCR}(1 - \text{PAR}) \end{cases}, \delta < 1 \quad (15)$$

(11)식으로 HS의 구현이 가능하다. (11)식과 (15)식으로  $x_i^N$ 을 결정하게 되면 (14)식의 대역폭은  $x_i^j$ 을 중심으로 폭의 크기가 대칭적인 기존 대역폭과 다르게  $|L_i - x_i^j|$ 와  $|U_i - x_i^j|$  중 큰 쪽의 대역폭이 더 큰 비대칭적인 구조가 되며 이러한 결과는 매우 합리적이라 할 수 있다.

$p=1$ 인 전역탐색의 경우 기존과 달리 HM에 저장된 해를 이용하게 되어 알고리즘의 정보 활용성이 증대된다. 이와 더불어 새로운 방식은 대역폭의 초기화가 필요하지 않으며 전역탐색과 지역탐색이 하나로 통합되어 알고리즘 수행속도가 빠르게 된다.

#### IV. 예시 및 성능 비교

기존 알고리즘(HS)과 제시된 알고리즘(FHS) 성능 비교를 위해 4차원 이하 저차원 함수 3개와  $n=10$ 인 고차원 함수 2개를 사용하였다. 다음 표 1은 예시에 사용한 함수들이다. 두 종류 함수 모두에 대해 100번 반복 실행하였다.

표 1. 예시에 사용한 함수

Table 1. Functions for Numerical example

Function Name	Expression	Optimal Value	Search Range
Rosenbrock	$f(x) = 100(x_2 - x_1^2) + (1 - x_1)^2$	$f^*(1, 1) = 0$	$-10 \leq x_i \leq 10$
Wood	$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2]$	$f^*(1, 1, 1, 1) = 0$	$-10 \leq x_i \leq 10$
Goldstein and Price 1	$f(x) = \left\{ 1 + (x_1 + x_2 + 1) \left( \frac{19 - 14x_1 + 3x_1^2 - 14x_2}{3x_2^2} \right) \right\} \times \left\{ 30 + (2x_1 - 3x_2)^2 \left( \frac{18 - 32x_1 + 12x_1^2}{48x_2 - 36x_1x_2 + 27x_2^2} \right) \right\}$	$f^*(0, -1) = 3.0$	Multimodal $-10 \leq x_i \leq 10$
Rastrigin	$f(x) = \sum_{i=1}^{n=10} (x_i^2 - 10\cos(2\pi x_i)) + 10$	$f(0) = 0$	$-5.12 \leq x_i \leq 5.12$
Ackely	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{10}\sum_{i=1}^{n=10} x_i^2}\right) - \exp\left(\frac{1}{10}\sum_{i=1}^{n=10} \cos(2\pi x_i)\right) + 20 + \exp(1)$	$f^*(0) = 0$	Multimodal $-32 \leq x_i \leq 32$

알고리즘 수행을 위해 사용한 매개변수 값은 HMS = 10, HMCR = 0.85, PAR = 0.75 이다. 두 알고리즘의 성능평가를 위한 지표로 알고리즘 평균실행 시간을 통해 수행속도를, 알고리즘 효율성 평가를 위해 평균오차, 표준편차, 최고해 및 허용오

차에 대한 성공률을 비교하였다. 이를 위해 저차원 함수의 경우 알고리즘 실행시수를 Maxit = 70,000로 허용오차를 하고 고차원 함수의 경우 Maxit = 100,000로, 로 설정하였다. 다음 표 2는 그 결과를 나타낸다.

표 2. HS vs. FHS 성능 비교

Table 2. performance comparison of HS vs. FHS

	Rosenbrock		Wood		Goldstein and Price 1		Rastrigin		Ackely	
	HS	FHS	HS	FHS	HS	FHS	HS	FHS	HS	FHS
Average Execution time(sec)	0.3828	0.2035	0.5719	0.2192	0.3823	0.1891	1.6068	0.4477	1.6604	0.4586
Average error	0.0084	1.2193e-06	1.5991	1.7378	6.3984	0.8115	3.2480e-05	1.0038e-07	2.8095e-04	6.9721e-06
Best error	7.9260e-12	5.8808e-12	1.8273e-08	4.7174e-09	3.1086e-15	0	7.2195e-06	3.0803e-08	1.2783e-04	3.9777e-06
SD	0.0552	1.2048e-05	3.8134	3.1334	15.4605	4.6288	9.4505e-06	2.7319e-08	4.0910e-05	1.2053e-06
Success rate(%)	96	99	38	60	64	94	2	100	0	100

표 2에서 보듯이 기존 HS보다 FHS가 저차원과 고차원 모든 함수와 평가지표에서 우수함을 알 수 있다. 본문에서 주장한 것과 같이 알고리즘 수행시간의 차이가 매우 크다는 것을 알 수 있다. 저차원 함수의 경우 평균 2.17배, 고차원 함수의 경우 평균 3.6배 차이가 난다. 허용오차에 대한 성공률 또한 전반적으

로 많은 차이가 있으며 차원이 높은 경우 그 차이가 크게 나타남을 보여주고 있다. 목적변수의 수가 많을수록 더 큰 차이를 보이고 있다. 수행속도의 차이는 FHS가 HS 참조와 무작위 선택을 하나로 통합한 결과로 나타나는 효과이다.

## V. 결 론

본 연구에서는 기존 HS에서 서로 독립적인 관계를 갖는 HM 참조와 무작위 선택을 하나의 과정으로 통합하였다. 이를 통하여 기존 HS에서 필수적인 대역폭 설정이 필요 없게 되었고 새로운 목적변수 값을 결정하는 알고리즘 수행시간을 줄일 수 있었다. 또한 전역탐색을 HM을 기반으로 함으로써 알고리즘의 기존 정보 활용성을 높여 전역탐색의 효율성을 높일 수 있었다.

제시한 알고리즘을 종래 HS와 비교한 결과 알고리즘 수행 속도가 매우 빨라졌음을 확인 할 수 있었으며 이러한 효과는 목적변수 수가 많을수록 더욱 크게 나타났다. 아울러 다른 성능평가 지표들도 HS보다 우수함을 확인 할 수 있었다. 이것은 FHS에 새롭게 적용되는 매개변수가 지역탐색과 전역탐색에 매우 효과적임을 보여준다.

향후 FHS에서 도입된 매개변수  $p$ 의 선택방법 개선을 통해 FHS의 성능을 한 층 더 개선 할 수 있을 것이다. 이와 더불어 IHS 등 그동안 발표된 HS 성능 개선 방식에 FSH를 적용하면 더 좋은 결과가 있을 것으로 기대된다.

## References

[1] T. Bäck, D. Fogel, and Z. Michalewicz, Handbook of Evolutionary Computation. London, U.K.: xford Univ. Press, 1997.

[2] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing. New York: Springer-Verlag, 2003.

[3] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence. Hoboken, NJ: Wiley, 2006.

[4] J. Kennedy, R. C. Eberhart, and Y. Shi, Swarm Intelligence. San Francisco, CA: Morgan Kaufmann, 2001.

[5] Slowik, A., Kwasnicka, H. Evolutionary algorithms and their

applications to engineering problems. Neural Comput. & Applic. 32, 12363–12379, 2020.

[6] M. Dorigo and G. Di Caro, “Ant colony optimization: a new meta-heuristic,” Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 1999, pp. 1470-1477, Vol. 2.

[7] EBERHART, Russell; KENNEDY, James. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. 1995. pp.1942-1948.

[8] GOLDBERG, David E.; HOLLAND, John H. Genetic algorithms and machine learning, Machine learning, 1988, 3.2: pp.95-99.

[9] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: Harmony search,” J. Simul., vol. 76, no. 2, pp. 60–68, Feb. 2001.

[10] A. A. Al-Omouh, A. A. Alsewari, H. S. Alamri and K. Z. Zamli, “Comprehensive Review of the Development of the Harmony Search Algorithm and its Applications,” in IEEE Access, vol. 7, pp. 14233-14245, 2019,

[11] MAHDAVI, Mehrdad; FESANGHARY, Mohammad; DAMANGIR, E. An improved harmony search algorithm for solving optimization problems, Applied mathematics and computation, Vol.188, No.2: pp.1567-1579, 2007.

[12] OMRAN, Mahamed GH; MAHDAVI, Mehrdad. Global-best harmony search, Applied mathematics and computation, Vol.198, No.2: pp.643-656, 2008,

[13] WANG, Chia-Ming; HUANG, Yin-Fu. Self-adaptive harmony search algorithm for optimization, Expert Systems with Applications, Vol.37, No.4, pp.2826-2837, 2010.

[14] T. B. Lee, “Analytic Approach to Determin Algorithm Parameters of HS Optimization,” Trans. of KIEE vol. 68, no. 4, pp. 200-206, 2019.



**이 태 봉 (Tae-Bong Lee)**

1986년 홍익대학교 전자공학과 졸업.  
1989년 동 대학원 전자공학과 (석사/박사)  
1995년 ~ 현재 : 가천대학교 전자공학과 교수  
※ 관심분야 : 최적이론, 건설제어, 휴리스틱기법