

# 매개변수 자가적응 화음탐색 알고리즘의 성능 비교를 통한 최적해 탐색 효율 향상

최영환<sup>1</sup>, 이호민<sup>2</sup>, 유도근<sup>3</sup>, 김종훈<sup>1\*</sup>

<sup>1</sup>고려대학교 건축사회환경공학부, <sup>2</sup>고려대학교 방재과학기술연구소, <sup>3</sup>수원대학교 건설환경공학

## Improvement of Search Efficiency in Optimization Algorithm using Self-adaptive Harmony Search Algorithms

Choi Young Hwan<sup>1</sup>, Lee Ho Min<sup>2</sup>, Yoo Do Guen<sup>3</sup>, Kim Joong Hoon<sup>1\*</sup>

<sup>1</sup>School of Civil, Environmental, and Architectural Engineering, Korea University

<sup>2</sup>Research Center for Disaster Prevention Science and Technology, Korea University

<sup>3</sup>Department of Civil Engineering, University of Suwon

**요약** 다양한 공학분야의 최적화 문제를 해결하기 위해 적절한 매개변수를 설정하기란 번거로운 작업이며, 매개변수 민감도 분석을 통해 적절한 매개변수를 설정하더라도 설정된 매개변수가 모든 문제에 적절한지 판단하기에는 한계가 있다. 이러한 이유로 매개변수를 문제에 따라 적절하게 설정하는 매개변수 자동검보정 (Self-adaptive) 화음탐색 알고리즘이 개발되고 발전하고 있다. 본 연구에서는 지금까지 개발된 자가적응형 하모니시치를 조사하고 그의 특성을 해탐색, 설정 매개변수, 적용성 등으로 구분하였으며, 이 중 매개변수 설정의 번거로움을 없애고, 적절한 매개변수 설정을 통해 해의 성능 향상을 위해 개발된 6 가지 자가적응형 화음탐색 알고리즘을 선택하여 비교 분석을 수행하였다. 최적화 결과의 객관적인 비교를 위해 대표적인 수학적, 공학적 최적화 문제를 모두 적용 하였고, 다양한 성능 지수 (Performance index)를 사용하여 각 알고리즘의 성능을 정량적으로 비교하였다. 이것은 향후 신규 최적화 알고리즘을 개발하거나 해 탐색의 성능을 향상시키는 연구에 도움이 될 것으로 기대된다.

**Abstract** In various engineering fields, determining the appropriate parameter set is a cumbersome and difficult task when solving optimization problems. Despite the appropriate parameter setting through parameter sensitivity analysis, there are limits to evaluating whether the parameters are appropriate for all optimization problems. For this reason, kinds of a Self-adaptive Harmony searches have been developed to solve various engineering problems by the appropriate setting of algorithm's own parameters according to the problem. In this study, various types of Self-adaptive Harmony searches were investigated and the characteristics of optimization were categorized. Six algorithms with a differentiation of optimization process were applied and compared with not only the mathematical optimization problem, but also the engineering problem, which has been applied widely in the algorithm performance comparisons. The performance of each algorithm was compared, and the statistical performance indicators were used to evaluate the application results quantitatively.

**Keywords** : Algorithm, Harmony Search, Metaheuristic, Optimization, Self-adaptive approach

---

This work was supported by a grant from The National Research Foundation (NRF) of Korea, funded by the Korean government (MSIP) (no. 2016R1A2A1A05005306).

\*Corresponding Author : Kim Joong Hoon(Korea Univ.)

Tel: +82-2-3290-4724 email: jaykim@korea.ac.kr

Received December 15, 2017

Revised January 4, 2018

Accepted January 5, 2018

Published January 31, 2018

## 1. 서론

최적화 알고리즘은 수학 및 공학적인 문제를 해결하기 위해서 다양한 분야에서 개발/적용되었다. 그러나 최적화 알고리즘의 탐색성능 (지역탐색, 전역탐색)은 알고리즘의 매개변수 설정에 따라 많은 차이를 보이고 있다. 이러한 이유로 최적해를 효율적으로 탐색하기 위해 연구자들은 각 알고리즘의 매개변수를 다양하게 조합하여 최적에 가까운 해를 찾는 매개변수 민감도 분석을 통해 적절한 매개변수를 설정한다. 그러나 민감도 분석은 적절한 매개변수 조합을 탐색하기에 매우 번거로우며, 이렇게 결정된 매개변수가 여러 형태의 최적화 문제에 가장 적절한지에 대해서는 판단하기 어렵다 [1]. 이러한 매개변수 설정 문제를 개선하기 위해 매개변수 설정이 필요 없는 자가적응형 접근법 (Self-adaptive approach)과 같은 매개변수 제어 방법이 다양한 최적화 알고리즘에서 연구되고 있다 [2~4].

마찬가지로 화음탐색 알고리즘 (또는 화음탐색법, Harmony Search) 또한 이러한 매개변수 자동설정을 통해 해탐색의 효율성을 높이고 있다. 화음탐색 알고리즘은 음악의 즉흥연주 개념을 적용하여 최적화를 수행하는 기법으로 Geem et al.[5]과 Kim et al. [6]에 의해서 제안되었다. 화음탐색 알고리즘은 최적해 탐색을 위해 HM (Harmony memory), HMCR (Harmony memory considering rate), PAR (Pitch adjusting rate), Bw (Bandwidth)등을 매개변수로 사용하고 있다. HM이란 반복 훈련과정을 통해 얻은 화음 중 아름답다고 판단하는 화음을 모은 집합이며, HMCR은 새로운 화음을 생성할 때 아름답다고 생각되는 기존의 화음을 고려하여 무작위로 생성할 지를 고려하는 확률이다. PAR은 기존 하모니를 고려하여 새로운 메모리를 생성한 후 미세한 음조를 조절을 통해 해의 성능을 향상하는 최적화 과정이다. 음조를 조절하는 폭은 Bw라고 한다.

화음탐색 알고리즘은 이전 연구된 최적화 알고리즘 (Taboo Search [7], Simulated Annealing [8], Ant Colony Optimization [9], Particle Swarm Optimization [10])과 비교해서 수학적인 연산이 적고, 다양한 분야의 최적화 문제를 해결할 수 있다 [11, 12]. 그러나 화음탐색 알고리즘의 세 가지 매개변수 (i.e., HMCR, PAR, Bw)는 상수 값이며, 이러한 매개변수의 값은 해결하는 최적화 문제에 특성에 의존하기 때문에 많은 시행을 통해 결정되

어야 하는 번거로움이 있다. 이러한 이유로 최적해의 성능 향상과 적절한 매개변수의 설정을 위해 다양한 매개변수를 자동으로 설정하는 자가적응형 화음탐색 알고리즘 (Self-adaptive Harmony Search)이 제안되었다.

현재까지 개발된 자가적응형 화음탐색 알고리즘 성능에 크게 두 가지 범주의 방법을 적용하여 매개변수를 자동 설정하였다. 첫 번째는 새로운 가능해 (Feasible Solution)를 생성하기 위한 연산자를 변경하는 방법이다. 이는 기존 화음탐색 알고리즘에는 사용자에게 의해서 상수 값으로 입력되던 매개변수를 대체할 수 있는 새로운 연산자를 추가하여 최적해를 탐색하는 방법이다. 두 번째는 매개변수 설정을 위해 수학적 연산을 도입하여 매회 반복시산에서 적절한 새로운 매개변수를 설정하는 방법이다.

Pan et al. [13]은 Global Harmony Search [12]에서 매개변수의 자동설정이 가능한 Self-adaptive Global Harmony Search (SGHS) 알고리즘을 개발하였다. SGHS는 화음탐색 알고리즘의 매개변수 중 Bw의 값을 반복시산의 횟수가 증가함에 따라 감소하게 설정하여, 최적화가 진행하면서 지역탐색의 성능을 강화하였다. Wang and Huang [14]과 Degertekin [15]는 화음탐색 알고리즘의 매개변수 중 PAR과 Bw를 이용한 해탐색의 방법을 대체하는 새로운 방법을 제시하여 매개변수 설정의 번거로움을 제거하는 알고리즘인 Self-adaptive Harmony Search (SAHS)를 개발하였다. 이 방법은 해탐색 방법 중 음 조절(Pitch Adjustment)과정에서 이웃한 솔루션을 고려하여 최적해를 탐색하는 방법에서 각 결정변수의 HM의 최대값, 최소값을 이용한 탐색과정으로 대체하였다. 대체된 방법은 새로운 솔루션 생성 시 탐색공간을 HM으로 제한하여 최적탐색성능을 높인다.

Jing et al. [16]와 Luo [17]는 적절한 HMCR, Bw 설정을 위해 Novel Self-adaptive Harmony Search (NSHS)를 개발했다. NSHS는 결정변수의 개수가 증가함에 따라 HMCR의 값이 증가하도록 설정하였으며, Bw는 결정변수의 경계 조건과 반복시산이 증가함에 따라 감소하여 최적화가 진행되면서 지역탐색의 성능이 향상되도록 설정하였다. Geem [18]과 Geem and Cho [1]는 화음탐색 알고리즘의 구조변형을 통하여 매회 반복시산을 수행 할 때마다 자동으로 매개변수가 업데이트되는 Parameter-Setting-Free Harmony Search (PSF-HS) 방법을 제시하였다. PSF-HS는 새로운 Operation Type Memory

(OTM)을 도입하여 최적화 수행과정에서 Harmony Memory Consideration과 Pitch Adjustment 과정 수행여부를 고려하여, HMCR과 PAR을 매회 업데이트한다.

Jiang et al. [19]은 이전의 PSF-HS 방법에는 HMCR 및 PAR 매개변수만을 고려했지만, 추가적으로 Bw까지 고려하여 세 가지 매개변수를 모두 설정해주는 Almost-parameter-free Harmony Search (APF-HS)를 개발하였다. APF-HS에서 Bw는 매회 반복시산을 통해 생성된 결정변수와 경계조건의 변수에 의해 결정되도록 하였다.

본 연구는 최적화 탐색기법 중 하나인 화음탐색 알고리즘의 향상된 버전인 다양한 매개변수 자동검보정 알고리즘의 성능을 비교하고, 각 기법의 해 탐색 특징을 분석하였다. 또한, 23개의 자가적응형 화음탐색 알고리즘을 적용한 문헌을 Table 1과 같이 조사하였으며, 유사한 매개변수 설정방법을 사용한 알고리즘을 제외하면 6 가지 기술로 요약 될 수 있다. 지금까지 자가적응형 화음탐색 알고리즘에 대한 연구는 각자 방법론을 개발하고, 서로 다른 최적화 문제에 적용하여 제안된 방법론의 우수성을 보여주는 절차로 진행되었다. 본 연구에서는 매개변수 설정의 번거로움을 없애고, 적절한 매개변수 설정을 통

해 해의 성능 향상을 위해 개발 된 6 가지 자가적응형 화음탐색 알고리즘을 객관적으로 비교하기 위하여 대표적인 수학적, 공학적 최적화 문제를 모두 적용 하였고, 다양한 성능 지수 (Performance index)를 사용하여 각 알고리즘의 성능을 정량적으로 비교하였다. 이것은 향후 신규 최적화 알고리즘을 개발하거나 해 탐색의 성능을 향상시키는 연구에 도움이 될 것으로 기대된다.

## 2. 방법론

본 연구에서는 앞서 언급한 23종류의 자가적응형 화음탐색 알고리즘 중 해 탐색기법과 매개변수 설정하는 기법이 다른 6 가지 기법 (Parameter-setting-free Harmony Search [18], [20], Almost-Parameter-Free Harmony Search [19], Novel Self-adaptive Harmony Search [17], Parameter Adaptive Harmony Search [32], Self-adaptive Global-based Harmony Search Algorithm [28]) 을 선정하여 그 특성을 서로비교, 분석하였다.

Table 1. Category of Self-adaptive Harmony Search

| Algorithm                         | Application field                | Improvement   | Author |
|-----------------------------------|----------------------------------|---------------|--------|
| Single-objective Self-adaptive HS | Water Resource Eng.              | HMCR, PAR     | [18]   |
|                                   | Water Resource Eng.              | HMCR, PAR     | [1]    |
|                                   | Mathematics, Water Resource Eng. | HMCR, PAR     | [20]   |
|                                   | Economic dispatch                | HMCR, PAR     | [21]   |
|                                   | Water Quality Eng.               | HMCR, PAR, Bw | [19]   |
|                                   | Mathematics                      | Bw            | [13]   |
|                                   | Mathematics                      | Bw            | [14]   |
|                                   | Mathematics                      | HMCR, Bw      | [16]   |
|                                   | Mathematics                      | HMCR,PAR      | [22]   |
|                                   | Structures Eng.                  | Bw            | [15]   |
|                                   | Mathematics                      | PAR, Bw       | [23]   |
|                                   | Mathematics                      | HMCR, PAR, Bw | [17]   |
|                                   | Traffic Eng.                     | PAR, Bw       | [24]   |
|                                   | Mathematics                      | HMCR          | [25]   |
|                                   | Data mining                      | PAR           | [26]   |
|                                   | Economic dispatch                | PAR           | [27]   |
|                                   | Electricity system               | PAR, Bw       | [28]   |
|                                   | Electricity system               | -             | [29]   |
|                                   | Mathematics                      | HMCR, PAR     | [30]   |
|                                   | Mathematics                      | PAR, Bw       | [31]   |
| Mathematics                       | HMCR, PAR, Bw                    | [32]          |        |
| Multi-objective Self-adaptive HS  | Mathematics                      | HMCR, PAR, Bw | [33]   |
|                                   | Mechanical Eng.                  | HMCR, PAR, Bw | [34]   |

## 2.1 Harmony Search

화음탐색법 (Harmony Search, HS)은 화음이라는 인공적인 현상에서 착안하여 만들어진 최적화 알고리즘으로, 음악가들의 즉흥 연주 과정에서 발생하는 여러 가지 악기가 발생하는 소리는 하나의 화음을 생성하게 된다. 이렇게 만들어지는 화음 중에 잘 어울리는 화음이 있고, 불협화음을 내는 경우가 있지만 반복적인 연습과정을 통해 불협화음은 점차 사라지게 되고, 아름다운 화음을 구성하는 최상의 조합을 탐색하는 것이다. 화음탐색법에서는 총 4 가지 매개변수 (Harmony Memory (HM), Harmony Memory Considering Rate (HMCR), Pitch adjustment Rate (PAR), Bandwidth (Bw))을 고려하여 최적의 해를 탐색한다. HM이란 훈련의 과정 속에서 아름다운 화음이라고 생각되는 화음을 모은 집합으로 최적화 구동시 HM의 수를 사전에 결정하여야 한다. 최적화 과정시 새롭게 발생한 화음이 기존 화음보다 좋다면 새로운 화음은 추가되고 기존 화음 중 가장 좋지 못한 화음은 도태되게 한다. HMCR이란 새로운 화음을 만들 때, HM에 있는 화음을 바탕으로 만들 것인지 새로운 영역에서 무작위로 생성 할 지를 판단하는 확률이다. HMCR은 반복계산이 지속되면서 알고리즘에서 발생할 수 있는 지역해에 빠지지 않는 것을 방지하고, HM의 악기의 음조를 바탕으로 더욱 좋은 화음을 발생할 수 있는 가능성을 부여하는 역할을 한다. PAR은 좋은 화음을 만들기 위해 각 악기의 음조를 조절하는 것과 같이 우수한 해를 찾기 위해서 기존 해와 이웃하는 값을 고려하면서 HS의 성능을 향상시키는 인자로 여기서 음조를 조정하는 폭을 Bw라고 한다. 화음탐색법의 연산은 식(1), (2)와 같다.

$$x_i^{New} = \begin{cases} x_i \in [x_i^{Lower}, x_i^{Upper}] & \text{if } Rnd > HMCR \\ x_i \in HM = [x_i^1, x_i^2, \dots, x_i^{HMS}] & \text{if } Rnd \leq HMCR \end{cases} \quad (1)$$

$$x_i^{New} = \begin{cases} x_i^{New} & \text{if } Rnd > PAR \\ x_i^{New} + Bw & \text{if } Rnd \leq PAR \end{cases} \quad (2)$$

여기서,  $x_i^{New}$ 은 새로운 해의 결정변수이며,  $x_i^{Lower}$ 와  $x_i^{Upper}$ 는 결정변수의 경계조건이다.  $Rnd$ 는 [0,1]사이의 랜덤변수이다.

## 2.2 Parameter-Setting-Free Harmony Search

적합한 매개변수 설정의 번거로움을 줄이기 위해 Geem [18]에서 제안한 Parameter-setting-free Harmony Search (PSF-HS)는 매회 반복시산과정에서 결정변수에

다른 HMCR과 PAR을 설정함으로써 초기 최적화 과정에서는 전역탐색을 높이고, 후기 최적화 과정에서는 지역탐색비율을 증가하는 등의 해 탐색의 효율을 향상시켰다. PSF-HS는 매개변수를 갱신하기 위해서 식 (3)과 같이 매회 반복시산 과정에서 해탐색을 위한 연산 방법 (Random Search (RS), Harmony Memory Consideration (HMC), Pitch Adjustment (PA))을 기억하는 Operation Type Memory (OTM)을 소개했다. HS 연산자 (RS, HMC, PA) 중에서 새로운 솔루션을 생성하기 위해 이전 반복시산에서는 각 결정변수가 어떤 연산자를 통해 생성되었는지를 기억하여 각 결정변수의 HMCR 및 PAR을 식 (4)와 같이 갱신한다. 그러나 Geem and Shim [20]은 식(5)와 같이 각기 다른 갱신연산을 수행하여, 본 연구에서는 두 가지 계산 방법을 각각 고려하였다.

$$OTM = \begin{bmatrix} y_1^1 = RS & y_2^1 = PA & \dots & y_n^1 = HMC \\ y_1^2 = RS & y_2^2 = HMC & \dots & y_n^2 = PA \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{HMS} = RS & y_2^{HMS} = HMC & \dots & y_n^{HMS} = RS \end{bmatrix} \quad (3)$$

$$HMCR_i = \frac{n(y_i^j = HMC \text{ or } PA)}{HMS}$$

$$PAR_i = \frac{n(y_i^j = PA)}{n(y_i^j = HMC \text{ or } PA)} \quad (\text{by Geem, 2010}) \quad (4)$$

$$HMCR_i = \frac{n(y_i^j = HMC)}{HMS} \quad PAR_i = \frac{n(y_i^j = PA)}{n(y_i^j = HMS)}$$

(by Geem and shim, 2010) (5)

PSF-HS는 반복시산마다 매개변수를 갱신하였을 때, 반복 횟수가 증가함에 따라 일반적으로 HMCR은 증가하지만 PAR은 감소하게 된다. 이러한 경향은 HMCR과 PAR의 최대 최소 범위인 1과 0을 초과 할 수 있다. 이 문제를 방지하기 위해 PSF-HS에서는 식 (6)과 같이 Noise vector( $\emptyset$ )를 사용하여 HMCR과 PAR을 1과 0의 범위 내에서 설정될 수 있도록 한다.

$$HMCR_i = \frac{HMCR_i + \emptyset \times U(-1, 1)}{HMCR_i} \in [0, 1]$$

$$PAR_i = \frac{PAR_i + \emptyset \times U(-1, 1)}{PAR_i} \in [0, 1] \quad (6)$$

여기서,  $U(-1, 1)$ 은 -1부터 1사이의 무작위 수이며,  $\emptyset$ 는 Noise vector이다. 이 값은 일반적으로 0에서 1사이의 값을 사용한다.

### 2.3 Almost-Parameter-Free Harmony Search

Jiang et al. [19]에 의해 제안된 Almost-Parameter-Free Harmony Search (APS-HS)은 기존 PSF-HS [20] 방법에서 고려된 HMCR 및 PAR이 자동으로 설정되는 것을 포함하고, 추가적으로 동적 Bw 설정을 고려하는 PSF-HS의 수정된 버전이다. APS-HS는 PSF-HS와 동일하게 OTM의 개념을 적용하여 HMCR과 PAR의 설정을 식 (3), (5), (6)와 같이 계산한다. 또한 Bw는 식 (7)과 같이 HM의 최대 및 최소 결정변수 값에 따라 매 반복시산과정에서 동적으로 업데이트한다.

$$x_i^{j+1} = \begin{cases} x_i^j - [x_i^j - \min(x_i^j)] \times U(0,1) & \text{if, } Rnd > 0.5 \\ x_i^j - [\max(x_i^j) - x_i^j] \times U(0,1) & \text{otherwise} \end{cases} \quad (7)$$

여기서,  $x_i^j$ 는  $j$ 번째 반복시산의  $i$ 번째 결정변수 값이다.

### 2.4 Novel Self-adaptive Harmony Search

Novel self-adaptive harmony search (NSHS)는 Luo [17]에 의해서 최초로 제안되었으며, 이는 HMCR, PAR, Bw를 고정값으로 사용자가 설정해주는 기존의 HS에서 HMCR과 Bw는 결정변수의 개수와 적합도 (fitness)의 변화량에 따라 동적으로 업데이트하여, HMC와 PA의 해탐색 과정을 개량한 기법이다. 우선 HMCR의 값은 결정변수의 개수가 많은 복잡한 최적화 문제에서는 식 (8)과 같이 1에 가까운 HMCR의 값을 사용하여 전역탐색의 확률을 감소시켰다. 또한 HS의 해 탐색의 중요한 요소 중 하나인 Bw의 설정을 위해서 NSHS는 식 (9)와 같이 최적화가 진행됨에 따라 Bw의 값을 감소시키는 동적 Bw를 사용하였다. PA 절차는 기존 HS에서 Pitch를 조정하는 과정에서 적합도의 표준편차를 고려하여 개량된 새로운 해 탐색 방법을 제안하였고, 이 방법은 결정변수의 경계조건에 의해 새로운 해를 생성한다 (식 10).

$$HMCR = 1 - \frac{1}{n(DV) + 1} \quad (8)$$

$$BW_i^j = \begin{cases} \frac{Bw_i^{Upper} - Bw_i^{Lower}}{100} \times (1 - \frac{j}{NI}) & \text{if, } f_{std} < 0.0001 \\ 0.0001 & \text{if, otherwise} \end{cases} \quad (9)$$

$$x_i = \begin{cases} x_i + U[-1,1] \times Bw & \text{if, } HMCR \geq U(0,1) \\ LB + U[0,1] \times (UB - LB) \\ + U[-1,1] \times Bw & \text{if, } HMCR \geq U(0,1) \text{ and } f_{std} > 0.0001 \\ \min x_i + U[0,1] \times (\max x_i - \min x_i) \\ + U[-1,1] \times Bw & \text{if, } HMCR \geq U(0,1) \text{ and } f_{std} > 0.0001 \end{cases} \quad (10)$$

여기서,  $n(DV)$ 는 결정변수의 개수,  $f_{std}$ 는 적합도의 표준편차  $f_{std} = std(f(x^1), f(x^2), \dots, f(x^{HMS}))$ 이며,  $Bw_i^{Lower}$ ,  $Bw_i^{Upper}$ 는  $j$ 번째 반복시산의  $i$ 번째 Bandwidth,  $NI$ 는 총 반복시산 횟수이다.

### 2.5 Parameter Adaptive Harmony Search

Kumar et al. [32]는 HMCR과 PAR, Bw 값의 동적 변화를 제안하여 Parameter Adaptive Harmony Search (PAHS)라는 HS의 향상된 버전을 제안하였다. PAHS는 HS의 세 가지 매개변수를 경계조건과 반복시산이 진행됨에 따라 변화 하도록 설정하였다. 예를 들어 초기 반복시산에서는 HMCR의 값이 최소 경계 조건과 가깝게 설정되며, 후기 반복시산에서는 최대 경계 조건 값으로 설정하도록 하였으며, PAR의 경우는 이와 반대로 반복시산이 진행됨에 따라 값이 작아지도록 설정하였다. 또한 이러한 동적 매개변수의 설정을 선형 (Linear) 형태와 지수 (Exponential) 형태로 하여, 그 결과를 비교하였다. 그 결과 HMCR의 값은 선형 형태, PAR은 지수 형태의 경우에서 우수한 해를 얻었다. 따라서 본 연구에서는 식 (11), (12)과 같이 동적 HMCR, PAR을 적용하였다.

$$HMCR^j = HMCR^{\min} \times \left( \frac{HMCR^{\max} - HMCR^{\min}}{NI} \right) \times j \quad (11)$$

$$PAR^j = PAR^{\min} \times \exp\left( \frac{\ln(PAR^{\max} / PAR^{\min})}{NI} \right) \times j \quad (12)$$

### 2.6 Self-adaptive Global-based Harmony Search Algorithm

Shivaie et al.에서는 더 나은 최적해 탐색과 효과적인 매개변수 설정을 위해 2015년에 Self-adaptive Global-based harmony search (SGHSA)를 개발했다. SGHSA는 HS의 매개변수 중 Bw의 값을 변화하였으며, 초기 반복시산에서는 최대 경계조건을 적용하고, 반복시산 후기에는 최소 경계조건을 적용하여 지역탐색과 전역탐색의 비율을 조절하였으며, 이는 Bw의 경계조건과 반복시산 횟수에 영향을 받는다 (식 13).

$$Bw_i^j = \begin{cases} Bw_i^{\max} - \frac{Bw_i^{\max} - Bw_i^{\min}}{NI} \times 2 \times j & \text{if } j < \frac{NI}{2} \\ Bw_i^{\min} & \text{otherwise} \end{cases} \quad (13)$$

### 3. 적용 및 결과

본 연구에서는 Rastrigin [35], Dixon and Szego [36], Molga and Smutnicki [37]가 제안한 수학적 벤치마크 문제를 적용하여 6가지 알고리즘의 성능 및 특성에 대해 분석한 후 이를 공학적인 문제 중 상수도관망 최적설계에 적용하였다. Table 2는 최적화 알고리즘의 적용 및 검증에 많이 사용된 수학적 문제이다. 각 문제는 다양한 형태의 최적해 분포 형상을(Bowl-shaped, Valley-shaped, Many local optimum) 가지고 있어, 최적화 기법의 탐색 능력을 판단할 수 있으며, 이 결과를 정량적인 성능지표를 사용하여 알고리즘의 성능을 객관적으로 비교한다.

본 연구의 최적화 시뮬레이션은 Win7 (64X), 8GB, Intel 3.40 GHZ Core (TM) i7 환경에서 Visual Basic 6.0으로 구현되었으며, 4가지 최적화 문제와 6가지 결정 변수 (2, 5, 10, 30, 50, 100)조합을 이용하여 알고리즘을 성능을 비교, 분석하였다. 이때 최대 성능평가횟수 (Number of function evaluation, NFE)는 50,000번으로 설정하였으며, 최적화 결과의 신뢰성을 높이기 위하여 각 시행은 독립적으로 30회 실행하여 이를 통해 목적함수의 최소, 평균, 최대 (best, mean, worst)값 등의 통계값 도출을 수행하였다.

이때, 각 알고리즘의 초기해의 성능이 최적화에 미치는 영향을 최소화 하기 위하여, 무작위로 생성한 동일한 초기해는 모든 알고리즘에 동일하게 사용하였다. 적용 결과는 Fig. 1, Table 3과 같이 각 알고리즘의 성능을 결정변수 개수와 통계분석 결과에 따라 각각 구분하여 나타내었다.

Fig. 1의 경우 각 매개변수에 따른 최적해의 성능을 우수한 순서로 7점부터 1점으로 점수를 부여하였고, 이 점수들의 합을 도시하였다. 예를 들어 Fig.1(a)의 경우 모든 결정변수를 고려하여 최적화를 하였을 때, NSHS가 가장 우수한 결과를 나타내어 6가지 경우에서 7점을 부여 받았고, 총 점수가 35점으로 도시할 수 있다. 여기서 총 점수가 높은 값이 나올수록 알고리즘의 성능이 우수하다고 판단할 수 있다. Fig. 1을 통해서 전반적인 알고리즘의 성능을 판단할 수 있는데, Sphere function과 Griewank function에서는 NSHS가 Best, Mean, Worst 등 모든 측면에서 가장 우수한 결과를 나타냈다. Rosenbrock function에서는 SGHSA가 Best 결과에서는 가장 우수 하지만 Mean, Worst 솔루션은 NSHS가 높은 점수를 나타내는데, 이는 SGHSA이 낮은 차원 즉, 해 탐색범위가 상대적으로 좁은 최적해문제의 경우 해탐색의 효율성이 높다는 것을 확인 할 수 있다. 이러한 결과는 Table 3에서 구체적인 결과를 확인 할 수 있다.

Table 3에서는 Fig. 1의 결과 중 각각의 결정변수의 개수에 따른 Best 솔루션의 결과를 나타낸 것으로 이를 통해 각 알고리즘의 해 탐색범위 (결정변수의 수)와 성능에 대한 관계를 확인 할 수 있다.

Table 3의 결과에 따라 4가지 benchmark 문제에서 상대적으로 낮은 차원의 경우 (DV = 2, 5), SGHSA는 다른 알고리즘과 비교하였을 때, 우수한 성능을 보였다. 특히, 최적화 문제가 다수의 지역해를 가지는 (Many Local Optimal) 형태일 경우 이와 같은 특성이 확인히 나타났다.

Table 2. Mathematical benchmarks problems

| Name<br>(Characteristic)                   | formulation  | Dimension                | Search domain         | Global optimum |
|--|--|--------------------------|-----------------------|----------------|
| Sphere function<br>(Bowl-shaped)           | $\text{Min } f(x) = \sum_{i=1}^n x_i^2$  | 2, 5, 10, 30, 50,<br>100 | $[-\infty, \infty]^n$ | 0              |
| Rosenbrock function<br>(Valley-shaped)     | $\text{Min } f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$                                   |                          | $[-30, 30]^n$         | 0              |
| Rastrigin function<br>(Many local optimum) | $\text{Min } f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i)]$   |                          | $[-5.12, 5.12]^n$     | 0              |
| Griewank function<br>(Many local optimum)  | $\text{Min } f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ |                          | $[-600, 600]^n$       | 0              |

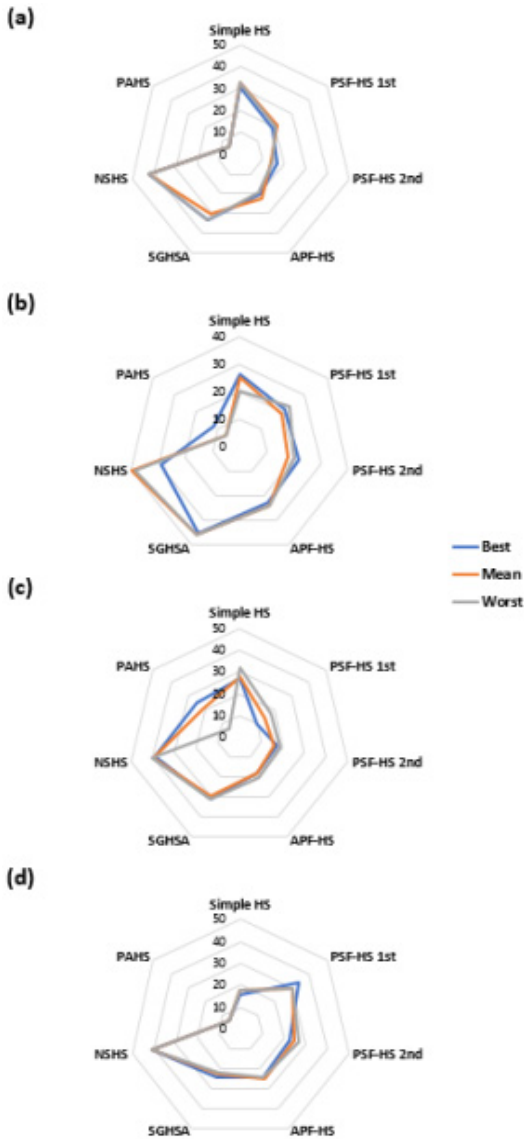


Fig. 1. Statistic optimization results of Mathematical problem (a) Sphere function, (b) Rosenbrock function, (c) Rastrigin function, (d) Griewank function

하지만, SGHSA는 결정변수의 수가 증가하는 경우에는 통계적으로 NSHS 보다는 좋지 않은 결과를 보인다. 또한 평균값을 기준으로 본다면, NSHS는 총 24가지 (4가지 문제 × 6가지 결정변수) 시행 중에서 20가지 시행에서 가장 우수한 성능을 보였다. 특히, 결정변수의 수가 많은 복잡한 문제의 경우 (DV = 30, 50, 100) 최적해 탐색의 성능 다른 6가지 알고리즘보다 우수하다는 것을 확

Table 3. Optimization results of Mathematical problem based on the number of decision variable (Best Error)

| # Decision variables       | 2                | 5                | 10               | 30               | 50               | 100              |
|----------------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <b>Sphere function</b>     |                  |                  |                  |                  |                  |                  |
| Simple HS                  | 6.43.E-11        | 1.21.E-07        | 5.08.E-07        | 2.38.E-03        | 1.39.E-02        | 4.75.E-02        |
| PSF-HS 1st                 | 1.27.E-11        | 7.49.E-06        | 5.29.E-04        | 1.82.E-02        | 3.97.E-02        | 1.00.E-01        |
| PSF-HS 2nd                 | 1.04.E-10        | 1.68.E-05        | 2.89.E-04        | 1.68.E-02        | 4.96.E-02        | 1.26.E-01        |
| APF-HS                     | 7.07.E-12        | 2.12.E-06        | 4.22.E-04        | 1.77.E-02        | 4.91.E-02        | 1.26.E-01        |
| SGHSA                      | 6.88.E-12        | 2.59.E-09        | 3.80.E-07        | 4.15.E-03        | 2.17.E-02        | 7.93.E-02        |
| NSHS                       | <b>1.14.E-15</b> | <b>1.72.E-10</b> | <b>3.64.E-09</b> | <b>4.30.E-08</b> | <b>1.69.E-07</b> | <b>7.80.E-07</b> |
| PAHS                       | 1.54.E-06        | 4.50.E-04        | 4.61.E-03        | 3.33.E-02        | 9.30.E-02        | 2.25.E-01        |
| <b>Rosenbrock function</b> |                  |                  |                  |                  |                  |                  |
| Simple HS                  | 2.03.E-08        | 6.49.E-06        | 8.21.E-06        | 1.77.E-02        | 4.60.E-02        | 1.42.E-01        |
| PSF-HS 1st                 | 4.13.E-10        | 1.72.E-05        | 1.46.E-03        | 3.37.E-02        | 8.46.E-02        | 2.62.E-01        |
| PSF-HS 2nd                 | 3.43.E-10        | 1.65.E-05        | 9.67.E-04        | 3.08.E-02        | 7.56.E-02        | 2.47.E-01        |
| APF-HS                     | 1.26.E-11        | 8.41.E-06        | 7.77.E-04        | 2.81.E-02        | 8.73.E-02        | 2.58.E-01        |
| SGHSA                      | <b>0</b>         | <b>0</b>         | 2.40.E-05        | 1.19.E-02        | 4.63.E-02        | 2.14.E-01        |
| NSHS                       | <b>0</b>         | <b>1.68.E-06</b> | <b>5.67.E-06</b> | <b>6.60.E-06</b> | <b>3.03.E-06</b> | <b>4.67.E-06</b> |
| PAHS                       | 1.67.E-04        | 5.86.E-05        | 1.87.E-04        | 3.77.E-04        | 1.58.E-06        | 8.25.E-04        |
| <b>Rastrigin function</b>  |                  |                  |                  |                  |                  |                  |
| Simple HS                  | 4.40.E-06        | 1.96.E-05        | 9.77.E-06        | 4.44.E-05        | 1.49.E-05        | 1.69.E-05        |
| PSF-HS 1st                 | 7.76.E-06        | 3.73.E-06        | 1.24.E-05        | 7.31.E-06        | 4.00.E-05        | 1.06.E-04        |
| PSF-HS 2nd                 | 5.97.E-06        | 2.25.E-05        | 1.89.E-05        | 1.41.E-06        | 2.39.E-05        | 3.81.E-05        |
| APF-HS                     | 2.37.E-07        | 1.13.E-05        | 4.01.E-06        | 5.42.E-05        | 8.23.E-05        | 8.29.E-05        |
| SGHSA                      | 5.02.E-06        | <b>1.47.E-09</b> | <b>2.26.E-07</b> | <b>6.90.E-06</b> | <b>7.07.E-06</b> | <b>3.12.E-06</b> |
| NSHS                       | <b>1.51.E-07</b> | 9.57.E-06        | 4.49.E-06        | 2.62.E-05        | 8.29.E-05        | 3.71.E-05        |
| PAHS                       | 1.67.E-04        | 5.86.E-05        | 1.87.E-04        | 3.77.E-04        | 1.58.E-06        | 8.25.E-04        |
| <b>Griewank function</b>   |                  |                  |                  |                  |                  |                  |
| Simple HS                  | 3.16.E-12        | 1.49.E-06        | 1.78.E-04        | 1.32.E-03        | 1.94.E-03        | 2.97.E-03        |
| PSF-HS 1st                 | 4.57.E-13        | 6.95.E-07        | 2.56.E-05        | 7.21.E-04        | 1.29.E-03        | 1.99.E-03        |
| PSF-HS 2nd                 | 6.69.E-11        | 5.09.E-06        | 3.52.E-05        | 7.62.E-04        | 1.31.E-03        | 2.08.E-03        |
| APF-HS                     | 4.54.E-12        | 1.14.E-06        | 4.43.E-05        | 7.24.E-04        | 1.47.E-03        | 2.22.E-03        |
| SGHSA                      | 1.49.E-14        | 3.28.E-07        | 9.29.E-05        | 1.41.E-03        | 1.89.E-03        | 2.86.E-03        |
| NSHS                       | <b>0</b>         | <b>1.82.E-11</b> | <b>3.49.E-10</b> | <b>2.46.E-09</b> | <b>4.48.E-09</b> | <b>1.08.E-08</b> |
| PAHS                       | 5.13.E-07        | 8.74.E-05        | 5.51.E-04        | 1.75.E-03        | 2.27.E-03        | 3.77.E-03        |

인할 수 있었다. 이러한 결과는 단목적 최적화 문제 (single objective optimization), 연속적 문제 (Continuous problem), 수학적 문제의 경우 NSHS의 해 탐색의 방법이 다른 자가적응형 화음탐색법에 비하여 적합함을 알 수 있다. 이는 NSHS가 솔루션의 품질에 따라 지역 탐색 확률을 높이는 매개변수를 설정함으로써 결정변수가 커질 경우에도 상대적으로 우수한 해를 안정적으로 찾을 수 있기 때문이라 판단된다.

수학적 최적화 문제의 적용에 이어, 대표적인 토목분야의 이산조합최적화 문제인 상수도 관경 최적설계 문제에 본 연구의 방법론을 적용하여 비교분석하였다.

상수도 관망은 인간의 활동에 필수적인 가장 기본적인

사회기반시설의 하나로 안전하고 신뢰할 수 있는 물 공급을 위해 필요한 요소이다. 상수도 관망의 주요 목적은 수원으로부터 수용가에 이르기까지 요구되는 유량을 적절한 수질과 수압을 만족시키며 공급하는 것이다. 일반적으로 상수도 관망의 구성에는 매우 큰 비용이 발생하며, 따라서 설계조건을 만족시키는 다양한 설계안 중 가장 낮은 비용을 나타내는 대안을 찾아내는 최적설계 과정이 필요하다. 이러한 이유로 본 연구에서는 Fig. 2와 같이 실제 상수도 관망 시스템인 S-city network를 적용하였다.



Fig. 2. Layout of S-city network

대상관망은 356개의 관로로 구성되어 있으며, Yoo et al. [38]에서 최초로 상수도 관망 최적설계의 대상으로 선택하였다. 본 연구에서 사용한 관 길이 당 비용 데이터는 Table 4와 같으며, 최적화를 위한 상업용 관경의 개수는 18개를 고려하였다. 총 356개의 관로로 구성된 S-city network의 최적설계에는 총 18356개의 가능해가 존재한다. 최적설계에 적용한 목적함수는 식 (14)와 같이 관경과 관로의 길이에 대한 함수로 구성된다. 수리학적 제한조건은 수리해석결과인 수압/유속조건을 사용하고, 절점의 최소압력수두(minimum nodal pressure)를 10m, 최대압력수두(maximum nodal pressure)를 35m으로 하였다.

Table 4. Commercial pipe diameters and costs

| Diameter (mm) | Construction costs (¥/m) | Maintenance costs (¥/m) | Material costs (¥/m) |
|---------------|--------------------------|-------------------------|----------------------|
| 80            | 65,000                   | 6,500                   | 15,000               |
| 100           | 65,999                   | 6,600                   | 27,583               |
| 150           | 76,410                   | 7,641                   | 40,686               |
| 200           | 86,028                   | 8,603                   | 58,716               |
| 250           | 96,135                   | 9,614                   | 81,160               |
| 300           | 105,325                  | 10,533                  | 103,231              |
| 350           | 113,818                  | 11,382                  | 125,107              |
| 400           | 126,797                  | 12,680                  | 148,836              |
| 450           | 136,250                  | 13,625                  | 155,522              |
| 500           | 147,792                  | 14,779                  | 181,823              |
| 600           | 171,991                  | 17,199                  | 211,396              |
| 700           | 211,413                  | 21,141                  | 273,528              |
| 800           | 307,640                  | 30,764                  | 339,740              |
| 900           | 359,048                  | 35,905                  | 384,619              |
| 1000          | 415,702                  | 41,570                  | 451,932              |
| 1100          | 482,074                  | 48,207                  | 547,224              |
| 1200          | 576,736                  | 57,674                  | 606,962              |
| 1350          | 687,390                  | 68,739                  | 716,075              |

유속조건으로는 최소유속(minimum pipe velocity)을 0.01m/sec, 최대유속(maximum pipe velocity)을 2.5m/sec로 설정하였으며, 이러한 수리학적 제약조건을 만족시키지 못하는 설계안에 대해서는 식 (15)와 같은 벌점함수를 적용하여 목적함수에 이를 합산, 해당 설계안이 도태되도록 한다.

$$Min. Cost = \sum_{i=1}^N (C_c(D_i) + C_M(D_i) + C_p(D_i))L_i + P_j \quad (14)$$

$$P_j = \alpha(h_{min} - h_j) + \beta, \text{ if } h_{min} > h_j \quad (15)$$

여기서,  $C_c$ ,  $C_M$ ,  $C_p$  는 관로  $i$ 의 단위길이(m)당 공사비, 유지관리비, 재료비의 비용함수를 의미하며,  $L_i$ 는 관로의 길이,  $D_i$ 는 관로  $i$ 의 관경,  $N$ 은 관로의 수를 의미하며,  $P_j$ 는 벌점함수를 의미한다.  $\alpha$ 와  $\beta$ 의 경우 벌점함수에 적용되는 상수를 의미하며 주어진 문제의 특성에 따라 제한조건을 위배하는 설계안을 도태시킬 수 있도록 충분한 값을 적용하며 본 연구에서는  $10^{10}$ 을 각각 적용하였다.

Table 5. Optimal design results and cost comparison of the initial plan

| Algorithm  | Design Cost    | Initial design cost (¥) | Optimal design cost (¥) | Variation (%) |
|------------|----------------|-------------------------|-------------------------|---------------|
| Simple HS  | 11,200,114,720 | 11,200,114,720          | 10,072,542,100          | 10.07         |
| PSF-HS 1st |                |                         | 10,068,009,456          | 10.11         |
| PSF-HS 2nd |                |                         | 10,066,397,849          | 10.12         |
| APF-HS     |                |                         | 10,006,768,400          | 10.65         |
| SGHSA      |                |                         | 9,975,644,245           | 10.93         |
| NSHS       |                |                         | 9,972,723,208           | 10.96         |
| PAHS       |                |                         | 9,986,321,140           | 10.84         |



앞선 방법론에서 제시한 6가지 알고리즘을 동일한 조건 하에서 성능을 평가하기 위해 50,000회의 목적함수 값 계산 횟수 (Number of function evaluations, NFES)를 종결조건으로 적용하였으며, 무작위과정을 통해 생성된 각 초기설계안을 동일하게 사용하여 최적화를 수행하였다. 또한 총 10회의 독립적인 최적설계안 탐색을 수행하여 그 평균 설계비용을 Table 5와 같이 나타내었다. 최적설계 수행 결과 기존설계안 (11,200,114,720 원)과 비교하였을 때, NSHS를 사용하여 최적설계를 진행한 경우 10.96% 감소된 비용 (9,972,723,208 원)으로 6가지 알고리즘의 중 최소 설계비용을 나타내었다. 이를 통해 앞선 수학적 benchmark 문제의 결과와 상수도관망 최적설계 문제의 결과를 비교해 보면 NSHS와 SGHSA는 연속적인 문제 (Continuous problem) 뿐만 아니라, 비연속 문제 (Discrete problem)에서도 우수한 성능을 확인할 수 있었다.

#### 4. 결론

본 연구에서는 최근 대표적인 최적화 알고리즘인 화음탐색법 중 매개변수 설정이 필요 없으며, 해 탐색 성능을 개선하는 자가적응형 화음탐색법을 비교하고, 그 성능을 검증하였다. 이를 위해 최적화 문제에서 많이 사용하는 수학적 벤치마크 문제와 상수도관망 최적설계에 적용하였으며, 공정한 비교 분석을 위해 동일한 초기 솔루션을 적용하였으며, 이를 정량적으로 평가하기 위해 다양한 시행을 통해 통계 분석을 수행하였다.

결과적으로 수학적 벤치마크 문제와 공학적인 문제에서 6가지 자가적응형 화음탐색 알고리즘 중 NSHS는 특히 search space가 큰 문제에서 좋은 해탐색 성능을 보였다. NSHS는 적합도의 표준편차를 사용하여 HM의 솔루션의 목적함수 값의 유사성이 높을 경우 Bw를 작은 값으로 조정해주며, PAR과정에서 높은 지역탐색에 비율을 유지하는 기능이 최적설계안을 탐색할 때 효과적인 것으로 판단된다. 이는 해탐색의 성능을 높일 뿐만 아니라, 화음탐색법의 3가지 매개변수 (HMCR, PAR, Bw)를 솔루션의 성능에 자동으로 설정하여 복잡한 매개변수 설정을 없앤다. 하지만, NSHS는 몇 가지 경계 조건 (BwUpper, BwLower)이 갖는 불확실성과 다양한 목적함수를 동시에 고려하는 다목적최적화 문제에서의 성능

이 본 연구의 성능과 유사하다고 판단할 수 없기 때문에 다양한 공학적 문제, 다목적최적화 문제에 적용 할 필요가 있다.

또한 본 연구를 통해 분석된 자가적응형 최적해 탐색 기법을 통해 화음탐색 알고리즘뿐만 아니라 다양한 알고리즘적용이 가능하며, 향후 연구에서는 이러한 매개변수 자동 보정기법의 특성을 고려하여 새로운 자가적응형 알고리즘을 개발하여 다양한 벤치 마크 문제 (연속성 및 이산 수학 문제, 실제 공학 문제)에 적용에 도움이 될 것이라 생각된다.

#### References

- [1] Z. W. Geem, Y. H. Cho, "Optimal design of water distribution networks using parameter-setting-free harmony search for two major parameters", *Journal of Water Resources Planning and Management* vol. 137, no. 4, pp. 377-380, 2010.  
DOI: [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000130](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000130)
- [2] K. Deb, H. G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover", *Evolutionary computation*, vol. 9, no. 2, pp. 197-221, 2001.  
DOI: <https://doi.org/10.1162/106365601750190406>
- [3] A. Ismail, A. P. Engelbrecht, "Self-Adaptive Particle Swarm Optimization", *Seal*, vol. 7673, 2012.  
DOI: [https://doi.org/10.1007/978-3-642-34859-4\\_23](https://doi.org/10.1007/978-3-642-34859-4_23)
- [4] M. G. Omran, A., Salman, A. P. Engelbrecht, "Self-adaptive differential evolution", *International Conference on Computational and Information Science. Springer, Berlin, Heidelberg*, 2005.
- [5] Z. W. Geem, J. H. Kim, G. V. Loganathan, "A new heuristic optimization algorithm: harmony search", *simulation*, vol. 76, no. 2, pp. 60-68, 2001
- [6] J. H. Kim, Z. W. Geem, E. S. Kim, "Parameter estimation of the nonlinear Muskingum model using harmony search", *JAWRA Journal of the American Water Resources Association*, vol. 37, no. 5, pp. 1131-1138, 2001.  
DOI: <https://doi.org/10.1111/j.1752-1688.2001.tb03627.x>
- [7] Glover, F. "Heuristics for integer programming using surrogate constraints." *Decision Sciences* 8.1 (1977): 156-166.  
DOI: <https://doi.org/10.1111/j.1540-5915.1977.tb01074.x>
- [8] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing", *science* vol. 220, no. 4598, pp. 671-680, 1983.
- [9] M. Dorigo, "Optimization, learning and natural algorithms", Ph. D. Thesis, Politecnico di Milano, Italy, 1992.
- [10] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", *Micro Machine and Human Science, Proceedings of the Sixth International*

- Symposium on IEEE*, 1995.  
DOI: <https://doi.org/10.1109/MHS.1995.494215>
- [11] M. Mahdavi, M. Fesanghary, E. Damangir, "An improved harmony search algorithm for solving optimization problems", *Applied mathematics and computation*, vol. 188, no. 2, pp. 1567-1579 2007.  
DOI: <https://doi.org/10.1016/j.amc.2006.11.033>
- [12] M. G. Omran, M. Mahdavi, "Global-best harmony search", *Applied mathematics and computation*, vol. 198, no. 2, pp. 643-656, 2008.  
DOI: <https://doi.org/10.1016/j.amc.2007.09.004>
- [13] Q. K. Pan, P. N. Suganthan, M. F. Tasgetiren, J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems", *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830-848, 2010.  
DOI: <https://doi.org/10.1016/j.amc.2010.01.088>
- [14] C. M. Wang, Y. F. Huang, "Self-adaptive harmony search algorithm for optimization", *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826-2837, 2010.  
DOI: <https://doi.org/10.1016/j.eswa.2009.09.008>
- [15] S. O. Degertekin, "Improved harmony search algorithms for sizing optimization of truss structures", *Computers and structures*, vol. 92, pp. 229-241, 2012.  
DOI: <https://doi.org/10.1016/j.compstruc.2011.10.022>
- [16] J. Chen, H. F. Man, Y. M. Wang, "Novel Self-adaptive Harmony Search Algorithm for continuous optimization problems", *Control Conference (CCC), 2011 30th Chinese. IEEE*, 2011.
- [17] K. Luo, "A novel self-adaptive harmony search algorithm", *Journal of Applied Mathematics*, vol. 2013, 2013.  
DOI: <http://dx.doi.org/10.1155/2013/653749>
- [18] Z. W. Geem, "Parameter estimation of the nonlinear Muskingum model using parameter-setting-free harmony search", *Journal of Hydrologic Engineering*, vol. 16, no. 8, pp. 684-688, 2010.  
DOI: [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0000352](https://doi.org/10.1061/(ASCE)HE.1943-5584.0000352)
- [19] S. Jiang, Y. Zhang, P. Wang, M. Zheng, "An almost-parameter-free harmony search algorithm for groundwater pollution source identification", *Water Science and Technology*, vol. 68, no. 11, pp. 2359-2366, 2013.  
DOI: <https://doi.org/10.2166/wst.2013.499>
- [20] Z. W. Geem, K. B. Sim, "Parameter-setting-free harmony search algorithm", *Applied Mathematics and Computation*, vol. 217, no. 8, pp. 3881-3889, 2010.  
DOI: <https://doi.org/10.1016/j.amc.2010.09.049>
- [21] Z. W. Geem, "Economic dispatch using parameter-setting-free harmony search", *Journal of Applied Mathematics*, vol. 2013, 2013.
- [22] S. Kulluk, L. Ozbakir, A. Baykasoglu, "Self-adaptive global best harmony search algorithm for training neural networks", *Procedia Computer Science* vol. 3, pp. 282-286, 2011.  
DOI: <https://doi.org/10.1016/j.procs.2010.12.048>
- [23] A. Kattan, A. Rosni, "A dynamic self-adaptive harmony search algorithm for continuous optimization problems", *Applied Mathematics and Computation*, vol. 219, no. 16, pp. 8542-8567, 2013.  
DOI: <https://doi.org/10.1016/j.amc.2013.02.074>
- [24] D. S. Rani, N. Subrahmanyam, M. Sydulu, "Self adaptive harmony search algorithm for optimal capacitor placement on radial distribution systems", *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on. IEEE*, 2013.  
DOI: <https://doi.org/10.1109/ICEETS.2013.6533580>
- [25] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, M. Fei, "An improved adaptive binary harmony search algorithm", *Information Sciences*, vol. 232, pp. 58-87, 2013.  
DOI: <https://doi.org/10.1016/j.ins.2012.12.043>
- [26] B. Naik, J. Nayak, H. S. Behera, A. Abraham, "A self adaptive harmony search based functional link higher order ANN for non-linear data classification", *Neurocomputing*, vol. 179, pp. 69-87, 2016.  
DOI: <https://doi.org/10.1016/j.neucom.2015.11.051>
- [27] A. Rajagopalan, V. Sengoden, R. Govindasamy, "Solving economic load dispatch problems using chaotic self adaptive differential harmony search algorithm", *International Transactions on Electrical Energy Systems*, vol. 25, no. 5, pp. 845-858, 2015.  
DOI: <https://doi.org/10.1002/etep.1877>
- [28] M. Shivaie, M. T. Ameli, M. S. Sepasian, P. D. Weinsier, V. Vahidinasab, "A multistage framework for reliability-based distribution expansion planning considering distributed generations by a self-adaptive global-based harmony search algorithm", *Reliability Engineering & System Safety*, vol. 139, pp. 68-81, 2015.  
DOI: <https://doi.org/10.1016/j.res.2015.03.001>
- [29] M. Z. Vahid, M. O. Sadegh, "A new method to reduce losses in distribution networks using system reconfiguration with distributed generations using self-adaptive harmony search algorithm", *Fuzzy and Intelligent Systems (CFIS), 2015 4th Iranian Joint Congress on. IEEE*, 2015.  
DOI: <https://doi.org/10.1109/CFIS.2015.7391655>
- [30] H. H. Yan, J. H. Duan, B. Zhang, Q. K. Pan, "Harmony search algorithm with self-adaptive dynamic parameters", *Control and Decision Conference (CCDC), 2015 27th Chinese. IEEE*, 2015.  
DOI: <https://doi.org/10.1109/CCDC.2015.7162104>
- [31] F. Zhao, Y. Liu, C. Zhang, J. Wang, "A self-adaptive harmony PSO search algorithm and its performance analysis", *Expert Systems with Applications*, vol. 42, no. 21, pp. 7436-7455, 2015.  
DOI: <https://doi.org/10.1016/j.eswa.2015.05.035>
- [32] V. Kumar, J. K. Chhabra, D. Kumar, "Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems", *Journal of Computational Science*, vol. 5, no. 2, pp. 144-155, 2014.  
DOI: <https://doi.org/10.1016/j.jocs.2013.12.001>
- [33] X. Dai, X. Yuan, Z. Zhang, "A self-adaptive multi-objective harmony search algorithm based on harmony memory variance", *Applied Soft Computing*, vol. 35, pp. 541-557, 2015.  
DOI: <https://doi.org/10.1016/j.asoc.2015.06.027>
- [34] P. Sabarinath, M. R. Thansekhar, R. Saravanan, "Multiobjective optimization method based on adaptive parameter harmony search algorithm", *Journal of Applied Mathematics*, vol. 2015, 2015.
- [35] L. A. Rastrigin, "Systems of extremal control. Theoretical

Foundations of Engineering Cybernetics Series", *Nauka, Moscow*, 1974.

- [36] L. C. W. Dixon, "The global optimization problem: an introduction", *Towards Global Optimiation* vol. 2, pp. 1-15, 1978.
- [37] M. Molga, C. Smutnicki, "Test functions for optimization needs", *Test functions for optimization needs*, 2005.
- [38] D. G. Yoo, H. M. Lee, A. Sadollah, J. H. Kim, "Optimal pipe size design for looped irrigation water supply system using harmony search: Saemangeum project area", *The Scientific World Journal*, vol. 2015, 2015. DOI: <http://dx.doi.org/10.1155/2015/651763>

**최영환(Choi Young Hwan) [정회원]**



- 2010년 2월 : 한양대학교 건설환경시스템공학과 (공학사)
- 2014년 3월 ~ 현재 : 고려대학교 건축사회환경공학과 수자원공학전공 석박사통합과정

<관심분야>  
수자원공학, 수자원시스템공학, 최적화 알고리즘

**이호민(Lee Ho Min) [정회원]**



- 2012년 2월 : 고려대학교 사회환경시스템공학과 (공학사)
- 2012년 3월 ~ 2017년 8월 : 고려대학교 건축사회환경공학과 수자원공학 (공학박사)

<관심분야>  
수자원공학, 수자원시스템공학, 최적화 알고리즘

**유도근(Yoo Do Guen) [정회원]**



- 2007년 2월 : 고려대학교 사회환경시스템공학과 (공학사)
- 2009년 2월 : 고려대학교 건축사회환경공학과 수자원공학 (공학석사)
- 2013년 2월 : 고려대학교 건축사회환경공학과 수자원공학 (공학박사)
- 2017년 9월 : 수원대학교 건설환경공학 교수

<관심분야>  
수자원공학, 수자원시스템공학, 최적화 알고리즘

**김중훈(Kim Joong Hoon) [정회원]**



- 1984년 2월 : 고려대학교 토목공학과 (공학사)
- 1986년 12월 : Virginia Polytechnic Institute & State Univ. (공학석사)
- 1992년 2월 : Univ. of Texas at Austin (공학박사)
- 1993년 3월 ~ 현재 : 고려대학교 건축사회환경공학부 교수

<관심분야>  
수문학, 수리학, 수자원공학