

새로운 메타 휴리스틱 최적화 알고리즘의 개발: Exponential Bandwidth Harmony Search with Centralized Global Search

김영남, 이의훈*
충북대학교 토목공학과

Development of the Meta-heuristic Optimization Algorithm: Exponential Bandwidth Harmony Search with Centralized Global Search

Young Nam Kim, Eui Hoon Lee*
Division of Civil Engineering, Chungbuk National University

요약 본 연구에서는 기존의 Harmony Search(HS)의 성능을 강화한 Exponential Bandwidth Harmony Search with Centralized Global Search(EBHS-CGS)를 개발하였다. EBHS-CGS는 HS의 성능 강화를 위해 총 두 가지 방법을 추가하였다. 첫 번째 방법은 지역탐색을 강화하기 위한 Bandwidth(bw) 개량방안이다. 이 방법은 기존 bw를 지수형태의 bw로 대체하여 적용함으로써 반복시산이 진행되면서 bw값을 줄인다. 이러한 형태의 bw는 정밀한 지역탐색을 가능하고, 이를 통해 알고리즘은 더욱 정밀한 값을 구할 수 있다. 두 번째 방법은 효과적인 전역탐색을 위한 탐색범위 축소이다. 이 방법은 Harmony Memory(HM) 내에서 가장 좋은 결정변수를 고려하여 탐색범위를 축소한다. 이를 Centralized Global Search(CGS)라 하며, 이 과정은 새로운 매개변수 Centralized Global Search Rate(CGSR)에 의해 HS의 전역탐색과는 별도로 진행된다. 축소된 탐색범위는 효과적인 전역탐색을 가능하게 하며, 이를 통해 알고리즘의 성능이 향상된다. EBHS-CGS를 대표적인 최적화 문제(수학 및 공학 분야)에 적용하고, 그 결과를 HS와 Improved Harmony Search(IHS)와 비교하여 제시하였다.

Abstract An Exponential Bandwidth Harmony Search with Centralized Global Search (EBHS-CGS) was developed to enhance the performance of a Harmony Search (HS). EBHS-CGS added two methods to improve the performance of HS. The first method is an improvement of bandwidth (bw) that enhances the local search. This method replaces the existing bw with an exponential bw and reduces the bw value as the iteration proceeds. This form of bw allows for an accurate local search, which enables the algorithm to obtain more accurate values. The second method is to reduce the search range for an efficient global search. This method reduces the search space by considering the best decision variable in Harmony Memory (HM). This process is carried out separately from the global search of the HS by the new parameter, Centralized Global Search Rate (CGSR). The reduced search space enables an effective global search, which improves the performance of the algorithm. The proposed algorithm was applied to a representative optimization problem (math and engineering), and the results of the application were compared with the HS and better Improved Harmony Search (IHS).

Keywords : Harmony Search, Exponential Bandwidth, Harmony Memory, Centralized Global Search, Optimization

본 논문은 2019년도 정부(교육부)의 재원을 한국 연구재단의 지원을 받아 수행된 연구임 (NRF-2019R1I1A3A01059929)

*Corresponding Author : Eui Hoon Lee(Chungbuk National Univ.)

email: hydrohydro@cbnu.ac.kr

Received October 21, 2019

Revised December 23, 2019

Accepted February 7, 2020

Published February 29, 2020

1. 서론

과거의 최적화 기법은 수학적 방법인 Linear Programming(LP), Non-Linear Programming(NLP) 및 Dynamic Programming(DP)을 사용하여 복잡한 문제를 해결하였다. 그러나 이러한 수학적 방법은 단순하여 복잡한 자연현상 문제를 해결하기 어려웠다. 따라서 복잡한 자연현상 문제를 해결하기 위해서는 단순화하는 과정이 필요했고, 이로 인해 정확성은 떨어지게 되었다. 이러한 한계점을 극복하기 위해 발견적 탐색법(heuristic method)이 제안되었다. 대표적인 발견적 탐색법은 유전자알고리즘(GA: Genetic Algorithm), 개미군집최적화(ACO: Ant Colony Optimization), 입자군집최적화(PSO: Particle Swam Optimization), 그리고 화음탐색법(HS: Harmony Search) 등이 있다[1-4].

앞서 언급한 알고리즘 중 HS는 최적화 과정이 간단하여 많이 사용되어 왔으며, 다양한 공학문제에 적용하기 위하여 알고리즘의 개량에 관한 많은 연구가 진행되었다. HS를 개량한 대표적인 알고리즘으로는 Improved Harmony Search(IHS), Global-best Harmony Search(GHS), 및 Self-adaptive harmony search 등이 있다[5-7].

알고리즘의 반복시산 과정 후반에는 세부적인 탐색과정이 필요하여 지역탐색이 중요해진다. 그러므로 최종적인 결과는 지역탐색으로 결정되는 경우가 많으며 지역해가 다수 존재하는 경우 지역해에 빠지기 쉽다. 이러한 문제는 HS에서도 발생하며 이를 해결하기 위해 여러 가지 방법이 제안되었다.

Random Distribution Harmony Search(RDHS)는 새로운 매개변수 Assign을 통해 무작위 분산을 수행하는 알고리즘이다. 새로운 계산과정인 무작위 분산은 기존 HS의 Harmony Memory Considering 과정에 추가되었다. 무작위 분산은 HS 진행 중 Harmony Memory(HM) 내에서 선택된 결정변수(A)와 HM 내에서 무작위로 선택된 결정변수(B)를 이용한 새로운 계산과정이다. 무작위 분산의 계산과정은 다음과 같다. 먼저 B에 0과 1 사이의 무작위 숫자(R)를 곱하여[BxR] A에 더해준다(A-1). 만일 A-1이 탐색범위를 넘어가게 되면 A-1이 넘어간 탐색범위의 상한 또는 하한값으로 교체한다. 그리고 A-1이 탐색범위를 넘어간 만큼을 B에서 빼준다. 반대로 A-1이 탐색범위를 넘어가지 않으면 B에 (1-R)을 곱한 값 [Bx(1-R)]을 B에서 빼준다. RDHS는 무작위 분산을 통한 전역탐색 능력 강화로 최적해에 근사한 값을 찾으나

정확한 최적해를 찾지 못하였다[8].

Dynamical Search Space Harmony Search(DSHS)는 탐색범위 축소(Shrinking the search space)와 보조(Ancillary) HM을 기존 HS에 추가하였다. DSHS는 기존 HS 계산과정을 주탐색(Main search)로 정의하고 탐색범위 축소를 보조탐색(Ancillary search)로 정의하였다. 탐색범위 축소는 초기의 결정변수와 현재 결정변수를 이용하여 탐색범위를 줄인다. 탐색범위 축소를 실행하기 위한 매개변수는 전체 HM에서 같은 값을 가지는 결정변수의 비율(α)이다.

반복시산 초반에 보조 HM은 기존 HM(Main HM)과 같은 결정변수를 가지고 있다. 하지만 DSHS에서 두 HM은 서로 다른 계산과정을 거치게 된다. 그 결과 보조 HM과 기존 HM은 서로 다른 결정변수를 가지게 된다. 만약 보조 HM의 결정변수와 기존 HM의 가장 좋지 않은 결정변수를 비교하였을 때 보조 HM의 결정변수가 더 좋다면 기존 HM의 결정변수를 보조 HM의 해당 결정변수로 교체하게 된다. 다만 DSHS에서는 각 반복시산마다 여러 개의 해를 생성하게 되므로 연산속도에 대한 정확한 비교가 어렵다[9].

본 연구는 HS의 문제점을 해결하고 성능을 개선하기 위해 지역탐색 및 전역탐색을 강화하였다. 본 연구에서 제안된 방법은 다음과 같다. 먼저 지역탐색을 강화하기 위해 제안된 방법은 HS에서 지역탐색의 범위에 해당하는 Bandwidth(bw)를 반복시산이 진행되면서 줄어드는 것이며, 이는 더 정밀한 지역탐색이 가능하도록 하였다. 다음으로 전역탐색을 강화하기 위해 제안된 방법은 HM 내에서 가장 좋은 결정변수를 고려하여 탐색범위를 줄이는 것이다. 이를 통해 기존 보다 효과적인 전역탐색이 가능했다. 앞서 언급한 두 가지 방법이 적용된 새로운 알고리즘을 기존 HS 및 IHS와 비교하여 알고리즘의 개선효과를 확인하였다.

2. 연구 방법

2.1 Harmony Search(HS)

HS는 Geem 등(2001)이 음악연주자들의 즉흥연주를 모방하여 제안한 발견적 탐색법이다. HS의 매개변수는 Harmony Memory Size(HMS), Harmony Memory Considering Rate(HMCR), Pitch Adjusting Rate(PAR), 그리고 bw가 있다.

HMS는 HM의 크기를 뜻한다. HMS에 따라 초기 무작위로 생성되는 해의 개수가 결정된다. HMCR은 HM 내에서 결정변수를 선택하는 확률이다. 기존 HM 내의 결정변수를 무작위로 조합하여 새로운 결정변수를 만들어내는 점은 GA의 연산과정과 비슷하다. 하지만 GA와 차이점이 존재하는데, 주요 차이점은 다음과 같다. GA는 반드시 2개의 결정변수에서 새로운 결정변수를 만들어낸다. 하지만 HS는 HMS만큼, 즉 2개 이상의 결정변수에서 새로운 결정변수를 만들어 낼 수 있다. 또 다른 점은 HS에서는 새로운 해를 생성하는 동안 각 결정변수를 독립적으로 선택할 수 있지만, GA에서는 유전자 구조를 유지해야 하므로 결정변수를 독립적으로 고려할 수 없다.

PAR은 HMCR을 통해 선택된 결정변수를 일정한 bw 만큼 조절하여 선택된 결정변수와 인접한 결정변수를 선택하는 확률이다. Fig. 1은 HS의 계산과정에 대한 간단한 모식도이다.

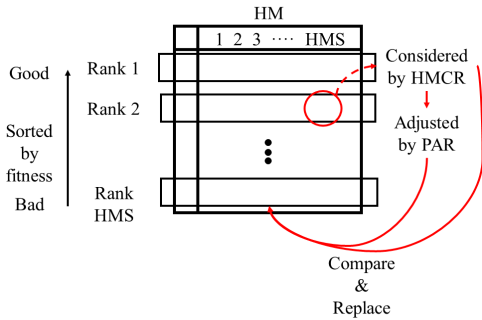


Fig. 1. Structure of HS

HS의 자세한 계산과정은 다음과 같은 절차로 진행된다.

- [step 1] HMS에 따라 초기 HM을 생성한다.
- [step 2] HM을 목적함수 값에 따라 정렬한다.
- [step 3] 0과 1 사이의 무작위 숫자(R)를 생성한다.
- [step 4-1] R이 HMCR보다 작으면 기존 HM 내에서 결정변수를 무작위로 선택한다.(A)
- [step 4-2] 새로운 R을 생성하고 PAR보다 작으면 [step 4-1]에서 선택된 결정변수를 bw를 적용하여 조정한다.(B)
- [step 4-3] R이 HMCR보다 크면 무작위로 결정변수를 생성한다.(C)
- [step 5] A, B 또는 C에서 생성된 결정변수로 새로운 해를 만든다.

[step 6] [step 5]에서 생성된 새로운 해를 HM 내에서 가장 좋지 않은 해와 비교하여 새로운 해가 더 좋으면 교체한다.

[step 7] 정지조건을 만족할 때까지 [step 1]~[step 6]의 과정을 반복한다.

2.2 Improved Harmony Search(IHS)

IHS는 Mahdavi 등(2007)이 제안한 개량된 HS이다. IHS는 HS의 지역탐색을 담당하는 두 가지 매개변수 PAR과 bw가 시산 횟수에 따라 변하도록 하여 성능을 개량하였다.

알고리즘은 반복시산 후반부에서 전역탐색보다 지역탐색의 영향을 더 크게 받는다. 그러므로 반복시산 후반부로 갈수록 지역탐색 확률을 증가시키면 알고리즘은 더 좋은 결과를 찾을 수 있다. 이를 위해 Mahdavi 등(2007)은 HS의 지역탐색 확률인 PAR이 시산 횟수에 따라 커지도록 하여 성능을 개선하였다. IHS에서는 Eq. (1)과 같이 PAR이 증가하도록 하였다.

$$PAR = PAR_{min} + \left(\frac{PAR_{max} - PAR_{min}}{N} \right) * k \quad (1)$$

여기서 N은 최대 반복시산 횟수, k는 현재 반복시산 횟수, PAR_{max} 는 최대 PAR, PAR_{min} 는 최소 PAR이다.

반복시산이 진행되면서 HM 내의 결정변수들은 목적함수 값이 가장 좋은 결정변수로 수렴하게 되어 결정변수 간의 차이가 줄어들는다. 따라서 정밀한 지역탐색을 하기 위해서는 반복시산이 진행됨에 따라 지역탐색의 범위를 줄이는 것이 유리하다. 그러므로 HS의 전역탐색범위에 해당하는 bw값을 시산 횟수에 따라 감소하도록 구성하면 더 정밀한 지역탐색이 가능하다. 이를 위해 Mahdavi 등(2007)은 Eqs. (2)-(3)과 같은 형태의 bw를 제안하여 시산 횟수에 따라 bw의 값이 감소하도록 하였다.

$$bw = bw_{max} + \text{Exp}(c*k) \quad (2)$$

$$c = \ln\left(\frac{bw_{min}}{bw_{max}}\right) / N \quad (3)$$

여기서 N은 최대 반복시산 횟수, k는 현재 반복시산 횟수, bw_{max} 는 최대 bw, bw_{min} 는 최소 bw이다.

2.3 Exponential Bandwidth Harmony Search with Centralized Global Search(EBHS-CGS)

IHS는 HS의 성능개선에 성공하였으나 지역해가 많은 경우 발생하는 문제를 해결하지는 못하였다. 그 이유는 IHS의 구조상 반복시산의 후반에 실행되는 계산과정이 거의 지역탐색이므로 지역해에 빠지면 이를 벗어나기 힘들기 때문이다. 본 연구는 이러한 문제점을 해결하기 위해 HS에 새로운 전역탐색을 추가하였고, 성능개선을 위한 bw의 형태도 제안하였다.

2.3.1 Exponential Bandwidth

앞서 언급한 바와 같이 정밀한 지역탐색을 하기 위해서는 지역탐색의 범위가 시산 횟수에 따라 줄어들어야 한다. 따라서 본 연구는 HS의 지역 탐색범위에 해당하는 bw의 새로운 형태를 제안한다. 제안된 bw의 형태는 Eq. (4)와 같으며 이를 지수 bw(Exponential bw)라 한다. 지수 bw는 음의 지수 형태로 반복시산의 후반부에는 매우 작은 값을 가진다. 따라서 HM 내의 결정변수 간의 차이가 거의 없는 후반부에도 지역탐색으로 결정변수를 조정할 수 있으며 이를 통해 정밀한 지역탐색이 가능하다.

$$bw = \text{EXP}\left(\frac{-k}{N \times \text{PAR} \times \text{HMCR}}\right) \quad (4)$$

여기서 N은 최대 반복시산 횟수, k는 현재 반복시산 횟수이다.

결정변수의 탐색범위를 고려하기 위해 Eq. (4)에 탐색 범위를 곱해준 Eq. (5)로 수정하여 적용한다.

$$bw = (xub - xlb) \times \text{Exp}\left(\frac{-k}{N \times \text{PAR} \times \text{HMCR}}\right) \quad (5)$$

여기서 N은 최대 반복시산 횟수, k는 현재 반복시산 횟수, xub는 결정변수 x의 탐색범위 상한, xlb는 결정변수 x의 탐색범위 하한이다.

Fig. 2에 따르면 약 20,000번의 시산 횟수까지 bw 값은 급격히 줄어들고 이후는 매우 미세하게 줄어든다. 매우 미세하게 줄어든 bw는 후반부에 더욱 정밀한 지역 탐색을 가능하게 한다. Fig. 2는 PAR이 0.05, HMCR이 0.95이며 최대 시산 횟수가 100,000번인 경우 Eq. (5)의 수렴그래프를 나타낸 것이다.

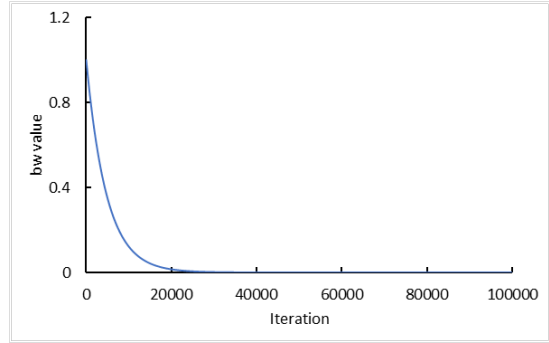


Fig. 2. Conversion shape of bw

2.3.2 Centralized Global Search

지역해가 많은 경우 알고리즘은 최적해를 구하기 힘들다. 따라서 이 문제를 해결하기 위해 전역탐색을 보다 효율적으로 할 필요가 있으며 이를 위해 탐색범위를 축소하였다. 탐색범위가 작아지면 기존 보다 더 효과적인 전역탐색이 가능해진다. 따라서 본 연구에서는 효과적인 전역탐색을 위해 최적 결정변수를 고려한 탐색범위의 축소 방법을 제안하였다. 본 연구에서 제안하는 탐색범위 축소 과정은 다음과 같다.

- [step 1] HM 내의 최적 결정변수와 탐색범위의 중간 값 간의 거리를 구한다.
- [step 2] 구한 거리만큼 중간값에서 빼주어 최적 결정 변수의 반대 값을 생성한다.
- [step 3] 최적 결정변수와 그 반대 값을 탐색범위의 상한과 하한으로 설정한다.
- [step 4] 새로운 탐색범위로 전역탐색을 실시한다.
- [step 5] 기존 최적해와 비교 및 교환한다.

위와 같은 과정을 Centralized Global Search (CGS)라고 하며, CGS는 새로운 매개변수 Centralized Global Search Rate(CGSR)를 통해 실행된다. CGSR은 0과 1 사이의 값을 가지는 실수이다. CGS는 반복시산이 진행되면서 탐색범위가 중앙값을 기준으로 축소되는 특징을 가지고 있다. 이러한 특징은 CGS의 계산과정이 탐색범위의 중간값에서 현재 최적 결정변수까지의 거리를 통해 새로운 탐색범위를 설정하기 때문에 나타난다. 좁아진 탐색범위로 인해 효과적인 전역탐색이 가능하며 지역해에 빠지는 것을 방지한다. Fig. 3은 CGS의 구조를 간단하게 표현한 것이다.

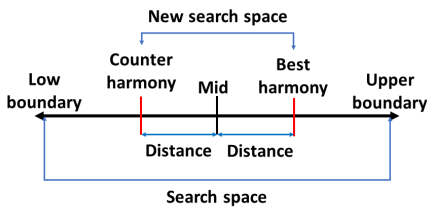


Fig. 3. The concept of changing search space

본 연구는 CGS와 지수 bw를 결합하여 HS의 전반적인 성능 개량을 하였다. 지수 bw를 통하여 정밀한 지역 탐색이 가능하며, 알고리즘이 최적해에 가까운 값을 찾거나 최적해를 찾도록 하였다. 만일 알고리즘이 지역해에 빠지게 되면 CGS를 통한 효과적인 전역탐색으로 알고리즘이 지역해를 빠져나오도록 한다. 제안된 방법을 결합한 EBHS-CGS의 의사코드는 Fig. 4와 같다.

```

Generate initial HM
Sorting HM by objective function
R = rnd(1)

If R bigger than CGSR
R = rnd(1)
If R smaller than HMCR
Random Select harmony in HM
R = rnd(1)
If R smaller than PAR
Adjusting harmony using Exponential bw
end if
Or
Generate new harmony in search space
end if
Or
Reset search space by CGS
Generate new harmony
end if

Compare harmony and replace harmony

Continue algorithm until satisfying stop condition
    
```

Fig. 4. Pseudocode of EBHS-CGS

Fig. 5를 보면 CGS는 기존 HS와 별도로 운영된다. 만일 CGS가 HS의 기존 전역탐색을 대체한다면 전체 탐색 범위에 대한 전역탐색이 실행되지 않는다. 따라서 CGS는 HS와 별도로 진행되어야 한다. Fig. 5는 EBHS-CGS의 탐색과정을 나타낸 흐름도이며 P는 계산 과정 중 매번 무작위로 생성되는 0과 1 사이의 숫자이다.

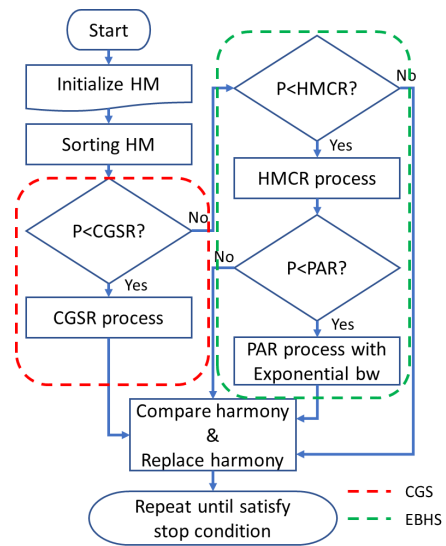


Fig. 5. Flowchart of EBHS-CGS

3. 연구 결과

EBHS-CGS의 성능을 검토하기 위해 결정변수가 2개인 수학문제 2개, 결정변수가 30개인 수학문제 2개, 그리고 공학문제 1개에 적용하였다. 각 문제의 최적해와 변수개수는 Table 1과 같다.

Table 1. Specification of benchmark problems

Benchmark problems	Optimal value	Number of decision variable
Easom	-1	2
Bartels Conn	1	2
Rastrigin	0	30
Griewank	0	30
Three-bar truss design	-	2

최적값(Optimal value)이 정해져 있는 수학문제에서는 알고리즘이 구한 결과값과 최적값의 오차(Error)로 결과를 산출하였다. 오차가 0에 가까울수록 정해진 최적값에 가까운 값을 찾는 것을 의미한다. 따라서 수학문제의 경우 결과 산출을 오차값으로 하였다. 반면 공학문제의 경우 정해진 최적값이 없기 때문에 알고리즘이 구한 결과값으로 결과를 산출하였다.

3.1 2변수 수학적문제

먼저 결정변수가 2개인 수학적문제에 EBHS-CGS를 적용하여 성능을 검증하여 보았다. 본 연구는 HS와 IHS 역시 같은 문제에 적용하여 EBHS-CGS와 결과를 비교하였다.

3.1.1 Easom

Eq. (6)은 Easom의 식을 나타낸 것이고 Table 2는 Easom에 적용한 알고리즘의 매개변수를 나타낸다. Easom은 최대 시산 횟수를 50,000번으로 하여 100번 반복하였다.

$$\begin{aligned} \text{Min } f(x) = & -\cos(x_1)\cos(x_2) \times \\ & \text{Exp}(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \\ \text{Subject to } & -100 < x_1 < 100 \\ & -100 < x_2 < 100 \end{aligned} \quad (6)$$

여기서 x_1 과 x_2 는 Easom의 결정변수이다.

Table 2. General setting for the simulation(Easom)

Parameter	HS	IHS	EBHS-CGS
HMS	30	30	30
HMCR	0.8	0.8	0.8
PAR	0.05	-	0.05
maxPAR	-	0.99	-
minPAR	-	0.05	-
maxbw	-	0.001	-
minbw	-	1.0E-05	-
CGSR	-	-	0.01

Table 3은 Easom에서 HS, IHS와 EBHS-CGS의 적용결과를 나타낸다. 적용결과를 살펴보면 가장 좋은 값은 HS를 제외한 나머지 IHS와 EBHS-CGS에서는 최적값을 찾아냈다. 하지만 IHS의 표준편차(SD: Standard Deviation)가 HS보다 크게 나오므로 최적값을 찾았다고 볼 수 없다. 반면 EBHS-CGS는 표준편차가 0으로 모두 일정하게 최적값을 찾아낸다고 할 수 있다. 따라서 EBHS-CGS는 Easom에서 안정적으로 최적값을 찾는 것을 확인할 수 있다.

Table 3. Comparison of Error for application of Easom

	HS	IHS	EBHS-CGS
Mean Error	0.297685	0.093226	0
Best Error	1E-07	0	0
Worst Error	0.832252	0.999996	0
SD	0.231074	0.254809	0

Fig. 6은 Easom에서 각 알고리즘의 변화곡선을 나타낸 것이다. Fig. 6에 따르면 초반부의 EBHS-CGS는 거의 일정한 값을 가지다가 10,000번의 반복시산 이후 급격히 감소하여 오차값이 0이 된다. IHS는 10,000번의 반복시산 이내에 오차값이 급격한 감소하지만, 최적값에 수렴하지 못하였다. HS의 경우 지속적으로 값이 줄어 들기는 하지만 최적값에 수렴하지 못하였다. 이를 통해 EBHS-CGS는 IHS보다 수렴속도는 느리지만, 결과는 더 좋은 것을 확인할 수 있다.

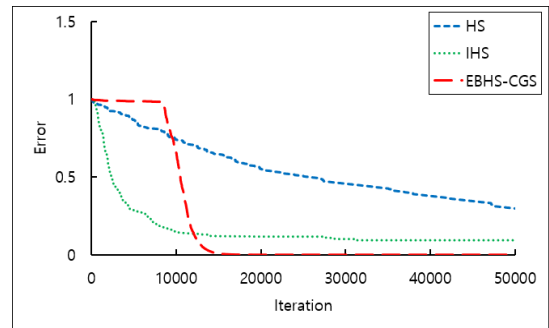


Fig. 6. Conversion shape for Easom

3.1.2 Bartels Conn

Eq. (7)은 Baetels Conn이다. Table 4는 Baetels Conn에 적용한 알고리즘의 매개변수를 나타낸다. Bartels Conn은 최대 시산 횟수를 50,000번으로 하며, 100번 반복하였다.

$$\begin{aligned} \text{Min } f(x) = & |x_1^2 + x_2^2 + x_1x_2| \\ & + |\sin(x_1)| + |\cos(x_2)| \\ \text{Subject to } & -500 < x_1 < 500 \\ & -500 < x_2 < 500 \end{aligned} \quad (7)$$

여기서 x_1 과 x_2 는 Bartels Conn의 결정변수이다.

Table 4. General setting for the simulation(Bartels Conn)

Parameter	HS	IHS	EBHS-CGS
HMS	30	30	30
HMCR	0.95	0.95	0.95
PAR	0.05	-	0.05
maxPAR	-	0.99	-
minPAR	-	0.05	-
maxbw	-	0.001	-
minbw	-	1.0E-05	-
CGSR	-	-	0.05

Table 5는 Bartels Conn에서 HS, IHS와 EBHS-CGS의 적용결과를 나타낸다. 적용결과를 살펴보면, EBHS-CGS는 최적값을 찾아냈으나 나머지 IHS와 HS는 최적값을 찾지 못했다. HS의 경우는 최적값을 전혀 찾지 못하였고 표준편차 역시 크게 나타났다. 반면 IHS는 표준편차가 작게 나타나 오차는 일정한 값을 가지지만 0에 수렴하지 못한다. 마지막으로 EBHS-CGS는 표준편차가 0으로 모두 일정하게 최적값을 찾아낸다. 따라서 EBHS-CGS는 다른 두 알고리즘에 비해 안정적으로 최적값을 찾는 것을 확인할 수 있다.

Table 5. Comparison of Error for application of Bartels Conn

	HS	IHS	EBHS-CGS
Mean Error	84.62426	0.006641	0
Best Error	3.914447	0.00545	0
Worst Error	566.0324	0.065918	0
SD	100.9723	0.007802	0

Fig. 7은 각 알고리즘의 Bartels Conn에서의 변화곡선을 나타낸 것이다. Fig. 7에 따르면 세 알고리즘 모두 초기 300번의 반복시작 내에서 급격히 감소하는 경향을 보인다. 수렴한 값이 작은 순서대로 나열해 보면 EBHS-CGS, IHS, 그리고 HS 순이다. 이를 통해 EBHS-CGS가 다른 두 알고리즘에 비해 수렴속도가 빠르다는 것을 알 수 있다.

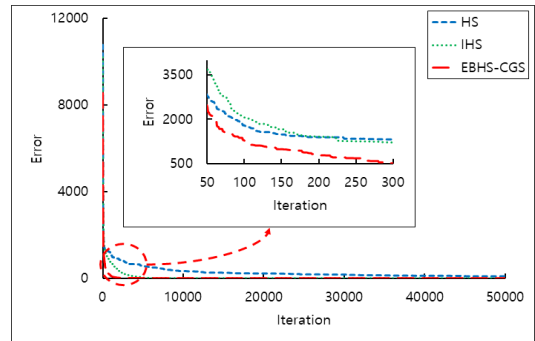


Fig. 7. Conversion shape for Bartels Conn

종합적으로 2변수 수학문제의 결과를 살펴보면 EBHS-CGS는 모든 문제에서 안정적으로 최적값을 찾아내고 있다.

3.2 30변수 수학문제

변수가 많은 문제에 대한 알고리즘의 성능을 비교하기 위해 2가지 30변수 수학문제에 적용해 보았다. 2변수와 마찬가지로 동일한 문제에 HS, IHS 그리고 EBHS-CGS를 동시에 적용하여 그 차이를 확인해 보았다.

3.2.1 Rastrigin

Eq. (8)은 Rastrigin의 식을 나타낸 것이고 Table 6은 Rastrigin에 적용한 알고리즘의 매개변수를 나타낸다. Rastrigin은 최대 시산 횟수를 100,000번으로 하여 100번 반복하였다.

$$\text{Min } f(x) = 10N + \sum_{i=1}^{30} (x_i^2 - 10\cos(2\pi x_i))$$

$$\text{Subject to } -5.12 < x_i < 5.12 \quad (8)$$

여기서 i 는 결정변수의 숫자, x_i 는 Rastrigin의 i 번째 결정변수이다.

Table 6. General setting for the simulation(Rastrigin)

Parameter	HS	IHS	EBHS-CGS
HMS	30	30	30
HMCR	0.8	0.8	0.8
PAR	0.05	-	0.05
maxPAR	-	0.99	-
minPAR	-	0.05	-
maxbw	-	0.001	-
minbw	-	1.0E-05	-
CGSR	-	-	0.5

Table 7은 Rastrigin에서 HS, IHS와 EBHS-CGS의 적용결과를 나타낸다. 적용결과를 살펴보면 HS는 모든 계산에서 최적값을 찾지 못하였고 표준편차도 크게 나타나 값이 일정하지 못하였다. IHS는 표준편차가 작아 HS보다 일정한 값을 가지지만 최적값은 찾지 못하였다. 반면 EBHS-CGS는 표준편차가 0으로 모두 일정하게 최적값을 찾아낸다. 따라서 EBHS-CGS는 다른 두 알고리즘에 비하여 더 좋은 성능을 보인다.

Table 7. Comparison of Error for application of Rastrigin

	HS	IHS	EBHS-CGS
Mean Error	63.78128	0.001123	0
Best Error	27.88105	6.65E-05	0
Worst Error	108.4651	0.003249	0
SD	14.64638	0.000852	0

Fig. 8은 각 알고리즘의 Rastrigin에서의 변화곡선을 나타낸 것이다. Fig. 8에 따르면 EBHS-CGS는 약 500번의 반복시산 내에 최적값에 수렴하였다. 반면 HS와 IHS는 반복시산 후반까지도 최적값에 도달하지 못하였다. 그 중 IHS는 반복시산 횟수가 약 500번일 때 처음으로 오차값이 급격히 감소하고, 이후 반복시산 횟수가 약 6,000번일 때 오차값이 한 번 더 급격히 감소하였으며 그 후로 지속해서 감소하여 약 0.001 정도의 값을 나타

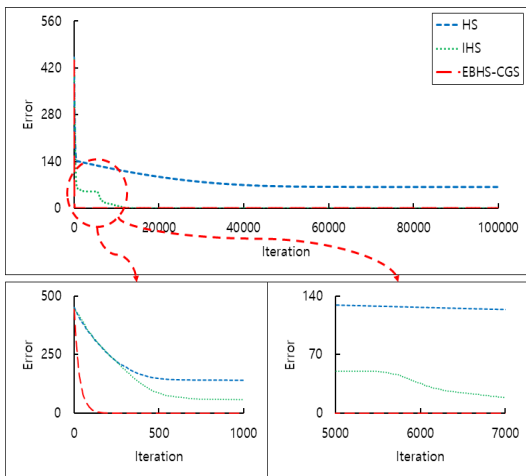


Fig. 8. Conversion shape for Rastrigin

낸다. 반면 EBHS-CGS는 매우 빠르게 최적값에 수렴한다. 따라서 EBHS-CGS는 다른 알고리즘에 비하여 매우 빠른 수렴속도를 가지고 있고, 결과값 역시 더 좋은 것을 확인할 수 있었다.

2.3.1 Griewank

Eq. (9)는 Griewank이고 Table 8은 Griewank에 적용한 알고리즘의 매개변수를 나타낸다. Griewank은 최대 시산 횟수를 100,000번으로 하여 100번 반복하였다.

$$\text{Min } f(x) = 1 + \sum_{i=1}^{30} \frac{x_i}{4000} - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

$$\text{Subject to } -600 < x_i < 600 \quad (9)$$

여기서 i 는 결정변수의 숫자, x_i 는 Griewank의 i 번째 결정변수이다.

Table 8. General setting for the simulation(Griewank)

Parameter	HS	IHS	EBHS-CGS
HMS	30	30	30
HMCR	0.95	0.95	0.95
PAR	0.05	-	0.05
maxPAR	-	0.99	-
minPAR	-	0.05	-
maxbw	-	0.001	-
minbw	-	1.0E-05	-
CGSR	-	-	0.5

Table 9는 Griewank에서 HS, IHS와 EBHS-CGS의 적용결과를 나타낸다. 적용결과를 살펴보면 HS와 IHS 모두 최적값을 전혀 찾지 못하는 것으로 나타났다. 반면 EBHS-CGS는 모든 계산에서 최적값을 찾았고 표준편차가 0으로 모두 일정하게 최적값을 찾아낸다. 따라서 EBHS-CGS는 Griewank에서 안정적으로 최적값을 찾아낸다.

Table 9. Comparison of Error for application of Griewank

	HS	IHS	EBHS-CGS
Mean Error	82.58758	89.55436	0
Best Error	25.77713	38.91589	0
Worst Error	160.9633	186.8211	0
SD	25.98218	32.20971	0

Fig. 9는 각 알고리즘의 Griewank에서의 변화곡선을 나타낸 것이다. Fig. 9에 따르면 400번의 반복시산 내에 모든 알고리즘은 일정한 값에 수렴하였다. 하지만 EBHS-CGS는 최적값에 가까운 값으로 수렴하고, HS와 IHS는 최적값을 찾지 못하였다. 따라서 EBHS-CGS는 다른 알고리즘에 비하여 비교적 빠른 수렴속도를 가지고 있고, 결과값도 좋은 것을 확인할 수 있었다.

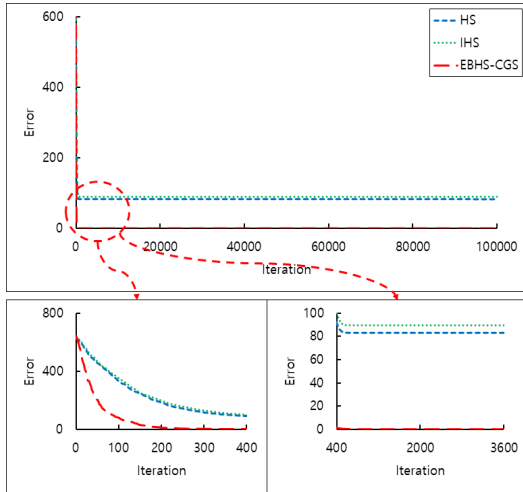


Fig. 9. Conversion shape for Griewank

종합적으로 30변수 수학문제의 결과를 살펴보면 EBHS-CGS는 모든 문제에서 안정적으로 최적값을 찾아내고 있다.

3.3 공학문제(Three-bar truss design)

$$\text{Min } f(x) = (2\sqrt{2}x_1 + x_2) \times l$$

$$\text{subject to } g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - P - \sigma \leq 0$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - P - \sigma \leq 0$$

$$g_3(x) = \frac{1}{\sqrt{2}x_1^2 + x_1} - P - \sigma \leq 0$$

$$0 \leq x_i \leq 1, \quad i = 1, 2$$

$$\text{where } l = 100\text{cm}, P = 2\text{kN/cm}^2, \sigma = 2\text{kN.cm}^2$$

(10)

여기서 x_1 과 x_2 는 Three-bar truss design의 결정변수이다.

실제 공학문제에 적용하여 EBHS-CGS의 성능을 알아 보았다. Eq. (10)은 Three-bar truss design의 식이고 Table 10은 Three-bar truss design에서 적용한 각 알고리즘의 매개변수를 표로 정리한 것이다. Three-bar truss design은 최대 시산 횟수를 100,000번으로 하여 100번 반복하였다.

Table 10. General setting for the simulation (Three-bar truss design)

Parameter	HS	IHS	EBHS-CGS
HMS	30	30	30
HMCR	0.8	0.8	0.8
PAR	0.1	-	0.1
maxPAR	-	0.99	-
minPAR	-	0.1	-
maxbw	-	0.001	-
minbw	-	1.0E-05	-
CGSR	-	-	0.05

Table 11은 Three-bar truss design에서 HS, IHS와 EBHS-CGS의 적용결과를 나타낸다. 적용결과에 따르면 표준편차는 EBHS-CGS가 가장 적게 나왔는데 이를 통해 EBHS-CGS가 안정적인 결과를 낸다는 것을 알 수 있다. 가장 좋은 결과값은 IHS와 EBHS-CGS가 비슷하게 나타나지만, 표준편차는 IHS가 더 높으므로 EBHS-CGS가 더 정밀하다. 따라서 EBHS-CGS는 Three-bar truss design에서 안정적이고 정밀한 값을 찾아낸다.

Table 11. Comparison of Value for application of Three-bar truss design

	HS	IHS	EBHS-CGS
Mean Value	264.0017	263.9742	263.9312
Best Value	263.9023	263.8959	263.8958
Worst Value	264.2545	264.3255	264.0548
SD	0.069243	0.095658	0.040766

Fig. 10은 각 알고리즘의 Three-bar truss design에서의 변화곡선을 나타낸 것이다. Fig. 10에 따르면 반복시산 초반에 EBHS-CGS는 다른 알고리즘보다 더 좋은 값을 나타내었다. 그러나 약 1,100번의 반복시산이 지나

고 나서는 HS가 가장 좋은 값을 나타내다가 약 50,000 번 이후로는 다시 EBHS-CGS가 가장 좋은 값을 나타낸다.

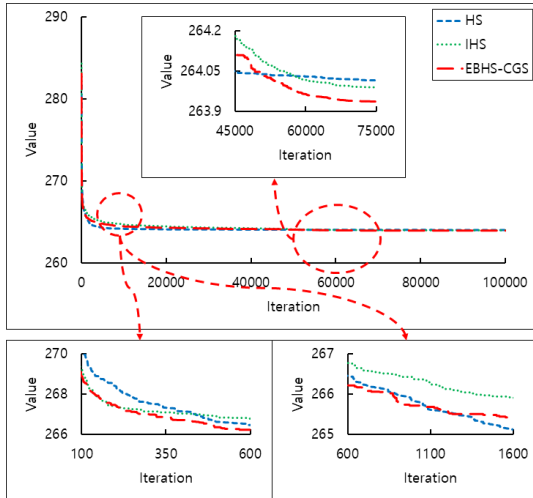


Fig. 10. Conversion shape for Three-bar truss design

모든 2변수, 30변수 수학적문제, 그리고 공학문제에서 EBHS-CGS가 안정적이며 가장 좋은 결과를 보였다.

3. 결론

본 연구는 CGS를 통해 탐색범위를 줄여 효과적인 전역탐색을 가능하게 하였고 지수 bw를 통해 반복시산 횟수가 늘어가면서 bw가 줄어들도록 하여 정밀한 지역탐색을 가능하게 하였다. 이를 통해 EBHS-CGS는 기존 HS보다 성능이 향상되었다.

EBHS-CGS는 지수 bw를 적용하여 반복시산이 진행되면서 bw의 값이 줄었다. 이를 통해 더욱 정밀한 지역탐색을 가능하게 하였고, 기존의 알고리즘보다 더 정밀한 값을 찾을 수 있었다. 또한 EBHS-CGS는 CGS를 적용하였다. CGS는 가장 좋은 결정변수를 고려하여 탐색범위를 축소한다. 반복시산이 진행되면서 가장 좋은 결정변수는 일정한 값으로 수렴하고, 그에 따라 CGS는 탐색범위를 축소하며, 효과적인 전역탐색이 가능하다는 것을 확인할 수 있었다.

EBHS-CGS의 성능을 확인하기 위해 결정변수가 각각 2개, 30개인 수학적문제와 실제 공학문제에 적용하여 결과를 확인하였다. 모든 문제에서 EBHS-CGS는 안정적으로 좋은 결과를 보이는 것을 확인할 수 있었다. 다만

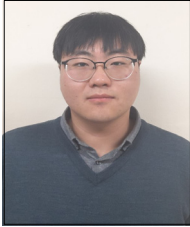
EBHS-CGS는 기존 HS에서 새로운 매개변수를 추가했기 때문에 기존보다 사용법이 복잡하다는 단점이 발생한다. 따라서 향후 연구를 통해 CGSR이 현재 최적 결정변수에 따라 자동으로 결정되는 구조로 개량하여 사용법을 단순화할 필요가 있다.

Reference

- [1] GOLDBERG, David E.; HOLLAND, John H. Genetic algorithms and machine learning, Machine learning, 1988, 3.2: pp.95-99. DOI: <http://dx.doi.org/10.1023/A:1022602019183>
- [2] DORIGO, Marco; DI CARO, Gianni. Ant colony optimization: a new meta-heuristic, In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). IEEE, 1999. pp.1470-1477. DOI: <http://dx.doi.org/10.1109/CEC.1999.782657>
- [3] EBERHART, Russell; KENNEDY, James. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. 1995. pp.1942-1948. DOI: <http://dx.doi.org/10.1109/icnn.1995.488968>
- [4] GEEM, Zong Woo; KIM, Joong Hoon; LOGANATHAN, Gobichettipalayam Vasudevan, A new heuristic optimization algorithm: harmony search. simulation, Vol.76, No.2, pp.60-68, 2001. DOI: <http://dx.doi.org/10.1177/00375497010760020>
- [5] 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. DOI: <https://doi.org/10.1016/j.amc.2006.11.033>
- [6] OMRAN, Mahamed GH; MAHDAVI, Mehrdad. 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>
- [7] 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. DOI: <https://doi.org/10.1016/j.eswa.2009.09.008>
- [8] GU, Jiadong; WU, Defeng. A random distribution harmony search algorithm, In: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI). IEEE, pp.432-437, 2018. DOI: <https://doi.org/10.1109/ICACI.2018.8377498>
- [9] JING, Wang, et al. A dynamical search space harmony search for unconstrained optimization problems, In: 2013 9th Asian Control Conference (ASCC). IEEE, pp.1-6, 2013. DOI: <https://doi.org/10.1109/ASCC.2013.6606037>

김 영 남(Young-Nam Kim)

[정회원]



- 2018년 2월 : 충북대학교 토목공학부 (공학사)
- 2019년 3월 ~ 현재 : 충북대학교 토목공학부 수공학전공 석사과정

<관심분야>

수문학,수리학,수자원공학

이 의 훈(Eui-Hoon Lee)

[정회원]



- 2003년 2월 : 고려대학교 토목환경공학과 (공학사)
- 2005년 2월 : 고려대학교 토목환경공학과 수자원공학 (공학석사)
- 2017년 2월 : 고려대학교 건축사회환경공학과 수자원공학 (공학박사)
- 2018년 9월 ~ 현재 : 충북대학교 토목공학부 교수

<관심분야>

수문학,수리학,수자원공학