

# Hybrid Vision Correction Algorithm의 개발

류용민<sup>1</sup>, 이의훈<sup>2\*</sup>

<sup>1</sup>충북대학교 토목공학과, <sup>2</sup>충북대학교 토목공학부

## Development of Hybrid Vision Correction Algorithm

Yong Min Ryu<sup>1</sup>, Eui Hoon Lee<sup>2\*</sup>

<sup>1</sup>Department of Civil Engineering, Chungbuk National University

<sup>2</sup>School of Civil Engineering, Chungbuk National University

**요약** 메타휴리스틱 탐색법은 주어진 정보의 부족 및 시간의 제약을 받는 상황에서 다양한 목적함수를 가진 문제를 해결하기 위해 개발되었다. 본 연구에서는 기존의 최적화 알고리즘인 Vision Correction Algorithm(VCA)의 성능을 강화한 Hybrid Vision Correction Algorithm(HVCA)을 개발하였다. HVCA는 기존의 알고리즘의 성능을 개선하기 위해 두 가지 방법을 적용하였다. 첫 번째 방법으로 사용자가 입력해야 하는 매개변수를 자가적응형 매개변수로 개선하였다. 두 번째 방법으로 Exponential Bandwidth Harmony Search With Centralized Global Search(EBHS-CGS)의 CGS 구조를 HVCA에 추가하였다. CGS 구조의 추가로 인해 HVCA 내부는 CGS와 VCA의 두 가지 구조로 구성되어 있다. 두 가지 구조를 효율적으로 사용하기 위해 반복시산을 진행하면서 최적값이 나오는 구조의 선택확률을 증가시키는 방법을 적용하였다. 제안된 HVCA의 성능을 확인하기 위해 최적화 문제에 적용하고, 그 결과를 Harmony Search(HS), Improved Harmony Search(IHS) 및 VCA와 비교하여 나타내었다. 적용결과 수학문제와 공학문제에서 HVCA는 HS, IHS 및 VCA보다 최적값 및 100번의 반복실행 중 최적값을 찾는 횟수가 많았으며, 최적값에 수렴하는 반복시산횟수도 낮았다. 이를 통해 HVCA가 성능이 개선되었다는 것을 확인할 수 있었다. 제안된 HVCA는 적용한 수학문제 및 공학문제 이외에도 많은 분야에 대해 좋은 결과를 나타낼 것으로 기대된다.

**Abstract** Metaheuristic search methods have been developed to solve problems with a range of purpose functions in situations lacking information and time constraints. In this study, the Hybrid Vision Correction Algorithm (HVCA), which enhances the performance of the Vision Correction Algorithm (VCA), was developed. The HVCA has applied two methods to improve the performance of VCA. The first method changes the parameters required by the user for self-adaptive parameters. The second method, the CGS structure of the Exponential Bandwidth Harmony Search With a Centralized Global Search (EBHS-CGS), was added to the HVCA. The HVCA consists of two structures: CGS and VCA. To use the two structures, a method was applied to increase the probability of selecting the structure with the optimal value as it was performed. The optimization problem was applied to determine the performance of the HVCA, and the results were compared with Harmony Search (HS), Improved Harmony Search (IHS), and VCA. The HVCA improved the number of times to find the optimal value during 100 repetitions compared to HS, IHS, and VCA. Moreover, the HVCA reduced the Number of Function Evaluations (NFEs). Therefore, the performance of the HVCA has been improved.

**Keywords** : Optimization, Self-adaptive, Improved Harmony Search, Centralized Global Search, Vision Correction Algorithm.

본 논문은 2020년도 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 September 15, 2020

Accepted January 8, 2021

Revised October 6, 2020

Published January 31, 2021

## 1. 서론

최적화는 주어진 조건 내에서 목적함수의 최댓값 및 최솟값을 찾아내는 과정이다. 목적함수는 설계비용 및 설계를 위한 부재의 수치 등 사용자가 다양하게 적용할 수 있다. 많은 분야에 사용 가능하다는 최적화의 장점으로 인해 최적화에 대한 연구가 진행되었다. 메타휴리스틱 탐색법은 주어진 정보의 부족과 시간의 제약을 받는 상황에서 최적화문제를 해결하기 위해 개발되었다. 개발된 탐색법은 군집지능기반 알고리즘(Swarm intelligence based algorithms), 생체인식 알고리즘(Bio-inspired(Not SI-based) algorithms), 물리화학기반 알고리즘(Physics and Chemistry based algorithms) 및 기타 알고리즘(Other algorithms) 등으로 분류할 수 있다[1].

분류된 알고리즘 중 군집지능기반 알고리즘은 자연에서 볼 수 있는 집단들에게 영감을 받은 알고리즘이다. 개미군집최적화(Ant Colony Optimization) 및 입자군집최적화(Particle Swarm Optimization)이 군집지능기반 알고리즘의 대표적인 예이다[2,3].

생체인식 알고리즘은 군집지능기반 알고리즘보다 더 넓은 의미이다. 군집지능기반 알고리즘은 자연 집단들에게 영감을 받았으나 생체인식 알고리즘은 자연에서 영감을 받은 알고리즘이다. 유전자알고리즘(Genetic Algorithm)이 대표적인 예이다[4].

물리화학기반 알고리즘은 중력, 전하 및 하천 시스템 등 물리적 및 화학적 법칙을 모방하여 개발되었다. 화음탐색법(Harmony Search, HS) 및 Visual Correction Algorithm(VCA) 등이 물리화학기반 알고리즘의 대표적인 예이다[5,8].

앞서 언급한 알고리즘 중 물리화학기반 알고리즘의 화음탐색법(HS)은 간단한 최적화 과정으로 인해 많이 사용되었으며, Improved Harmony Search(IHS), Exponential Bandwidth Harmony Search With Centralized Global Search(EBHS-CGS) 등의 개량된 알고리즘들이 제안되었다[6,7].

HS를 개량한 알고리즘들 중 IHS는 기존 HS의 Pitch Adjusting Rate(PAR)를 일정 값이 아닌 반복시산횟수가 증가할수록 PAR이 증가하는 방법을 이용하여 지역탐색의 성능을 개선하였다. 또한 반복시산횟수가 증가함에 따라 band width(bw)를 감소하다가 수렴하는 지수형 함수 형태로 적용하였다. 앞서 언급한 방법을 적용하여 IHS는 반복시산횟수가 증가할수록 세밀한 지역탐색이 가능해졌다[6].

EBHS-CGS의 경우 알고리즘의 성능 개선을 위해 다음과 같은 방법을 이용하였다. 첫 번째 방법으로 Centralized Global Search(CGS) 구조를 추가하였다. CGS는 전역탐색을 진행할 경우 탐색범위를 축소해 최적값을 찾는 방법이다. CGS는 현재시산의 최적 결정변수와 최적 결정변수의 반대 값을 이용하여 새로운 탐색범위를 설정한다. 새롭게 설정한 탐색범위를 이용하여 전역탐색을 실시하는 방법이다. 두 번째 방법으로 지수 bw를 추가하였다. CGS에 적용된 지수 bw는 음의 지수 형태로 반복시산의 후반부에는 매우 작은 값을 가진다. 이를 통해 반복시산의 후반부에도 정밀한 지역탐색을 가능하게 한다[7].

VCA는 시력교정과정에서 고안된 최적화 알고리즘이다. VCA는 검안을 통해 시력검사, 시력교정을 하는 과정을 최적해 탐색에 적용하였다. VCA는 반복적인 시력검사와 교정을 통해 나온 최적의 렌즈조합을 최적해로 한다. 또한 최적화 과정의 효율을 높이기 위하여 다음과 같은 방법을 이용한다. 최적해를 찾는 과정 중 지역해에 빠지는 문제를 해결하고자 탐색결과에 따라 지역탐색과 전역탐색을 유동적으로 수행하는 방법을 사용한다. VCA는 지역탐색 시 Modulation Transfer Function(MTF), Compression Factor(CF) 및 Astigmatic Correction(AF)을 이용해 미세조정을 한다. MTF와 CF를 통한 미세조정은 각각 Modulation Transfer Rate(MR) 및 Astigmatic Rate(AR)을 통해 적용된다. VCA는 좋은 결과를 보여주었으나 사용자가 입력해야 하는 많은 매개변수들로 인해 설정 시 어려움이 있다는 단점이 있다[8].

본 연구에서는 VCA의 성능 및 문제점을 개선한 Hybrid Vision Correction Algorithm(HVCA)을 제안했다. 기존 알고리즘의 문제점을 개선하기 위해 HVCA는 사용자가 입력해야 하는 매개변수를 자가적응형 매개변수로 개선함으로써 사용성을 증대시켰다. VCA의 미세조정확률을 담당하는 AR 및 MR을 자가적응형 매개변수로 개선하기 위해 IHS의 PAR개념을 적용한 HR(Hybrid Rate)로 변경하였다. HR은 PAR의 최솟값인 MinPAR에서 PAR의 최댓값인 MaxPAR까지 반복시산횟수가 증가함과 함께 PAR이 증가하는 방법을 사용한다. HR은 최솟값 0부터 최댓값 1까지 현재시산횟수를 이용해 확률을 증가시키는 방법이다. HVCA는 HR을 통해 반복시산횟수가 증가할수록 지역탐색 시 미세조정을 강화하는 방법을 적용했다. 추가적으로 HVCA의 전역탐색 성능을 강화하기 위해 EBHS-CGS의 CGS 구조를 추가했다. CGS 구조의 추가로 인하여 HVCA는 크게 두 가지 구조로 나

눌 수 있다. HVCA 내에 있는 두 가지 구조를 최적해 탐색에 효율적으로 적용하기 위해 다음과 같은 방법을 적용하였다. HVCA는 반복시산이 진행되는 동안 VCA의 구조와 CGS 구조 중 현재시산의 최적값을 찾아내는 구조의 선택확률을 높이는 방법을 적용하였다. HVCA는 반복시산횟수 및 반복시산을 진행하는 동안 최적값을 찾아낸 횟수를 이용하여 최적값을 찾을 수 있다. 앞서 제안된 방법이 적용된 알고리즘의 성능을 확인하기 위해 HVCA를 HS, IHS 및 VCA와 비교하였다.

## 2. 연구방법

### 2.1 Improved Harmony Search(IHS)

IHS는 HS의 개량형 알고리즘이다. HS의 성능을 개선하기 위해 IHS는 두 가지 방법을 적용하였다. 첫 번째 방법으로 PAR을 변경하였다. HS의 PAR은 일정 값을 적용하였다. HS는 지역탐색 실행 시 PAR을 이용해 지역탐색에 대한 미세조정을 실시하게 된다. IHS는 일정한 PAR을 적용하지 않고 PAR의 최솟값인 MinPAR부터 PAR의 최댓값인 MaxPAR까지 현재시산에 대해 증가하는 방법을 적용했다. 증가하는 PAR을 이용해 IHS는 반복시산의 후반부에서 미세조정하는 확률을 증가시켰다. 두 번째 방법으로는 bw의 개선이다. IHS는 Maxbw부터 Minbw까지 현재시산에 대해 감소하는 방법을 적용하여 반복시산의 후반부에는 매우 작은 값을 가지게 한다. IHS는 두 가지 방법을 적용해 반복시산의 후반부에서 지역탐색 후 미세조정을 실시하는 확률을 높임과 동시에 작은 bw를 이용한다. 이를 통해 반복시산의 후반부에서도 정밀한 지역탐색이 가능하게 한다. IHS의 특징 중 현재시산에 대해 증가하는 PAR개념을 HVCA의 일부 매개변수에 적용하였다. 이 방법을 통해 HVCA에서 반복시산의 후반부에서도 정밀한 지역탐색이 가능하게 했다.

### 2.2 Exponential Bandwidth Harmony Search With Centralized Global Search(EBHS-CGS)

EBHS-CGS는 HS의 개량형 알고리즘이다. EBHS-CGS는 두 가지 방법을 이용해 성능을 개선하였다. 첫 번째 방법으로 CGS 구조를 추가하였다. CGS는 탐색범위를 축소하여 효과적인 전역탐색을 실행한다. CGS는 현재시산의 최적 결정변수와 최적 결정변수의 반대값을 이용하여 새로운 탐색범위를 설정하여 탐색을 실

행한다. 두 번째 방법으로 지수형 bw를 적용하였다. 지수형 bw는 음의 지수 형태로 적용되며, 반복시산이 진행될수록 매우 작은 값을 가지게 된다. 이를 통해 반복시산의 후반부에도 정밀한 지역탐색이 가능하게 되었다.

EBHS-CGS는 HS에 전역탐색 방법 CGS를 추가한 형태이다. HS의 전역탐색을 CGS가 대체한다면 전체 탐색 범위에 대한 전역탐색을 실행하지 못하기 때문이다. CGS 구조를 추가하기 위해서 Centralized Global Search Rate(CGSR)를 적용하였다. CGSR을 통해 CGS를 이용한 탐색과 HS를 이용한 탐색 중 한 가지 탐색을 실행하게 된다. HS를 이용한 탐색 실행 시 HMCR과 PAR을 통해 최적해를 탐색한다. EBHS-CGS는 CGS구조를 이용해 전역탐색을 강화하였다. 반복시산의 초기에는 최적해 탐색이 지역탐색보다 전역탐색의 영향이 더 크기 때문에 CGS구조를 반영하여 HVCA의 전역탐색에 대한 성능을 증가시켰다.

### 2.3 Vision Correction Algorithm(VCA)

VCA는 시력교정과정을 최적해 탐색에 적용한 발견적 탐색법이다. VCA는 총 6개의 매개변수를 입력하게 된다. 기존 HS와 같은 알고리즘들은 고정적인 확률매개변수들을 통해 전역탐색 및 지역탐색을 실행한다. VCA는 전역탐색과 지역탐색의 유동적인 탐색을 하기 위해 Division Rate(DR)를 적용하였다. DR은 현재시산에서 최적값을 찾아낸 탐색방법의 선택확률을 높이는 방법을 적용하기 위한 매개변수이다. VCA는  $DR_1$  및  $DR_2$ 를 사용한다.  $DR_1$ 은 VCA가 지역탐색과 전역탐색 중 현재시산의 탐색방법을 정하게 되는 매개변수이다.  $DR_2$ 는 전역탐색 실행 시 최적값을 찾는 방향을 정하는 매개변수이다.  $DR_2$ 로 인해 정하게 되는 전역탐색 방향은 (+) 방향과 (-) 방향 중 한 가지이다.  $DR_1$  및  $DR_2$ 를 적용하여 VCA는 지역해에 빠져 최적해를 찾아내지 못하는 문제를 해결하였다. VCA는 지역탐색과 전역탐색을 실행한 후 미세조정을 실시한다. 미세조정은 MTF, CF 및 AF를 통해 계산되어 적용된다. MTF를 적용한 미세조정은 MR을 통해 적용된다. 마찬가지로 AF를 적용한 미세조정도 AR을 통해 적용된다.

### 2.4 Hybrid Vision Correction Algorithm (HVCA)

본 연구에서는 기존 VCA의 성능 및 문제점을 개선하기 위해 HVCA를 제안하였다. HVCA는 VCA의 매개변수 중 미세조정을 실시하는 AR 및 MR을 HR로 변경하

였다. 기존의 AR 및 MR은 사용자가 직접 입력해야 하는 매개변수인 반면 HR은 반복시산이 진행되는 동안 현재 시산횟수를 전체 반복시산횟수로 나누는 방법을 적용한 자가적응형 매개변수이다. 반복시산의 초기에는 지역탐색보다 전역탐색의 영향이 더 크며, 반복시산 후기로 갈수록 지역탐색의 중요성이 높아진다. 따라서 HR을 이용해 반복시산이 진행될수록 지역탐색의 미세조정 확률을 높이는 것이다. 반복시산의 후반부에 미세조정 확률을 높여 반복시산 후반부에서도 정밀한 지역탐색을 실시했다. 적용한 HR의 형태는 식 (1)과 같다. 분자는 현재 시산횟수이며, 분모는 전체 반복시산횟수이다. 이를 통해 반복시산이 진행될수록 HR의 값이 증가한다.

$$HR = \frac{\text{Current iteration}}{\text{Total iteration}} \quad (1)$$

HR은 HVCA에서 지역탐색을 실행한 뒤 미세조정 실시 여부에 대한 확률을 나타내는 매개변수이다. HVCA는 해당 매개변수들을 통한 미세조정으로 지역탐색의 성능을 증가시키는 역할을 한다. 기존의 AR 및 MR은 0과 1 사이의 실수를 입력한다. 그러나 HVCA의 HR은 최소값인 0부터 최대값인 1까지 현재시산횟수에 따라 증가하게 된다.

Table 1. Pseudocode of HVCA

```

Generate initial lens
Sort lens by objective function

If (rand > CGSR)
  If (rand < DR1)
    If (rand < DR2)
      Generate new lens in Myopia search
    else
      Generate new lens in Hyperopia search
    end if
  else
    Adjust lens using Compression factor
    If (rand < HR)
      Adjust lens using MTF
    end if
    If (rand < HR)
      Adjust lens using Astigmatism
    end if
  end if
else
  Generate new lens using CGS
end if

Compare lens and replace better lens
Continue algorithm until satisfying stop condition
    
```

HVCA는 매개변수의 개선 및 CGS 구조의 추가를 적용하여 기존 VCA의 성능 및 문제점을 개선하였다. 반복시산의 초기에 최적해를 찾을 때 지역탐색의 영향보다 전역탐색의 영향이 더 크다. 전역탐색을 강화시키기 위해 효율적으로 전역탐색을 실행하는 CGS구조를 추가하였

다. 매개변수의 개선을 통해 사용성 및 지역탐색을 강화하였고, CGS 구조의 추가를 통해 전역탐색을 강화하였다. HVCA의 의사코드는 Table 1과 같다.

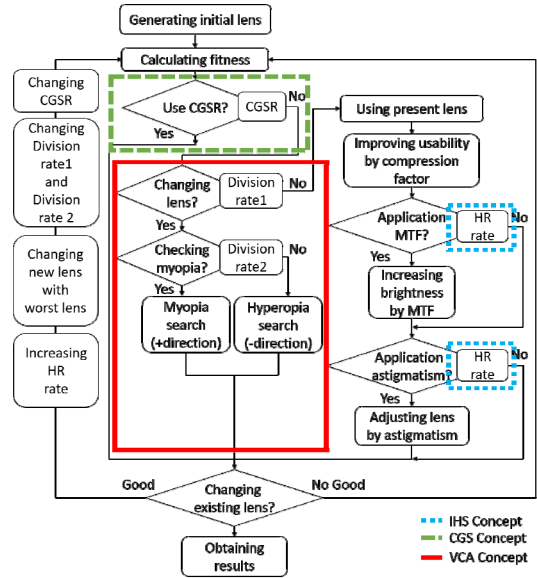


Fig. 1. Flow Chart of HVCA

Fig. 1은 HVCA의 전체 흐름도이다. Fig. 1을 보면 CGS 구조의 추가로 인해 HVCA는 크게 두 가지 구조로 구성된다는 것을 알 수 있다. 두 가지 구조를 사용하기 위해 CGSR을 추가하였다. EBHS-CGS의 CGSR은 0부터 1 사이의 실수로 입력된다. 반면에 HVCA에 적용된 CGSR은 VCA의 DR<sub>1</sub> 및 DR<sub>2</sub>와 같이 유동적으로 적용된다. CGS 구조와 VCA 구조 중 현재시산의 최적값이 나온 구조의 선택확률을 높이는 방법을 적용하였다. CGSR은 확률매개변수로 0부터 1사이의 값을 가지게 된다. VCA 구조를 이용해 최적값을 찾게되는 경우 DR<sub>1</sub>을 통해 VCA 구조 내의 지역탐색과 전역탐색을 선택하여 실행하게 된다. VCA 구조의 지역탐색을 실행하게 되는 경우 전체 흐름도 상의 우측 과정을 거치게 된다. 지역탐색 시 CF, MTF, AF에 의해 미세조정이 적용되고, 미세조정은 HR의 확률에 의해 적용된다. VCA 구조의 전역탐색을 실행하는 경우 DR<sub>2</sub>에 의해 (+) 방향으로 탐색 및 (-) 방향으로의 탐색을 실시한다. 기존 VCA의 구조는 전역탐색 및 지역탐색을 실행한 뒤 모두 미세조정을 적용하였으나, HVCA는 지역탐색에만 미세조정을 적용하였다. 반복시산이 진행되면서 지역탐색 후 미세조정을 하는 확률이 증가하기 때문에 HVCA의 지역탐색성능이 강화

되었다.

예를 들어 반복시산이 진행되면서 현재시산의 최적값이 이전 시산의 최적값보다 상대적으로 좋은 값이 나오게 되는 경우 CGSR, DR<sub>1</sub> 및 DR<sub>2</sub>는 최적값이 나온 구조의 선택확률을 증가시킨다. HR은 식 (1)의 방법을 적용하여 반복시산의 후반부로 갈수록 확률이 증가하게 된다.

HVCA의 매개변수는 총 6개로 구성되어있다, 매개변수의 종류와 기능은 Table 2와 같다. HVCA의 매개변수들 중 4개는 반복시산이 진행되면서 변경되는 자가적응형 매개변수이다. 자가적응형 매개변수는 확률로 적용되며, 이외의 매개변수는 정수형태로 계산에 적용된다. DR<sub>1</sub>, DR<sub>2</sub> 및 CGSR은 현재시산의 최적값을 찾아내는 구조의 선택확률을 조절한다. 이로 인해 지역해에 빠지는 문제를 해결할 수 있음과 동시에 최적값이 존재할 확률이 높은 구조로 탐색을 할 수 있도록 한다. 지역탐색을 진행할 경우 HR을 통해 미세조정 실행 여부를 판단한다.

Table 2. Definition of Parameters in Hybrid Vision Correction Algorithm

Parameters	Definition	Range (Default)
DR <sub>1</sub>	Probability rate of global search	Self-adaptive
DR <sub>2</sub>	Probability rate of positive search in global search phase	Self-adaptive
HR	Probability rate of MTF process and astigmatic process	Self-adaptive
AF	Integer value of astigmatic angle	0 ~ 180 (1)
CF	Searching range in MTF process	0 ~ 100 (20)
CGSR	Probability rate of CGS process	Self-adaptive

### 3. 연구결과

제안된 HVCA의 성능을 검증하기 위해 최적화문제에 적용하였다. 적용한 최적화문제는 결정변수가 2개인 수학문제 2개(Batels Conn 및 Drop-wave), 결정변수가 30개인 수학문제 2개(Ackley 및 Griewank), 마지막으로 공학문제인 Welded beam design 문제에 적용하였다. 2변수 및 30변수 수학문제는 알고리즘이 구한 결과

와 최적값(Optimal value)을 비교하여 오차로 결과를 나타냈다. 오차가 0에 근접할수록 알고리즘이 구한 결과가 최적값에 근접한 값을 찾았다고 할 수 있다. 각 문제별로 최적값과 결정변수의 수는 Table 3과 같다.

Table 3. Specification of benchmark functions

Benchmark problem	Optimal value	Number of decision variables
Bartels conn	1	2
Drop-wave	-1	2
Ackley	0	30
Griewank	0	30
Welded beam design	1.724852	4

Table 4는 결정변수가 2개 및 30개로 구성된 수확문제에 대해 적용한 HS의 매개변수 및 IHS의 매개변수이다.

Table 4. General Setting of HS and IHS for the Application of 2D and 30D benchmark function

	HS	IHS
HM	10	10
HMCR	0.9	0.9
PAR	0.5	-
MaxPAR	-	0.99
MinPAR	-	0.35
Maxbw	-	0.05
Minbw	-	0.00001

MaxPAR 및 MinPAR은 IHS에서 사용하는 매개변수이다. HS 및 IHS는 PAR을 이용해 지역탐색 시 미세조정 여부에 대한 결정을 내린다. HS는 일정한 PAR을 적용하지만 IHS는 PAR을 MinPAR부터 MaxPAR까지 현재시산횟수에 따라 증가하는 방법을 적용한다. HS 및 IHS는 bw를 이용해 지역탐색 시 미세조정을 한다. Maxbw 및 Minbw는 IHS에서 사용하는 매개변수이다. IHS는 bw를 Maxbw부터 Minbw까지 현재시산횟수에 따라 감소하는 방법을 적용하였다.

Table 5는 결정변수가 2개 및 30개로 구성된 수확문제에 대해 적용한 VCA 및 HVCA의 매개변수이다.

HS 및 IHS에서 사용되는 Harmony Memory(HM)와 VCA 및 HVCA에서 사용되는 Candidate Glasses(CG)가 동일한 역할을 하기 때문에 같은 값으로 설정하였다.

Table 5. General Setting of VCA and HVCA for the Application of 2D and 30D benchmark function

	VCA	HVCA
CG	10	10
CGSR	-	0.1
MaxHR	-	1
MinHR	-	0
DR <sub>1</sub>	0.2	0.2
DR <sub>2</sub>	0.5	0.5
MR	0.1	-
CF	20	20
AR	0.1	-
AF	45	1

최적화문제에 대한 각 알고리즘 별 성능을 비교하기 위해 Number of Function Evaluations(NFEs)를 이용하였다. NFEs는 설정된 전체 반복시산횟수와 목적함수에 의해 생성되는 새로운 해의 갯수를 곱하여 나타내는 값이다. NFEs는 최적해를 얼마나 빨리 찾아내는지 알 수 있는 정량적 지표이다.

### 3.1 2변수 수확문제

결정변수가 2개인 수확문제(Bartels conn, Drop-wave)에 대해 HVCA를 적용하였다. 성능을 검증하기 위해 HVCA의 적용결과를 HS, IHS 및 VCA의 적용결과와 비교하였다.

#### 3.1.1 Bartels conn

식 (2)는 Bartels conn의 식을 나타낸 것이다. Bartels conn 문제를 해결하기 위해 NFEs를 50,000으로 설정한 후 100회 반복실행하였다.

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

여기서  $x_1, x_2$ 는 Bartels conn의 결정변수이다. Table 6은 Bartels conn 문제를 HS, IHS, VCA 및 HVCA에 적용한 결과이다.

적용결과를 살펴보면 가장 좋은 최적값을 찾아낸 알고리즘은 HVCA이다. HVCA는 표준편차(Standard Deviation, SD)가 0으로 100회의 반복실행동안 모두

Table 6. Comparison of Error for Application of 2D benchmark function(Bartels conn)

	HS	IHS	VCA	HVCA
Best Error	7.05E-05	2.41E-05	1.90E-13	0
Worst Error	0.004875	0.002250	3.84E-06	0
Mean Error	0.001604	0.000675	2.48E-07	0
SD	0.001092	0.000425	5.73E-07	0
Mean NFEs	49,512	49,988	28,004	402

최적값을 찾았다. HVCA를 제외한 HS, IHS 및 VCA는 100회의 반복실행동안 모두 최적값을 찾지 못했다. HS와 IHS의 경우 NFEs는 약 50,000에 가깝게 나왔다. HS와 IHS는 설정된 50,000의 NFEs에서 연산이 계속되고 있다. HVCA는 다른 알고리즘들에 비해 낮은 NFEs에서 수렴했다. HVCA는 반복시산 초기에 CGS 구조를 통한 최적값 탐색이 진행된다. CGS 구조는 탐색범위의 중간값과 탐색범위의 최댓값 및 최솟값과의 거리를 이용해 탐색범위를 재설정한다. 탐색범위의 중간값과 최적값이 비슷한 경우 CGS 구조를 통한 최적해 탐색이 빠르게 이루어진다. 이러한 CGS의 장점으로 인해 NFEs 초기에 CGS 구조를 통한 전역탐색에서 오차가 빠르게 줄어든다. CGS 구조를 통해 최적값과 근접한 값을 찾아낸 후 VCA 구조의 선택확률이 높아짐과 동시에 CGS 구조의 선택확률은 줄어든다. VCA 내 지역탐색의 선택확률이 증가하여 미세조정을 통해 최적값을 빠르게 찾을 수 있다. 따라서 Bartels conn 문제에 대해 HVCA는 안정적

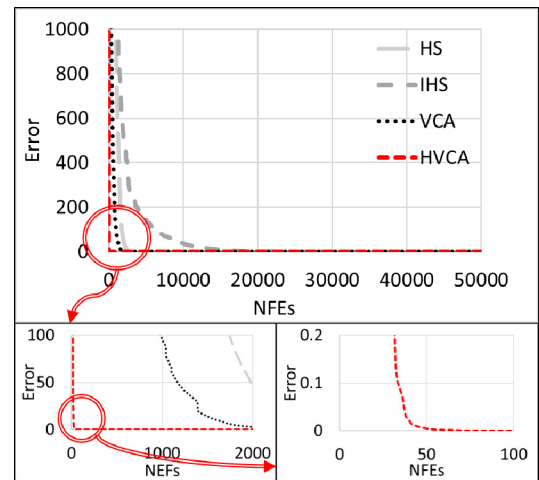


Fig. 2. Conversion shape for Bartels conn

이며 빠르게 최적값을 찾아낸다는 것을 알 수 있다.

Fig. 2는 Bartels conn 문제에 대해 알고리즘을 적용 하였을 때 각 알고리즘 별 최적값을 찾아내는 변화곡선을 나타낸 것이다. 변화곡선은 100회의 반복실행결과의 평균값을 이용하였다.

HS의 경우 약 5000의 NFEs 이전까지 빠르게 감소하다가 완만하게 수렴을 하는 것을 확인할 수 있다. 그러나 최적값에 수렴하지는 못하였다. IHS, VCA 및 HVCA는 반복시산이 시작함과 동시에 약 2000의 NFEs 이전까지 빠른 속도로 감소한다. IHS 및 VCA는 빠른 속도로 감소하였으나 최적값에 수렴하지는 못하였다. HVCA는 CGS 구조를 통한 전역탐색을 이용해 약 70의 NFEs 이전까지 빠르게 감소하였다. Fig. 2는 HVCA가 약 100의 NFEs에서 수렴하는 형태를 보이나, CGS의 전역탐색이후 VCA의 지역탐색 미세조정을 통해 오차가 감소하여 402의 NFEs에서 최적값에 수렴을 한다. 이를 통해 HVCA는 HS, IHS 및 VCA보다 수렴속도 및 결과가 모두 좋다는 것을 확인할 수 있다.

### 3.1.2 Drop-wave

식 (3)은 Drop-wave의 식을 나타낸 것이다. Drop-wave 문제를 해결하기 위해 NFEs를 50,000으로 설정한 후 100회 반복실행하였다.

$$\text{Min } f(x) = \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{(0.5(x_1^2 + x_2^2) + 2)} \quad (3)$$

$$\text{Subject to } \begin{aligned} -5.2 < x_1 < 5.2 \\ -5.2 < x_2 < 5.2 \end{aligned}$$

여기서  $x_1, x_2$ 는 Drop-wave의 결정변수이다. Table 7은 Drop-wave 문제를 HS, IHS, VCA 및 HVCA에 적용한 결과이다.

Table 7. Comparison of Error for Application of 2D benchmark function(Drop-wave)

	HS	IHS	VCA	HVCA
Best Error	1.47E-08	1.2E-10	0	0
Worst Error	0.063755	3.9E-07	0	0
Mean Error	0.061205	8.3E-08	0	0
SD	0.012493	8.8E-08	0	0
Mean NFEs	50,000	50,000	27,620	19,208

적용결과를 살펴보면 가장 좋은 최적값을 찾아낸 알고리즘은 VCA와 HVCA이다. HS와 IHS의 경우 최적값과의 오차가 있지만 VCA와 HVCA는 최적값과의 오차가 없음과 동시에 SD가 0으로, 100회의 반복실행에서 모두 최적값을 찾았다. HVCA는 VCA에 비해 NFEs값이 낮은 것을 볼 수 있다. 이를 통해 HVCA가 VCA에 비해 최적값을 빠르게 찾았다는 것을 확인할 수 있다. HS와 IHS의 경우 NFEs는 50,000이다. HS와 IHS는 50,000의 NFEs에서 연산을 진행하고 있는 상태이다. VCA는 27,620의 NFEs에서 수렴을 하였으며, HVCA는 19,208의 NFEs에서 수렴을 했다. VCA와 달리 HVCA는 CGS 구조를 통한 전역탐색을 진행한다. NFEs 초반에 전역탐색을 실시한 이후 VCA 구조를 사용한다. CGS 구조를 통한 전역탐색과 VCA를 통한 지역탐색으로 인해 HVCA는 기존 VCA보다 NFEs가 낮다. Drop-wave 문제에 대해 HVCA는 HS, IHS 및 VCA에 비해 최적값을 안정적으로 찾음과 동시에 빠르게 찾는다는 것을 알 수 있다.

Fig. 3은 Drop-wave 문제에 대해 알고리즘을 적용 하였을 때 각 알고리즘 별 최적값을 찾아내는 변화곡선을 나타낸 것이다. 변화곡선은 100회의 반복실행결과의 평균값을 이용하였다.

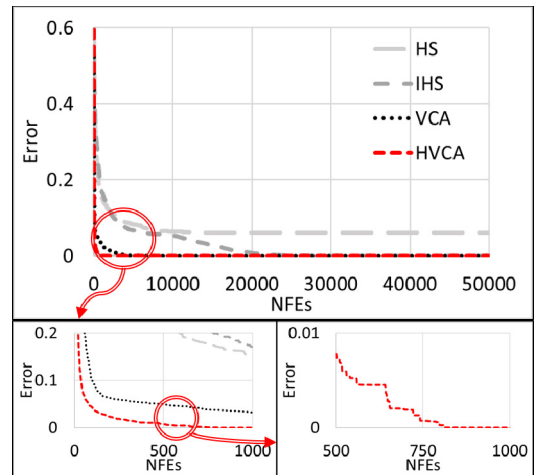


Fig. 3. Conversion shape for Drop-wave

모든 알고리즘이 약 5,000의 NFEs 동안 빠르게 감소하는 것을 볼 수 있다. 그러나 HS는 약 5,000의 NFEs 이후 감소하지 않는 모습을 보였다. IHS의 경우 약 10,000의 NFEs 동안 일정하다가 다시 감소 및 수렴하는 것을 확인할 수 있다. 그러나 최적값에 수렴하지는 못했다. HS와 IHS의 경우 설정된 50,000의 NFEs에서 현재

연산을 진행하고 있는 상태이다. VCA와 HVCA는 반복 시산 시작과 동시에 빠른 속도로 감소하였다. 약 100의 NFEs부터 완만하게 수렴하는 것을 볼 수 있다. HVCA는 VCA에 비해 앞선 NFEs에서 수렴을 시작하는 것을 볼 수 있다. Fig. 3의 그림에서는 HVCA가 약 1,000의 NFEs에서 빠르게 오차가 감소하는 모습을 보인다. NFEs 초기에 CGS 구조를 통한 전역탐색으로 인해 오차가 급격하게 줄어든다. 오차가 줄어든 이후 VCA의 선택확률이 증가하게 되면서 지역탐색을 통한 미세조정으로 최적값의 오차가 작게 감소한다. 매우 작게 오차가 감소하면서 수렴이 진행되기 때문에 Fig. 3의 그림에서는 약 1,000의 NFEs에서 수렴을 하는 것처럼 보이나 약 19,200의 NFEs에서 수렴을 하게 된다. 이를 통해 HVCA는 HS, IHS 및 VCA에 비해 빠르게 최적값을 찾는다라는 것과 최적값을 안정적으로 찾을 수 있다는 것을 알 수 있다.

2변수 수학문제들에 대해 HS, IHS, VCA 및 HVCA를 적용한 결과 HVCA가 다른 알고리즘들에 비해 최적값을 안정적으로 찾음과 동시에 빠르게 찾아내고 있다.

### 3.2 30변수 수학문제

결정변수가 30개인 수학문제(Ackley, Griewank)에 대해 HVCA를 적용하였다. 성능을 검증하기 위해 HVCA의 적용결과를 HS, IHS 및 VCA의 적용결과와 비교하였다.

#### 3.2.1 Ackley

식 (4)는 Ackley의 식을 나타낸 것이다. Ackley 문제를 해결하기 위해 NFEs를 100,000으로 설정한 후 100회 반복실행하였다.

$$\begin{aligned}
 \text{Min } f(x) = & -20 \text{Exp} \left( -0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2} \right) \quad (4) \\
 & - \text{Exp} \left( \frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i) \right) \\
 & + 20 + \text{Exp}(1) \\
 \text{Subject to } & -32 < x_i < 32 \\
 & i = 1, 2, \dots, 30
 \end{aligned}$$

여기서  $i$ 는 결정변수의 수이고,  $x_i$ 은 Ackley의 결정변수이다. Table 8은 Ackley 문제를 HS, IHS, VCA 및 HVCA에 적용한 결과이다.

Table 8. Comparison of Error for Application of 30D benchmark function(Ackley)

	HS	IHS	VCA	HVCA
Best Error	15.851357	1.157248	0	0
Worst Error	19.287443	18.999402	3.55E-15	0
Mean Error	17.946714	3.887985	3.55E-16	0
SD	0.700060	3.188924	1.07E-15	0
Mean NFEs	52,385	99,996	15,308	698

적용결과를 살펴보면 가장 좋은 최적값을 찾아낸 알고리즘은 VCA와 HVCA이다. HS의 경우 NFEs가 52,385로 IHS보다 좋은 결과가 나왔다. 그러나 가장 낮은 오차값과 가장 높은 오차값이 심하게 발생하였다. HS의 경우 반복시산 초기에 전역탐색을 통한 최적값탐색이 기능을 하지 못했다. 전역탐색이 제대로 실행되지 않은 상태에서 지역탐색이 진행되었기 때문에 최적값을 찾지 못했다. IHS의 경우 HS보다 좋은 오차결과를 보였지만 NFEs가 99,996의 결과를 보였다. IHS는 현재 연산을 진행하는 과정이며, 설정된 100,000의 NFEs에서는 수렴을 하지 않는다. VCA의 경우 좋은 최적값을 찾아냈으나 SD가 1.07E-15로 100회의 반복실행에서 모두 최적값을 찾지는 못하였다. 그러나 HVCA의 경우 가장 좋은 최적값을 찾음과 동시에 SD가 0이다. 따라서 HVCA는 100회의 반복실행동안 모두 최적값을 찾았으며, NFEs값이 다른 알고리즘에 비해 낮았다. HVCA는 CGS 구조를 통한 전역탐색을 이용해 NFEs초기에 최적값에 근접한 값을 찾아낸다. 최적값에 근접한 값을 찾아낸 후 VCA의 선택확

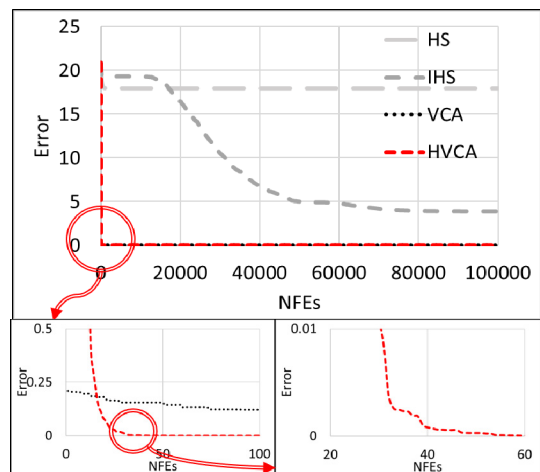


Fig. 4. Conversion shape for Ackley



률이 증가하게 되면서 VCA의 지역탐색확률이 증가한다. HVCA는 CGS와 VCA의 적절한 선택으로 인해 다른 알고리즘보다 낮은 NFEs가 나왔다.

Fig. 4는 Ackley 문제에 대해 알고리즘을 적용하였을 때 각 알고리즘 별 최적값을 찾아내는 변화곡선을 나타낸 것이다. 변화곡선은 100회의 반복실행결과와 평균값을 이용하였다.

HS의 경우 반복시산이 시작된 후 감소하지 않고 일정하게 진행되는 것을 볼 수 있다. IHS의 경우 약 20,000의 NFEs 이후 감소를 하다가 약 50,000의 NFEs부터 천천히 수렴하는 것을 볼 수 있다. HS와 IHS의 경우 최적값을 찾지 못하는 것을 알 수 있다. VCA와 HVCA의 경우 반복시산이 시작함과 동시에 빠르게 감소하는 것을 확인할 수 있다. VCA는 반복시산 시작과 동시에 낮은 값에서 수렴이 진행되었으나, HVCA가 약 20의 NFEs 이후 VCA보다 빠르게 최적값에 수렴한다는 것을 알 수 있다. Fig. 4에서 HVCA는 NFEs 초반에 CGS 구조를 통해 오차가 빠르게 줄어든다. 오차가 줄어든 이후 VCA 구조를 통한 지역탐색의 선택확률이 증가한다. 지역탐색이 실행되면서 미세조정확률을 통해 미세조정이 실행된다. 미세조정으로 인해 오차가 매우 작은 차이를 보이며 감소한다. 작은 차이를 보이며 감소하기 때문에 그래프상에서는 수렴하는 모습을 보이지만 HVCA는 약 700의 NFEs에서 수렴을 한다. 따라서 HVCA가 다른 알고리즘에 비해 빠르고 안정적으로 최적값을 찾는다라는 것을 알 수 있다.

### 3.2.2 Griewank

식 (5)는 Griewank의 식을 나타낸 것이다. Griewank 문제를 해결하기 위해 NFEs를 100,000으로 설정한 후 100회 반복실행하였다.

$$\text{Min } f(x) = 1 + \sum_{i=1}^{30} \frac{x_i^2}{4000} - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (5)$$

$$\text{Subject to } -600 < x_i < 600 \\ i = 1, 2, \dots, 30$$

여기서  $i$ 는 결정변수의 수이고,  $x_i$ 은 Griewank의 결정변수이다. Table 9는 Griewank 문제를 HS, IHS, VCA 및 HVCA에 적용한 결과이다.

적용결과를 살펴보면 가장 좋은 최적값을 찾아낸 알고리즘은 VCA 및 HVCA이다. HS 및 IHS의 경우 100회의 반복실행동안 최적값을 찾아내지 못했다. HS와 IHS는

Table 9. Comparison of Error for Application of 30D benchmark function(Griewank)

	HS	IHS	VCA	HVCA
Best Error	0.010125	7.71E-03	0	0
Worst Error	0.154069	0.249669	0	0
Mean Error	0.052357	5.60E-02	0	0
SD	0.032746	0.044953	0	0
Mean NFEs	99,994	99,998	8,260	322

설정된 NFEs인 100,000과 거의 근접한 값을 나타내고 있다. HS와 IHS는 현재 연산을 진행하는 과정이다. HVCA의 경우 VCA와 다르게 전역탐색을 실행하는 CGS 구조가 추가적으로 있다. CGS 구조는 NFEs 초기에 효율적인 전역탐색을 실행한다. CGS 구조는 탐색범위를 이용해 전역탐색 범위를 새롭게 지정하여 탐색을 한다. 새로운 탐색범위를 통한 지역탐색으로 인해 NFEs 초기에 오차가 빠르게 줄어든다. HVCA는 CGS 구조를 통한 전역탐색 이후 VCA 구조의 선택확률이 증가한다. HVCA는 CGS 구조와 VCA 구조를 통해 VCA보다 낮은 NFEs 결과를 나타낸다. 이를 통해 HVCA가 HS, IHS 및 VCA보다 최적값을 빠르고 안정적으로 찾는다라는 것을 알 수 있다.

Fig. 5는 Griewank 문제에 대해 알고리즘을 적용하였을 때 각 알고리즘 별 최적값을 찾아내는 변화곡선을 나타낸 것이다. 변화곡선은 100회의 반복실행결과와 평균값을 이용하였다.

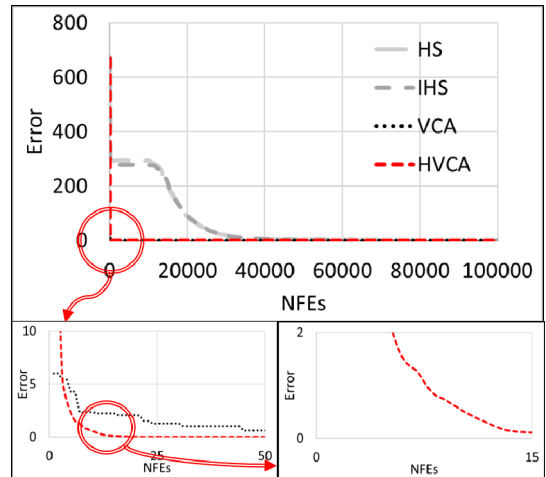


Fig. 5. Conversion shape for Griewank

HS 및 IHS의 경우 비슷한 형상을 보이는 것을 확인할 수 있다. 약 10,000의 NFEs 이후로 감소하다가 약 30,000의 NFEs 이후 수렴하였다. VCA 및 HVCA의 경우 반복시산이 시작한 직후부터 빠르게 감소하였다. HVCA는 CGS 구조를 통해 NFEs 초기에 빠르게 오차가 줄어들게 되고, 최적값에 근접하게 된다. 이후 VCA구조의 선택확률이 증가한다. CGS 구조를 통한 전역탐색과 VCA 구조 내 지역탐색 및 지역탐색의 미세조정을 통한 최적값의 감소가 계속 진행된다. HVCA가 VCA에 비해 시작단계에서의 오차 값은 컸으나, 반복시산 초기에 VCA보다 오차가 빠르게 수렴하는 것을 확인할 수 있다. 따라서 HVCA가 HS, IHS 및 VCA에 비해 최적값을 안정적으로 찾음과 동시에 빠르게 찾는다라는 것을 알 수 있다.

30변수 수학문제에 대해 HS, IHS, VCA 및 HVCA를 적용한 결과 HVCA가 다른 알고리즘들에 비해 최적값을 안정적으로 찾음과 동시에 빠르게 찾아내고 있다.

2변수 및 30변수 수학문제들을 HS, IHS, VCA 및 HVCA에 적용한 결과 모든 수학문제에서 HVCA는 최적값을 찾는 횟수, 최적값 및 최적값으로 수렴하는 속도가 HS, IHS, VCA보다 좋은 결과를 나타내고 있다. HVCA는 두가지 구조를 이용해 최적값을 빨리 찾기 때문이다. HVCA의 CGS 구조는 새로운 탐색범위를 설정할 때 기존 탐색범위의 중간값을 사용한다. 중간값과 최댓값 및 최솟값의 거리가 동일할 경우 CGS를 통해 오차가 빠르게 줄어든다. 오차가 빠르게 줄어든 이후 VCA 구조의 선택확률이 증가한다. VCA 구조를 통한 지역탐색을 통해 최적값을 찾는다. HVCA가 다른 알고리즘들에 비해 낮은 NFEs 결과를 나타낸 이유이다. 수학문제에 대한 결과를 통해 HVCA의 성능이 다른 알고리즘들에 비해 좋다는 것을 알 수 있다.

### 3.3 공학문제(Welded beam design)

HVCA의 성능을 확인하기 위해 결정변수가 4개인 공

Table 10. General Setting of HS and IHS for Application of engineering benchmark function

	HS	IHS
HM	100	100
HMCR	0.95	0.95
PAR	0.5	-
MaxPAR	-	0.99
MinPAR	-	0.35
Maxbw	-	0.05
Minbw	-	0.00001

학문제 Welded beam design에 적용하였다. HVCA의 성능을 검증하기 위해 HS, IHS 및 VCA에 Welded beam design을 적용한 결과와 비교하였다. Table 10은 Welded beam design에 대해 적용한 HS의 매개변수 및 IHS의 매개변수이다.

Table 11은 Welded beam design에 대해 적용한 VCA 및 HVCA의 매개변수이다. CGS 구조는 탐색범위의 중간값과 탐색범위의 최댓값 및 최솟값과의 거리를 이용해 탐색범위를 재설정하여 탐색을 실행하는 방법이다. 중간값과 거리가 비슷할 경우 CGS구조의 효율이 높다는 장점이 있으나, 거리가 비슷하지 않을 경우 CGS 구조의 효율이 낮아진다. 이런 단점으로 인해 HVCA는 공학문제에서 VCA구조를 집중적으로 사용했다. HVCA는 VCA구조의 HR을 통해 반복시산이 진행되면서 미세조정확률을 증가시킨다.

Table 11. General Setting of VCA and HVCA for Application of engineering benchmark function

	VCA	HVCA
CG	100	100
CGSR	-	0
MaxHR	-	1
MinHR	-	0
DR <sub>1</sub>	0.1	0.1
DR <sub>2</sub>	0.5	0.5
MR	0.1	-
CF	25	25
AR	0.1	-
AF	45	1

식 (6)은 Welded beam design의 식을 나타낸 것이다. Welded beam design을 해결하기 위해 알고리즘의 NFEs를 100,000으로 설정한 후 100회 반복실행하였다.

$$\text{Min } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

Subject to

$$g_1(x) = \tau(x) - \tau_{max} \leq 0$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0$$

$$g_3(x) = x_1 - x_4 \leq 0$$

$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0$$

$$g_5(x) = 0.125 - x_1 \leq 0$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0$$

$$g_7(x) = P - P_c(x) \leq 0$$

$$0.1 \leq x_i \leq 2, \quad i = 1, 4$$

$$0.1 \leq x_i \leq 10, \quad i = 2, 3$$

where

$$\begin{aligned} \tau(x) &= \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \\ \tau'' &= \frac{MR}{J}, \quad M = P(L + \frac{x_2}{2}), \\ R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \\ J &= 2 \left\{ \sqrt{2}x_1x_2 \left[ \frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\}, \\ \sigma(x) &= \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_3^3x_4}, \\ P_c(x) &= \frac{4.013E\sqrt{(x_3^2x_4^6/36)}}{L^2} + \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}}\right) \\ P &= 6000 \text{ lb}, \quad L = 14 \text{ in}, \quad E = 30 \times 10^6 \text{ psi}, \\ G &= 12 \times 10^6 \text{ psi}, \quad \tau_{\max} = 13,600 \text{ psi}, \\ \sigma_{\max} &= 30,000 \text{ psi}, \quad \delta_{\max} = 0.25 \text{ in} \end{aligned} \quad (6)$$

여기서  $i$ 는 결정변수의 수이고,  $x_i$ 은 Welded beam design의 결정변수이다.  $x_1, x_2, x_3$  및  $x_4$ 는 Welded beam design의 결정변수이다. Table 12는 Welded beam design을 HS, IHS, VCA 및 HVCA에 적용한 결과이다.

Table 12. Comparison of Result for Application of engineering benchmark function(Welded beam design)

	HS	IHS	VCA	HVCA
Best Result	1.778264	1.725000	1.724852	1.724852
Worst Result	3.051782	1.934531	1.724999	1.724857
Mean Result	1.870171	1.751978	1.724866	1.724852
SD	0.187076	0.049829	0.000027	0.000001
Mean NFEs	99,900	100,000	85,660	70,535

적용결과를 살펴보면 HS와 IHS는 최적값을 찾아가는 과정이다. NFEs의 값이 설정된 100,000과 거의 근접하기 때문이다. HS와 IHS는 최적값을 찾기 위해 100,000의 NFEs보다 많은 NFEs를 필요로 한다는 것을 알 수 있다. VCA 및 HVCA가 가장 좋은 최적값을 나타낸다. 그러나 VCA의 경우 HVCA보다 SD가 높다는 것을 알 수 있다. 이를 통해 HVCA가 VCA에 비해 더 안정적으로 최적값을 찾아낸다고 볼 수 있다. HVCA는 100회의 반복

실행 중 가장 큰 최적값인 1.724857이 1회 발생하였다. 1회를 제외한 나머지 값들은 낮은 최적값과 유사하게 나왔기 때문에 Mean Result에 대한 영향이 크지 않았다. 크지 않은 영향으로 인해 Mean Result와 Best Result 값은 Table 12에서 같은 값으로 나왔으나 확인 결과 7.46E-08의 차이를 보였다. NFEs의 경우 HVCA가 VCA에 비해 낮은 것을 확인할 수 있다. HVCA는 VCA와 다르게 NFEs가 증가할수록 HR의 확률이 증가하기 때문에 정밀한 지역탐색이 가능하다. 기존 VCA는 AR과 MR이 일정한 확률을 가지고 있으나, HVCA는 HR을 통한 미세조정확률이 높아지기 때문에 HVCA가 지역탐색의 미세조정을 통한 최적해를 찾아낼 확률이 높아진다. 따라서 HVCA가 HS, IHS 및 VCA에 비해 최적값을 안정적으로 찾음과 동시에 빠르게 찾아낸다는 것을 알 수 있다.

HVCA는 수학문제에 비해 공학문제의 NFEs가 높은 것을 확인할 수 있다. CGS 구조 선택확률의 차이가 있기 때문이다. 수학문제의 경우 CGS 구조의 장점으로 인해 최적값을 빠르게 찾지만, 공학문제의 경우 CGS 구조의 장점이 나타나지 못한다. 탐색범위의 중간값과 탐색범위의 최댓값 및 최솟값과의 거리가 다르기 때문이다. HVCA는 CGS 구조를 제외한 VCA 구조를 집중적으로 사용한다. 수학문제의 경우 CGS 구조를 통한 NFEs 초반의 오차 감소가 있었으나, 공학문제에서는 CGS의 초기 선택확률이 낮다. 이런 이유로 공학문제에 적용한 HVCA의 NFEs는 수학문제에 적용한 HVCA의 NFEs보다 크다.

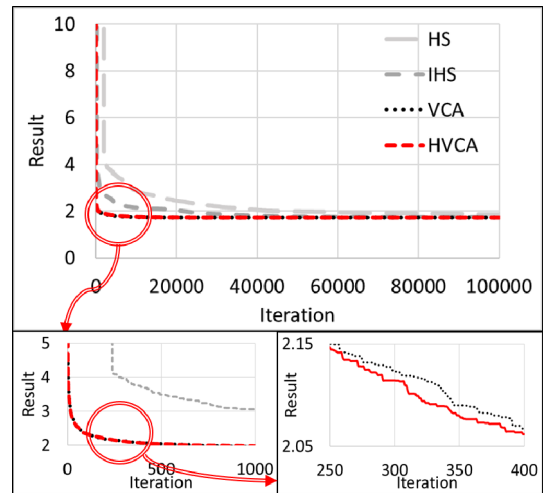


Fig. 6. Conversion shape for Welded beam design

Fig. 6은 Welded beam design에 대해 알고리즘들을 적용하였을 때 각 알고리즘 별 최적값을 찾아내는 변화곡선을 나타낸 것이다. 변화곡선은 100회의 반복실험 결과의 평균값을 이용하였다.

반복시산이 시작한 직후 모든 알고리즘이 빠르게 감소하여 수렴하는 모습을 확인할 수 있다. 그러나 알고리즘들 중 HVCA가 가장 빠르게 감소하는 것을 확인할 수 있다. HVCA의 수렴곡선은 VCA와 비슷한 형태를 보였으나, VCA보다 더 빠르게 최적값에 수렴하는 것을 확인할 수 있다.

#### 4. 결론

본 연구는 VCA를 기반으로 두 가지 방법을 적용하여 문제점 및 성능을 개선한 HVCA를 제안하였다. 첫 번째 방법은 기존의 매개변수를 자가적응형 매개변수로 개선하는 방법이다. 두 번째 방법은 CGS 구조를 추가하여 전역탐색을 강화하는 방법이다.

기존의 매개변수를 자가적응형 매개변수로 개선하기 위해 HVCA는 반복시산횟수에 따라 매개변수가 변경되는 방법을 적용하였다. HVCA의 자가적응형 매개변수인 HR은 반복시산횟수가 증가할수록 지역탐색 시 미세조정을 확률을 증가시킨다. 이를 통해 반복시산 후반부에서도 지역탐색을 강화시킬 수 있었다. 지역탐색의 성능강화와 동시에 CGS 구조의 추가로 HVCA의 전역탐색을 강화하였다. HVCA는 CGS구조와 VCA구조를 사용하기 위해 현재시산의 최적값을 찾아내는 구조의 선택확률을 높이는 방법을 적용하였다.

HVCA의 개선된 성능 및 결과 비교를 위해 최적값을 찾아내는 횟수 및 NFEs를 제시하여 나타내었다. 최적값을 찾아내는 횟수는 적용한 문제에 대해 알고리즘이 안정적으로 최적값을 찾아내는지 알 수 있다. NFEs는 최적값을 얼마나 빨리 찾을 수 있는지 알 수 있는 정량적인 지표이다. 최적값을 찾아내는 횟수 및 NFEs를 통해 알고리즘이 적용문제에 대해 최적값을 찾아내는 속도와 정확도를 확인할 수 있으며 성능을 비교할 수 있다.

HVCA는 CGS 구조를 통해 반복시산의 초반부에서 최적값에 근접한 값을 빠르게 찾아낸다. 이후 반복시산이 진행되면서 VCA 구조의 미세조정으로 최적값을 찾아낸다. VCA 구조의 미세조정은 자가적응형 매개변수인 HR을 통해 진행된다. 반복시산이 진행되면서 HR의 값이 증가하기 때문에 미세조정을 실행하는 확률이 증가하게 된다.

HVCA의 성능을 확인하기 위해 결정변수가 2개 및 30개로 구성된 수학문제와 결정변수가 4개로 구성된 공학문제에 적용하여 HS, IHS 및 VCA와 결과를 비교하였다. 결과를 비교하기 위해 각 알고리즘의 최적값 및 NFEs를 이용하였다. HVCA는 다른 알고리즘들에 비해 최적값을 안정적으로 찾음과 동시에 NFEs가 낮은 것을 확인할 수 있었다. 적용한 최적화 문제에서 HVCA는 기존 VCA보다 NFEs가 최소 17.7%에서 최대 98.6% 정도 감소하였다. 감소한 NFEs와 최적값 및 오차의 결과를 통해 HVCA의 성능이 기존 VCA보다 개선되었다는 것을 확인할 수 있었다.

기존 VCA의 경우 사용자가 입력해야 하는 매개변수의 수가 많은 문제점이 있었다. HVCA는 VCA의 문제점을 해결하기 위해 매개변수의 개선 및 성능을 강화하기 위한 구조적 변화가 있었다. 문제점 및 성능 개선을 위한 방법이 적용된 HVCA의 결과와 VCA의 결과 비교를 통해 기존 문제점의 개선 및 성능을 강화하였다는 것을 확인할 수 있었다.

HVCA는 CGS 구조를 추가하여 알고리즘 내 구조가 두 가지가 되었다. 구조를 추가하는 과정에서 알고리즘 내 많은 연산이 추가되었고, 따라서 기존 VCA보다 시간이 더 소요된다는 단점이 있다. 향후 연구를 통해 HVCA의 계산과정 및 연산을 간소화한다면 사용자가 편리하게 사용할 수 있을 것이다.

#### References

- [1] I. Fister Jr, X.S. Yang, I. Fister, J. Brest, D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization", *Elektrotehniški vestnik* Vol.80, No.3, pp.1-7, Jul 2013.
- [2] D. E. Goldberg, J. H. Holland, "Genetic algorithms and machine learning", *Machine learning*, Vol.3, pp.95-99, Oct 1988.  
DOI: <http://dx.doi.org/10.1023/A:1022602019183>
- [3] M. Dorigo, G. Di Caro, "Ant colony optimization: a new meta-heuristic", *In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*. IEEE, pp.1470-1477, Jul 1999.  
DOI: <http://dx.doi.org/10.1109/CEC.1999.782657>
- [4] J. Kennedy, R. Eberhart, "Particle swarm optimization", *In: Proceedings of the IEEE international conference on neural networks*, pp.1942-1948, Dec 1995.  
DOI: <http://dx.doi.org/10.1109/icnn.1995.488968>
- [5] Z. W. Geem, J. H. Kim, G.V. Loganathan, Gobichettipalayam Vasudevan, "A new heuristic

optimization algorithm: harmony search", *Simulation*, Vol. 76, Issue.2, pp.60-68, Feb 2001.

DOI: <https://doi.org/10.1177/003754970107600201>

- [6] M. Mahdavi, M. Fesanghary, E. Damangir, "An improved harmony search algorithm for solving optimization problems", *Applied mathematics and computation*, Vol.188, Issue.2, pp.1567-1579, May 2007. DOI: <https://doi.org/10.1016/j.amc.2006.11.033>
- [7] Y. N. Kim, E. H. Lee, "Development of the Meta-heuristic Optimization Algorithm: Exponential Bandwidth Harmony Search with Centralized Global Search.", *Journal of the Korea Academia-Industrial cooperation Society*, Vol.21, Issue.2 pp. 8-18, Feb 2020. DOI: <https://doi.org/10.5762/KAIS.2020.21.2.8>
- [8] E. H. Lee, H. M. Lee, D. G. Yoo, J. H. Kim, "Application of a Meta-heuristic Optimization Algorithm Motivated by a Vision Correction Procedure for Civil Engineering Problems", *KSCE Journal of Civil Engineering*, Vol.22, No.7 pp. 2623-2636, Sep 2017. DOI: <https://doi.org/10.1007/s12205-017-0021-3>

류 용 민(Yong Min Ryu)

[준회원]



- 2020년 2월 : 충북대학교 토목공학부 (공학사)
- 2020년 2월 ~ 현재 : 충북대학교 토목공학부 수공학전공 석사과정

<관심분야>

수리학,수문학,수자원공학

이 의 훈(Eui Hoon Lee)

[종신회원]



- 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