

공학설계 최적화 문제 해결을 위한 GA-VNS-HC 접근법⁺

(GA-VNS-HC Approach for Engineering Design Optimization Problems)

윤 영 수^{1)*}
(YoungSu Yun)

요 약 본 연구에서는 공학설계 최적화 문제 해결을 위한 혼합 메타휴리스틱(Hybrid Meta-heuristic) 접근법을 제안된다. 공학 설계 최적화 문제는 다양한 형태의 변수를 가지며, 복잡한 제약조건들하에서 그 최적해를 구하는 문제로 이미 많은 기존 연구들을 통해 다양한 접근법들이 개발되어져 왔다. 하지만 그 효율성은 아직까지 크게 개선되지 못하고 있는 실정이다. 따라서 본 연구에서는 이러한 효율성을 개선하기 위한 새로운 접근법을 제안한다. 제안된 혼합 메타휴리스틱 접근법은 탐색 공간에 대한 전역적 탐색을 위해 유전알고리즘(Genetic Algorithm: GA) 접근법, 지역적 탐색을 위해 변동이웃탐색(Variable Neighborhood Search: VNS) 접근법과 언덕오르기(Hill Climbing: HC) 접근법을 혼합(GA-VNS-HC)하였다. 사례 연구에서는 다양한 형태의 공학설계 최적화 문제를 이용하여 본 연구에서 제안한 GA-VNS-HC 접근법의 우수성을 입증하였다.

핵심주제어: 공학설계 최적화 문제, 메타휴리스틱, 유전알고리즘, 변동이웃탐색, 언덕오르기

Abstract In this study, a hybrid meta-heuristic approach is proposed for solving engineering design optimization problems. Various approaches in many literatures have been proposed to solve engineering optimization problems with various types of decision variables and complex constraints. Unfortunately, however, their efficiencies for locating optimal solution do not be highly improved. Therefore, we propose a hybrid meta-heuristic approach for improving their weaknesses. the proposed GA-VNS-HC approach is combining genetic algorithm (GA) for global search with variable neighborhood search (VNS) and hill climbing (HC) for local search. In case study, various types of engineering design optimization problems are used for proving the efficiency of the proposed GA-VNS-HC approach

Keywords: Engineering design optimization problem, Meta-heuristics, Genetic algorithm, Variable neighborhood search, Hill climbing

* Correspponding Author: ysyun@chosun.ac.kr

+ 이 논문은 2020학년도 조선대학교 학술연구비의 지원을 받아 연구되었음.

Manuscript received January 12, 2022 / revised January 25, 2022 / accepted January 26, 2022

1) 조선대학교 경상대학 경영학부

1. 서 론

일반적으로 공학설계 최적화(Engineering Design Optimization) 문제는 다양한 형태의 설계변수(예

를 들어, 이산변수(Discrete Variable), 정수변수(Integer Variable), 실수변수(Continuous Variable), 0-1변수 등)들이 함께 사용되며, 이들 설계변수들로 구성된 복잡한 제약조건을 만족하는 상황에서 정밀한 최적해를 구하는 문제라고 정의할 수 있다. 특히 사용되는 설계변수의 형태가 다양하고 함께 사용되는 경우가 많기 때문에 이들 설계변수를 효과적으로 다루기 위한 다양한 접근법들이 개발되어져 왔다(Amir and Hasegawa, 1989, Sandgran 1990; Fu et al., 1991; Kuo et al., 2001; Yarushkina, 2002).

대표적인 기존 연구들을 살펴보면 다음과 같다. Amir and Hasegawa (1989)는 강화 콘크리트 빔(Reinforced Concrete Beam) 설계문제를 해결하기 위한 접근법을 제시하였다. 이 문제의 경우 세 개의 설계변수가 사용되며, 그중 하나는 사전에 설정된 이산변수 형태를 가지며, 나머지 두 개는 실수변수와 정수변수를 각각 가진다. 이들의 연구에서는 MROP(Modified Rosenbrocks Orthogonalization Procedure) 접근법을 사용하여 실수변수 및 정수변수를 전부 이산변수 형태로 변경한 후 그 최적해를 구하였다. 하지만 이 접근법의 경우, 변수의 형태를 변경하는 과정에서 실수변수의 소수점 이하 값들 일부가 손실되는 문제가 있으며, 결국 구해진 최적해가 전역적 최적해가 아닌 경우가 종종 발생한다는 단점이 존재한다.

Sandgran (1990)은 이산변수, 정수변수, 실수변수가 함께 사용되는 기어트레인(Gear Train) 설계문제와 압력용기(Pressure Vessel) 설계문제를 해결하기 위해 분지한계법(Branch and Bound Method)을 사용하였다. 이 접근법을 적용하기 위해 이산변수와 정수변수를 전부 실수변수로 변경하였다. 하지만 이러한 변경 과정에서 이산변수와 정수변수가 가지는 원래의 탐색범위를 조정하게 되며, 결국 적정해(Feasible Solution)는 구할 수 있지만, 구해진 적정해가 최적해가 아닐 수 있다는 단점이 존재한다.

Sandgran (1990)과 마찬가지로 Fu et al. (1991) 또한 기어트레인 설계 문제와 압력용기 설계 문제를 다루었다. 여기서는 이산변수와 정수변수에 대해 벌금함수(Penalty Function)를 도

입하여 변수의 형태를 변경한 후 IDCNLP (Integer - Discrete-Continuous Non-Linear Programming) 접근법을 이용하여 최적해를 구하였다. 하지만 이러한 변수 형태의 변경과정에서 이산변수와 정수변수가 가지는 원래의 탐색범위가 조정되며, 결국 최적해를 찾지 못하고 국소해(Local Optimal Solution)에 빠지거나 조기수렴(Premature Convergence)되는 문제점을 가지고 있다.

위에서 언급한 기존 연구들의 경우 사용되는 다양한 변수형태(이산변수, 정수변수, 실수변수)를 변경하여 사용하고 있다. 하지만 이러한 변경과정에서 구해진 해의 품질이 떨어지거나 (즉, 최적해를 찾지 못한다) 조기 수렴되는 한계점이 존재한다. 이러한 한계점을 개선하기 위해 메타휴리스틱(Meta-Heuristics) 접근법을 이용한 다양한 연구가 활발히 진행되었다. 이들 메타휴리스틱 접근법을 이용할 경우, 공학설계 최적화 문제에서 사용되는 다양한 변수형태를 변형시키지 않고 그대로 사용할 수 있다는 장점을 가지고 있다 (Lin and Hajela, 1991; Wu and Chow, 1995; Yokota et al., 1999; Gandomi et al., 2013; Maputi and Arora, 2019). 대표적인 기존 연구들을 살펴보면 다음과 같다.

Wu and Chow(1995)는 이산변수, 실수변수, 정수변수가 함께 혼합되어 사용되는 코일 압축 스프링(Coil Compression Spring) 설계문제를 제시하였다. 해당 설계문제에 대해 유전알고리즘(Genetic Algorithm: GA)을 적용하여 그 최적해를 구하였으며, 기존 연구들 (Amir and Hasegawa, 1989; Sandgran, 1990; Fu et al. 1991)에서 제안된 접근법들과의 비교분석을 통해 구해진 최적해의 우수성을 입증하였다. Lin and Hajela (1991), Yokota et al. (1999) 또한 GA를 이용하여 공학설계 최적화 문제를 해결한 방법론을 제시하였다. Gandomi et al. (2013)는 메타휴리스틱 접근법 중의 하나인 쿠쿠서치(Cuckoo Search)를 이용하여 주름진 격벽 설계(Corrugated Bulkhead Design) 문제, 3개의 막대 트러스 설계(Three-bar Truss Design) 문제 등 다양한 공학설계 최적화 문제를 해결하였다. 비교적 최근의 연구로는 Maputi and Arora (2019) 이 제안한 TLBO(Teaching Learning-Based

Optimization) 접근법이 있다. 이 접근법은 교사와 학생의 지도, 토론 및 학습 과정을 이용하여 최적해로의 탐색을 유도해 주는 탐색적 접근법이며, 메타휴리스틱 접근법 중의 하나이다. 이 연구에서는 가중치를 가진 기어 설계(Weight Gear Design)문제, 기어 트랜스미션 시스템 설계(Gear Transmission System Design) 문제 등에 TLBO 접근법을 적용하여 그 최적해를 구하였으며, 이를 기존의 다양한 메타휴리스틱 접근법들(GA, Simulated Annealing(SA), Particle Swarm Optimization(PSO) 등)과의 비교연구를 수행하였다.

위에서 언급된 메타 휴리스틱 접근법을 이용하여 공학설계 최적화 문제를 해결한 기존연구들(Wu and Chow, 1995; Lin and Hajela, 1991; Yokota et al., 1999; Gandomi et al., 2013; Maputi and Arora, 2019) 대부분은 이산변수, 정수변수, 0-1변수, 실수변수의 형태를 변경시키지 않고 그대로 해당 문제에 적용할 수 있기 때문에 Amir and Hasegawa (1989), Sandgran (1990), Fu et al. (1991), Kuo et al. (2001), Yarushkina (2002) 등이 제안한 접근법들보다 더 우수한 해를 구할 수 있다는 장점이 존재한다. 하지만 이러한 장점에도 불구하고 GA, SA, TLBO, PSO 접근법 등과 같은 기존의 단일(Single) 메타휴리스틱 접근법들을 개별적으로 적용할 때보다 각각의 장점을 서로 연계하여 함께 사용하는 혼합(Hybrid) 메타휴리스틱 접근법들이 최근에 개발되고 있다. 예를 들어 GA접근법의 경우 전역적 탐색(Global Search) 능력은 우수하지만, 지역적 탐색(Local Search) 능력은 떨어진다. 따라서 지역적 탐색 능력이 우수한 SA접근법, 언덕 오르기(Hill Climbing: HC) 접근법 등을 GA접근법과 함께 사용할 경우 전역적 탐색능력과 지역적 탐색능력을 함께 가지기 때문에 기존의 단일 메타휴리스틱 접근법을 사용할 경우보다 더 우수한 해를 구할 수 있다는 연구들이 발표되고 있다. (Castelli and Vanneschi, 2014; Babaie-Kafaki et al., 2016; Dib et al., 2017; Gen et al., 2018; Yun et al., 2020; Sbai et al., 2020)

따라서 본 연구에서도 혼합 메타휴리스틱 접근법을 이용하여 공학설계 최적화 문제를 해결하고

자 한다. 제안된 GA-VNS-HC 접근법은 전역적 탐색을 위해 GA 접근법을 사용하고, GA 접근법을 적용한 후 수렴된 해의 주변에 대해 지역적 탐색을 실시하는 변동이웃탐색(VNS) 접근법과 HC 접근법을 각각 적용하여 최적해를 좀 더 효율적으로 찾도록 하였다.

본 연구의 구성은 다음과 같다. 1장에서는 본 연구 수행의 배경과 목적을 설명하였고, 2장에서는 공학설계 최적화 문제의 특징에 대해 간략히 살펴본다. 3장에서는 본 연구의 핵심 내용으로 제안된 GA-VNS-HC 접근법의 특징 및 구체적인 이행전략을 제시한다. 4장에서 다양한 형태의 공학설계 최적화 문제를 제시하고 이를 3장에서 제시한 GA-VNS-HC 접근법에 적용한다. 또한 기존의 다양한 단일 및 혼합 메타휴리스틱 접근법을 함께 적용하여 본 연구에서 제안한 GA-VNS-HC 접근법의 우수성을 입증하고자 한다. 마지막 5장은 본 연구의 결론으로서 전체 연구 수행에 대한 요약과 향후 개선방향 등에 대해 논의한다.

2. 공학설계 최적화 문제의 수리모형

최소화문제를 고려할 경우, 공학설계 최적화 문제의 일반적인 구조는 다음과 같다. (Yun and Moon, 2003)

$$\text{minimize } f(X) \quad (1)$$

$$\text{subject to } g_i(X) = 0 \quad i=1, \dots, k \quad (2)$$

$$g_i(X) \geq 0 \quad i=k+1, \dots, I \quad (3)$$

$$x_j^L \leq x_j \leq x_j^U \quad j=1, \dots, J \quad (4)$$

where

f : a scalar objective

$$X = \{x_1, x_2, \dots, x_J\}^T = \{X^C, X^Z, X^I, X^D\}^T$$

x_j^L : set the lower bounds on the design variables x_j

x_j^U : set the upper bounds on the design variables x_j

$X^C \in R^C$: feasible subset of continuous variables

$X^Z \in R^Z$: feasible subset of 0-1 variables

$X^I \in R^I$: feasible subset of integer variables

$X^D \in R^D$: feasible subset of discrete variables

수식 (1)은 공학설계 최적화 문제에서 다루어지는 설계 비용의 최소화 혹은 제작비용의 최소화 관련되며, 주로 선형(Linear) 혹은 비선형(Non-Linear) 함수 형태로 표현된다. 만일 수식 (1)이 비선형 함수 형태로 표현되고, 사용되는 설계변수가 이산변수, 정수변수, 0-1변수, 실수변수 형태로 다양하게 사용될 경우, 그 최적해를 구하기 위해서는 설계변수의 형태가 하나의 단일 형태로 전부 변경되는 것이 일반적이다(Amir and Hasegawa, 1989; Sandgran, 1990; Fu et al., 1991) 하지만, 이미 서론에서도 언급하였듯이 이렇게 변수의 형태를 하나로 단일화하여 적용할 경우, 구하고자 하는 해의 품질이 떨어지거나, 최적해를 찾지 못하고 조기 수렴되는 문제점이 존재한다. 수식 (2)-(4)는 수식 (1)의 최소화 문제를 해결하기 위해 사용되는 제약조건과 각 설계 변수의 형태를 표시하고 있다.

3. GA-VNS-HC 접근법

2장에서 이미 언급하였듯이 공학설계 최적화 문제에서 사용되는 수리모형의 형태가 선형 혹은 비선형 함수로 표현되며, 설계변수의 형태가 다양하고 혼합되어 사용되기 때문에 그 최적해를 효율적으로 구하기 위해서는 설계변수의 형태를 변형시키지 않고 그대로 사용하는 것이 보다 더 효율적이다(Lin and Hajela, 1991; Wu and Chow, 1995; Yokota et al., 1999; Gandomi et al., 2013; Maputi and Arora 2019).

메타 휴리스틱 접근법들은 공학설계 최적화 문제의 설계변수 형태를 그대로 사용하여 그 최적

해를 구할 수 있으며, 구해진 최적해가 기존의 접근법들을 사용한 연구들(Amir and Hasegawa, 1989, Sandgran, 1990; Fu et al., 1991; Kuo et al., 2001; Yarushkina, 2002)보다 더 효율적이라고 증명되었다. 하지만 기존의 단일 메타 휴리스틱 접근법들이 개별적으로 적용될 때보다 각각의 장점을 함께 사용하는 혼합 메타휴리스틱 접근법이 최적해를 찾는 데 더 효율적이라고 알려져 있다(Castelli and Vanneschi, 2014; Babaie-Kafaki et al., 2016; Dib et al., 2017; Gen et al., 2018; Yun et al., 2020; Sbai et al., 2020).

따라서 본 연구에서도 이러한 경향에 따라 공학설계 최적화 문제를 효율적으로 해결하기 위한 혼합 메타휴리스틱 접근법을 제안한다. 제안된 GA-VNS-HC 접근법은 먼저 해의 탐색공간 전체에 대해 전역적 탐색을 실시하는 GA 접근법을 사용하고, GA 접근법을 적용한 후 수렴된 해의 주변에 대해 정밀하게 지역적 탐색을 실시하는 VNS 접근법과 HC 접근법을 각각 적용하여 최적해를 좀 더 효율적으로 찾도록 하였다. GA와 VNS 접근법을 함께 사용한 혼합 메타휴리스틱 접근법의 경우, 이미 기존 연구들(Castelli and Vanneschi, 2014; Wen et al., 2011; Sbai et al., 2020)에서 제안되었다. 이들의 연구와 본 연구에서 제안한 GA-VNS-HC 접근법의 차이점은 다음과 같다.

첫째, 기존 연구들에서 제안한 접근법은 GA와 VNS 접근법만을 연계한 것이다. 하지만 본 연구에서 제안한 GA-VNS-HC 접근법은 GA와 VNS 접근법에 추가하여 HC 접근법을 함께 사용하였다. 전자 및 후자의 접근법들 모두 전역적 탐색을 실시하는 GA 접근법과 지역적 탐색을 실시하는 VNS 접근법을 공통으로 적용하고 있다. 하지만 VNS 접근법의 경우 그 탐색과정에서 특정 지역을 집중적으로 깊이 있게 탐색하기보다는 일정한 횟수만큼만 탐색을 실시한 후, 만일 우수해가 발견되지 않을 경우, 현재의 탐색공간에서 이웃(Neighborhood)하는 새로운 탐색공간으로 이동한 후 다시 탐색을 실시하는 특징을 가지고 있다. 즉 보다 다양한 탐색공간을 활용하여 탐색을 실시할 수는 있지만 특정 지역에 집중적으로 깊이 있게 탐색하는 능력은 떨어진다. 이러

한 VNS 접근법의 적용상 한계점을 극복하기 위해 본 연구에서는 VNS 접근법의 적용이후 수립된 탐색공간에 대해 보다 집중적으로 재 탐색을 실시할 수 있는 HC 접근법을 추가적으로 사용한다. GA와 VNS 접근법을 함께 사용하는 기존연구들의 수행도와 GA, VNS 및 HC 접근법을 함께 사용한 본 연구의 접근법의 수행도는 4장 사례연구에서 비교분석된다.

둘째, 기존의 연구들과 본 연구에서 제안하는 GA-VNS-HC 접근법들 모두는 GA와 VNS 접근법을 함께 사용하고 있다. 하지만 VNS 접근법의 적용상의 방법에는 약간의 차이가 존재한다. 즉 Wen et al. (2011)의 연구와 본 연구에서는 GA 접근법 수행 이후 생성된 새로운 집단의 해들 중에서 우수해 10%만을 이용하여 VNS 접근법을 적용한다. 하지만 Castelli and Vanneschi (2014)의 연구에는 100%(즉, GA 접근법 수행 이후 생성된 새로운 집단의 해들 전부)를 사용하고, Sbai et al. (2020)의 연구에서는 GA 접근법의 적용과정에서 사용하는 유전 연산자(Genetic Operator)인 돌연변이 연산자(Mutation Operator)를 사용하지 않고 그 대신에 VNS 접근법을 사용한다는 차이점이 존재한다. 이러한 VNS 접근법의 적용상 차이점은 최적해로의 탐색과정에서 해의 품질을 결정하는 요인이 될 수 있다. 기존 연구들(Wen et al., 2011; Castelli and Vanneschi, 2014; Sbai et al., 2020)의 수행도와 본 연구의 수행도 차이점은 4장 사례연구에서 비교분석된다.

본 연구에서 제안하는 GA-VNS-HC 접근법의 이행과정을 간략히 설명하면 다음과 같다.

Step 1: GA 접근법

Step 1.1: 부모집단을 랜덤하게 생성한다.

Step 1.2: 생성된 부모집단에 대해 교차변이 연산자(Crossover Operator)와 돌연변이 연산자(Mutation Operator) 연산자를 적용한다. 이러한 과정을 통해 새로운 자손집단이 생성된다.

Step 1.3: 자손집단의 해들 중에서 일정비율의 우수해를 선택한다.

Step 2: VNS 접근법

Step 2.1: Step 1.3을 통해 선택된 우수해에 대해 그 주변지역에서 랜덤하게 하나의 해를 생성한다.

Step 2.2: 생성된 랜덤해에 대해 삽입 연산자(Insertion Operator) 혹은 교체 연산자(Swap Operator) 연산자를 적용하여 새로운 해를 생성한다.

Step 2.3: 생성된 새로운 해가 Step 2.1의 우수해보다 더 우수할 경우에는 새로운 해가 우수해를 대체하고, 그렇지 않을 경우에는 우수해의 주변지역을 새롭게 변경하여 다시 탐색을 실시한다.

Step 2.4: 선택된 모든 우수해에 대해 Step 2.1에서 Step 2.3까지의 절차를 반복한다. 이러한 반복과정을 통해 가장 우수한 해를 저장한다.

Step 3: HC 접근법

Step 3.1: Step 2.4 적용 이후 저장된 가장 우수해의 주변지역에서 새로운 해를 랜덤하게 생성한다.

Step 3.2: 생성된 새로운 해가 우수해보다 더 우수할 경우에는 새로운 해가 우수해를 대체한다.

Step 3.3: 사전에 설정된 반복수 만큼 Step 3.1에서 Step 3.2까지의 절차를 반복한다.

Step 4: 해의 개선 및 우수해 저장

Step 4.1: Step 3의 적용과정 이후 생성된 우수해를 Step 1.3의 자손집단에 포함시키고 자손집단 전체의 해 중에서 가장 우수한 해를 저장한다.

Step 4.2: 선택연산자(Selection Operator)를 Step 1.1의 부모집단과 Step 4.1의 자손집단에 적용하여 새로운 부모집단을 생성한다.

Step 4.3: 사전에 설정된 전체 반복수 만큼 Step 1.2에서 Step 4.2까지의 절차를 반복한다. 이러한 반복과정을 통해 가장 우수한 해를 저장한다.

GA-VNS-HC 접근법의 보다 구체적인 이행과정을 프로그램 설계코드(Pseudo Code)로 나타내면 Fig. 1과 같다.

```

procedure: GA-VNS-HC approach
input:  $t$ : number of total iterations;
         $PP(t)$ : parent population;
         $OP(t)$ : offspring population;
         $C_{best}$ : global best solution;
         $L_{best}$ : local best solution;
         $C_{best}$ : current best solution
begin
   $C_{best} = 0$ ;
   $t \leftarrow 1$ ;
  randomly generate  $PP(t)$  under satisfying all
  constraints;
  while (not pre-determined termination condition)
    apply non-uniform arithmetic crossover operator
    (Gen and Cheng, 1997) and random mutation
    operator (Gen and Cheng, 1997) to  $PP(t)$  for
    obtaining  $OP(t)$ ;
    sorting  $OP(t)$ ;
    extract some best solutions  $B_i$  from  $OP(t)$ ;
    update  $B_i$  using the procedure of VNS
    approach;
    // goto VNS approach
    insert the updated  $B_i$  into  $OP(t)$  and store  $C_{best}$ 
    in  $OP(t)$ ;
    update  $C_{best}$  using the procedure of HC
    approach;
    // goto HC approach
    insert the updated  $C_{best}$  into  $OP(t)$ ;
    re-sort and evaluate  $OP(t)$ ;
     $L_{best} \leftarrow$  best solution in  $O(t)$ ;
    if  $C_{best} < L_{best}$  then
       $C_{best} \leftarrow L_{best}$ 
    end
     $t = t + 1$ 
    generate new  $PP(t)$  after applying elitist
    selection strategy in enlarged sampling space
    (Gen and Cheng 1997) to  $PP(t)$  and  $OP(t)$ ;
  end
output:  $C_{best}$ 
end

```

Fig. 1 Detailed Imprementation Procedure of GA-VNS-HC Approach

Fig. 1에서 사용되는 VNS와 HC 접근법의 이행 절차는 각각 Fig. 2, Fig. 3과 같다.

4. 사례 연구

사례연구에서는 공학설계 최적화 문제에서 많이 다루어진 여섯 가지 형태의 문제를 이용하며, 이를 간략히 정리하면 Table 1과 같다. Table 1에서 Case 1, 3의 경우에는 이산변수, 실수변수, 정수변수가 혼합되어 사용되었고, Case 4의 경우에는 이산변수와 실수변수가 혼합되어 사용되었다. 하지만 Case 2의 경우에는 전부 정수변수가 사용되었고, Case 5, 6의 경우에는 전부 실수변

수가 사용되었다. Table 1에서 제시된 여섯가지 공학설계 최적화 문제들은 본 연구에서 제안된 GA-VNS-HC 접근법을 포함하여 두 개의 기존 단일 메타휴리스틱 접근법(GA, VNS)과 세 개의 기존 혼합 메타휴리스틱 접근법(GA-VNS1, GA-VNS2, GA-VNS3)에 적용된다. 이를 간략히 정리하면 Table 2와 같다.

```

procedure: VNS Approach
input:  $B_{num}$ : number of  $B_i$ ;
         $S_{max}$ : maximum number of neighborhood structure;
         $I_{max}$ : maximum number of iterations in
        neighborhood structure
begin
  for  $i = 1$  to  $B_{num}$ 
     $j = 1$ 
    while  $j < S_{max}$ 
       $k = 1$ 
      while  $k < I_{max}$ 
        randomly generate a new solution  $B_{nbr}$ 
        from  $j^{\text{th}}$  neighborhood of  $B_i$ ;
        randomly select one operator between
        insertion and swap operators(Wang et
        al., 2018);
        apply the selected operator to  $B_{nbr}$  for
        generating a new solution  $B_{new}$ ;
        if  $f(B_i) < f(B_{new})$  then
           $B_i \leftarrow B_{new}$ ,  $j \leftarrow 1$ ,  $k \leftarrow k + 1$ 
          if  $k = I_{max}$  then
             $j \leftarrow j + 1$ 
          end
        else
           $j \leftarrow j + 1$ ,  $k \leftarrow I_{max}$ 
        end
      end
    end
  next
output:  $B_i$ 
end

```

Fig. 2 Detailed Implementation Procedure of VNS Approach

```

procedure: HC Approach
input:  $H_{max}$ : maximum number of iterations;
         $N_{size}$ : size of neighborhood structure
begin
  for  $i = 1$  to  $H_{max}$ 
    randomly generate a new solution  $C_{new}$  within
     $N_{size}$  of  $C_{best}$ ;
    if  $C_{best} < C_{new}$  then
       $C_{best} \leftarrow C_{new}$ 
    end
     $i \leftarrow i + 1$ 
  next
output:  $C_{best}$ 
end

```

Fig. 3 Detailed Implementation Procedure of HC Approach

Table 2에서 보듯이 GA와 VNS 접근법을 비교대상으로 사용한 이유는 본 연구에서 제안된 GA-VNS-HC 접근법이 GA와 VNS 접근법을 기본적으로 포함하기 있기 때문이며, 단일 메타휴리스틱 접근법(즉 GA, VNS)의 수행도와 본 연구에서 제안된 혼합 메타휴리스틱 접근법(즉 GA-VNS-HC)의 수행도와의 비교를 위한 것이다. 또한 세 종류의 혼합 메타휴리스틱 접근법(즉 GA-VNS1, GA-VNS2, GA-VNS3)을 비교대상으로 사용한 이유는 본 연구에서 제안된 GA-VNS-HC 접근법이 GA와 VNS 접근법과 함께 추가적으로 HC 접근법을 가지고 있기 때문에 HC 접근법을 추가적으로 가지고 있으나 여부에 따라 그 수행도를 비교분석해 보기 위함이다.

Table 1 Engineering Design Optimization Problems for Comparison

Case	Problem	Reference
1	Reinforced Concrete Beam Problem	Amir and Hasegawa (1989)
2	Gear Train Problems	Sandgren (1990)
3	Coil Compression Spring Problem	Wu and Chow (1995)
4	Pressure Vessel Problem	Fu et al. (1991)
5	Corrugated Bulkhead Problem	Kvalie (1967)
6	Speed Reducer Problem	Gandomi and Yang (2011)

Table 2 Various Approaches for Comparison

Approach	Description
GA	Single meta-heuristic approach using GA (Gen and Cheng, 1997)
VNS	Single meta-heuristic approach using VNS (Mladenović and Hansen, 1997)
GA-VNS1	Hybrid meta-heuristic approach using GA and VNS (Sbai et al. 2020)
GA-VNS2	Hybrid meta-heuristic approach using GA and VNS (Wen et al., 2011)
GA-VNS3	Hybrid meta-heuristic approach using GA and VNS (Castelli and Vanneschi, 2014)
GA-VNS-HC	Hybrid meta-heuristic approach using GA, VNS and HC (This paper)

GA-VNS1, GA-VNS2, GA-VNS3 접근법 모두는 GA와 VNS 접근법을 동일하게 가지고 있지만 GA와 VNS를 어떻게 결합하느냐에 따라 약간의 차이가 있다. 먼저 GA-VNS1의 경우, GA 접근법을 적용할 때 사용하는 유전연산자 중에서 돌연변이 연산자를 사용하지 않고 그 대신에 VNS 접근법을 사용한다(Sbai et al., 2020). GA-VNS2와 GA-VNS3의 차이점은 전자의 경우에는 GA 접근법 적용 이후 생성된 자손집단 중에서 우수해를 10%만 사용(Wen et al., 2011)

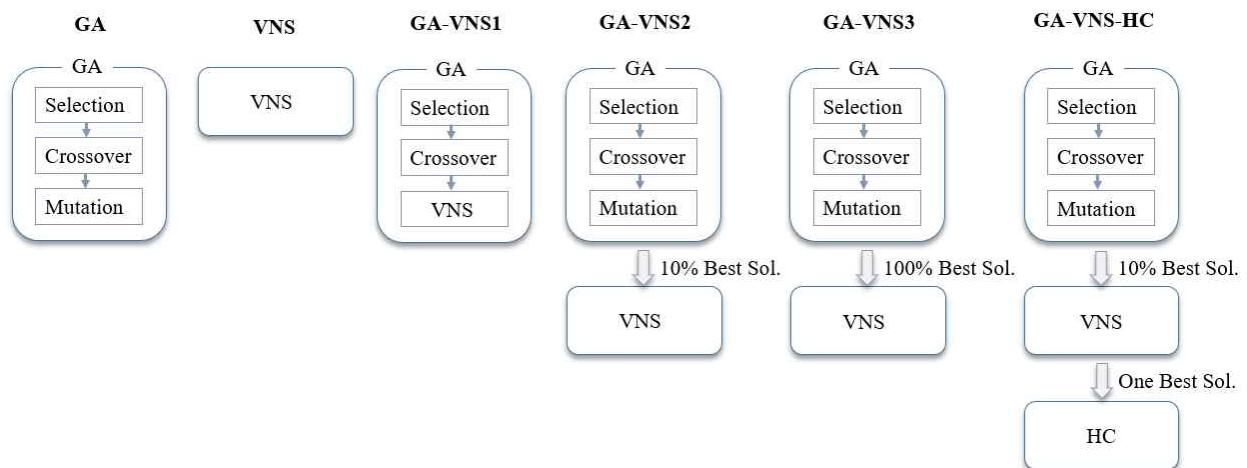


Fig. 4 Detailed Comparison among All Approaches

하고 후자의 경우에는 우수해 전체(100%)를 사용 (Castelli and Vanneschi, 2014)한다. GA-VNS2 접근법과 본 연구에서 제안된 GA-VNS-HC 접근법은 GA 적용 후 생성된 자손집단 중에서 우수해 10%를 사용하고 있다는 측면에서는 동일하지만, 전자의 경우에는 VNS 접근법 적용 이후 수렴된 탐색공간에 대해 보다 집중적으로 재 탐색을 실시할 수 있는 HC 접근법을 사용하지 않는다. 하지만 후자의 경우에는 이를 함께 사용한다는 것이 가장 큰 차이점이라고 할 수 있다. Fig. 4는 Table 2에서 제시한 접근법들 간의 중요한 차이점을 보다 상세히 묘사하고 있다.

Fig. 4에 제시된 모든 접근법들은 동일한 컴퓨터 환경(MATLAB: Ver. 2014b, CPU: 1.3 GHz Intel Core I5-1600, RAM: 4GB, OS: OS-X EI)에서 프로그래밍되고 실행되었다. GA, GA-VNS1, GA-VNS2, GA-VNS3, GA-VNS-HC 접근법에서 공통으로 사용된 GA 접근법의 파라미터 설정은 다음과 같다. 집단의 크기: 10, 교차변이 비율: 0.3, 돌연변이 비율: 0.2. VNS, GA-VNS1, GA-VNS2, GA-VNS3, GA-VNS-HC 접근법에서 공통으로 사용된 VNS 접근법의 파라미터 설정은 다음과 같다. 이웃 구조의 수 (Number of Neighborhood Structure): 10, 이웃 구조에서 반복수(Number of Iteration in Neighborhood Structure): 2,000. 또한 GA-VNS-HC 접근법에서 단독으로 사용된 HC 접근법의 파라미터 설정은 다음과 같다. 반복수 (Number of Iteration): 2,000, 이웃 구조의 크기 (Size of Neighborhood Structure): 10. 모든 접근법들은 총 2,000번의 반복수를 동일하게 가지며, 프로그램 실행 중에 발생하는 다양한 랜덤성을 줄이기 위해 총 10회 반복 수행하였다.

Fig. 4에 제시된 모든 접근법들은 Table 3에서 제시된 수행도 척도에 의해 비교 분석된다.

Table 2의 모든 접근법들은 Table 3에서 제시한 세 가지 수행도 척도를 이용하여 실험을 실시하였으며 그 결과는 Table 4, 5와 같다. Table 4, 5에서 제시된 Gap 1 및 Gap 2의 값들은 GA-VNS-HC 접근법의 BS 및 AS 결과와 기존의 접근법들(GA, VNS, GA-VNS1, GA-VNS2, GA-VNS3)의 BS 및 AS 결과의 차이를 각각 나타내는 것으로, 이 값이 만일 + 값으로 나타날 경우에는 GA-VNS-HC 접근법의 결과가 기존 접근법들의 결과보다 더 우수하다는 것을 의미이며, - 값으로 나타날 경우에는 그 반대로 해석하면 된다.

Table 3 Performance Measures

Measure	Description
BS	Best solution(BS) in all trials
AS	Average solution(AS) through all trails
CPU	Averaged CPU time (CPU) through all trails (unit: sec.)

Table 4, 5의 결과를 분석하면 다음과 같다. Case 1의 경우에 있어서 수행도 척도 BS의 결과에 의하면, GA-VNS-HC 접근법에서 364.857의 값을 구하였고, 이 값은 GA, VNS, GA-VNS1, GA-VNS2, GA-VNS3 접근법에서 각각 구해진 364.899(0.01%), 365.039(0.05%), 365.366(0.14%), 365.728(0.24%), 364.982(0.03%)의 값과 거의 같거나 약간 더 우수한 것을 알 수 있다. 또한 수행도 척도 AS의 결과에 의하면, GA-VNS-HC 접근법에서 366.056의 값을 구하였고, 이 값은 GA, GA-VNS2에서 각각 구해진 368.840(0.76%), 369.389(0.915%)의 값과 거의 같거나 약간 더 우수한 것을 알 수 있다. 하지만 전자의 값은 VNS, GA-VNS1, GA-VNS3 접근법에서 각각 구해진 373.382(2%), 371.182(1.40%), 373.570

Table 4 Computation Results among GA, VNS, and GA-VNS1

Case	GA					VNS					GA-VNS1				
	BS	AS	CPU	Gap 1	Gap 2	BS	AS	CPU	Gap 1	Gap 2	BS	AS	CPU	Gap 1	Gap 2
1	364.899	368.840	1.100	0.01%	0.76%	365.039	373.382	0.520	0.05%	2.00%	365.366	371.182	0.480	0.14%	1.40%
2	1.17E-10	2.72E-07	0.009	4217.58%	4711.46%	9.92E-10	7.43E-04	0.0034	36634.93%	13136281.88%	2.73E-08	2.02E-03	0.126	1009376.05%	35740292.96%
3	2.491	4.373	67.300	19.57%	94.68%	2.115	2.293	57.700	1.51%	2.08%	2.210	3.431	63.400	6.10%	52.75%
4	7216.367	7258.080	0.720	0.21%	0.34%	7209.214	7384.043	0.770	0.11%	2.08%	7230.674	7375.936	0.680	0.41%	1.97%
5	7.093	7.494	0.096	7.43%	12.76%	7.266	8.056	0.0092	10.05%	21.22%	6.959	7.194	0.120	5.39%	8.24%
6	3002.660	3006.601	0.042	3.53%	3.09%	3126.468	3314.213	0.044	7.80%	13.64%	3058.736	3263.021	0.025	5.47%	11.88%

Table 5 Computation Results among GA-VNS2, GA-VNS3, and GA-VNS-HC

Case	GA-VNS2					GA-VNS3					GA-VNS-HC		
	BS	AS	CPU	Gap 1	Gap 2	BS	AS	CPU	Gap 1	Gap 2	BS	AS	CPU
1	365.728	369.389	0.498	0.24%	0.91%	364.982	373.570	0.508	0.03%	2.05%	364.857	366.056	1.600
2	2.36E-09	2.63E-06	0.112	87192.31%	46293.01%	2.36E-09	1.29E-07	0.162	87192.31%	2183.97%	2.70E-12	5.66E-09	0.412
3	2.371	2.831	57.600	13.80%	26.06%	2.272	2.835	58.100	9.06%	26.22%	2.0834	2.2461	71.900
4	7222.186	7385.508	0.520	0.29%	2.10%	7251.418	7389.961	0.440	0.70%	2.16%	7201.14	7233.580	1.090
5	7.055	7.926	0.077	6.85%	19.27%	7.039	7.555	0.0132	6.61%	13.67%	6.6026	6.6458	0.592
6	3148.749	3258.778	0.028	8.57%	11.74%	3079.680	3196.646	0.024	6.19%	9.61%	2900.18	2916.411	0.168

(2.05%)의 값 보다 상당히 우수한 것을 알 수 있다. 수행도 척도 CPU의 결과에 의하면, GA-VNS1, GA-VNS2, GA-VNS3, VNS, GA 접근법이 각각 0.480, 0.498, 0.508, 0.520, 1.100 순서로 빠른 수행도를 보여주었으며, GA-VNS-HC 접근법의 경우 1.600으로 가장 느린 수행도를 보여주고 있다.

Case 2의 경우에는 BS에 있어서 GA-VNS-HC 접근법에서 구해진 2.70E-12 값이 기존 접근법들 (GA, VNS, GA-VNS1, GA-VNS2, GA-VNS3)에서 각각 구해진 1.17E-10, 9.92E-10, 2.73E-08, 2.36E-09, 2.36E-09 값들보다 훨씬 더 우수한 수행도를 보여주고 있으며, AS에 있어서도 GA-VNS-HC 접근법에서 구해진 값이 기존 접근법들에서 각각 구해진 값들보다 훨씬 더 우수한 수행도를 보여주고 있는 것을 알 수 있다. 하지만 CPU 수행도 척도에서는 VNS 접근법의 결과값이 가장 우수하고, GA-VNS-HC 접근법의 결과값이 가장 열등한 것을 알 수 있다. Case 3, 4, 5, 6의 경우에 있어서 BS 및 AS에서 GA-VNS-HC 접근법의 결과값이 가장 우수하며, CPU에서는 GA-VNS-HC 접근법의 결과값이 가장 열등한 것을 알 수 있다. Table 4, 5의 결과를 종합적으로 분석하면 BS, AS에서는 GA-VNS-HC 접근법이 가장 우수하지만, CPU에서는 GA-VNS-HC 접근법이 가장 열등한 것을 알 수 있다.

Fig. 5, 6은 Case 5, 6에 대해 BS에서 각 접근법들의 수렴과정을 보여주고 있다. Fig. 5를 살펴보면, GA 및 GA-VNS1 접근법의 경우에는 반복수가 각각 약 35회 전후와 약 70회 전후에서 급격한 수렴현상을 보이지만, 그 이후에는 별다른 수렴현상을 보이지 않고 있다. 하지만 VNS, GA-VNS2, GA-VNS3, GA-VNS-HC 접근법들

의 경우에는 반복 수가 증가할수록 지속적인 수렴현상을 보여주고 있다.

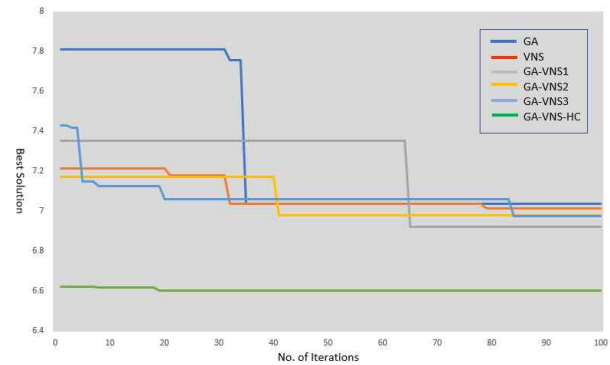


Fig. 5 Convergence Behaviors of BS for Case 5

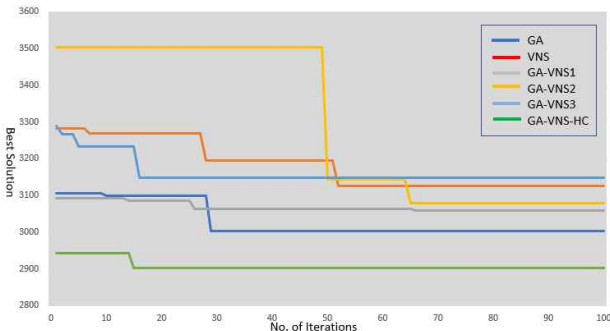


Fig. 6 Convergence Behaviors of BS for Case 6

특히 GA-VNS-HC 접근법들의 경우에는 초기 반복수 단계에서부터 다른 접근법들보다 더 우수한 적합도 값을 가지며 지속적인 수렴현상을 보이고 있는 것을 알 수 있다. Fig. 6을 살펴보면, GA-VNS3 접근법을 제외하고 비교대상인 모든 접근법들이 초기 반복수 단계부터 급격하고 빠른 수렴 현상을 지속적으로 보여주고 있다. 이 중에서도 GA-VNS-HC 접근법의 수행도가 초기 반복수 단계부터 더 우수한 수렴현상을 보여주고 있는 것을 알 수 있다. Fig. 5, 6의 수렴현상에 대한 비교분석결과를 종합하면, 본 연구에서 제

안한 GA-VNS-HC 접근법이 기존 접근법들보다 더 빠르고 신속한 수렴현상을 보여주고 있으며, 이러한 결과는 Table 4, 5에서 Case 5, 6의 BS의 계산결과를 뒷받침하고 있다.

Fig. 7, 8에서는 Case 5, 6에 대해 GA-VNS3 및 GA-VNS-HC 접근법의 AS에서 수렴과정을 보여주고 있다.

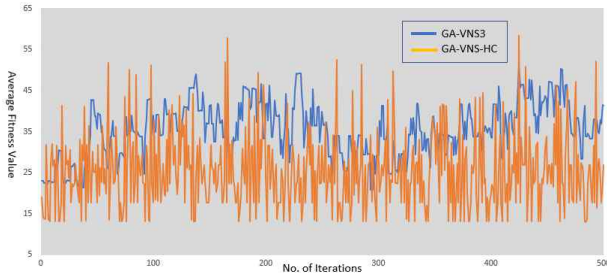


Fig. 7 Average Fitness Behaviors of GA-VNS3 and GA-VNS-HC Approaches for Case 5

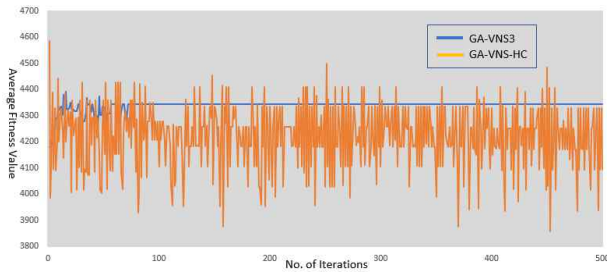


Fig. 8 Average Fitness Behaviors of GA-VNS3 and GA-VNS-HC Approaches for Case 6

Fig. 7, 8은 GA-VNS3 및 GA-VNS-HC 접근법들이 최적해를 탐색해 가는 과정에서 평균적합도 값이 어떻게 변화하고 있는지를 보여주고 있다. 여기서 평균적합도값은 GA-VNS3 접근법의 경우에는 GA 탐색과정 이후의 집단 평균적합도값과 VNS 탐색과정 이후의 집단 평균적합도값을 평균한 값의 변화를 의미하며, GA-VNS-HC 접근법의 경우에는 GA 탐색과정 이후의 집단 평균적합도값과 VNS 및 HC 탐색과정 이후의 집단 평균적합도값을 평균한 값의 변화를 의미한다.

Fig. 7을 살펴보면, GA-VNS3 및 GA-VNS-HC 접근법들은 반복수 증가에 따라 평균적합도값이 다양한 변동폭을 가지며 변화하고 있는 것을 알 수 있다. 하지만 GA-VNS-HC 접근법이

GA-VNS3 접근법보다 더 변동폭이 넓고, 더 다양한 변동현상을 보여주고 있는 것을 알 수 있다. 이것은 GA-VNS-HC 접근법이 최적해를 탐색하는 과정에서 GA-VNS3 접근법보다 탐색공간을 더 넓게, 더 다양하게 탐색할 수 있다는 것을 의미한다. Fig. 8을 살펴보면, GA-VNS3 접근법의 경우에는 초기 단계에서 약간의 변동현상을 보이고 있지만 그 단계 이후에는 전혀 변동현상을 보여주지 못하고 있다. 하지만 GA-VNS-HC 접근법은 Fig. 7과 마찬가지로 더 넓은 변동폭과 다양한 변동현상을 보이는 것을 알 수 있다. 이러한 분석결과는 결국 GA-VNS-HC 접근법이 GA-VNS3 접근법보다 최적해를 찾는 데 더 효율적이라는 것을 의미하며, Table 4, 5의 Case 5, 6에서 GA-VNS-HC 접근법이 GA-VNS3 접근법보다 더 우수한 Best Solution(BS)을 제공한다는 것이 이를 증명하고 있다.

5. 결론

본 연구에서는 공학설계 최적화 문제를 효율적으로 해결하기 위한 혼합형 메타휴리스틱(Hybrid Meta-heuristics) 접근법을 제안하였다. 공학설계 최적화 문제는 사용되는 설계변수의 형태가 이산변수, 정수변수, 실수변수, 0-1변수 등으로 다양하게 혼합되어 사용되는 복잡한 선형 혹은 비선형 최적화 문제로 정의될 수 있으며, 기존의 많은 연구들을 통해 이들 설계변수를 효과적으로 다루기 위한 다양한 접근법들이 개발되어져 왔다. 하지만 기존에 제안된 접근법들 대부분은 다양한 변수형태를 변경하여 사용하고 있으며, 이러한 변경과정에서 구해진 해의 품질을 떨어지거나 조기 수렴되는 한계점이 존재한다. 이와 같은 한계점을 개선하기 위해 메타휴리스틱(Meta-Heuristics) 접근법을 이용한 다양한 연구가 활발히 진행되었다. 이들 메타휴리스틱 접근법을 이용할 경우, 공학 설계 최적화 문제에서 사용되는 다양한 설계변수 형태를 변형시키지 않고 그대로 사용할 수 있다는 장점이 존재한다. 하지만 메타휴리스틱 접근법 또한 이들 접근법들이 단일 형태로 사용되었을 때보다 각각의 접근

법들이 가진 장점을 혼합한 혼합형 메타휴리스틱 접근법을 사용하는 것이 최적해로의 탐색에 더 유리하다 것이 기존의 많은 연구들을 통해 증명되었다.

본 연구에서 제안한 혼합 메타휴리스틱 접근법은 탐색 공간에 대한 전역적 탐색을 실시하는 GA 접근법, 지역적 탐색을 실시하는 VNS 접근법과 HC 접근법을 혼합(GA-VNS-HC)하였다. 사례 연구에서는 여섯 가지 형태의 공학설계 최적화 문제를 이용하여 본 연구에서 제안한 GA-VNS-HC 접근법과 기존 연구들에서 제안된 단일 메타휴리스틱(GA, VNS) 접근법 및 혼합 메타휴리스틱(GA-VNS1, GA-VNS2, GA-VNS3) 접근법의 수행도를 함께 비교분석하였다. 분석결과를 수행도 척도 BS, AS, CPU 측면에서 살펴보면 다음과 같이 요약할 수 있다.

첫째, 수행도 척도 BS 및 AS에서는 GA-VNS-HC 접근법의 수행도가 비교대상인 기존의 메타휴리스틱 (GA, VNS, GA-VNS1, GA-VNS2, GA-VNS3) 접근법들의 수행도보다 더 우수한 결과를 보여주었다.

둘째, 수행도 척도 CPU에서는 GA-VNS-HC 접근법의 탐색속도가 기존의 메타휴리스틱 접근법들의 탐색속도보다 더 열등한 결과를 보여준 것을 알 수 있다. 이러한 결과는 전자의 경우, 세 가지 형태의 단일 메타휴리스틱(GA, VNS, HC) 접근법을 함께 사용하여 탐색을 실시하기 때문에 기존의 단일 메타휴리스틱(GA, VNS) 접근법을 사용하는 경우와 두 가지 형태의 메타휴리스틱(GA-VNS1, GA-VNS2, GA-VNS3) 접근법을 함께 사용하여 탐색을 실시하기 경우보다 더 느린 탐색속도를 나타내는 것으로 분석할 수 있다.

비록 본 연구에서 제안한 GA-VNS-HC 접근법이 사례연구에서 고려된 공학설계 최적화 문제에 대해 기존의 메타휴리스틱 (GA, VNS, GA-VNS1, GA-VNS2, GA-VNS3) 접근법들보다 더 우수한 실험결과를 보여주었지만, 탐색속도 측면에서는 좀 더 개선의 여지가 있을 것으로 파악된다. 또한 사례연구에서 고려되지 않은 좀 더 다양한 공학설계 최적화 문제들에 대해서도 기존의 메타휴리스틱 (GA, VNS, GA-VNS1, GA-VNS2, GA-VNS3) 접근법들과 비교적 최근

에 개발된 Maputi et al. (2019)의 GA-TLBO (Teaching Learning-based Optimization) 접근법을 포함하여 본 연구에서 제안한 GA-VNS-HC 접근법에 대해 실험을 실시하고 그 수행도를 비교분석해 보는 노력이 필요할 것으로 본다. 이러한 두 가지 본 연구의 한계점은 추후 개선 사항으로 고려하고자 한다.

References

- Amir, H. M., and Hasegawa, T. (1989). Nonlinear mixed-discrete structural optimization. *Journal of Structural Engineering*, 115(3), 626-646.
- Babaie-Kafaki, S., Ghanbari, R., and Mahdavi-Amiri, N. (2016). Hybridizations of genetic algorithms and neighborhood search metaheuristics for fuzzy bus terminal location problems, *Applied Soft Computing*, 46, 220-229.
- Castelli, M., and Vanneschi, L. (2014). Genetic algorithm with variable neighborhood search for the optimal allocation of goods in shop shelves, *Operations Research Letters*, 42, 355-360.
- Dib, O., Manier, M.-A., Moalic, L., and Caminada, A. (2017). Combining VNS with genetic algorithm to solve the one-to-one routing issue in road networks, *Computers and Operations Research*, 78, 420-430.
- Fu, J. F., Fenton, R. G., and Cleghorn, W. L. (1991). A mixed integer - discrete-continuous programming method and its applications to engineering design optimization, *Engineering Optimization*, 17, 263-280
- Gandomi, A. H., and Yang, X-S. (2011). *Benchmark problems in structural optimization*, In Koziel, S. and Yang, X-S. (Eds.), *Computational Optimization, Methods and Algorithms*, Springer-Verlag,

- Berlin, 267 - 291.
- Gandomi, A. H., Yang, X-S., and Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Engineering with Computers*, 29, 17-35.
- Gen, M., Lin, L., Yun, Y. S., and Inoue, H. (2018). Recent advances in hybrid priority-based genetic algorithms for logistics and SCM network design, *Computers and Industrial Engineering*, 125, 394-412
- Kuo, W., Prasad, V. R., Tillman, F., and Hwang, C. L. (2001). *Optimal Reliability Design: Fundamentals and Applications*, Cambridge Univ. Press, 2001.
- Kvalie, D. (1967). *Optimization of Plane Elastic Grillages*, PhD Thesis, Norges Teknisk Naturvitenskapelige Universitet, Norway.
- Lin, C.-Y., and Hajela, P. (1991). Genetic algorithm in optimization problems with discrete and integer design variables, *Engineering Optimization*, 16,(4), 309-327.
- Maputi, E. S., and Arora, R. (2019). Multi-objective spur gear design using teaching learning-based optimization and decision-making techniques, *Cogent Engineering*, 6, 1665396, doi.org/10.1080/23311916.2019.1665396
- Mladenović, N., and Hansen, P. (1997). Variable neighborhood search, *Computers and Operations Research*, 24(11), 1097-1100.
- Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization, *ASME Journal of Mechanical Design*, 112(2), 223-229.
- Sbai, I., Krichen, S., and Limam, O. (2020). Two meta-heuristics for solving the capacitated vehicle routing problem: The case of the Tunisian post office, *Operational Research*, 20, 2085-2108.
- Wen, Y., Xu, H., and Yang, J. (2011). A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system, *Information Sciences*, 181, 567-581.
- Wu, S. J., and Chow, P. T. (1995). Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization, *Engineering Optimization*, 24, 137-159.
- Yarushkina, N. G. (2002). Genetic algorithms for engineering optimization: theory and practice, *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02)*, 357-362.
- Yokota, T., Taguchi, T., and Gen, M. (1999). A solution method for optimal cost problem of welded beam by using genetic algorithms, *Computers and Industrial Engineering*, 37(1-2), 379-382.
- Yun, Y. S., Chuluunsukh, A., Gen, M. (2020). Sustainable closed-loop supply chain design problem: A hybrid genetic algorithm approach, *Mathematics*, 8(1), 84, https://doi.org/10.3390/math8010084.
- Yun, Y. S., and Moon, C. U. (2003). Comparison of adaptive genetic algorithms for engineering optimization problems, *International Journal of Industrial Engineering*, 10(4), 584-590.



윤영수 (YoungSu, Yun)

- 종신회원
- 대구대학교 산업공학과 학사
- 건국대학교 산업공학과 석사, 박사
- Waseda University 정보생산 시스템연구과 박사

- 현재: 조선대학교 경영학부 교수
- 관심분야: 물류/SCM, 유전알고리즘, 생산최적화