

# 실수코딩 유전자 알고리즘과 개체 군집 최적화를 이용한 구조물의 결함탐지 시뮬레이션

## Simulation of Structural Damage Localization through RCGA and PSO

최 종 헌\*·정 민 중\*\*·고 봉 환†

Jong-Hun Choi, Minjoong Jeong and Bong-Hwan Koh

### 1. 서 론

본 연구에서는 구조물의 동적특성 중 결함에 대한 고유진동수의 민감도를 활용하여 대상 구조물에 임의로 발생시킨 결함의 위치와 크기를 알아내기 위하여 RCGA(Real Coded Genetic Algorithm), PSO(Particle Swarm Optimization) 그리고 LHS-RCGA-IT-14 등 다양한 최적화 알고리즘과 유한요소모델 개선기법을 활용하여 실제 외팔보 구조물에 존재하는 결함의 위치와 크기를 실제 실험을 통해 확인하였다.

### 2. 최적화 알고리즘

#### 2.1 실수코딩 유전자 알고리즘(RCGA)

실수코딩 유전자 알고리즘(RCGA)의 구조는 일반적인 유전자 알고리즘의 구조와 구성이 동일하지만 RCGA는 해 집단의 염색체를 비트 스트링 형태로 표현하는 일반적인 유전자 알고리즘과 달리 해 집단의 염색체를 직접 실수로 표현하여 문제의 설계변수와 정확하게 일치하는 집단을 구성할 수 있다. 또한 해의 정밀도가 향상되며 해 집단과 비트 스트링 염색체 사이의 변환 과정인 인코딩과 디코딩 과정이 생략되어 탐색 속도가 향상된다.

본 연구에서 제안된 알고리즘(LHS-RCGA-IT-14)은 일반적인 RCGA에 바탕을 두고 있으며, 초기에 해 집단을 생성할 때 랜덤방식이 아닌 일정한 순열규칙에 의하여 샘플링을 하는 LHS(Latin Hypercube Sampling)방식을 도입하였으며, 적합도 함수에 따른 선택은 가장 높은 적합도를 갖는 개체를 선발하는 방식(tournament selection)과 달리 적합도를 통해 개체의 순위를 정하여 그 중 상위 14%에 해당하는 개체들 중 임의로 선발하는

방식을 사용하였다. 이렇게 선발된 개체들을 이용하여 IT(Intermediate Tendency) 교배 연산을 통하여 새로운 개체를 생성하였으며, 이 교배 방식은 기존의 사용된 균일 교배 방식보다 보다 폭 넓은 전역적인 탐색을 수행할 수 있는 장점을 갖고 있다. 일반적인 RCGA와 LHS-RCGA-IT-14 알고리즘을 Table 1에 다음과 같이 정리하였다. 정의된 최대 세대 수(maximum generation) N에 도달할 때까지 앞서 제시한 일련의 연산 과정을 수행하도록 하였다.

Table 1 The specific parameter in algorithm

	Simple RCGA	LHS-RCGA-IT-14
Sampling	random	latin hypercube
Selection	tournament	14%(ranking)
Crossover	$p1=f+r*(m-f)$	$p1=f+r*(m-other)$
	$p2=m-r*(f-m)$	$p2=m-r*(f-other)$
Mutation	$new\_p = var_{Lower} + r*(var_{Upper}-var_{Lower})$	

여기서, f와 m은 선택 방식에 의해 선발된 개체를 말하며 r은 0과 1사이의 임의의 수를 p1과 p2는 교배 연산을 통해 생성된 개체를 의미하며 LHS-RCGA-IT-14에서 other는 전체 개체 집단 내에서 임의로 뽑힌 개체를 나타낸다. new\_p는 돌연변이 연산을 통해 새롭게 생성된 개체를 말하며 var<sub>Lower</sub>, var<sub>Upper</sub>는 설계변수가 갖을 수 있는 최소, 최대값을 의미한다.

#### 2.2 개체 군집 최적화(PSO)

PSO는 군집생활을 이루는 물고기나 새 무리의 행동양식을 응용한 기법으로 군집을 구성하는 개체가 군집의 궁극적 목표에 도달하기 위하여 개체 자신과 이웃하는 개체 주변의 지역적 경험을 공유 및 상호작용하여 점진적으로 군집의 목표 즉, 최적의 해를 찾게 되는 과정으로 요약된다.

### 3. 유한요소모델 개선기법

† 교신저자: 동국대학교 기계로봇에너지공학과  
E-mail : bkoh@dongguk.edu  
Tel : (02) 2260-8591, Fax : (02) 2263-9379

\* 동국대학교 기계로봇에너지공학과 대학원

\*\* 한국과학기술정보연구원 선도연구팀

일반적으로 구조물 결함 탐지 문제는 유한요소해석으로 해석된 고유진동수와 진동 실험을 통하여 측정된 실제 구조물의 고유진동수간의 차이를 최소화하는 것으로 정의될 수 있다.

$$J = W_{\omega} J_{\omega} \quad (1)$$

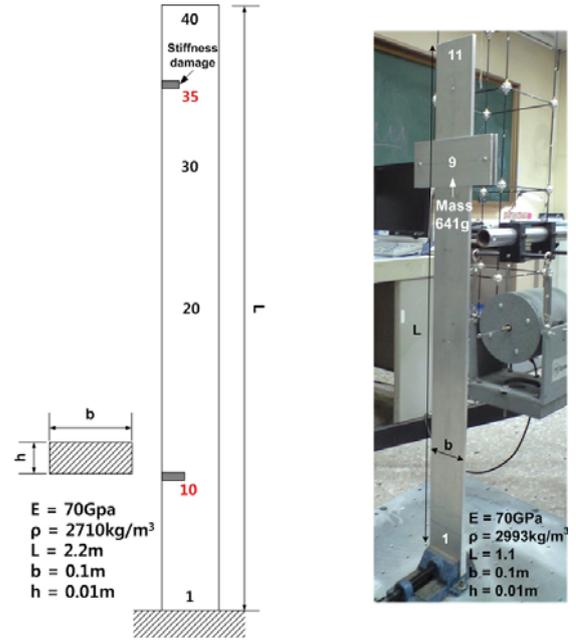
$$J_{\omega} = \sum_{t=1}^n \left( \frac{\omega_t^m - \omega_t^a}{\omega_t^m} \right)^2 \quad (2)$$

여기서  $J_{\omega}$ 는 고유 진동수와 관련된 목적함수가 된다.  $W_{\omega}$ 는  $J_{\omega}$ 의 고유모드 적용 시 사용되는 가중치상수이며 여기서 고유진동수의 유효숫자를 고려하여  $W_{\omega} = 1000$ 을 선택하였다. 식 (2)에서  $n$ 은 목적 함수에 사용된 고유 진동수의 차수를 나타낸다.  $\omega_t^m$ 는 실험 데이터 또는 기준 데이터(reference data)를 말하며  $\omega_t^a$ 는 모델 개선으로 얻어진 데이터를 말한다.

#### 4. 해석 및 실험결과

앞서 제시한 최적화 알고리즘에 대하여 결함 탐지 문제를 검증하기 위하여 Fig 1.(a)과 같이 40개의 요소 중 두 곳의 요소[10,35]에 각각 20%, 10%에 해당하는 강성 결함을 임의로 발생시킨 가상의 외팔보 모델과 Fig 1.(b)와 같이 실제 구조물을 대상으로 결함탐지 문제를 검증하기 위하여 11개의 요소 구분된 외팔보에 한 요소 [9]에 질량을 증가시켜 결함을 발생시킨 모델을 선정하였다. Fig 1.(a)과 같은 가상 구조물의 경우 발생된 결함을 강성요소(stiffness element)의 변화와 동일하다고 가정하여 요소의 강성 변화량을 설계 변수로 설정하였으며, Fig 1.(b)와 같이 실제 구조물의 경우는 질량요소(mass element)의 변화를 결함으로 인식하여 요소의 질량(밀도) 변화량을 설계 변수로 설정하였다. 또한 가상의 외팔보 모델은 12개의 해당하는 고유진동수를 목적 함수에 적용하였고 실제 모델의 경우 Fig. 2와 같이 진동실험을 통하여 측정된 5개의 고유진동수를 목적함수에 적용하여 제시한 알고리즘과 유한요소모델 개선기법을 각각 10회씩 수행하였다. 그 결과 결함 탐지 성공에 대한 각각의 알고리즘의 횟수를 Table 2에 나타내었다.

Table 2를 살펴보면 가상 구조물 모델의 경우 본 연구에서 제시한 LHS-RCGA-IT-14(6번)가 PSO(1번)보다 매우 높은 성공 횟수를 보였다. 그러나 일반적인 RCGA의 경우 외팔보 문제에서 단 한 번도 결함의 위치와 크기를 산출하지 못하였다. 그리고 실제 구조물 모델의 경우 LHS-RCGA-IT-14와 PSO는 동일한 결함 탐지 성능(10번)을 나타내었다.



(a) Stiffness damages (b) Mass damage

Fig. 1 Cantilevered beam

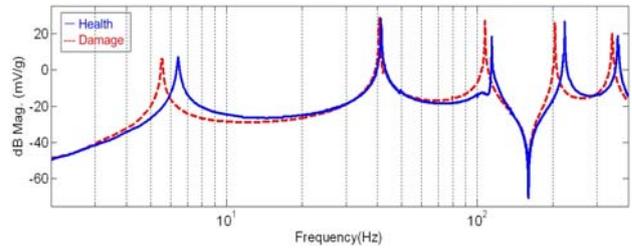


Fig. 2 Measured natural frequencies of cantilever beam

Table 2 Results of the cantilever beam localization

Algorithms	Number of successful	
	Virtual Model	Real Model
Simple RCGA	0	4
LHS-RCGA-IT-14	6	10
PSO	1	10

#### 5. 결론

최근 최적화 문제에 많이 사용되는 PSO와 Simple RCGA, 또 본 연구에서 제안한 LHS-RCGA-IT-14 알고리즘을 가상 또는 실제 구조물의 유한요소모델 개선기법에 적용하여 임의로 발생시킨 강성 또는 질량 결함의 위치와 크기를 실험적으로 찾아내는 문제에 적용하였다. 그 결과 PSO가 Simple RCGA보다 비교적 우수한 성능을 보였으며 LHS-RCGA-IT-14 알고리즘이 PSO보다 상대적으로 더 강건함을 입증하였다.