

구조적 설계문제 최적화를 위한 혼합유전알고리즘

운영수* · 이상용*

Hybrid Genetic Algorithm for Optimizing Structural Design Problems

Young-Soo Yun* · Sang-Yong Yi*

Abstract

Genetic algorithms(GAs) are suited for solving structural design problems, since they handle the design variables efficiently. This ability of GAs considers them as a good choice for optimization problems. Nevertheless, there are many situations that the conventional genetic algorithms do not perform particularly well, and so various methods of hybridization have been proposed.

Thus, this paper develops a hybrid genetic algorithm(HGA) to incorporate a local convergence method and precision search method around optimum in the genetic algorithms. In case study, it is showed that HGA is able consistently to provide efficient, fine quality solutions and provide a significant capability for solving structural design problems.

1. 서론

구조적 설계문제는 실제 생산현장에서 사용되는 각종 기계구성부품, 설비등의 설계에 있어 설계변수가 정수, 연속 혹은 이산값이 혼합되어

사용되는 경우에 이를 최적으로 설계하기 위한 문제이다. 일반적인 최적화 문제에서는 설계변수가 대부분 연속으로 고려되지만 구조적 설계 문제에서는 이러한 사실이 잘 적용되지 않는다. 즉 대부분의 설계변수는 실제 제조공정상의 제약과 미리 정의된 표준크기의 구성부품을 사용

* 건국대학교 산업공학과

하기 때문에 정수 혹은 이산값을 가진다. 예를 들면 기어톱니의 수, 기어 모듈값, 트러스(truss)에 사용되는 빔의 수, 구조물에 사용되는 리벳, 나사의 수 등등이다.

이러한 비연속변수가 연속변수와 함께 사용될 때 최적해를 위한 탐색은 상당히 복잡해진다. 따라서 정수·이산·연속설계변수가 포함된 구조적 설계문제를 다루기 위한 신뢰성있고 효율적인 기법이 개발되어져야 한다.

이러한 구조적 설계문제는 기존의 많은 연구들을 통해서 정식화되었고 그 해법 또한 다양하게 개발되어 왔다. 그러나 대부분의 기존 연구들은 탐색과정의 효율은 높였지만 설계변수들이 정수, 실수, 그리고 이산값을 함께 가지며 제약조건과 목적함수에 서로 혼합되어 사용되는 조합최적화문제(combination optimal problem)와 여러개의 지역적 최적점(local optimum)이 존재하는 경우에 전역적 최적점(global optimum)을 구하는 문제, 그리고 다목적함수(multi-objective function)의 설계시 최적해 집합의 효율적인 구현문제들에서는 아직까지 효과적이고 광범위하게 적용될 수 있는 방법이 개발되어 있지 못하다. 따라서 이러한 문제점을 쉽게 해결할 수 있는 대안으로 유전 알고리즘(genetic algorithm)에 의한 연구가 활발히 이루어지고 있다.

유전알고리즘은 생태계의 진화원리에 바탕을 둔 최적해에 대한 탐색 알고리즘의 하나로 설계변수의 형태에 관계없이 다양한 종류의 문제들에 대해 효과적으로 적용가능하며, 전역적 최적점에 가까운 해를 구할 수 있다는 장점이 있어 구조적 설계문제 최적화 모델에 많이 적용되고 있다. 하지만 기존의 유전알고리즘은 그 적용에 있어 지역적 수렴으로 인한 미성숙 수렴현상

(premature convergence), 비우수개체의 유전현상등으로 인한 문제점이 있어 최근에는 기존의 지역적 수렴기법과의 연계가 활발히 진행되고 있다[1,2].

따라서 본 연구에서도 기존의 유전알고리즘이 가졌던 문제점을 개선하며 지역적 최적점으로 빠르게 수렴하는 기존의 지역적 탐색법인 랜덤탐색·검사법(random search and test algorithm)과 유전알고리즘 수행 이후 최적점 부근에 수렴한 해 주변에 대한 정밀탐색법인 포복법(creeping method)을 함께 고려한 혼합유전알고리즘(hybrid genetic algorithms: HGA)을 설계하였으며, 이를 두가지의 구조적 설계문제에 적용하여 본 연구에서 개발한 HGA의 유효성을 보이고자 한다.

2. 구조적 설계문제

2.1 수리적 표현

구조적 설계문제는 구조물의 최적배치나 최적설계를 결정하는 것으로 모든 제약조건을 만족하며 선정된 목적함수를 최대화 혹은 최소화시키는 것이다. 따라서 이들 문제들은 비선형계획법(nonlinear programming: NLP)모델로 제시되는 것이 일반적이며 수리적 형태는 다음과 같이 표시가능하다.

$$\begin{aligned} \text{Find } X &= \{x_1, x_2, \dots, x_n\}^T \\ &= \{X^d, X^c, X^z, X^i\}^T \\ \text{to minimize } &F(x) \end{aligned}$$

subject to

$$g_j(x) \leq 0 \quad j=1, 2, \dots, k$$

$$x_{i \min} \leq x_i \leq x_{i \max} \quad i=1, 2, \dots, n$$

여기에서 X 는 설계변수의 벡터이고, $X^d \in R^d$ 는 이산설계변수의 가능부분집합, $X^c \in R^c$ 는 연속설계변수의 가능부분집합, $X^z \in R^z$ 는 0-1 설계변수의 가능부분집합, $X^i \in R^i$ 는 정수설계변수의 가능부분집합이며 k 는 g_j 의 갯수, n 은 독립설계변수의 수, $F(x)$ 는 목적함수, $g_j(x)$ 는 j 번째 부등제약조건, $x_{i \max}$, $x_{i \min}$ 은 각각 x_i 의 상한값과 하한값이다.

이러한 형태의 MDNLP[10](mixed-discrete nonlinear programming)모델은 기본적으로 MD-LP(mixed-discrete linear programming)문제를 해결하기 위한 기법에서 도출되었으며 기존에 다양한 기법이 개발되었다.

이러한 모델에 대해 유전알고리즘을 이용한 연구로는 Jenkins[9]는 여러 가지 구조물의 최소 중량설계와 트러스 구조물의 기하학적 형상 설계(geometrical design)에 대해 유전알고리즘을 적용하여 최적해를 찾으려는 시도를 하였다. 하지만 그 적용에 있어 위반함수(penalty function)의 적용 정도, 개체의 표현정도에 대해 설계자가 신중한 주의를 기울일 것을 강조하였다. Rajeev[13]는 이산화변수(discrete sizing variable)를 이용하여 구조적 최적화 문제(structural optimization problem)의 하나인 트러스 구조물과 전송타워(transmission tower)에 대해 유전알고리즘을 적용하였지만 계산시간이 오래걸리며 유전연산자중 돌연변이 연산의 사용을 배제하였다는 단점이 있다.

Suresh등[16]과 Tate, Smith[17]는 유전알

고리즘을 이용하여 설비배치문제를 해결하고자 하였는데 설비배치 문제의 특성을 그다지 고려하지 않았고 이전세대의 우수개체가 자손세대에 최대한 복제되지 못한다는 단점이 있다. Lin과 Hajara[10]는 이산·정수·연속설계변수 등이 혼합된 비선형 최적 설계문제에 대해 유전알고리즘을 적용하였다. 이 연구에서는 특수한 부문에 국한된 문제에만 적용하고 있기 때문에 일반화된 최적설계문제에 대해서는 적용이 곤란하다는 단점이 있다

3. 혼합유전알고리즘

(Mixed Genetic Algorithm)

3.1 혼합유전알고리즘의 형태

혼합유전알고리즘의 가장 일반적인 형태중의 하나는 선택과 재조합을 수행하는 표준적인 유전알고리즘 수행과정에 부가적으로 지역적 탐색 기법을 포함시키는 것이다.

이러한 지역적 탐색기법은 탐색공간에서 랜덤하게 새롭게 발생하는 개체들을 유전알고리즘의 집단속으로 투입하기 이전에 지역적 최적점으로 이동시키는 것이며 이러한 지역적 최적화 기법이 각 개체 주위의 지역적 탐색을 위해 사용되어진다면 유전알고리즘은 이렇게 수렴된 집단 가운데에서 전역적 탐색을 수행하기 위해 사용되어질 수 있다[4].

유전알고리즘과 기존의 지역적 탐색기법의 이러한 상호 보완적 특성 때문에 혼합접근법이 유전알고리즘과 지역적 탐색기법이 각각 개별적

으로 적용될 때 보다 더 우수한 결과를 가져올 수 있으며 본 연구에서도 이러한 특성을 이용하여 혼합접근법으로서의 유전알고리즘을 개발하고자 한다.

먼저 최적점 부근까지 빠르게 수렴하는 기존의 지역적 탐색기법인 랜덤탐색·검사법(random search and test method)과 수렴된 해에 대한 집단 전체의 탐색을 실시하는 유전알고리즘을 결합하고 보다 정밀한 탐색을 위해서 수렴된 해 주변에 대한 정밀탐색을 실시하는 포복법(creeping method)을 적용한 새로운 혼합유전알고리즘(hybrid genetic algorithm:HGA)을 개발하고자 한다.

본연구에서 적용하게 될 이러한 세가지 탐색기법의 특징은 다음과 같다.

3.1.1 지역적 수렴(Local convergence)기법

본 연구에서 적용할 지역적 수렴기법은 랜덤탐색·검사법으로 지역적 수렴점으로 수렴하는 과정에서 생성되는 우수한 개체들을 선택하여 집단으로 만든다. 만일 이렇게 수렴된 개체가 미리 정의된 집단의 크기보다 작으면 그 나머지 개체들에 대해서는 랜덤발생시키고 집단의 크기보다 크면 개체들을 분류(sorting)하여 적합도가 우수한 개체순으로 집단의 크기만큼 형성한다.

이러한 선택전략은 유전알고리즘에서 개체의 선택에 적용되는 확률적 토너먼트 선택전략(stochastic tournament selection strategy)을 변형한 것이다[6].

구체적인 탐색과정은 3.1.4절의 Step 1~4에 제시되어 있다.

3.1.2 유전알고리즘(Genetic algorithms)

본 연구에서 적용하고자 하는 유전알고리즘은 기존에 많이 사용되는 비트(0-1 bit)발생법을 사용하지 않고 실수(floating point)발생법[12]을 사용하여 변수를 설계하였다. 이것은 비트발생법이 비트발생이 아닌 대부분의 기존 지역적 수렴기법과 연계가 잘 되지 않는다는 단점 때문이며 실수발생법이 설계변수에 대해 정밀한 탐색이 가능하다는 장점 때문이기도 하다.

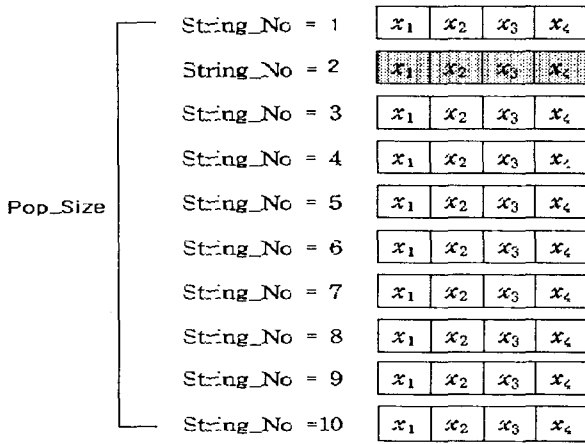
실수발생법을 이용한 유전연산자(genetic operator)의 적용은 다음과 같다.

1) 선택 (Selection)

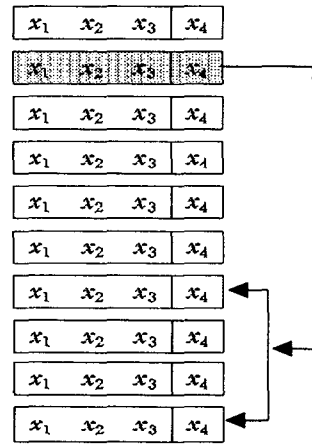
선택단계란 각 개체의 적합도에 따라 보다 환경에 적합한 개체를 확률적으로 선택하는 단계이며 일반적으로 많이 사용되고 있는 선택기법인 룰렛휠(roulette wheel)선택전략[7]은 각 개체의 적합도에 비례하여 확률적으로 개체를 선택하는 기법이다. 그러나 룰렛휠 선택전략은 그 특성상 같은 개체가 동시에 새로운 집단속으로 선택될 수 있다. 이럴 경우 서로 교차변이가 수행되더라도 다양한 개체가 발생되지 않기 때문에 지역적 탐색이 된다는 단점이 있다[1].

이러한 단점을 개선하기 위해서 HGA에서는 등급기준(rank-based) 선택전략과 엘리트리스트(elitist)선택전략[6] 개념을 응용한 새로운 선택전략을 개발하였으며 그 구체적인 과정은 다음과 같다.

첫째, HGA의 초기에 실행되는 랜덤탐색·검사법에서 지역적 최적점으로 수렴되는 과정에서 생성되는 개체들을 유전알고리즘 수행을 위한 집단으로 선택한다.



[그림 1] 일반적인 교차변이의 예



[그림 2] 전역적 탐색 교차변이

둘째, 유전알고리즘 적용이후 생성된 개체들 중에서 적합도가 가장 우수하며 이 개체가 이전 단계인 랜덤탐색·검사법에서 생성된 가장 우수한 개체의 적합도보다 더 우수할 경우만 유전알고리즘의 집단속으로 투입시킨다. 이때 기존에 투입된 개체중에서 적합도가 가장 낮은 개체 하나를 집단에서 탈락시킴으로서 집단의 크기를 일정하게 유지할 수 있다.

이러한 과정을 반복함으로써 좀더 적합도가 우수한 개체들로 이루어진 집단을 구성할 수 있다.

2) 교차변이(Crossover)

교차변이란 선택단계이후 집단내의 두 개체간의 정보를 교환하여 새로운 개체를 만들어 내는 방법으로 교차변이될 개체는 일반적으로 무작위적으로 선택된다. 만일 이러한 무작위 개체선택에서 적합도가 비슷한 개체간의 교차변이는 지역적 탐색이 되고, 적합도가 상이한 개체간의 교차변이는 전역적 탐색이 된다[1].

유전알고리즘에 적용되는 교차변이는 유전알고리즘이 가진 전역적 탐색능력을 보장해야 하기 때문에 지역적 탐색능력을 줄이고 전역적 탐

색능력을 증가시키도록 교차변이를 설계하여야 한다. 하지만 기존에 개발된 대부분의 기법들은 집단의 개체를 무작위적으로 선택하여 쌍을 만들기 때문에 적합도가 비슷한 개체가 선택될 경우 유전알고리즘의 전역적 탐색능력을 보장할 수 없다.

예를 들어 [그림 1]은 각 개체의 염색체구조를 표현하고 있으며 집단의 크기(Pop_Size)가 10인 경우 적합도에 따라 개체들을 분류(sorting)한후 임의로 선택된 개체번호(String_No)가 2라면 이 개체와 함께 교차변이시킬 개체는 집단에 포함된 전체 개체중에서 이미 선택된 개체번호 2를 제외한 나머지 개체 (1, 3, 4, 5, 10)중에서 임의로 하나를 택하여 교차변이 연산을 수행한다. 만일 임의로 선택된 개체의 번호가 이미 선택된 개체의 번호와 비슷한 적합도를 가질 경우 (예를들어 String_No =1, 3, 4) 이들 개체끼리의 교차변이는 전역적 탐색능력을 보장할 수 없다. 왜냐하면 적합도가 비슷한 개체끼리의 교차변이는 해의 탐색공간에서 다양하게 개체를 발생시킬 수 없기 때문이다[1].

이러한 문제점을 개선하기 위해 본 연구에서는 적합도가 상의한 개체끼리의 교차변이를 수행하도록 하는 방법을 개발하였으며 그 구체적인 교차변이 방법은 다음의 절차를 따른다.

첫째, 집단(Pop_size)내의 개체들을 적합도의 크기에 따라 정렬한후 이 집단에서 랜덤하게 하나의 개체를 선택하며 만일 선택된 개체의 번호($String_No$)가 아래의 식(3-1)과 같다면

$$String_No \leq \frac{Pop_size}{2} \quad (3-1)$$

이 개체와 함께 교차변이 시킬 개체의 번호($String_No$)는 아래식 (3-2)의 범위에서 하나의 개체를 선택하고

$$\left[String_No + \frac{Pop_size}{2}, Pop_size \right] \quad (3-2)$$

그렇지 않을 경우 아래식 (3-3)의 범위에서 하나의 개체를 랜덤발생시킨후 교차변이 시킨다.

$$\left[1, String_No - \frac{Pop_size}{2} \right] \quad (3-3)$$

예를들어 집단의 크기가 10인 경우 랜덤하게 발생된 개체의 번호($String_No$)가 2라면 위의 식 (3-1) (3-2)에 의해 교차변이될 개체의 번호는 [그림 2]에서 보여지는 것처럼 $String_No = 7, 8, 9, 10$ 중에서 하나를 랜덤하게 선택하여 교차변이 시킨다.

이러한 방법은 교차변이 시킬 개체쌍의 적합도가 상당히 다른 개체들 끼리의 교차변이가 되기 때문에 이렇게 해서 생성된 개체들은 적합도가 비슷한 개체들 끼리의 교차변이 경우보다 상당히 넓은 영역을 탐색할 수 있다는 장점이 있

다. 즉 전역적 탐색이 가능하게 된다.

이렇게 하여 선택된 개체에 대해 교차변이 연산자를 적용한다. 적용할 방법은 Gen과 Cheng[6]이 제안한 불균등 수리교차 변이법(non-uniform arithmetic crossover method)으로 2개의 부모개체들에 대해 2개의 새로운 자손개체를 생성한다.

3) 돌연변이(Mutation)

돌연변이 연산은 집단의 다양성을 유지하기 위하여 혹시 있을지도 모르는 유용한 개체를 잠정적으로 복구하는 유전연산자로 일반적으로 탐색의 방향이 지나치게 무작위적으로 바뀌지 않아야 함으로 돌연변이 확률을 낮게 준다

본 연구에서 적용하게될 돌연변이법은 균등 돌연변이(uniform mutation)[6]로 돌연변이 시킬 개체를 설계변수의 규정된 범위내에서 랜덤하게 발생시켜 만든다.

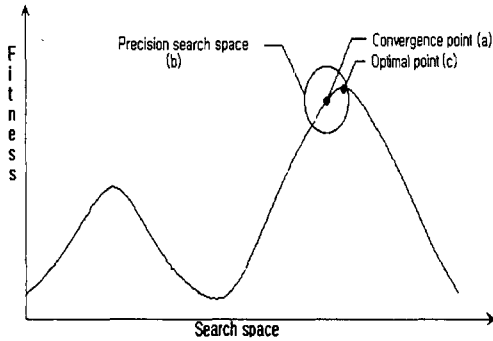
위와같은 유전연산자를 이용한 구체적인 탐색과정은 3.1.4절의 Step 5~8에 제시되어 있다.

3.1.3 정밀탐색(Precision search)법

정밀탐색법은 기존의 지역적 수렴기법과 유전알고리즘을 적용한 후에 사용하는 기법으로 수렴된 해 주위를 집단의 크기만큼 정밀하게 재탐색하여 더 우수한 해로 이동시키는 기법이다. 이 기법은 기존의 유전알고리즘으로는 그 주위를 정밀하게 탐색할 방법이 없기 때문에 이를 HGA에 적용할 경우 유전알고리즘과 기존의 지역적 탐색기법이 각각 개별적으로 적용될 때 보다 더 우수한 해를 찾을 수 있다.

예를들면 [그림 3]에서 처럼 만일 유전알고리즘 수행이후 수렴된 해(a)가 최적점(c) 부근이라면 이 부근에 대해 그 주위(b)의 위·아래

를 정밀하게 탐색하여 최적점을 찾을 수 있다는 원리이다.



[그림 3] 정밀 탐색 공간

구체적인 탐색과정은 3.1.4절의 Step 9~11에 제시되어 있다.

3.1.4 HGA의 절차

- Step 1 : 각 설계변수에 대해 미리 정의된 집단 크기만큼 랜덤하게 각 개체들을 생성하고 이를 제약조건에 대입한 후 제약조건을 만족하면 Step 2로 이동하고 그렇지 않으면 탈락시킨다.
- Step 2 : 제약조건을 만족한 개체들을 목적함수에 대입하여 가장 우수한 개체 하나만을 유전알고리즘 수행을 위한 집단속으로 투입시킨다.
- Step 3 : Step 1~2 과정을 반복수(Iteration No)만큼 실행하며, 이 과정에서 이전에 투입된 개체보다 더 우수한 개체가 생성되면 이를 유전알고리즘 수행을 위한 집단속으로 투입시킨다.
- Step 4 : Step 1~3까지의 과정을 수행한 후

생성된 집단의 크기가 유전알고리즘 수행을 위한 집단의 크기보다 작으면 그 나머지 개체들에 대해서는 랜덤발생시켜 투입시키며, 집단의 크기보다 크면 적합도의 크기에 따라 순서대로 정렬한 후 집단의 크기만큼 생성한다.

- Step 5 : Step 1~4의 과정에서 생성된 집단을 이용해 유전알고리즘을 실행한다. 유전연산자인 선택과 교차변이 단계는 본 연구에서 개발된 기법을 적용하며 돌연변이 단계에서는 기존에 개발된 균등돌연변이법을 적용한다.
- Step 6 : Step 5를 거치면서 생성된 개체들을 제약조건에 대입하여 이를 위반하면 탈락시키고 만족하면 목적함수에 대입하여 가장 우수한 하나의 개체만을 저장한다.
- Step 7 : Step 6에서 생성된 개체와 Step 1~4에서 생성된 개체들을 비교하여 Step 6에서 생성된 개체의 적합도가 더 우수하면 이를 Step 4의 유전알고리즘 적용을 위한 집단 속으로 투입시킨다. 이때 기존에 투입된 개체 중에서 적합도가 가장 낮은 개체 하나를 집단에서 탈락시킨다.
- Step 8 : Step 5~7 과정을 반복수 만큼 실행한다.
- Step 9 : Step 8의 실행 후 생성된 집단에서 가장 우수한 개체에 대해 그 주변을 정밀하게 탐색하는 포복법을 실시한다. 예를들어 설계변수 x_1 의 수렴된 값이 15.2이고 설계자가 정해준

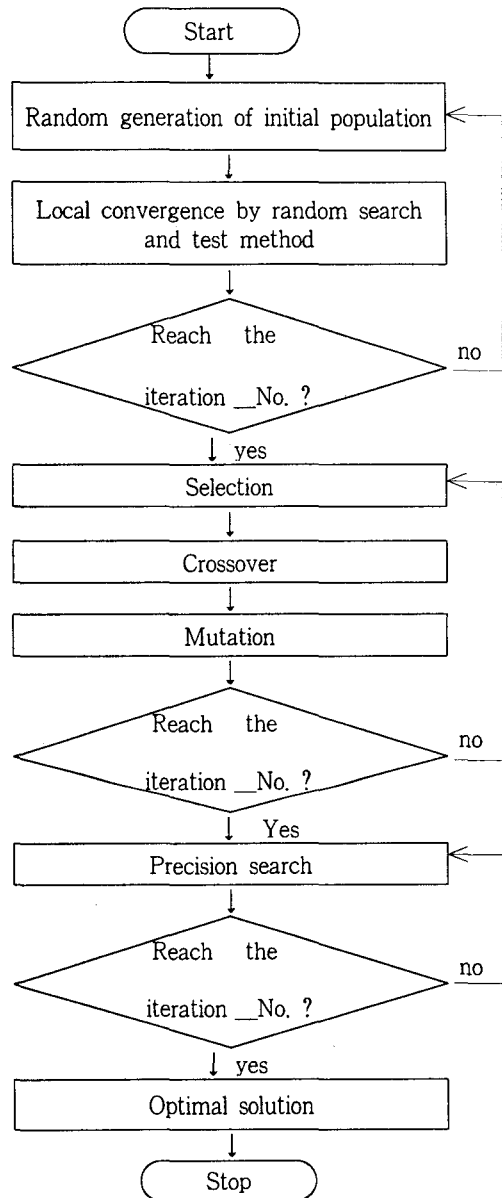
포복범위(*creeping-range*)가 1.5라면 x_1 은 13.7~16.7의 범위내에서 정밀하게 재탐색을 실시한다. 이에 대한 변수설계는 아래와 같다.

```
for i = 1 to pop_size
  temp = (2 × creeping_range
  × rnd) - creeping_range
   $x_1(i) = x_{1\_value} + temp$ 
next i
```

여기에서 *pop_size*는 집단의 크기이며, *temp*는 임시지역장소, *rnd*는 0, 1사이의 랜덤발생값이고, x_{1_value} 는 HGA수행에서 구해진 지역적 최적 x_1 의 값이다.

Step 10 : Step 9에서 생성된 개체들에 대해 제약조건을 위반하면 탈락시키고 만족하면 목적함수에 대입하여 적합도가 가장 우수한 개체와 Step 5~8에서 생성된 가장 우수한 개체와의 비교를 통해 Step 9에서 생성된 개체의 적합도가 더 우수할 경우만 저장시킨다.

Step 11 : Step 9~10을 반복수만큼 실행한다. 본 연구에서 제시한 HGA의 수행과정을 흐름도로 나타내면 [그림 4]와 같다.



[그림 4] HGA의 수행 흐름도

4. 사례연구

사례연구에서는 구조적 설계문제에서 많이 사용되는 압축코일스프링 최적설계모델과 압력용기 최적설계 모델을 제시하였으며 이를 기준

의 최적화 기법들과 비트발생 GA 및 본 연구에서 개발한 HGA와의 비교를 수행하였다.

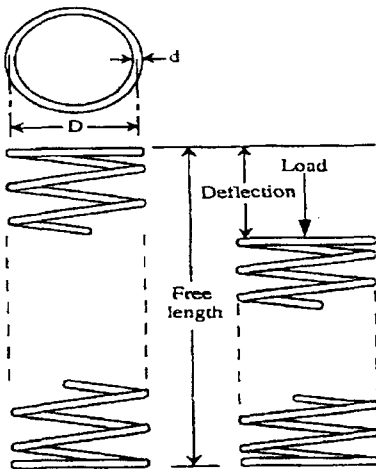
분석방법으로는 각 기법들을 설계모델에 적용하여 수렴해의 우수성을 비교하며, 그 수렴과정을 그래프로 나타내어 구체적인 비교·분석을

실시하였다.

본 연구에서 개발한 HGA는 Windows환경 (Pentium-90Mhz, RAM :32M)하에서 Visual Basic 5.0을 이용하여 구현하였다.

4.1 코일압축스프링 설계(Design of coil compression spring)

코일압축스프링은 일정한 부하에서 작동되는 나선형 압축스프링으로 기계설계부분에서 중요한 부품으로 취급된다. 본 연구에서 제시하고자 하는 모델은 Siddall[15]이 제안한 것으로 목표함수는 스프링제조에 사용되는 와이어의 부피를 최소화하는 것이며, 제약조건으로는 [그림 5]에서 보여지는 d, D 와 코일의 수(N)이다. 즉 정수로 정의된 코일의 수(N), 연속값으로 정의된 코일의 지름(D), 이산값으로 정의된 와이어의 지름(d)등이다. 여기에서 와이어의 지름(d)는 미국재료시험협회(American society for testing and materials :ASTM)에 규정된 크기만을 사용해야 되기 때문에 <표 1>과 같은 이산값을 가진다.



[그림 5] 코일압축스프링의 구조

<표 1> 코일압축스프링 설계모델에서 사용되는 이산값

0.0090	0.0150	0.0280	0.0720	0.1620	0.2830
0.0095	0.0162	0.0320	0.0800	0.1770	0.3070
0.0104	0.0173	0.0350	0.0920	0.1920	0.3310
0.0118	0.0180	0.0410	0.1050	0.2070	0.3620
0.0128	0.0200	0.0470	0.1200	0.2250	0.3940
0.0132	0.0230	0.0540	0.1350	0.2440	0.4375
0.0140	0.0250	0.0630	0.1480	0.2630	0.5000

계산의 편의상 각 설계변수를 식(4-1)과 같이 나타낸다.

$$\begin{aligned} \text{Find } X &= [N, D, d] \\ &= [x_1, x_2, x_3] \end{aligned} \quad (4-1)$$

이에 대한 수리적 모델은 식 (4-2)~(4-10)과 같다.

$$\begin{aligned} \text{Minimize } F(x) &= \\ &= \pi^2 x_2 x_3^2 (x_1 + 2) / 4 \end{aligned} \quad (4-2)$$

Subject to

$$G_1(x) = \frac{8C_f F_{\max} x_2}{\pi x_3^3} - S \leq 0 \quad (4-3)$$

$$G_2(x) = l_f - l_{\max} \leq 0 \quad (4-4)$$

$$G_3(x) = d_{\min} - X_3 \leq 0 \quad (4-5)$$

$$G_4(x) = x_2 + x_3 - D_{\max} \leq 0 \quad (4-6)$$

$$G_5(x) = 3.0 - \frac{x_2}{x_3} \leq 0 \quad (4-7)$$

$$G_6(x) = \delta_p - \delta_{pm} \leq 0 \quad (4-8)$$

$$\begin{aligned} G_7(x) &= \delta_p + \frac{F_{\max} - F_p}{K} \\ &+ 1.05(x_1 + 2)x_3 - l_f \leq 0 \end{aligned} \quad (4-9)$$

$$G_8(x) = \delta_w - \frac{F_{\max} - F_p}{K} \leq 0 \quad (4-10)$$

where

$$C_f = \frac{4(x_2/x_3) - 1}{4(x_2/x_3) - 4} - \frac{0.165x_3}{x_2}$$

$$K = \frac{Gx_3^4}{8x_1x_2^3}$$

$$l_f = \frac{F_{max}}{K} + 1.05(x_1 + 2)x_3$$

$$\delta_p = \frac{F_p}{K}$$

$$F_{max} = 1000 \text{ lb}$$

$$S = 189,000 \text{ psi}$$

$$l_{mx} = 14.0 \text{ inch}$$

$$d_{min} = 0.2 \text{ inch}$$

$$D_{max} = 3.0 \text{ inch}$$

$$F_p = 300.0 \text{ lb}$$

$$\delta_{pm} = 6.0 \text{ inch}$$

$$\delta_w = 1.25 \text{ inch}$$

$$G = 11.5 \times 10^6 \text{ psi}$$

이상과 같이 설계변수가 이산·연속·정수 등으로 조합된 최적화 문제에 대해 HGA를 적용하였으며 기존연구와의 비교결과를 <표 2>에 나타내었다. 또한 구체적인 수렴과정을 살펴보기 위해 Meta_GA와 HGA를 비교한 것이 [그림 6]

에 제시되어 있다.

[그림 6]에 제시된 HGA의 수행과정은 각 반복회수에 대해 HGA의 과정을 수행한 결과를 모아서 하나의 그래프로 나타낸 것이다.

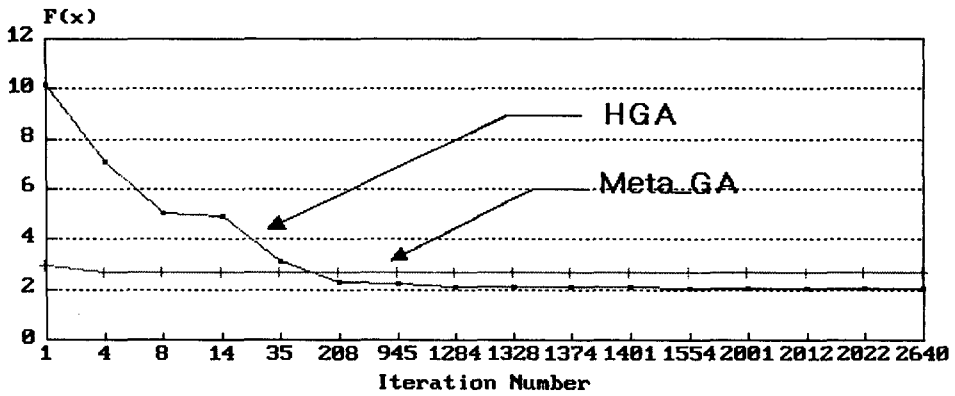
본 연구의 HGA에 사용된 모수는 다음과 같다.

<i>Iteration_No</i>	: 3,000
<i>Pop_Size</i>	: 30
<i>Crossover_Rate</i>	: 0.5
<i>Mutation_Rate</i>	: 0.1
<i>Creeping_Range</i>	: 0.01

<표 2>를 분석해 보면 Sandgren[14]은 비선형분지한계법(Nonlinear branch and bound)을 이용하였는데 이 기법은 정확하게 부과된 한계 제약에 따라 그 변수값이 변화되어지도록 허용되며 모든 이산변수가 탐색될 때까지 계속 수행됨으로 이 문제와 같이 이산·연속·정수변수가 혼합된 경우에는 탐색과정에서 명확한 해의 개선은 기대할 수 없다는 단점이 있다. 또한 기존의 비트발생GA로 탐색을 실시한 Chen과 Tsao의 연구[3]와 Wu 와 Chow[18]의 Meta_GA의 경우는 연속변수를 표현할때 정밀도의 값 선정에 문제점이 있으며 이산변수의 경우도 대응 관계를 올바르게 표현하지 못하여 지역적 최적

<표 2> 코일압축스프링 설계모델의 적용결과

Method	Nonlinear Branch and Bound	GA	Meta_GA	HGA	Type of variables
x_1	10	9	9	9	Integer
x_2	1.180701	1.2287	1.227411	1.1092	Continuous
x_3	0.283	0.283	0.283	0.263	Discrete
$F(x)$	2.7995	2.6709	2.6681	2.0824	
Run time (Sec.)				12	



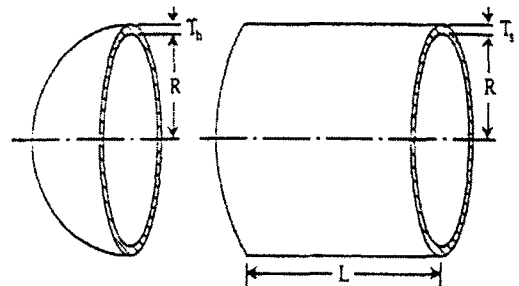
[그림 6] 코일압축스프링 설계모델에서 Meta_GA와 HGA의 수렴과정

점으로 수렴한 것으로 분석된다. 그러나 HGA에서는 지역적 최적점으로 빠르게 수렴하는 랜덤탐색법과 수렴된 해 주변에 대한 정밀탐색을 GA에 혼합하였기 때문에 기존연구들보다 더 우수한 해를 구할 수 있었다. 또한 수렴시간을 살펴보면 기존연구들이 workstation 이상의 고성능 컴퓨터를 사용한 것과는 달리 HGA는 일반 PC로 수행하여 12초의 수행시간을 얻었기 때문에 기존연구들과 비교는 곤란하지만 상당히 우수한 것으로 분석된다.

Meta_GA와 HGA의 수렴과정을 제시한 [그림 6]에서도 세대수가 약 100세대 이전까지는 Meta_GA가 우수하지만 100세대 이후에는 HGA가 수렴과정이 계속 진행되어 더 우수한 결과를 얻을 수 있었으며 Meta_GA의 경우는 더 이상의 해의 수렴현상은 없었다. 이러한 결과는 기존의 GA연구가 설계변수의 비트표현에서 올바르게 못한 정밀도의 사용과 위반된 해에 대한 위반함수(penalty function)의 적용이 문제가 있고, 일단 수렴된 해에 대해 그 주위를 정밀하게 탐색하지 못했기 때문에 HGA보다 더 우수한 결과를 얻을 수 없었던 것으로 분석된다.

4.2 압력용기설계(Design of pressure vessel)

압력용기는 운전 중에 일어나는 압력 및 온도를 기준으로 설계하는데 재료의 허용인장응력, 사용철판의 최소두께, 사용재료의 부식여유, 동체의 두께등을 고려하여 어떤 용도로 사용되는가에 따라 그 재질과 두께 등이 달라진다. 본 연구에서 사용하고자 하는 압력용기[18]는 [그림 7]에 제시되었으며 최소 750 feet의 부피와 3000 psi의 압력에도 견딜 수 있는 압축공기 저장탱크이다. 이러한 압력용기는 미국기계학회(American society of mechanical engineers: ASME)에서 정한 보일러와 압력용기 코드의 규정에 따라 설계되어야 한다.



[그림 7] 압력용기의 구조

[그림 7]에서 제시된 압력용기의 두께(T_h , T_s)는 이산값으로 0.0625"의 정수배수(integer multiples)로 규정되어지며, 각각 2"를 넘지 못한다. 또한 두께(T_s)는 1.1"보다는 커야 하며 T_h 는 0.6"보다 커야한다. 압력용기의 길이(L)와 반지름(R)은 각각 40"~80", 20"~60" 내에서 설계되어야 하며, 그밖에 필요한 상수와 변수값은 압력용기설계에 기본적으로 필요하며 설계 규정에 따라 미리 정해진다.

계산의 편의상 설계변수들은 식(4-11)과 같이 표시한다

$$\begin{aligned} \text{Find } X &= [T_s, T_h, R, L]^T \\ &= [x_1, x_2, x_3, x_4]^T \end{aligned} \quad (4-11)$$

이상과 같은 제약조건을 식으로 나타내면 (4-12) ~ (4-17)과 같다.

Subject to

$$G_1(X) = x_1 - 0.0193x_3 \geq 0 \quad (4-12)$$

$$G_2(X) = x_2 - 0.00954x_3 \geq 0 \quad (4-13)$$

$$\begin{aligned} G_3(X) &= \pi x_3^2 x_4 + \frac{4}{3} \pi x_3^3 \\ &\quad - 750 \times 1728 \geq 0 \end{aligned} \quad (4-14)$$

$$G_4(X) = -x_4 + 240 \geq 0 \quad (4-15)$$

$$G_5(X) = x_1 - 1.1 \geq 0 \quad (4-16)$$

$$G_6(X) = x_2 - 0.6 \geq 0 \quad (4-17)$$

이러한 제약하에서 압력용기의 제작에 필요한 총제조비용을 최소화하는 것이 목적이며 식(4-18)과 같이 표시된다.

Minimize

$$\begin{aligned} F(x) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\ &\quad + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \end{aligned} \quad (4-18)$$

이상과 같은 이산·연속변수가 포함된 구조적 설계문제에 대해 HGA를 적용하며 여기에 사용되는 모수는 다음과 같다.

Iteration_No : 9600

Pop_Size : 20

Crossover_Rate : 0.6

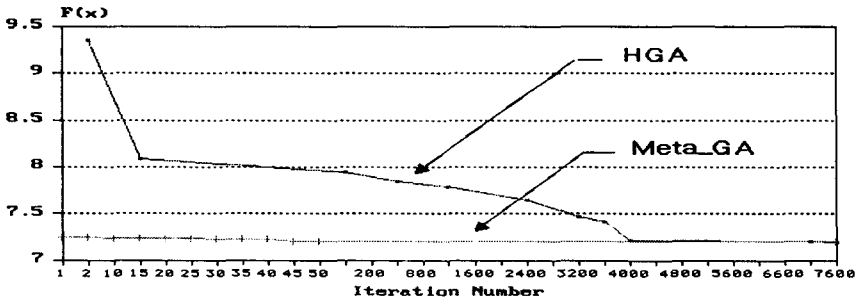
Mutation_Rate : 0.1

Creeping_Range : 0.62

<표 3>은 HGA와 기존연구들을 비교·분석한 결과이며 구체적인 탐색과정을 살펴보기 위해 Meta_GA와 HGA의 수렴과정을 [그림 8]에 나타내었다.

<표 3> 압력용기 설계모델의 적용결과

Method	Nonlinear branch and bound	IDCNLP	Meta_GA	HGA	Type of variables
x_1	1.125	1.125	1.125	1.125	Discrete
x_2	0.625	0.625	0.625	0.625	"
x_3	48.97	48.3807	58.1978	58.2894	Continuous
x_4	106.72	111.7449	44.2930	43.7346	"
$F(x)$	7982.5	8048.619	7207.49	7199.76	
Run Time (sec.)				12	



[그림 8] 압력용기 설계모델에서 Meta_GA와 HGA의 수렴과정

[그림 8]에 제시된 HGA의 수행과정은 각 반복회수에 대해 HGA의 과정을 수행한 결과를 모아 하나의 그래프로 나타낸 것이다.

<표 3>을 분석해 보면 Sandgren[14]은 비선형 분지한계법을 이용하여 해를 구하고 있지만 적용기법의 특성상 초기상태에서는 전체 마디(node)수를 알 수 없기 때문에 프로그램 작성시에 실제 필요한 저장공간보다 더 많은 저장공간을 미리 확보하여야 하며, 문제의 규모가 커질 경우 해법상 해결과정이 복잡하기 때문에 지역적 수렴되어 더 이상의 개선은 없었다. Fu등[5]은 IDCNLP(integer-discrete-continuous nonlinear programming)알고리즘을 이용하여 해결하였는데, 이 방법은 일반적인 비선형계획법의 해결절차에 근거를 두고 있으며 문제 해결과정에서 초기입력변수의 값을 선택하는데 어려움이 있다. 또한 Wu 와 Chow[18]의 연구결과는 코일압축스프링 모델에서 적용된 것과 같은 Meta_GA 기법을 이용했지만 역시 지역적 수렴되어 더 이상의 개선은 없었다.

반면에 본 연구의 HGA는 기존연구들보다 더 우수한 해를 구현하였다. 이러한 현상은 기존의 연구 모두가 국소해로 수렴하였지만 HGA는 초

기 랜덤탐색·검사법의 적용시에 해의 다양성을 실현하고 전역적 탐색능력을 향상시킨 유전연산자를 적용하여 미성숙 수렴현상을 방지하였기 때문에 빠른 시간 내에 더 우수한 해를 구할 수가 있었다고 분석된다.

그리고 수렴과정을 표시한 [그림 8]을 살펴보면 Meta_GA의 경우는 탐색초기에는 HGA보다 우수한 해를 구하였지만 탐색과정이 진행될수록 HGA는 해를 다양하게 발생시키면서 우수해로 향해 계속 수렴하고 Meta_GA의 경우는 탐색과정의 진행결과에 대해 거의 해를 개선시키지 못하고 있는 것을 볼 수 있다. 이러한 현상은 Meta_GA가 각 설계변수들의 최적 조합을 찾지 못하여 탐색공간에 대한 해의 발생을 다양하게 시키지 못하기 때문으로 분석된다.

5. 결 론

본 연구에서는 구조적 설계문제 최적화를 위한 혼합유전알고리즘(Hybrid genetic algorithms :HGA)을 설계하였다. HGA에서는 초기 집단형성시 지역적 탐색법을 수행하여 여기에서 수

렴된 우수개체와 랜덤발생의 비우수 개체를 함께 혼합하여 집단을 형성하는 랜덤탐색·검사법을 개발하였으며 유전알고리즘에서는 전역적 탐색능력을 향상시킨 선택전략과 교차변이법을 개발하였고 유전알고리즘 수행이후 수렴된 해 주변의 정밀탐색을 위해 포복법을 개발하였다. 이의 우수성을 입증하기 위해 두가지의 구조적 설계문제를 제시하였고 그 분석결과는 기존의 연구들보다 HGA가 전역적 탐색능력이 뛰어나며 비록 탐색초기에는 우수한 해를 얻지 못하였지만 탐색이 진행되면서 기존의 연구들보다 점차적으로 우수해로 수렴되는 현상을 보이고 있다. 이러한 현상은 HGA가 변수발생에 있어 기존연구들보다 집단을 다양하게 발생시킨 결과로 분석된다.

이상의 결과로 보아 본 연구의 HGA는 문제의 유형이나 규모, 사용되는 변수의 형태에 관계없이 다양하게 적용할 수 있으며 변수표현에서도 기존의 복잡한 비트발생이 아니라 변수형태를 그대로 표현하는 실수발생을 사용하였기 때문에 프로그램 설계가 쉽다는 장점이 있다.

참 고 문 헌

- [1] 기타노 히로아키, 유전자알고리즘, 대칭컴퓨터월드, 1996.
- [2] 이동곤, 김수영, 이창억, "최적화 효율향상을 위한 유전해법과 직접탐색법의 혼용에 관한 연구," 산업공학, 제8권, 제1호, (1995) pp.23-30.
- [3] Chen, J. L., and Tsao, Y. C., "Optimal Design of Machine Elements using Genetic Algorithms," *Journal of the Chinese Society of Mechanical Engineers*, Vol.14, No.4(1993), pp.193-199.
- [4] Davis, L., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufman Publishers, Los Altos, 1987.
- [5] Fu, J. F., Fenton, R. G., and Cleghorn, W. L., "A Mixed Integer-Discrete-Continuous Programming Method and Its Applications to Engineering Design Optimization," *Engineering Optimization*, Vol. 17(1991), pp.263-280.
- [6] Gen, M., and Cheng, R., *Genetic Algorithms and Engineering Design*, John Wiley & Sons, 1997.
- [7] Goldberg, D., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.
- [8] Holland, J. H., *Adaptation in Natural and Artificial Systems*, Ann Arbor, Michigan, The University of Michigan Press, 1975.
- [9] Jenkins, W. M., "Towards Structural Optimization Via The Genetic Algorithms," *Computer and Structures*, Vol.40, No.5(1991), pp.1321-1327.
- [10] Lin, C. Y. and Hajala P., "Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables," *Engineering Optimization*, Vol.19(1992), pp.309-327.
- [11] Loh, H. T., and Papalambros, P. Y., "A Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems," *Journal of Mechanical*

- Design*, Vol.113(1991), pp.325-334.
- [12] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programming*, Second Extended Edition, Springer-Verlag, 1994.
- [13] Rajeev, S. and Krishnambros, C. S., "Discrete Optimization of Structures Using Genetic Algorithms," *ASCE, Journal of Structural Engineering*, Vol.118, No.5(1992), pp.1233-1250.
- [14] Sandgren, E., "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *ASME Journal of Mechanical Design*, Vol.112, No.2(1990), pp.223-229.
- [15] Siddall, J. N., *Optimal Engineering Design: Principles and Applications*, New York: Marcel Dekker, 1982.
- [16] Suresh, G., Vinod, V. V. and Sahu, S., "A Genetic Algorithm for Facility Layout," *International Journal of Production Research*, Vol.33(1995), pp.3411-3423.
- [17] Tate, D. and Smith, A., "Unequal-Area Facility Layout by Genetic Search," *IIE Transactions*, Vol.27(1995), pp.465-472.
- [18] Wu, S. J. and Chow, P. T., "Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems via Meta-Genetic Parameter Optimization," *Engineering Optimization*, Vol.24(1995), pp.137-159.