

## 구조최적화를 위한 병렬유전자 알고리즘 Parallel Genetic Algorithm for Structural Optimization on a Cluster of Personal Computers

이 준 호\*  
Lee, Joon Ho

박 효 선\*\*  
Park, Hyo Seon

### ABSTRACT

One of the drawbacks of GA-based structural optimization is that the fitness evaluation of a population of hundreds of individuals requiring hundreds of structural analyses at each GA generation is computational too expensive. Therefore, a parallel genetic algorithm is developed for structural optimization on a cluster of personal computers in this paper. Based on the parallel genetic algorithm, a population at every generation is partitioned into a number of sub-populations equal to the number of slave computers. Parallelism is exploited at sub-population level by allocating each sub-population to a slave computer. Thus, fitness of a population at each generation can be concurrently evaluated on a cluster of personal computers. For implementation of the algorithm, a virtual distributed computing system in a collection of personal computers connected via a 100 Mb/s Ethernet LAN. The algorithm is applied to the minimum weight design of a steel structure. The results show that the computational time required for serial GA-based structural optimization process is drastically reduced.

### 1. 서론

1990년대 이후에 급속도로 발전해온 컴퓨터 관련기술은 공학분야에 많은 변화를 가져왔다. 구조최적화 문제에도 컴퓨터의 발달과 함께 많은 실용적 최적화 알고리즘들이 소개되었으며 철골 구조물과 같은 이산변수형 최적화 문제에 적용되어 그 효율성이 평가되고 있다. 특히 기존의 최적성 규준법과 함께 조합최적화 (combinatorial optimization) 기법인 GA<sup>1)</sup>, SA<sup>2)</sup>, Tabu Search<sup>3)</sup> 등이 개발되어 구조최적화에 적용되고 있다. 그러나 구조최적화를 위한 조합최적화 기법은 최적값을 찾는 동안 컴퓨터의 기억 용량과 계산시간의 한계로 인한 문제점 때문에 일반적인 개인용 컴퓨터에서 사용하는데 실용적 한계를 가지고 있다.

특히 유전알고리즘은 다른 최적화 알고리즘에 비하여 많은 반복수가 요구되며 해집단을 운용하기 때문에 적응도 평가를 위하여 해집단의 수만큼 구조해석을 필요로 한다. 또한 전체 최적화 수행시간 중 적응도 평가를 위한 구조해석에 소요되는 시간이 90-95%를 차지하기 때문에 일반 PC상에서 구조최적화를 위한 유전 알고리즘의 효율성은 구조해석의 효율성에 의존적이며 대형구조물의 최적화설계에 적용시 현실적으로 불가능한 일이 된다. 따라서 내재적인 병렬성을 가진 유전자알고리즘의 병렬화가 요구된다.<sup>5)</sup>

병렬알고리즘은 유전 알고리즘의 개발 초기인 1960년대부터 고려되었으며 수행시간의 단축과 더불어 해의 개선의 목적으로 이루어 졌다. 그러나 대부분의 알고리즘은 Super computer<sup>4)5)</sup>기반으로 개발되거나 유닉스환경의 네트워크기반으로 구성된 workstation cluster<sup>6)</sup>로 개발되었다. 그러나 국내 여건을 고려하면 Super computer나 workstation cluster를 사용하는 것이 보편화되어 있지 않다. 이러한 이유들은 구조최적화 문제를 위한 유전 알고리즘의 적용을 어렵게 하고 있으며 PC상에서 대형구조물을 위한 구조최적화에 적용하는

\* 영남대학교 건축공학과 석사과정

\*\* 연세대학교 사회환경·건축공학부 조교수

예가 드물다.

따라서 본 논문에서는 기존 GA알고리즘의 단점을 보완하기 위하여 최적알고리즘 수행시간의 90%이상을 차지하는 적응도 평가(구조해석)부분을 다수의 컴퓨터가 연결된 가상의 병렬연산시스템을 이용하여 병렬화하여 효율적인 구조최적화기법을 개발하고자한다. 개발된 최적화알고리즘은 철골구조물의 최소중량설계에 적용하여 그 성능을 평가한다.

## 2. 구조 최적화를 위한 정식화

본 논문에서 구조최적화 문제는 구조물의 총 중량을 최소화하는 것을 목적함수로 하고 있으며 m개의 부재로 이루어진 구조물에 대한 중량은 다음과 같이 나타낼 수 있다.

$$\text{Minimize } f(X) = \sum_{i=1}^m \rho A_i L_i \quad (1)$$

여기서  $A$ :단면적  $L$ :부재의 길이  $\rho$ :부재의 단위 하중이다.

일반적으로 구조 최적화에서 목적함수는 응력, 변위, 시공성에 관한 제약을 받고 있다.

### ① 변위제약

최대변위 및 층간변위 제약

$$\Delta_T \leq \Delta_{allow} \quad (\Delta_T : \text{최상층 변위}, \Delta_{allow} : \text{허용 최대변위})$$

$$\Delta_i - \Delta_{i-1} \leq \Delta_{allow}^s \quad (\Delta_i : i \text{ 층 변위}, \Delta_{allow}^s : \text{허용층간변위}) \quad (2)$$

### ② 응력 제약

$$\sigma \leq \sigma_{allow} \quad (\sigma : \text{작용응력}, \sigma_{allow} : \text{허용응력}) \quad (3)$$

### ③ 시공성 제약

동일 수직선상에 위치하는 기둥의 시공성을 고려하여 기둥과 가새에 대한 연결성을 제약한다.

$$A_{c_i} \leq A_{c_i} \quad (A_{c_i} : i \text{ 번째 층의 기둥단면적})$$

$$A_{br_i} \leq A_{br_i} \quad (A_{br_i} : i \text{ 번째 층의 가새단면적}) \quad (4)$$

그러나 유전알고리즘은 기본적으로 무제약 최적화 문제에 적용할 수 있기 때문에 제약이 있는 문제를 벌칙함수를 적용하여 무제약 최소화 함수로 변형해야 한다. 유전 알고리즘에서 벌칙함수의 적용은 가능(feasible)해를 보존하기 위한 방법으로 쓰이며 구속조건에 따라서 어떠한 구속조건에 가중치를 두어 벌칙함수를 적용하는가 하는 것이 중요하다.

$$\begin{aligned} \text{Minimize } F(X) &= f(X) & X \in \text{feasible} \\ F(X) &= f(X) + \text{penalty}(X) & X \in \text{infeasible} \end{aligned} \quad (5)$$

벌칙함수는 식(6)과 같이  $X$ 가 제약을 만족하면 0이고 제약을 만족하지 않으면 양의 값을 가지는 것으로 벌칙 상수와 위배량 제곱의 곱으로 표현된다. 벌칙함수는 어떠한 선택전략을 사용하는가에 따라서 달라질 수 있다. 룰렛휠에 의한 선택을 할 경우 목적함수의 값에 따른 선택확률에 따라서 선택되어지므로 벌칙상수와 제약조건간의 표준화에 따라서 해의 수렴성이 좌우된다. 아직까지 유전알고리즘을 포함한 진화프로그램에서 벌칙 함수를 위한 일반적인 규칙이 없으며 적절한 벌칙상수를 결정하는 것은 많은 시간과 노력이 필요하다.

$$\text{penalty}(X) = f(X_f) + \frac{1}{2} \left\{ r_1 \left[ \sum_{j=1}^n \frac{\delta_j}{\delta_j^a} \right]^2 + r_2 \left[ \sum_{j=1}^m \frac{\sigma_j}{\sigma_j^a} \right]^2 \right\} \quad X \in \text{infeasible}$$

$$X_f : \text{설계 가능영역중 중량이 가장 큰 개체} \quad m : \text{부재수} \quad n : \text{횡변위 자유도수} \quad (6)$$

본 논문에서는 Powell과 Skolnick이 제안한 방법을 수정하여 위의 식(6)과 같이 벌칙함수를 적용하였다. 또한 설계비가능영역의 적응도가 설계가능영역의 적응도보다 크게 평가되는 경우가 있음으로 해의 수렴이 벌칙상수에 너무 의존적이며 해집단내의 목적함수값의 표준화도 신중히 고려되어야 한다. 따라서 본 논문에서

서는 목적함수의 값의 크기보다 우열에 의하여 다음세대로 존속시킬 개체를 선택하는 방법인 토너먼트 선택 방법을 사용하였다.

### 3. 직렬 유전알고리즘

유전자 알고리즘은 다윈이 주장한 진화 법칙인 적자생존과 자연도태의 원리를 토대로 개발된 알고리즘이다. 진화론에 의하면 특정지역에 분포해 있는 개체들은 지역적 환경에 적합할수록 보존될 확률이 높으며 교배와 돌연변이를 거치면서 우성이 형질이 보존되고 열성인 형질은 도태된다. 다시 말하면 어떤 세대(Generation)를 구성하는 개체(Individual)들의 집합, 즉 개체군(Population)에서 환경에 대한 적합도(Fitness)가 높은 개체가 높은 확률로 살아남아 재생(Reproduction)할 수 있게 되며 이때 교배(Crossover) 및 돌연변이(Mutation)로서 다음 세대의 개체군을 형성하게 된다. 유전알고리즘이 다른 발견적 알고리즘(Heuristic Algorithm)이나 최적화 기법과 다른 점은 다음과 같다.

첫째, 설계변수를 코딩한 것을 직접 사용한다. 유전 조작이 용이한 0과 1로 이루어진 문자열을 설계변수 하나씩 나열하여 염색체 형태로 구성된 다음 교차와 돌연변이 등의 유전조작을 설계변수에 직접 가한다. 둘째, 기존의 고전적 최적화 기법이나 SA등 Heuristic 최적화 기법들이 하나의 해를 운용해 이웃해를 탐색해 나가는 방법인데 비해 유전알고리즘은 여러 개의 후보해 집단을 운용한다는 것이다. 이러한 군 탐색은 국부 최적점에 수렴하는 것을 방지해주고 전역적 탐색이 가능하도록 하여준다. 셋째, 최적해를 탐색해 나가는데 목적함수의 값만 이용할 뿐 경사도 벡터나 다른 정보를 필요로 하지 않는다. 특히 경사도 벡터를 필요로 하지 않기 때문에 유전알고리즘을 다양한 문제에 광범위하게 적용할 수 있다. 또한 유전 알고리즘은 내재적인 병렬성을 가지는 알고리즘이다. 다윈은 종 형성의 한 요인으로 지리적 격리를 강조했는데 자연 선별과 교배와 돌연변이등은 전체 집단이 아니라 대부분 제한된 한 지역을 중심으로 이루어진다. 이것은 다수의 부분 모 집단을 운용하여 각 부분 모집단간에 독립적으로 진화할 수 있다는 것을 나타낸다. 이러한 병렬화는 직렬알고리즘을 수행하는 경우 계산시간이 많이 걸리는 단점을 보완할 수 있다. 또한 각 세대간과 개체간에 의존성이 없으므로 독립적인 적용도 평가가 가능하다.

기본적인 유전알고리즘의 흐름도는 다음과 같다.

```

Initialize (Population)
Evaluate Fitness (Population)
DO WHILE (Stopping Criteria satisfied)
    Mating Pool = Reproduce (Population) by selection
    Inversion Pool = crossover (Mating Pool)
    Mutation Pool = Inversion (Inversion Pool)
    Population = Mutation (Mutation Pool)
    Evaluate (Population)
ENDDO
    
```

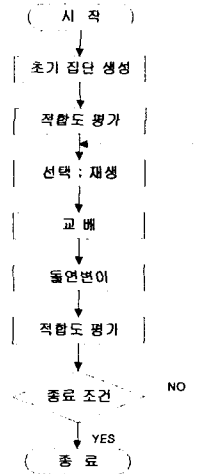


그림 1 SGA 흐름도

### 4. 병렬 유전알고리즘

#### 4.1 병렬시스템

병렬시스템은 메모리 방식이나 데이터 방식 그리고 프로그래밍방식에 따라서 분류할 수 있는데 본 연구에서는 SIMD(Single Instruction Multiple Data)방식으로 windows 98을 운영체제로 하는 PC를 Cluster로 구성하여 WMPI(Message Passing Interface for Windows) 패키지를 이용하여 통신한다. 각 Computer는

100Base-T의 Ethernet Network Card로 switching hub에 연결하며 한 대의 Master Computer와 여러 대의 Slave Computer를 tree방식으로 연결한다. 각 컴퓨터는 Pentium III 500M Hz의 CPU가 탑재되어 있으며 Windows 98을 운영체제로 하며 128M의 메모리를 가지고 있다. Master Computer에서는 유전연산을 담당하고 Slave Computer에서는 구조물의 물성치를 입력받고 구조해석과 제약조건의 위반량을 계산한다.

#### 4.2 병렬 유전알고리즘

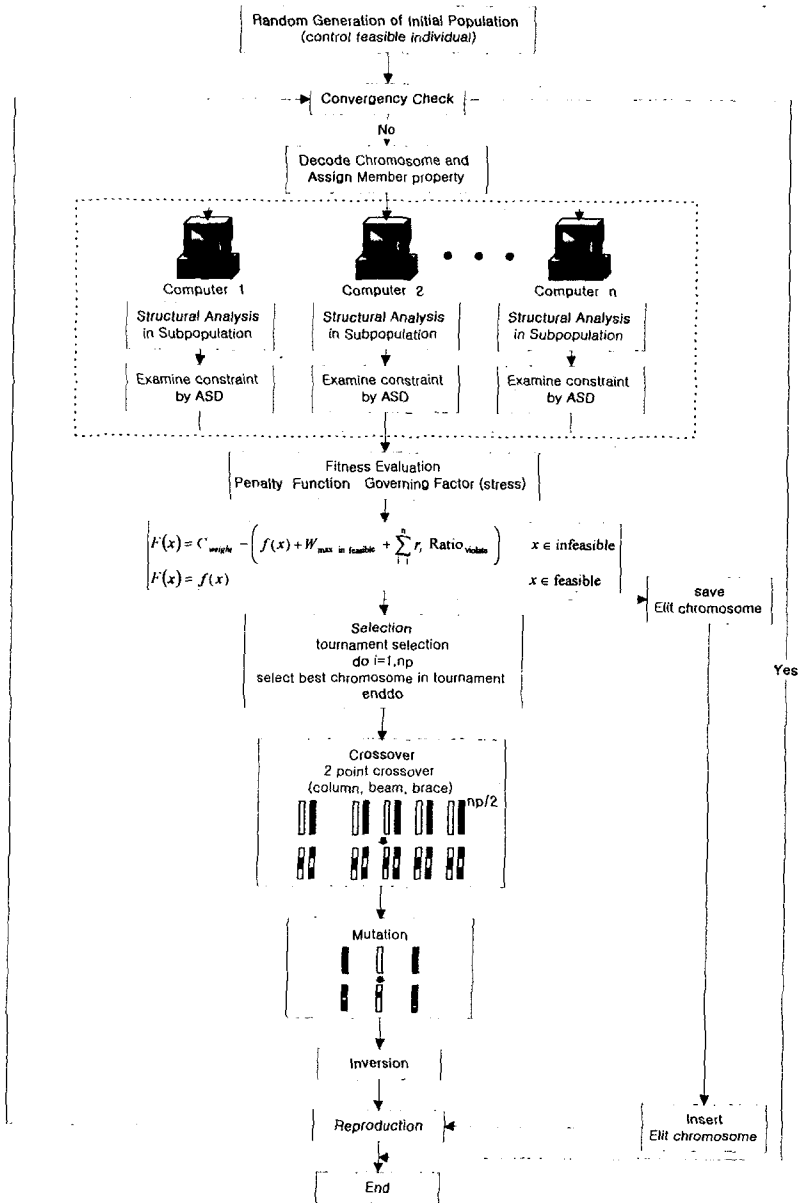


그림 2 병렬 유전알고리즘 흐름도

유전 알고리즘의 내재적인 병렬성은 알고리즘의 부분적인 병렬에서 전체로 적용하는 섬(island)모델까지 Parallel Region을 다양하게 적용할 수 있다. 구조 최적화를 위한 유전 알고리즘에서 연산시간의 대부분을 차지하는 것은 적응도 평가를 위한 구조해석이며 이 부분의 연산을 병렬화 함으로 전체 수행시간을 단축할 수 있다. 각 해집단 간의 적응도는 서로 의존적이지 않으며 서로 독립적으로 평가 될 수가 있다. 따라서 전체 해집단은 다수의 부분 모집단(Sub-Population)으로 나누어 질 수 있으며 이러한 하나의 부분 모집단에 대한 적응도 평가를 가상의 병렬 시스템상에서 slave computer가 담당하게 된다.

전체 모집단을 부분 모집단으로 분리하여 부분 모집단간에 독립적인 유전 알고리즘을 수행하는 섬(island) 모델의 경우 연산시간을 단축시킬 수 있지만 전체 모집단으로 운용하는 것과 같은 방식의 탐색을 하지 않으며 부분 모집단 수만큼 병렬탐색을 수행하여 개체를 이주시키기 때문에 또 다른 유전 파라미터 설정을 필요로 한다. 따라서 본 연구에서는 구조 최적화를 위한 직렬 유전 알고리즘에서 대부분의 연산시간을 차지하는 구조해석부분을 병렬화하여 적용하였다. 이것을 단계별로 정리하면 다음과 같다.

step 1 난수 발생기에서 0과 1의 수를 임의로 발생시켜 설계변수로 이어진 이진 string의 염색체를 구성하여 초기 모집단을 생성한다. 이때 설계 비 가용영역(Infeasible Region)에서 탐색하는 것을 방지하기 위하여 최소한 하나의 가용영역(Feasible Region)의 해를 둔다. 이를 위하여 부채의 제일 큰 단면을 가지는 개체를 하나 삽입한다.

step 2 수렴성 여부를 검토하여 수렴 조건을 만족할 경우 종료하고 만족하지 않을 경우 다음세대로 한세대 증가시킨다. 수렴성은 전체 설계가용영역중 최고의 적응도를 가지는 개체가 50%이상 차지할 때 수렴하는 것으로 가정하였다.

step 3 전체 모집단을 slave 수만큼 부분 모집단으로 분할하여 이진 스트링에서 decoding한 후보개체의 부재번호를 각 slave 프로세서로 전송한다.

step 4 각각의 Slave Computer는 부분모집단 개수만큼의 후보해에 대한 구조해석을 수행하여 제약조건에 대한 위배량을 계산한다. Slave Computer는 계산된 위배량을 Master Computer로 전송한다.

step 5 위배량을 전송받은 Master Computer는 목적함수 값을 유전알고리즘에 직접적용하기 위하여 제약이 있는 최적화 문제를 무제약 최적화 문제로 변환한 뒤 가목적 함수를 구성한다.

무제약 최적화 함수로 변환하기 위한 벌칙함수는 구조최적화를 위한 유전 알고리즘에서 선택전략과 의존적인 관계에 있다. 기존의 적응도 비례선택은 벌칙함수의 벌칙상수에 따라 수렴성이 큰 차이를 보이기 때문에 적절한 벌칙상수를 찾아야 하는데 이것은 반복실행과 많은 시간을 요하게 된다. 그러나 토너먼트 선택의 경우 목적함수의 값보다 해집단간의 우열에 따라 선택이 결정되어지므로 벌칙상수에 의존적으로 수렴성이 결정되지 않는다. 따라서 본 연구에서는 Powell의 방법을 수정한 아래와 같은 벌칙함수와 토너먼트 선택을 사용하였다.

$$\begin{aligned} \text{Minimize} \quad & F(X) = f(X) & X \in \text{feasible} \\ & F(X) = f(X) + \text{penalty}(X) & X \in \text{infeasible} \end{aligned}$$

$$\text{penalty}(X) = f(X_f) + \frac{1}{2} \left\{ r_1 \left[ \sum_{j=1}^n \frac{\delta_j}{\delta_j^a} \right]^2 + r_2 \left[ \sum_{j=1}^m \frac{\sigma_j}{\sigma_j^a} \right]^2 \right\}$$

$$x_f : \text{설계 가용영역중 중량이 가장 큰 개체} \quad m : \text{제약조건수} \quad n : \text{횡변위 자유도수} \quad (7)$$

위의 가목적 함수는 설계비가용영역의 개체가 가용영역의 개체보다 적응도가 높게 평가되지 못하도록 하기 위해 설계 설계가용 영역중 중량이 가장 큰 개체의 값을 더하고 제약 위배 비율에 따라 벌칙 함수를 적용하였다. 이때 기둥의 연결성 제약의 위배 여부에 따라  $f(X_f)_{X_f \in \text{Feasible}}$  을 더한다. 유전 알고리즘은 적응도가 높은 개체를 존속시키므로 가목적 함수를 아래와 변형시켜 목적함수를 구성한다.

$$\text{Maximize} \quad \text{Fitness}(X) = C_{\max} - F(X) \quad (8)$$

일반적으로  $C_{\max}$  는 상수로서 Fitness(X) 가 음이 되지 않을 만큼 큰 수를 택한다.

step 6 평가된 적응도를 바탕으로 최고의 적응도를 가지는 개체와 최저의 적응도를 가지는 개체를 구하여

최고의 적응도를 가지는 개체를 다음세대에 삽입시킬 엘리트 개체로서 보존한다. 그리고 전체 해집단중 설계 가용영역 해의 개수중 최고의 적응도를 가지는 개체가 중복해서 몇 개나 존재하는지 파악하여 수렴성 여부를 검토한다.

step 7 전체 개체가운데 임의의 개체를 토너먼트 개수만큼 뽑아서 그중 적응도가 가장 높은 개체를 다음 세대에 잔존시킨다. 이것을 다음 세대로 잔존시킬 개체수가 찰 때까지 반복한다.

step 8 설계변수로 구성된 이진스트링가운데 기둥과 보 그리고 가새 영역에서 각각 2점 교차를 실시한다. 각 영역에서 하나의 설계변수 안에서 교차가 일어나지 않도록 교차를 위해 선택되는 임의의 점이 설계변수의 string수보다 크게 한다.

step 9 돌연변이와 2점 역위(inversion)를 실시한다.

## 5. 예제적용

### 3경간 21층 가새골조 구조물 (35 Variable)

병렬의 효율성을 평가하기 위하여 slave computer의 개수가 2, 4, 6, 8 개별로 직렬알고리즘과 비교하였다. 그림3과 같이 입면상 대칭되는 위치에서 같은 구조적 역할을 하는 기둥, 보, 가새에 대하여 3개 층씩 그룹화 하였으며, 하중조건으로는 수직하중으로 고정하중과 활하중을 각각 보에 3290 kg/m, 1260 kg/m로 작용시키고 횡력은 KS규준에 의거 풍력을 산정하여 각 층당 절점에 작용시켰다. 하중조합은 횡력과 자중 고정하중 활하중이 동시에 작용할 때 최대 층변위와 층간변위에 대한 허용한계를 구조물의 층높이의 1/400로 제한하였으며 대한 건축학회 강구조 계산규준(허용 응력 설계법)에 의하여 부재에 대한 강도를 평가하였다. 사용 강재는 SS400 ( $F_y = 2.4 t/cm^2$ )으로 하였으며 탄성계수는  $2.04 \times 10^6 kg/cm^2$ , 단위중량은  $7.85 t/m^3$ 로 가정하였다. 각부재가 선택할 수 있는 단면은 기둥과 가새에 광폭(wide shape) 계열을 보는 세폭(narrow shape) 계열을 각각 16개 씩 사용하였다.

유전 파라미터의 설정은 De Jong의 Standard Setting을 기준으로 하여 반복실험을 통하여 적절한 파라미터를 설정하였으며 전체 해집단중 최고의 적응도를 가지는 개체가 Feasible Region가운데서 50%이상 차지하는 경우가 2회이상 반복될 때 수렴하는 것으로 가정하여 종료되도록 설정하였다. 유전 파라미터 설정은 표2와 같다.

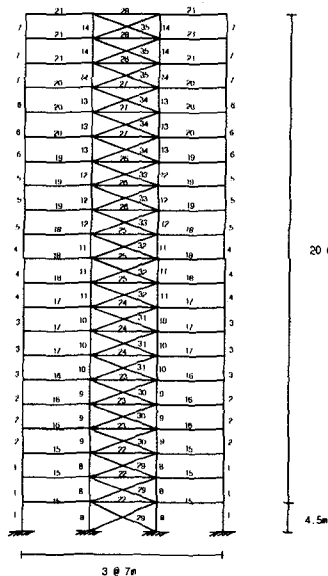


그림 3 3경간 21층 평면 가새골조

표 1 부재 단면 List (1~16:기둥, 17~32:보, 33~48:가새)

No	단면	단면적 (cm <sup>2</sup> )	단면2차모멘트 (cm <sup>4</sup> )	단면2차반경 (cm)
1	H-400*400*45*70	770.10	298000.00	11.10
16	H-150*150*7*10	40.14	1640.0	3.75
17	H-600*200*12*20	152.50	90400.00	5.11
32	H-250*125*5*8	32.68	3540.0	2.79
33	H-400*400*45*70	770.10	298000.00	11.10
48	H-150*150*7*10	40.14	1640.0	3.75

표 2 유전 파라미터

Population Size	120
Crossover rate	0.6
Mutation rate	0.01
Inversion rate	0.01
2-point Inversion rate	0.01
Tournament Size	8

직렬 알고리즘과 Slave의 개수가 2, 4, 6, 8개 일때의 병렬 알고리즘에 대하여 각 3번씩 실험하였으며 반복 횟수가 100번일때의 평균 소요시간을 측정하였다. 실험결과 그림4, 그림5와 같은 중량수렴곡선을 보이고 있다. 최적화 과정을 수행하는데 있어서 총 실험횟수에 대한 최종 수렴값의 편차가 10%미만의 신뢰성을 보였으며 수렴곡선의 경향은 병렬 Processor 수에 따라서 동일한 양상을 나타내었다.

Slave Processor개수별로 소요되는 시간은 개수의 정수배로 감소하고 있으며 Ideal Speedup과 근사한 경향을 보이고 있다. 그러나 Ideal Speedup보다 다소 높게 나타내는 경향을 보이는데 이것은 직렬유전 알고리즘이 병렬 알고리즘과 완전히 동일하지 않으며 단일 Processor사용으로 메모리 소요가 많기 때문이라 생각된다.

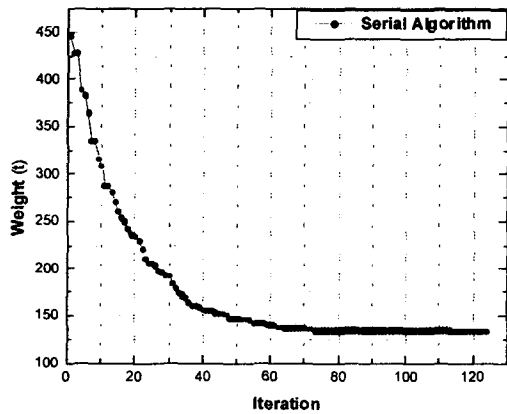


그림 4 직렬 유전알고리즘 수렴곡선

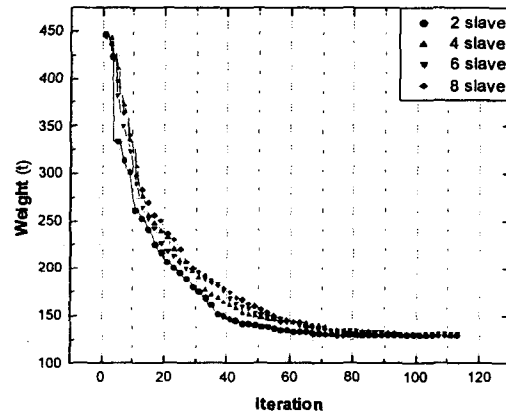


그림 5 병렬 유전 알고리즘 수렴곡선

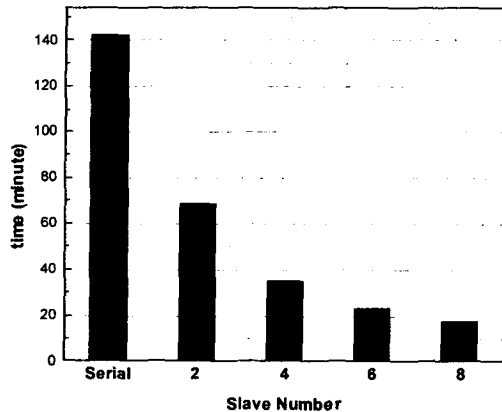


그림 6 최적화 수행시간

표 3 전체 최적화과정 수행시간 비교

	Time	weight
Serial	8549.4 sec	131.6 t
2 Slave	4119.1 sec	128.9 t
4 Slave	2092.9 sec	129.2 t
6 Slave	1400.4 sec	132.4 t
8 Slave	1039.4 sec	129.8 t

## 6. 결론

본 연구에서 구조 최적화를 위한 PC 상에서 병렬알고리즘을 개발하여 21층 평면 가새골조에 적용한 결과 다음과 같은 결론을 얻었다.

1) 각 세대별 전체 개체들을 병렬연산시스템에 연결된 slave 컴퓨터의 수에 해당하는 부분모집단으로 나누어 전체 개체들의 적응도 평가를 각 컴퓨터에서 동시에 수행함으로써 최적화에 소요되는 시간을 선형적으로 감소시켜 기존 GA알고리즘의 단점을 극복할 수 있었다.

2) 직렬 알고리즘과 slave 2, 4, 6, 8개의 병렬 알고리즘에 대하여 각 3번씩 실험하였으며 반복 횟수가 100번 일때의 평균 소요시간을 측정하여 평가한 결과 본 알고리즘은 slave 컴퓨터의 수에 무관하게 병렬성능이 발휘되는 scalability를 확보하고 있음을 확인할 수 있었다.

3) Powell의 방법을 수정한 벌칙함수와 토너먼트선택전략을 적용한 결과 총 실험횟수에 대한 최종 수렴값의 편차가 10%미만의 신뢰성을 보였다.

4) 다수의 개인용 컴퓨터를 사용한 가상병렬연산시스템을 기반으로 개발된 병렬유전알고리즘은 ideal speedup에 근사한 효율성을 얻을 수 있었으나 부구조 기법을 이용한 병렬 해석을 이용할 경우 전체 해석시간을 더욱 단축시켜 대형구조물의 최적설계에 적합한 실용적 유전자 알고리즘을 개발할 수 있으리라 사료된다.

## 감사의 글

본 논문은 1999년도 한국학술진흥재단의 연구비에 의하여 연구되었음. (KRF-99-041-E00598)

## 참고문헌

1. Goldberg, D. E. (1989), "Genetic Algorithms in search, Optimization and machine learning", Addison-Wesley.
2. Richard J. Balling (1991), "Optimal Steel Frame Design By Simulated Annealing", J. of Structural Engineering, Vol. 117, 1991, pp1780-1795
3. 김여근, 윤복식, 이상복 (1997), "메타 휴리스틱", 영지문화사
4. Adeli, H., and Cheng, N.T. (1994b), "Concurrent Genetic Algorithms for Optimization of Large Structures", J. of Aerospace Engineering, ASCE, Vol. 7, No3. pp. 276-296
5. Adeli, H., Kumar, S. (1999), "Distributed Computer-Aided Engineering", CRC Press. pp115-146.
6. Areibi, S., and Vannelli, A.(1998). "Distributed Advanced Search Techniques for Circuit Partitioning." Proc., 11th Can. Conf. on Electr. and Comp. Engineering., IEEE, Piscataway, N. J., Vol. 2, 553-556.
7. Jenkins, W.M, "Structural Optimization with the genetic algorithm" , the structural engineer, col 69, No. 24, 17 December 1991, pp. 418-422
8. Adeli, H. and Cheng, N.T. (1993), "Integrated Genetic Algorithms for Optimization of Space Structures", J. of Aerospace Engineering ASCE, Vol, No 4, pp 315-328
9. Adeli, H., and Cheng, N.T. (1994a), "Augmented Lagrangian Genetic Algorithm for Structural Optimization", J. of Aerospace Engineering ASCE, Vol. 7 NO 1, pp. 104-118