

분산 복합유전알고리즘을 이용한 구조최적화

Distributed Hybrid Genetic Algorithms for Structural Optimization

우 병 헌†

Woo, Byung-hun

박 호 선*

Park, Hyo Seon

(논문접수일 : 2003년 7월 10일 ; 심사종료일 : 2003년 11월 14일)

요 지

최근 구조최적화분야에서 활발하게 사용되고 있는 유전알고리즘은 해집단을 운용하기 때문에, 많은 반복수와 적응도 평가를 위하여 해집단의 수에 해당하는 구조해석을 필요로 하며, 또한 교배율과 돌연변이율 등의 파라미터에 따라 알고리즘의 성능이 변화하므로 문제에, 따라 적합한 파라미터 설정이 필요한 근본적인 단점을 지니고 있다. 본 연구에서는 기존 유전알고리즘의 단점을 극복할 수 있는 복합유전알고리즘을 마이크로유전알고리즘과 단순유전알고리즘을 결합한 형식으로 그리고, 최적화에 요구되는 연산을 다수의 개인용 컴퓨터에서 동시에 분산하여 수행할 수 있는 고성능 분산 복합유전알고리즘으로 개발하였다. 개발된 알고리즘은 철골 가새골조 구조물의 최소중량설계에 적용하여 그 성능을 평가하였다.

핵심용어 : 구조최적화, 유전알고리즘, 고성능연산, 분산연산, 철골구조

Abstract

Even though several GA-based optimization algorithms have been successfully applied to complex optimization problems in various engineering fields, GA-based optimization methods are computationally too expensive for practical use in the field of structural optimization, particularly for large-scale problems. Furthermore, a successful implementation of GA-based optimization algorithm requires a cumbersome and trial-and-error routine related to setting of parameters dependent on a optimization problem. Therefore, to overcome these disadvantages, a high-performance GA is developed in the form of distributed hybrid genetic algorithm for structural optimization on a cluster of personal computers. The distributed hybrid genetic algorithm proposed in this paper consist of a simple GA running on a master computer and multiple μ -GAs running on slave computers. The algorithm is implemented on a PC cluster and applied to the minimum weight design of steel structures. The results show that the computational time required for structural optimization process can be drastically reduced and the dependency on the parameters can be avoided.

Keywords : structural optimization, genetic algorithms, high-performance computing, distributed computing, steel structures

1. 서 론

최근 급속도로 발전해온 컴퓨터 관련기술은 공학 분야 전반에 많은 변화를 가져왔으며, 구조최적화 문제에도 이에 따라 많은 영향을 받았다. 그 결과 최근에는 컴퓨터를 기반으로 한 조합최적화(combinatorial optimization) 기법

인 Genetic Algorithms(GAs),¹⁾ Simulated Annealing(SA),²⁾ Tabu Search³⁾등이 구조최적화 문제에 빈번히 적용되어 그 효율성이 평가 되고 있다. 그러나 컴퓨터 관련 기술이 급속도로 발전하는 현재에도 구조최적화를 위한 조합최적화 기법은 최적해를 찾는 동안 최소한의 구조 정보를 이용하는 알고리즘의 특성 때문에 반복회수가 많아

† 책임저자, 정회원 · 롯데건설(주) 건축기사
전화: 051-581-9485; Fax: 051-581-9485
E-mail: uswest75@orgio.net

* 정회원 · 연세대학교 건축·도시공학부 부교수

* 이 논문에 대한 토론을 2004년 3월 31일까지 본 학회에 보내주 시면 2004년 6월호에 그 결과를 게재하겠습니다.

저 계산시간의 과다로 인하여 실무 적용에 한계를 가지고 있다.

특히 유전알고리즘은 해집단을 운용하기 때문에 다른 최적화 알고리즘에 비하여 많은 반복수와 적응도 평가를 위한 해집단의 수만큼 구조해석을 필요로 한다.⁴⁾ 그리고 유전알고리즘의 안정성과 수렴성은 교배율과 돌연변이율 등의 파라미터에 따라 변화하므로 문제에 따라 적합한 파라미터 설정이 필요하다. 그러므로 유전알고리즘은 해석시간의 증가에 따른 전체 알고리즘의 수행 시간이 증가 하는 문제와 더불어 각 문제에 따라 각기 다른 파라미터 설정이 필요하다라는 두 가지 큰 단점을 가지고 있다.

본 연구에서는 과다한 수행시간과 문제에 따라 요구되는 파라미터 설정 등의 기존 유전알고리즘의 단점을 극복할 수 있는 복합유전알고리즘을 마이크로유전알고리즘(μ -GAs)과 단순유전알고리즘(SGA)을 결합한 형식으로 그리고 최적화에 요구되는 연산을 다수의 개인용 컴퓨터에서 동시에 분산하여 수행할 수 있는 고성능 분산 복합유전알고리즘(distributed hybrid genetic algorithms, DHGAs)의 형식으로 개발하고자 한다. 개발된 알고리즘은 철골 가새 골조 구조물의 최소중량설계에 적용하여 그 성능을 평가하고자 한다.

2. 구조 최적화를 위한 정식화

본 논문에서 구조최적화 문제는 구조물의 총 중량을 최소화하는 것을 목적함수로 하고 있으며, M 개의 부재로 이루어진 구조물에 대한 중량은 식(1)과 같이 나타낼 수 있다.

$$\text{Minimize } f(\mathbf{X}) = \sum_{i=1}^M \rho_i A_i L_i \quad (1)$$

여기서 ρ 는 부재의 단위 중량, A 는 단면적, L 은 부재의 길이를 나타내며 M 은 전체 부재의 개수를 의미한다. 본 논문에서의 구조 최적화를 위한 정식화에는 응력, 변위, 부재간의 총 연결성에 관한 제약이 포함되어 있으며 각 제약함수의 수식은 식(2), (3), (4), (5), (6)과 같다.

$$g_1 = \frac{\Delta^T}{\Delta_a^T} \leq 1.0 \quad (2)$$

$$g_2 = \frac{\Delta^i - \Delta^{i-1}}{\delta_a^i} = \frac{\delta^i}{\delta_a^i} \leq 1.0 \quad i=1, N \quad (3)$$

$$h = \frac{\sigma_j}{\sigma_a^j} \leq 1.0 \quad j=1, M \quad (4)$$

$$k_1 = \frac{A_c^{i-1}}{A_c^i} \leq 1.0 \quad i=1, N \quad (5)$$

$$k_2 = \frac{A_r^{i+1}}{A_r^i} \leq 1.0 \quad i=1, N \quad (6)$$

여기서 Δ^T 와 Δ_a^T 는 최상층 그리고 허용최대변위이고 Δ^i 와 δ^i 는 i 번째 층 변위 및 층간변위이며 δ_a^i 는 i 번째 층의 허용층간변위이다. σ^i 와 σ_a^i 는 i 번째 부재에 발생하는 응력과 허용응력을 나타낸다. A_c^i 와 A_c^{i+1} 그리고 A_r^i 와 A_r^{i+1} 는 동일 수직선상에 있는 i 와 $i+1$ 층의 기둥 및 가새의 단면적을 나타낸다. M 은 전체 부재의 개수이며 N 은 전체 층의 수이다.

기본적으로 유전알고리즘은 무제약 최소화 문제이므로, 식(1)의 목적함수 및 변위와 응력에 대한 제약조건 식(2), (3), (4)를 벌칙함수를 사용하여 식(7)과 같이 무제약 최소화 문제로 변환할 수 있다.

$$\begin{aligned} &\text{Minimize} \\ &F(\mathbf{X}) = f(\mathbf{X}), \quad \mathbf{X} \in \text{feasible} \\ &F(\mathbf{X}) = f(\mathbf{X}) + P(\mathbf{X}), \quad \mathbf{X} \in \infty\text{feasible} \end{aligned} \quad (7)$$

본 논문에서는 외부벌칙함수법(exterior penalty function method)⁹⁾를 이용하여 벌칙함수를 식(8)과 같이 표현하였다.

$$\begin{aligned} P(\mathbf{X}) = &f(\mathbf{X}_j) \\ &+ \frac{1}{2} \left(r_1 \sum_{i=1}^N \left(\frac{\delta^i}{\delta_a^i} \right)^2 + r_2 \sum_{i=1}^M \left(\frac{\sigma^i}{\sigma_a^i} \right)^2 \right) \end{aligned} \quad (8)$$

여기서 r_1 과 r_2 는 벌칙상수이며 $f(\mathbf{X}_j)$ 는 설계가용 영역 중량이 가장 큰 개체의 목적함수이며, 식(5)와 (6)은 부 제약조건(side constraints)으로 프로그램을 구성하였다.

3. 유전알고리즘

3.1 유전알고리즘(GAs)

유전알고리즘은 다윈이 주장한 진화 법칙인 적자생존과 자연도태의 원리를 토대로 개발된 알고리즘이다. 유전알고리즘은 2진수로 코딩된 유전자를 사용하며, 최적해를 찾아가는 동안 목적함수 값만을 사용한다. 그러므로 다양한 문제에 적용하기 쉬운 장점이 있다. 또한 유전알고리즘은 해집단 내에서 각 개체간의 상호의존성이 없으므로 병렬화하기 용이한 알고리즘이다.⁴⁾ 유전알고리즘은 유전자 부동을 방지하고 안정적으로 최적해를 탐색하기 위하여 30개 이상의 개체를 사용하여야 하며, 각 문제에 따른 적절한 파라미터 설정이 필요하므로 유전알고리즘은 모든 문제에 일반적으로 적용하기 쉽지 않다는 단점이 있다.¹⁾

3.2 마이크로 유전알고리즘(μ -GAs)

개체군이 증가함에 따라 유전알고리즘은 안정적이며 신속한 최적해로의 수렴양상을 보이는 경향이 있다.^{1),7)} 그러나 개체군의 증가는 안정적인 수렴을 보장하지만 알고리즘의 수행시간의 증가라는 양면성을 가지고 있다. 이와 같이 개체 수 증가에 따라 해석 시간이 증가하는 문제를 해결하기 위하여, Goldberg¹⁾는 직렬유전알고리즘(serial GA) 안에서 작은 집단을 이용하여 효율적인 탐색을 할 수 있는 방법을 제시하였다.^{1),7)} 이 접근방법에 기초하여 Krishnakumar⁵⁾는 매우 작은 집단을 운용하는 마이크로 유전알고리즘(μ -GAs) 제안하였다.

Krishnakumar가 제안한 마이크로 유전알고리즘은 진화의 연료라 불리는 돌연변이 연산을 배제하며, 1.0의 고정된 교배율을 사용한다. 마이크로 유전알고리즘에서는 돌연변이 연산을 배제하는 대신 재시동 연산을 도입하여 개체군의 다양성을 보존하도록 하였다. 그러므로 초기 개체군을 10개 내외에서 설정하는 문제를 제외하면 기타의 유전연산에 필요한 파라미터의 설정에 따른 번거로움은 사라지게 된다. 그러나 마이크로 유전알고리즘은 디자인 변수가 증가하고 디자인 영역이 넓어지는 경우 안정적인 수렴에 한계를 보이는 특성을 가지고 있다.

4. 분산 유전알고리즘

4.1 분산연산시스템

그림 1은 본 논문에서 사용하는 분산연산시스템의 구성도이며 각 PC를 Ethernet 방식의 LAN으로 연결하여 구성하였다. 프로세서부분에 해당하는 각 PC는 1.6GHz의 펜티엄 IV Northwood 프로세서와 256MB의 SDRAM을 탑재하고 있으며, 7200RPM 10GB의 하드디스크 용량을 가지고 있다. 각 PC의 운영체제는 윈도우98을 사용한

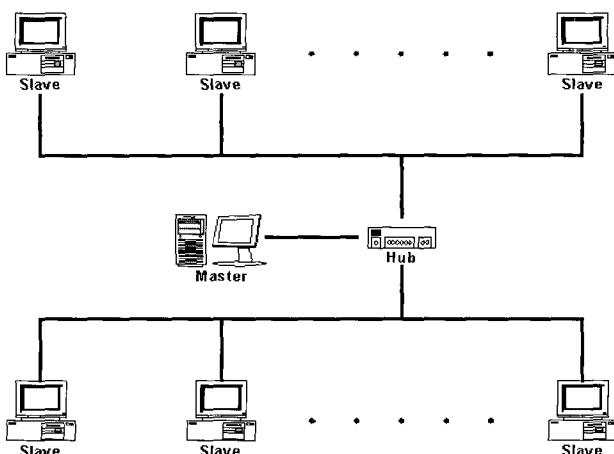


그림 1 분산연산시스템

다. 병렬시스템의 네트워크 구성은 100Mbit/s의 속도를 낼 수 있는 Faster Ethernet 방식을 사용하였으며, 다수의 슬레이브 프로세서(slave Processor)들은 스위칭 허브를 통해 마스터 프로세서(master Processor) 및 전체 네트워크에 연결되어 있으며 프로그램은 WMPI 1.3 Library를 사용하여 구성되어 있다.

4.2 분산 복합 유전알고리즘(DHGAs)의 구성요소

DHGAs의 마스터 프로세서에서는 μ -GAs가 그리고 다수의 슬레이브 프로세서들에서는 각각 독립적인 SGAs가 그림 2의 개념도와 같이 수행된다. 마스터 프로세서에서는 소수의 개체를 운용할 수 있으며 유전연산에 필요한 파라미터의 설정에 따른 번거로움을 해결할 수 있는 μ -GAs의 장점을 유지할 수 있게 된다. 또한 각 슬레이브에서 독립적으로 진화된 우수한 엘리트(elite) 개체들을 마스터 프로세서에 이주(migration) 연산을 통하여 공급하여 제한적 탐색영역을 운영하는 μ -GAs의 단점을 극복할 수 있다는 개념으로 구성된다.

4.2.1 마스터 프로세서에서의 마이크로 유전알고리즘

Krishnakumar가 제안한 마이크로 유전알고리즘의 특징은 3.2절에서 서술한 바와 같으며 재시동 연산을 수행할 때 엘리트 개체를 제외한 전체 개체를 임의로 생성하여 새로운 개체군을 형성한다. DHGAs에서는 재시동 연산 시에 엘리트 개체를 제외하고 새롭게 생성되는 개체를 임의로 생성하는 것이 아니라, 각각의 슬레이브 프로세서에서의 엘리트 개체를 이용하여 새로운 개체군을 형성하므로 유도된 초기화 해당하며 그 결과 계산시간 및 수렴속도를 증가시킬 수 있다는 장점을 가지게 된다.⁷⁾

4.2.2 슬레이브 프로세서에서의 단순 유전알고리즘

슬레이브 프로세서에서는 단순유전알고리즘이 사용되며, 각 유전연산에 있어서의 파라미터는 De Jong이 제시한 기

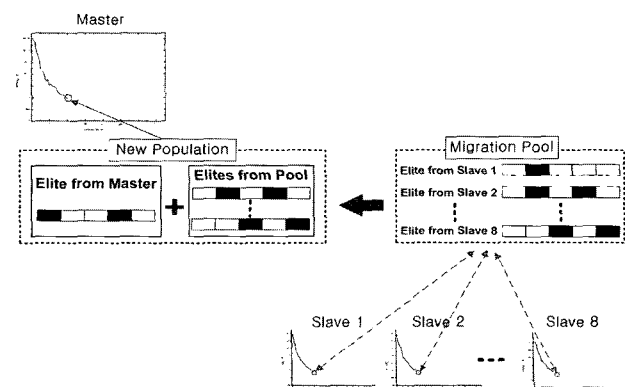


그림 2 DHGAs의 개념도

본 파라미터 설정이 사용된다.⁶⁾ 각 파라미터들은 문제에 따라 변화 하는 것이 아니라 고정된 파라미터로 본 논문에서는 일반적인 개체군 30개, 교배율 0.6, 돌연변이율 0.03이 사용된다. 각 슬레이브 프로세서들은 마스터 프로세서가 재시동하여 새로운 개체군을 구성하기 위하여 엘리트 개체를 이주하라는 메시지가 전달될 때까지 독립적인 진화 과정을 수행한다.

4.3 복합 분산 유전알고리즘(DHGAs)

그림 3은 그림 2의 개념도를 바탕으로 구체화한 DHGAs의 흐름도이다. 본 논문에서는 슬레이브 프로세서를 8개 까지 증가시키며 시뮬레이션 하였다. 마스터 프로세서와 슬레이브 프로세서에서는 기존의 마이크로 유전알고리즘과 단순유전 알고리즘이 각각 운용된다. 그러나 병렬로 결합된 전체 알고리즘은 마스터 프로세서에서 운용되는 마이크로 유전알고리즘에 의해서 제어 되므로 마이크로 유전알고리즘이 가지는 소수의 개체를 운용하며, 파라미터 설정이 간소하다는 장점들을 그대로 활용할 수 있다.

DHGAs에서는 분산시스템과 복합알고리즘을 구성하는 각각의 프로세서들을 연결하며, 상호보완적인 진화과정을 수행하기 위하여 이주 형태의 유전연산자를 가지고 있다. 유전연산자는 그림 3에서 마스터 프로세서와 슬레이브 프로세서의 사이에 존재한다. 마스터 프로세서가 재시동할 때 각각의 슬레이브 프로세서들에서 수집한 엘리트 개체를

저장하며 필요한 개체 수만큼 마스터 프로세서에 공급한다. 마스터 프로세서로의 공급이 종료되면 수집한 엘리트 개체를 각각의 슬레이브 프로세서에 다시 분배하는 역할을 하는 이러한 이주연산자는 DHGAs에 있어서 가장 중요한 역할을 하는 유전연산자라 할 수 있다.

일반적으로 병렬 유전알고리즘에서는 이주 빈도와 이주율 이라는 두 가지의 새로운 파라미터가 필요하며, 이는 병렬 유전알고리즘의 성능에 큰 영향을 미치는 중요한 파라미터가 된다. 그러나 본 논문의 DHGAs에서는 이 두 가지의 파라미터는 마스터 프로세서가 재시동시 이주하며, 각 슬레이브 프로세서의 엘리트 개체가 이주에 참여 하므로 이주 시기와 횟수 그리고 이주율에 관한 문제는 해결된다.

4.4 분산 복합 유전알고리즘의 검증

본 연구에서 새로이 개발된 분산 복합 유전알고리즘을 검증하기 위하여 기존 최적화 연구에서 많이 사용되었던 그림 4의 25부재 입체 트러스에 적용하여 최적해를 비교하였다. 그림 4의 25부재 입체 철골트러스 구조물은 탄성계수 $E=68.95$ GPa, 단위중량 $\rho=27.17$ kN/m³이며 제약조건으로는 1번과 2번 절점에서 수평변위가 ± 0.889 cm 이내로 하고, 각 부재의 최대 허용압축과 인장응력은 ± 275.6 MPa로 제한한다. 25부재 트러스 예제는 같은 구조적 거동을 하는 부재를 8개로 그룹으로 나누었다. 25부재 트러스 구조물의 하중은 표 1과 같다.

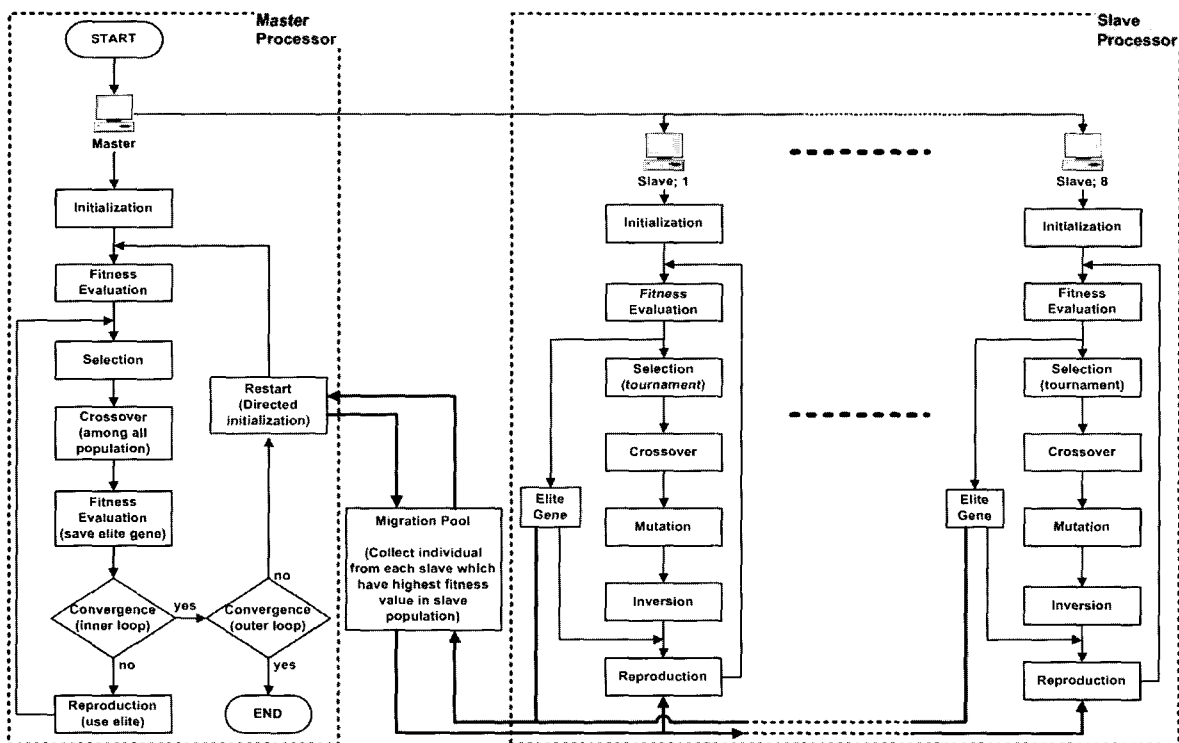


그림 3 DHGAs의 순서도

표 1 하중조건

절점	하중(Kips)	X	Y	Z
1	1.0	10.0	-10.0	
2	0	10.0	-10.0	
3	0.5	0	0	
6	0.6	0	0	

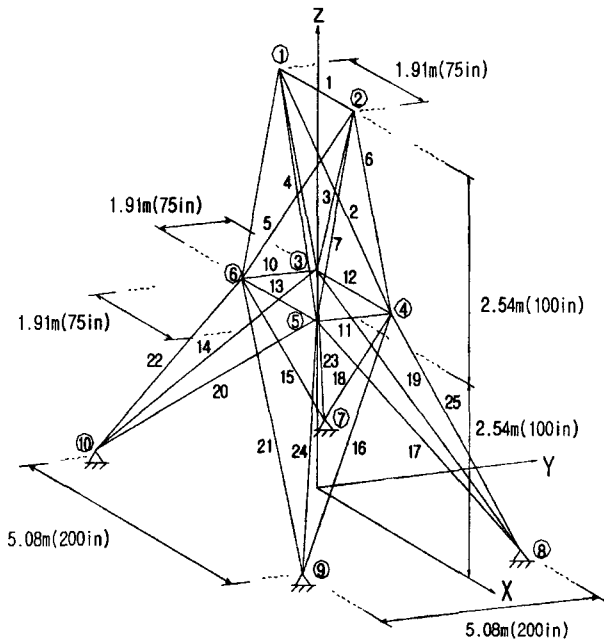


그림 4 25부재 입체 트러스 구조물

또한, 각 디자인 변수가 선택할 수 있는 단면은 {0.65, 1.29, 1.94, 2.58, 3.23, 3.87, 4.52, 5.16, 5.81, 6.45, 7.10, 7.74, 8.39, 9.03, 9.68, 10.32, 10.97, 11.61, 12.26, 12.90, 13.55, 14.19, 14.84, 15.48, 16.13, 16.77, 18.06, 19.35, 20.65, 21.94}(cm²)의 30개로서 이산형 단면을 사용한다.

표 2 25부재 입체 트러스의 최적해 비교

Variable (Member)	Optimum Solution, in ²				
	Zhu(1986) ¹⁰⁾	Rajeev(1992) ¹¹⁾	NDM(1998) ¹²⁾	DHGAs	
				Lowest	Highest
1 (1)	0.65 (0.1)	0.65 (0.1)	3.87 (0.6)	0.65 (0.1)	0.65 (0.1)
2 (2,3,4,5)	12.26 (1.9)	11.61 (1.8)	9.03 (1.4)	9.68 (1.5)	1.20 (0.2)
3 (6,7,8,9)	16.77 (2.6)	14.84 (2.3)	18.06 (2.8)	20.00 (3.1)	21.94 (3.4)
4 (10,11)	0.65 (0.1)	1.20 (0.2)	3.23 (0.5)	0.65 (0.1)	0.65 (0.1)
5 (12,13)	0.65 (0.1)	0.65 (0.1)	3.87 (0.6)	5.16 (0.8)	16.77 (2.6)
6 (14,15,16,17)	5.16 (0.8)	5.16 (0.8)	3.23 (0.5)	6.45 (1.0)	6.45 (1.0)
7 (18,19,20,21)	13.55 (2.1)	11.61 (1.8)	9.68 (1.5)	1.94 (0.3)	3.23 (0.5)
8 (22,23,24,25)	16.77 (2.6)	19.35 (3.0)	19.35 (3.0)	21.94 (3.4)	21.94 (3.4)
Weight, kN(lb)	2.504 (562.9)	2.429 (546.9)	2.420 (543.9)	2.127 (487.13)	2.246 (504.96)

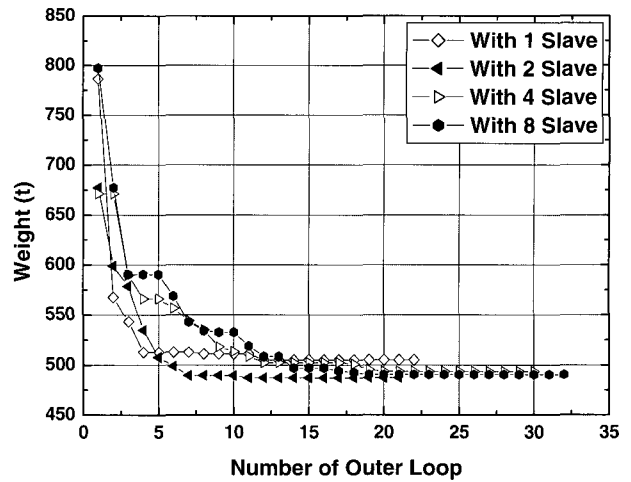


그림 5 25부재 입체 트러스의 수렴곡선

그림 5는 25부재 입체 철골 트러스 예제의 각 모델별 수렴곡선이다. 진화의 초기에는 각 모델별로 수렴 속도의 차이가 있으나 종국에는 모두 안정된 수렴 양상을 보이고 있다. 각 모델별 최적해는 최대값 2.246kN (504.96 lb), 최소값 2.127kN (487.13 lb)를 나타내며, 최대오차 3% 미만의 오차를 보이며 수렴한다. 표 2는 기존의 최적 알고리즘에 의한 25부재 입체 철골 트러스의 최적해를 비교한 표이다. 표에서 알 수 있듯이 분산 복합 유전알고리즘이 기존의 최적화 알고리즘들에 비하여 뛰어난 탐색 능력을 보임을 알 수 있다.

5. 예제적용

DHGAs의 효율성을 평가하기 위하여 강구조 정형 가새 골조 구조물의 최소 중량 설계에 적용하여 그 효율성을 평가 하였다. 1개의 마스터와 1개의 슬레이브로 구성되는 모델을 기본으로 하여 슬레이브 프로세서를 2, 4, 8개로 증가시키며 그 성능을 평가하였다.

5.1 12층 가새골조 구조물

그림 6의 예제는 12층 정형 가새골조 구조물로서 총 52 노드를 가지며 108개의 부재로 구성되어 있으며, 108개의 부재는 3개 층 단위로 그룹핑 하여 20개의 그룹으로 구분된다.

예제로 사용된 건물은 1층의 높이는 4.5 m이며 상부층은 3.75 m 로 구성되며 3개의 경간은 7 m 로 동일하다. 하중조건으로는 수직하중으로 고정하중과 적재하중을 각각 3.29 kN/m², 1.26 kN/m² 로 작용시키고, 횡력은 KS규준에 의거 35 m/sec 풍력을 산정하여, 각 층 당 절점에 작용시켰다. 하중조합은 횡력과 자중, 고정하중, 적재하중이 동시에 작용할 때 최대 층변위와 층간변위에 대한 허용한계를 구조물의 층높이의 1/400로 제한하였으며, 대한건축학회 강구조 계산규준(허용응력설계법)에 의하여 부재에 대한 강도를 평가 하였다.

사용강재의 탄성계수는 2.0×10⁸ kN/m², 단위중량은 77 kN/m³ 로 하였다. 각부재가 선택할 수 있는 단면은 표 3과 같이 기둥과 가새에 광폭(wide shape) 계열을, 보

표 3 부재 단면 List

No	단면	단면적 (cm ²)	단면2차 모멘트(cm ⁴)	단면2차 반경(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

(1~16:기둥, 17~32:보, 33~48:가새)

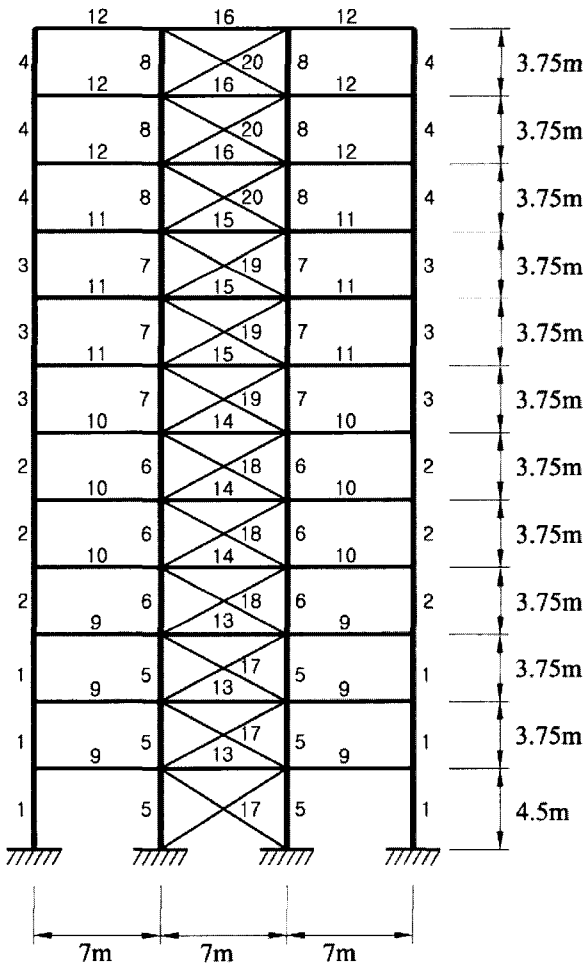


그림 6 12층 정형 가새골조

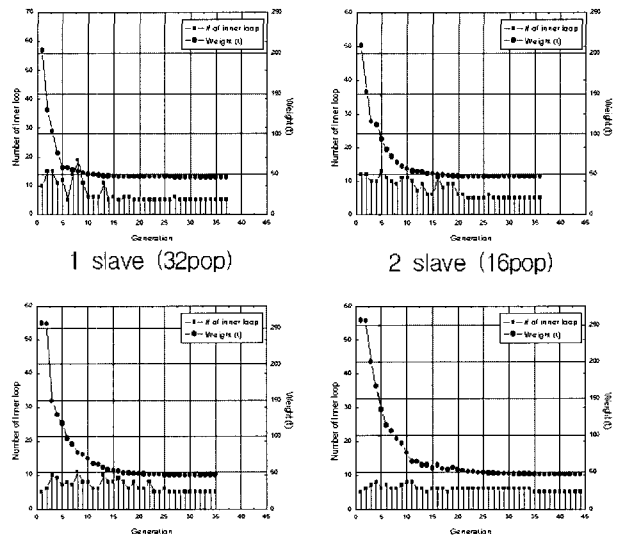


그림 7 각 모델별 수렴곡선 및 내부루프의 횡수

는 세폭(narrow shape) 계열을 각각 16 씩 사용하였다.

그림 7은 각 모델별 수렴곡선 및 내부루프의 횡수를 나타내는 그래프이다. 그림 7의 꺾은선 그래프는 목적함수를 나타내며 아랫부분의 막대그래프는 내부루프의 횡수를 나타낸다. 그림 7의 4가지의 그래프는 1-마스터 1-슬라이브 모델을 기준으로 슬라이브 프로세서의 개수를 2, 4, 8개로 증가시켜 가면서 구성된 모델의 수렴양상을 나타낸 그래프이다.

그림 7에서 볼 수 있듯이 각 모델별 목적함수의 수렴양상은 종료 시까지 외부루프의 최대 반복횟수 35회~45회 사이에서 모두 안정된 수렴양상을 보이며 종료함을 알 수 있다.

그림 7의 각 모델별 그래프의 내부루프의 횡수를 보면 내부루프의 최대횡수가 나타나는 양상이 서로 다름을 알 수 있다. 즉 가장 많은 내부루프가 나타나는 세대가 슬라이브 프로세서의 수가 증가하고, 마스터 프로세서가 운용하는 개체군이 적을수록 늦어짐을 알 수 있다. 또한 최대 반

복수와 최소반복수의 차이 또한 줄어드는 양상을 보인다. 내부루프의 횟수가 증가하는 이유는 진화가 빠르게 진행된다는 것을 의미한다. 즉 한번의 외부루프에 많은 진화가 있을 경우에 내부루프의 횟수가 증가하기 때문이다. 각 모델별 수렴 곡선을 서로 중복하여 나타낸 그림 8을 보면 진화 속도의 차이를 명확히 관찰 할 수 있다.

그림 8에서 나타나는 것처럼 진화의 초기 단계에서 내부루프의 횟수가 많은 1-마스터 1-슬라이브 모델의 경우 초기에 다른 모델에 비하여 빠른 진화 속도를 보여 주고 있다. 반대로 슬라이브 프로세서의 개수가 증가할수록 진화의 초기 단계에서 진화속도가 감소하는 양상을 보인다. 이와 같은 현상은 많은 개체수를 운용하는 모델의 경우 재시동 연산이 없이도 일정부분의 탐색이 활발함을 의미한다.

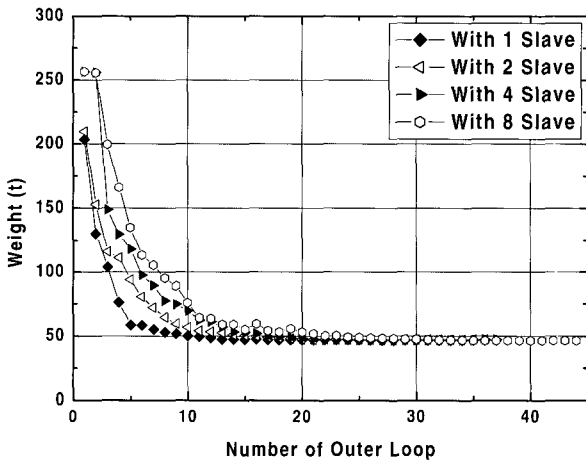


그림 8 각 모델별 수렴곡선

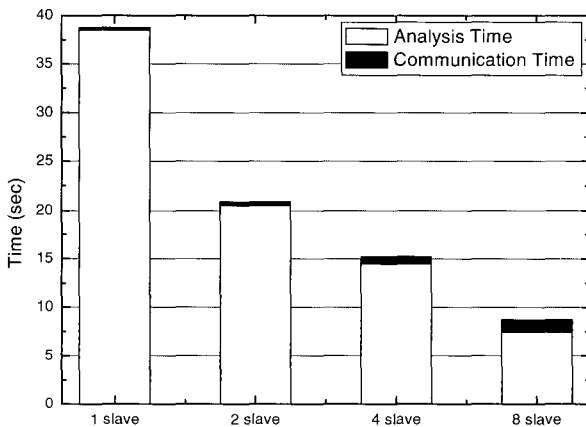


그림 9. 알고리즘 수행시간의 분석

표 4 최적해 및 알고리즘 수행시간

슬라이브 수	1	2	4	8	
중량 (t)	48.33	47.22	47.88	48.12	
시간 (sec)	전체	38.74	20.88	15.19	8.69
	통신	0.27 (0.67%)	0.38 (1.83%)	0.72 (4.74%)	1.26 (14.50%)

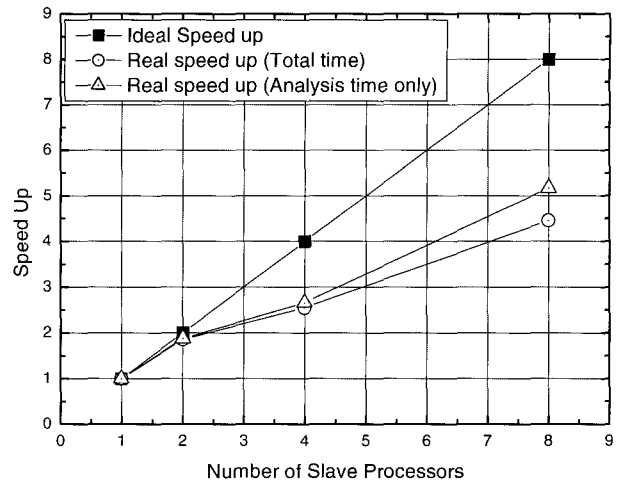


그림 10. Speed Up 그래프

하지만 진화의 중·후반으로 갈수록 진화 속도의 차이는 점차 감소하여 반복수 30회 정도에서는 동일한 양상을 보인다. 이는 초기 수렴속도가 떨어지는 모델의 경우라 할지라도 많은 슬라이브 프로세서를 운용함에 따르는 분산 유전알고리즘의 효과를 이용하여 세대가 증가할수록 진화속도를 가속화하기 때문이다.

그림 7에서 나타난 내부루프의 횟수의 그래프와 그림 8의 진화 속도를 비교하면, 내부루프의 횟수의 증가와 진화의 속도는 일치함을 알 수 있다. 표 4는 12층 정형 가새골조 구조물을 각 모델별로 시뮬레이션한 결과를 나타낸 표이다. 각 수치는 연속된 5회의 시뮬레이션 결과값을 평균한 값이다. 각 모델별 평균 최소 중량은 3% 미만의 오차를 보이며 안정된 수렴 양상을 보여 주고 있다. 표 4는 각 알고리즘을 5회 연속 시뮬레이션 하여 평균 수행시간을 보여 준다. 슬라이브 프로세서가 증가함에 따라 모델의 알고리즘 수행 시간은 감소하는 것을 알 수 있다. 알고리즘의 수행 시간은 전체 알고리즘수행 시간과 각 모델별 알고리즘의 수행 시간에서 통신 시간이 차지하는 시간과 비율을 표시하고 있다. 슬라이브 프로세서가 증가함에 따라 전체 알고리즘수행 시간은 감소하지만 알고리즘 수행시간에서 통신시간이 차지하는 시간은 증가하며 비율로 계산하면 통신시간의 증가는 매우 크다는 것을 알 수 있다.

그림 9는 표 4의 시간을 나타낸 그래프이며 슬라이브 프로세서가 증가함에 따라 알고리즘이 수행되는 시간이 감소하며 이와 반대로 통신시간이 증가하는 것을 보여준다.

표 4와 그림 9를 속도 향상의 그래프로 나타내면 그림 10과 같이 나타낼 수 있다. 그림 10은 1개 마스터 1개 슬라이브모델을 기본으로 하여 이상적인 속도향상과 실제 시뮬레이션을 통한 각 모델의 속도향상을 비교한 그래프이다. 1개 마스터 1개 슬라이브 모델을 기본모델로 하여 이상적인 속도 향상의 그래프와 실제 모델의 속도 향상을 비교하면 슬라이브 프로세서를 2개 이용할 때 까지는 이상적

인 속도 향상의 그래프와 유사한 속도향상을 보이다가 슬래이브 프로세서가 4, 8개로 증가할수록 이상적인 속도향상의 그래프에 비해 속도 개선의 효과가 감소하는 모습을 보인다.

일반적으로 분산 컴퓨팅의 경우 속도향상은 슬래이브 프로세서의 수가 증가함에 따라 통신량의 증가에 따른 통신 시간이 증가하므로 PC클러스터를 이용한 분산 시스템의 경우 속도향상은 이상적인 속도 향상에 비하여 감소한다. 그림 10에서 보는 바와 같이 전체 알고리즘의 수행시간을 이용하여 속도 향상을 비교한 그래프와 전체 알고리즘에서 통신시간을 제외한 알고리즘 수행시간을 이용한 속도 향상의 그래프는 서로 다른 속도 향상을 보여준다. 슬래이브 프로세서가 증가할수록 알고리즘 전체의 수행시간을 기준으로 한 속도 향상과 통신 시간을 제외한 알고리즘 수행시간을 기준으로 한 속도 향상의 차이는 점차 증가한다. 즉 슬래이브 프로세서가 증가할수록 증가하는 통신량에 의한 통신시간이 차지하는 시간의 증가가 그 원인이다. 슬래이브 프로세서가 증가함에 따라 증가하는 통신량의 관계는 표 4와 그림 9에 잘 나타나 있다.

그러나 통신시간의 증가를 고려하여도 속도 향상의 감소는 매우 큰 형태로 나타난다. 속도향상의 손실의 또 다른 부분은 각 프로세서에서 알고리즘을 수행할 때 이주연산의 신호를 받게 되면 수행하던 알고리즘의 진화를 중단하고 이주연산을 수행하기 위한 전·후 단계를 수행하는 과정에서 발생하는 시간손실이 슬래이브 프로세서가 증가함에 따라 증가하기 때문이다. 또한, 각각의 PC 클러스터를 연결하는 네트워크 기기의 상태 또한 속도향상의 손실에 큰 부분을 차지한다. 일반적으로 네트워크 카드의 경우 이주연산을 시작하라는 명령을 받고 실제로 데이터를 전송하기까지 지연시간이 걸리기 때문이다.

PC클러스터와 같이 프로세서가 각각 다른 기판에 존재하는 경우 발생하는 통신시간의 손실은 각 프로세서가 하나의 기판에 존재하는 슈퍼컴퓨터의 경우와 비교하여 그 손실은 매우 크다. 랜카드를 이용한 통신의 경우 이론상 100 Mbit/s 가 최고의 속도 이지만 프로세서가 같은 메인보드에 장착된 슈퍼컴퓨터의 프로세서간의 통신 속도는 1 Gbit/s 이므로 매우 큰 차이를 보인다. 하지만 PC를 이용한 클러스터는 경제적으로 사용비용을 감소시키면서 연산의 효율성을 증가시킬 수 있으므로 그 사용량이 점차 증가하며 본 논문에서도 실용적인 알고리즘 개발을 위하여 PC 클러스터를 사용하였다.

5.2 21층 가새골조 구조물

그림 11의 예제는 21층 정형 가새골조 구조물이며 88 노드를 가지며 189개의 부재로 구성되어 있으며 189개의 부재는 예제 1과 동일한 형태로 3개 층 단위로 그룹핑 하

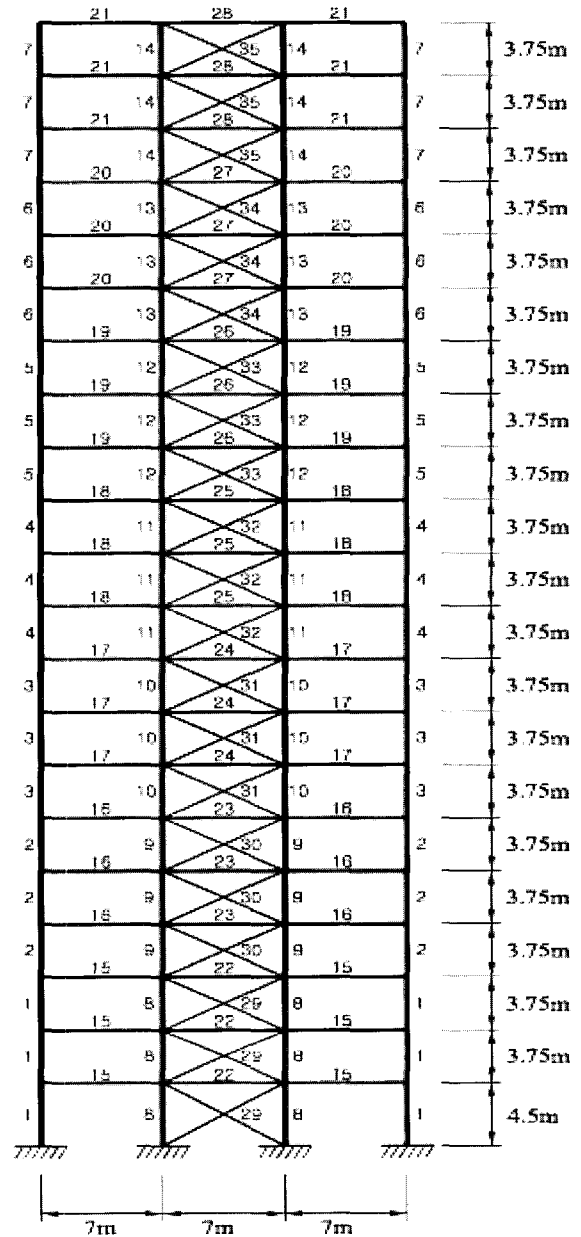


그림 11 21층 가새골조

여 35개의 그룹으로 구분된다. 21층 예제는 12층 예제와 유사한 형태를 가지고 있다. 1층의 높이는 4.5m이며 상부층은 3.75m로 첫 번째 예제와 동일한 형태로 총 21층으로 구성된다. 세 개의 경간은 7m로 동일하다. 하중조건과 사용된 부재의 물성치는 12층 예제와 동일하며 사용한 단면 또한 12층 예제와 동일한 표 3에 나타나는 단면을 사용하였다.

그림 12는 각 모델별 수렴곡선 및 내부루프의 횡수를 나타내는 그래프이다. 그림 10의 각 그래프는 12층 예제 그림 5와같이 꺾은선 그래프는 목적함수를 나타내며 아랫부분의 막대그래프는 내부루프의 횡수를 나타낸다. 시뮬레이션한 모델은 12층 예제와 동일한 네 가지의 모델을

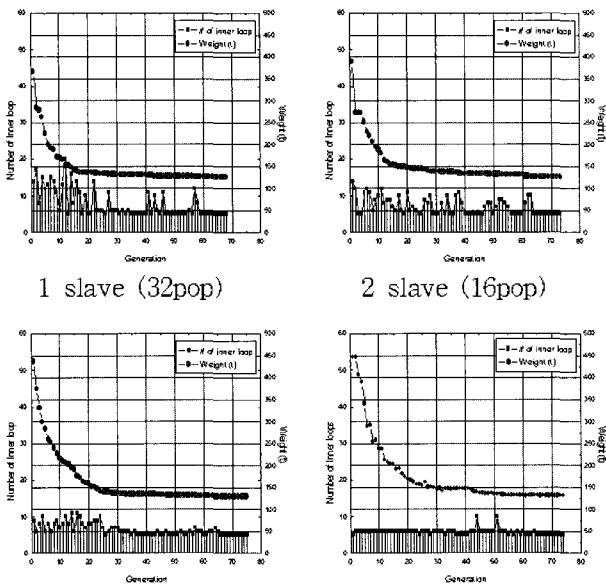


그림 12 각 모델별 수렴곡선 및 내부루프의 횟수

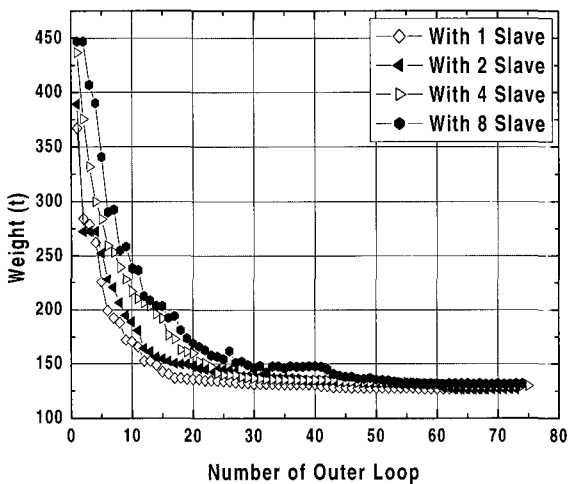


그림 13 각 모델별 수렴곡선

사용하여 1개 마스터 1개 슬레이브를 기준으로 슬레이브 프로세서의 개수를 2, 4, 8개로 증가시켜 가면서 구성하였다. 그림 10에서 볼 수 있듯이 각 모델별 목적함수의 수렴양상은 종료 시까지 외부루프의 최대 반복횟수 70회를 기준으로 하여 약간의 변동을 보이며 안정된 수렴양상을 보이며 종료함을 알 수 있다.

그림 12의 각 모델별 그래프의 내부루프의 횟수를 보면 12층 예제와 유사하게 내부루프의 최대횟수가 나타나는 양상이 서로 다를 수 있다.

즉 가장 많은 내부루프가 나타나는 세대가 슬레이브 프로세서의 수가 증가하고, 마스터 프로세서가 운용하는 개체군이 적을수록 늦어짐을 알 수 있다. 5.1장에서 설명한 바와 같이 내부루프의 횟수는 진화의 속도와 관련된다. 내부루프의 최대 횟수가 초기에 나타나는 모델의 경우와

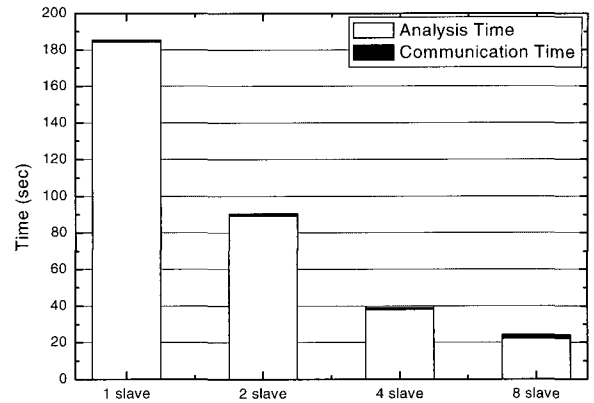


그림 14 알고리즘 수행시간의 분석

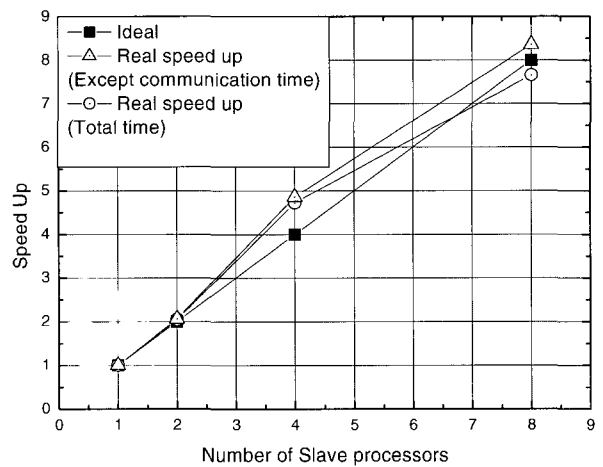


그림 15 Speed Up 그래프

표 5 최적해 및 알고리즘 수행시간

슬레이브 수		1	2	4	8
중량 (t)		129.54	129.54	133.49	131.91
시간 (sec)	전체	185.13	90.42	39.17	24.19
	통신	0.85 (0.46%)	1.14 (1.26%)	1.25 (3.19%)	2.14 (8.85%)

그렇지 않은 모델의 초기 진화 속도차이를 12층 예제의 그림 8에서와 같이 21층 예제 또한 그림 13에서 확인할 수 있다. 초기 수렴 속도의 차이는 있으나 12층 예제와 같이 진화의 중간부분에서 반복횟수 45회를 전후하여 그 수렴 속도는 동일한 양상을 나타낸다.

12층 예제와 21층 예제의 경우를 보아 초기에 수렴 속도가 빠른 이유는 마스터 프로세서의 개체군의 크기와 관련이 되어 있으며, 진화의 도중에 진화의 속도가 같아지는 현상의 진화가 진행될수록 즉, 최적값으로 수렴해 갈수록 슬레이브 프로세서의 역할이 증가 된다는 것을 의미한다.

표 5는 12층 정형 가새골조 구조물의 경우와 동일한 형태로 21층 정형 가새골조 구조물의 각 모델별 시뮬레이션 한 결과를 나타낸 표이다. 각 수치는 연속된 5회의 시뮬레

이선 결과값을 평균한 값이다. 각 모델별 평균 최소 중량은 3% 미만의 오차를 보이며 안정된 수렴 양상을 보여 주고 있다. 표 5에서 각 알고리즘을 5회 연속 시뮬레이션 하여 나타난 평균 수행시간을 보면 12층 예제와 동일한 양상을 보이며 슬레이브 프로세서가 증가함에 따라 모델의 알고리즘 수행 시간은 감소하며 반대로 통신시간은 증가하는 것을 알 수 있다. 하지만 알고리즘의 전체 수행시간에서 통신 시간이 차지하는 비중은 12층 예제와 비교하여 현저히 감소함을 알 수 있다. 알고리즘 수행시간을 분석한 내용을 그래프로 나타내면 그림 14와 같다. 그림 14는 21층 예제의 알고리즘 수행시간을 통신시간과 해석시간으로 구분하여 나타낸 그림이다. 12층 예제와 비교하여 전체 알고리즘수행시간에서 통신시간이 차지하는 비중이 감소함을 알 수 있다.

그림 15는 표 5와 그림 14에 나타는 알고리즘 시간을 속도 향상의 관점에서 표현한 그래프이다. 12층 예제와 동일하게 1개 마스터 1개 슬레이브 모델을 기본으로 하여 이상적인 속도향상과 실제 시뮬레이션을 통한 각 모델의 속도향상을 비교한 그래프이다. 그림 15는 12층 예제의 속도 향상 그래프 그림 10과 비교하여 속도 개선의 효과가 이상적인 그래프 이상으로 나타남을 알 수 있다. 5.1절에서 전술한바와 같이 속도 향상을 저해하는 요소는 슬레이브 프로세서가 증가함에 따르는 통신시간의 증가와, 각 통신기들이 통신을 시작하는데 소요되는 시간이 큰 비중을 차지한다. 그러나 21층 예제의 경우는 전체 알고리즘에서 해석이 차지하는 비중이 12층 예제에 비하여 크게 증가되었다. 디자인 변수의 개수를 기준으로 생각해 보면 20개의 디자인 변수를 사용하는 12층 예제의 1.75배인 35개의 디자인 변수를 운용함으로써 해석 시간의 증가가 속도향상에 직접적인 영향을 미친다. 또한 디자인 영역은 디자인 변수의 단면 개수의 거듭제곱 형태로 표현되므로 그 영역의 확장은 기하급수적으로 증가하게 된다. 이와 같은 이유로 21층 예제의 경우는 속도 향상이 크게 증가함을 알 수 있다. 일반적으로 분산 컴퓨팅의 경우 통신시간의 비중이 감소할수록 즉 통신시간이외에 해석시간이 증가할수록 알고리즘의 속도 향상의 폭은 증가하는 경향을 보인다. 그림 15에서도 그림 10에서와 마찬가지로 통신시간을 제외한 알고리즘 수행시간 만의 속도개선효과가 뛰어난 것을 볼 수 있다.

6. 결 론

본 연구에서 구조 최적화를 위해 개발한 DHGAs를 12층 및 21층 평면 가새골조에 적용한 결과 복합 알고리즘을 구성함으로써 인하여 파라미터 설정의 문제를 간소화 할 수 있었으며 분산 시스템을 도입함에 따라 알고리즘의 수행시간을 감소시킬 수 있었다. 각 결론을 분류하면 다음과 같은 결론을 얻을 수 있었다.

- 1) 적은 수의 개체를 운용함에도 안정적인 최적해를 얻을 수 있는 현상으로 보아 각 마스터 프로세서와 슬레이브 프로세서가 전역 및 지역 탐색을 효율적으로 분화하여 탐색하고 있다는 것을 알 수 있다.
- 2) 별도의 파라미터 설정 없이 De Jong이 제시하였던 일반적인 파라미터를 이용하여 모든 문제에 적용할 수 있음을 확인 할 수 있었다. 그러므로 모든 문제에 동일하게 적용이 가능한 알고리즘을 개발하였다.
- 3) 슬레이브 프로세서가 증가함에 따라 최적해의 변화 없이 선형적인 알고리즘의 수행시간 감소를 통해 효율적인 병렬화가 이루어 졌음을 알 수 있다.
- 4) 슈퍼컴퓨터 및 워크스테이션 클러스터가 아닌 PC 클러스터를 이용하여 상대적으로 저렴한 분산 시스템을 구축하여 적용하였다.

감사의 글

본 연구는 건설교통부의 산·학·연 공동연구개발사업 (과제번호: C102A1000004-02A0200-00410)의 일부로 수행되었으며, 이에 감사드립니다.

참 고 문 헌

1. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989
2. Park, H. S. and Sung, C. W., "Optimization of steel structures using distributed simulated annealing algorithm on a cluster of personal computers", *Computers & Structures*, Vol.80, 2002, pp.1305~1316
3. D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques*, Springer, 2000
4. Kamal C. Sarma, "Bilevel Parallel Genetic Algorithms for Optimization of Large Steel Structural", *Computer-Aided Civil and Infrastructure Engineering*, Vol.16, No.5, 2001, pp.295~304.
5. K. Krishnakumar, "Micro-genetic Algorithms for Stationary and Non-stationary Function Optimization", SPIE, *Intelligent Control and Adaptive Systems*, Vol.1196, 1989, pp.282~296
6. De Jong, K. A., *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral Dissertation, The University of Michigan, Ann Arbor, Michigan, 1975
7. Woo, B. H. and Park, H. S., "Distributed hy-

- brid genetic algorithms for structural optimization", *Preceding of CJK-OSM 2*, 2002, pp.117~122
8. 우병현, 박효선, "구조최적화를 위한 마이크로 유전알고리즘 기반 분산 복합 유전알고리즘", 대한건축학회 추계학술발표 논문집, 2002, pp.15~18
 9. Belegundu, A. D. and Chandrupatla, T. R., *Optimization Concepts and Applications in Engineering*, Prentice Hall, 1999
 10. Zhu, D. M., "An Improved Templeman's Algorithm for Optimum Design of Trusses with Discrete Member Sizes", *Engineering Optimization*, Vol.9, 1986, pp.303~312
 11. Rajeev, S. and Krishnamoorthy, C. S., "Discrete Optimization of Structures Using Genetic Algorithms", *Journal of Structural Engineering*, ASCE, Vol.118, No.5, 1992, pp.1233~1250
 12. Adeli, H. and Park, H. S., *Neurocomputing for Design Automation*, CRC Press, 1998