

구조최적화를 위한 분산 복합 유전알고리즘 Distributed Hybrid Genetic Algorithms for Structural Optimization

우 병 헌¹⁾ 박 효 선²⁾
Woo, Byung Hun Park, Hyo Seon

ABSTRACT

The great advantages on the Genetic Algorithms(GAs) are ease of implementation, and robustness in solving a wide variety of problems, several GAs based optimization models for solving complex structural problems were proposed. However, there are two major disadvantages in GAs. The first disadvantage, implementation of GAs-based optimization is computationally too expensive for practical use in the field of structural optimization, particularly for large-scale problems. The second problem is too difficult to find proper parameter for particular problem. Therefore, in this paper, a Distributed Hybrid Genetic Algorithms(DHGAs) is developed for structural optimization on a cluster of personal computers. The algorithm is applied to the minimum weight design of steel structures.

1. 서론

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 기법인 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 조합 최적화 기법은 최적 값을 찾는 동안 컴퓨터의 기억 용량과 계산시간의 한계로 인한 문제점 때문에 실용적 사용의 한계를 가지고 있다. 특히 유전알고리즘은 해집단을 운용하기 때문에 다른 최적화 알고리즘에 비하여 많은 반복수와 적응도 평가를 위하여 해집단의 수만큼 구조해석을 필요로 한다.⁴⁾ 유전알고리즘은 앞서 언급한 해석시간의 증가에 따른 전체 알고리즘의 수행 시간이 증가하는 문제와 각 문제에 따라 각기 다른 파라미터 설정이 필요하다는 두 가지 큰 단점을 가지고 있다. 적합한 파라미터 세팅은 안정적이고 신속한 수렴을 보장하므로 문제에 따른 적합한 파라미터 설정을 필수적인 요소이다.

본 연구에서는 유전알고리즘의 수행시간이 증가하는 문제와 또한 모든 문제에 일반적으로 적용할 수 있는 파라미터를 사용하는 Distributed Hybrid Genetic Algorithms(DHGAs)을 개발하였다. 새로이 개발된 알고리즘은 철골 가새골조 구조물의 최소중량설계에 적용하여 그 성능을 평가한다.

1) 연세대학교, 건축공학과, 석사과정

2) 연세대학교, 건축·도시공학부, 조교수

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

$$g_1 = \frac{\Delta^T}{\Delta^T_{allow}} \leq 1 \quad (2)$$

$$g_2 = \frac{\Delta^i - \Delta^{i-1}}{\Delta^i_{allow}} \leq 1 \quad i=1, N \quad (3)$$

$$h = \frac{\sigma^j}{\sigma^j_{allow}} \leq 1 \quad j=1, M \quad (4)$$

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

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

여기서 Δ^{Top} , Δ^{Top}_{allow} , Δ^i , Δ^i_{allow} 는 최상층 및 i 번째 층의 최대 수평변위 및 허용 수평변위 이다. σ^j , σ^j_{allow} 는 i 번째 부재에 발생하는 응력과 허용응력을 나타낸다. A_c^i , A_r^i 는 i 번째의 기둥과 가새의 단면적의 넓이를 나타낸다. M 은 전체 부재의 개수이며 N 은 전체 층높이 이다.

3. 유전알고리즘

3.1 유전알고리즘(GAs)

유전알고리즘은 다윈이 주장한 진화 법칙인 적자생존과 자연도태의 원리를 토대로 개발된 알고리즘이다. 유전알고리즘은 2진수로 코딩된 유전자를 사용하며 최적 해를 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 부동과 안정적으로 최적 해를 탐색하기 위하여 30개 이상의 개체를 사용하여야¹⁾ 하며 각 문제에 따른 적절한 파라미터설정이 필요하다. 파라미터 설정의 특이성으로 인하여 유전알고리즘은 모든 문제에 일반적으로 적용하기 쉽지 않다는 단점이 있다.

3.2 마이크로 유전알고리즘(μ -GAs)

개체군이 증가함에 따라 유전알고리즘은 안정적이며 신속한 최적해로의 수렴양상을 보이는 경향이 있다.¹⁷⁾ 그러나 개체군의 증가는 안정적인 수렴을 보장하지만 알고리즘의 수행시간의 증가라는 양면성을 가지고 있다. 이와 같이 개체수 증가에 따라 해석 시간이 증가하는 문제를 해결하기 위하여 Goldberg는 직렬 유전알고리즘(Serial GA) 안에서 작은 집단을 이용하여 효율적인 탐색을 할 수 있는 방법을 제시하였다.¹⁷⁾ 이 접근방법에 기초하여 Krishnakumar는 매우 작은 집단을 운용하는 마이크로 유전알고리즘(μ -GA)을 제안하였다.⁵⁾ 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. 하지만 마이크로 유전알고리즘은 디자인 변수가 증가하며 디자인 영역이 넓어지는 경우 안정적인 수렴에 한계를 보인다.

4. 병렬 유전알고리즘

4.1 병렬 시스템

그림 1은 본 논문에서 사용하는 분산시스템의 구성도이다. 본 논문에서 사용한 분산 시스템은 그림 1과 같이 각 PC를 Ethernet 방식의 랜으로 연결하여 구성하였다. 프로세서부분에 해당하는 각 PC는 1.6GHz의 펜티엄4 프로세서와 256MB의 SDRAM 을 탑재하고 있으며 10GB의 하드디스크 용량을 가지고 있다. 각 PC의 운영체제는 윈도우98을 사용한다. 병렬시스템의 네트워크 구성은 100Mbit/s의 속도를 낼 수 있는 Faster Ethernet 방식을 사용하였으며 각각의 Slave Processor들은 스위칭 허브를 통해 Master Processor 및 전체 네트워크에 연결되어 있다. 분산알고리즘을 수행하는 프로그램은 WMPI 1.3 Library를 사용하여 구성되어 있다.

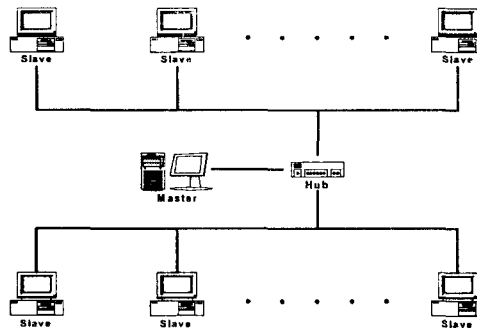


그림 1. 병렬 시스템

4.2 복합 병렬 유전알고리즘(Distributed Hybrid Genetic Algorithms, DHGAs)

본 논문에서 개발한 DHGAs의 구성은 기존은 SGAs¹⁾와 μ -GAs⁵⁾를 병렬로 구성하였다. 두 종류의 이질 유전알고리즘이 상호 병렬로 구성되어 서로의 단점을 보완하며 진화하게 된다. 그림 2는 DHGAs의 상세한 순서도 이다.

4.2.1 Master 프로세서에서의 마이크로 유전알고리즘

Krishnakumar가 제안한 마이크로 유전알고리즘의 특징은 3.2절에서 서술한 바와 같다. Krishnakumar가 제안한 마이크로 유전알고리즘에서는 재시동 연산을 수행할 때 엘리트 개체를 제외한 전체 개체를 임의로 새롭게 생성하여 새로운 개체군을 형성한다. 본 논문에서 개발한 HPGAs에서는 재시동 연산시에 엘리트 개체를 제외하고 새롭게 생성되는 개체를 임의로 생성하는 것이 아니라 각각의 Slave 프로세서에서의 엘리트 개체를 이용하여 새로운 개체군을 형성한다. 이는 유도된 초기화 기법에 해당하며 이에 따라 계산시간 및 수렴속도를 증가시킬 수 있다는 장점이 있다.⁷⁾

4.2.2 Slave 프로세서에서의 단순 유전알고리즘

Slave 프로세서에서는 단순유전알고리즘이 사용되며 각 유전연산에 있어서의 파라미터는 De Jong이 제시한 기본 파라미터 설정이 사용된다.⁶⁾ 각 파라미터들은 문제에 따라 변화하는 것이 아니라 고정된 파라미터가 사용된다. 개체군 30개, 교배율 0.6, 돌연변이율 0.03이 사용된다. 모든 Slave 프로세서의 동작은 Master 프로세서에서 제어되므로 별도의 종료조건을 두지 않았다.

4.3 복합병렬 유전알고리즘(HPGAs)

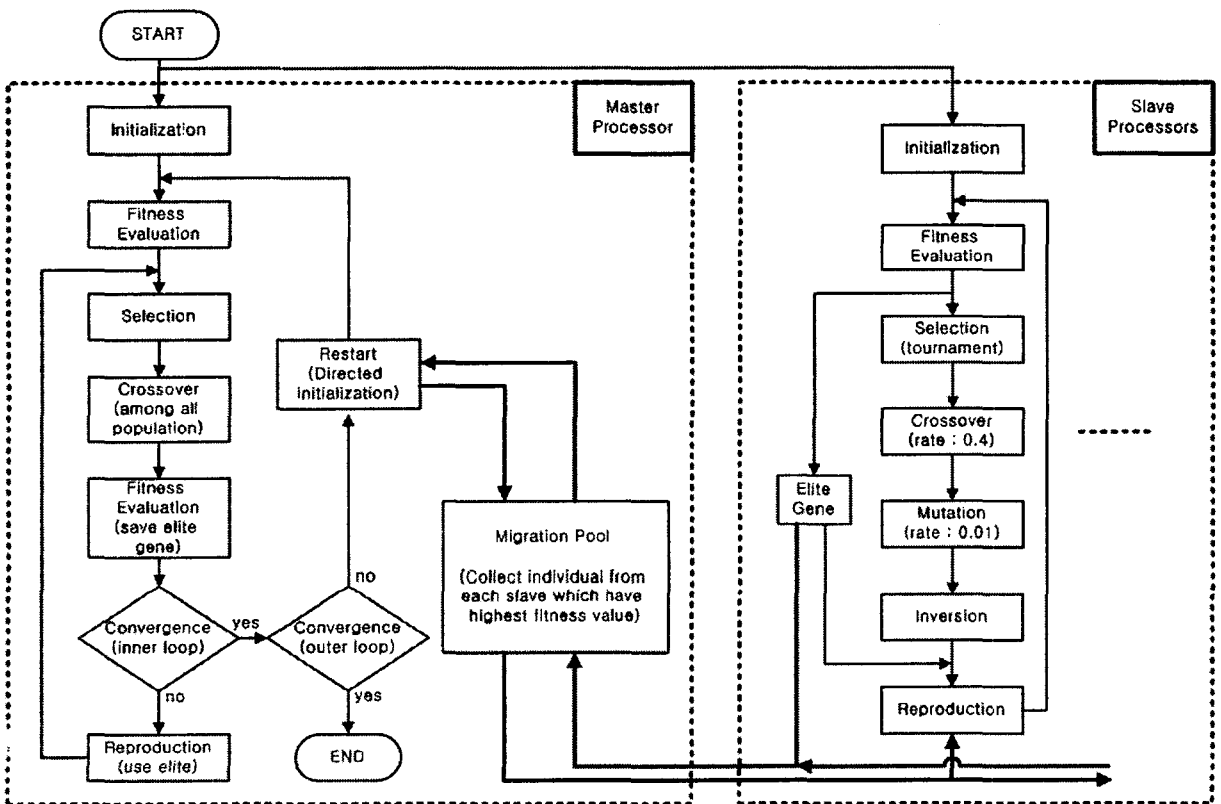


그림 2. Distributed Hybrid Genetic Algorithms(DHGAs)

그림 2는 본 논문에서 개발한 HPGAs의 순서도 이다. 본 논문에서 Slave 프로세서를 8개까지 증가시키며 시물레이션 하였다. Master 프로세서와 Slave 프로세서에서는 기존의 마이크로 유전알고리즘과 단순유전 알고리즘이 각각 운용된다. 하지만 병렬로 결합한 전체 알고리즘은 Master 프로세서에서 운용되는 마이크로 유전알고리즘에 의해서 제어되므로 마이크로 유전알고리즘이 가지는 소수의 개체를 운용하며 파라미터 설정이 간소하다는 장점들을 그대로 활용할 수 있다. 또한 병렬 유전알고리즘에서는 이주 빈도와 이주율이라는 두 가지의 새로운 파라미터가 필요하며 이는 병렬 유전알고리즘의 성능에 큰 영향을 미치는 중요한 파라미터가 된다. 하지만 본 논문에서 개발한 HPGAs에서는 이 두 가지의 파라미터는 Master 프로세서가 재시동시 이주하며 각 Slave 프로세서의 엘리트 개체가 이주에 참여 하므로 이주 시기와 횟수 그리고 이주율에 관한 문제는 해결된다.

5. 예제적용

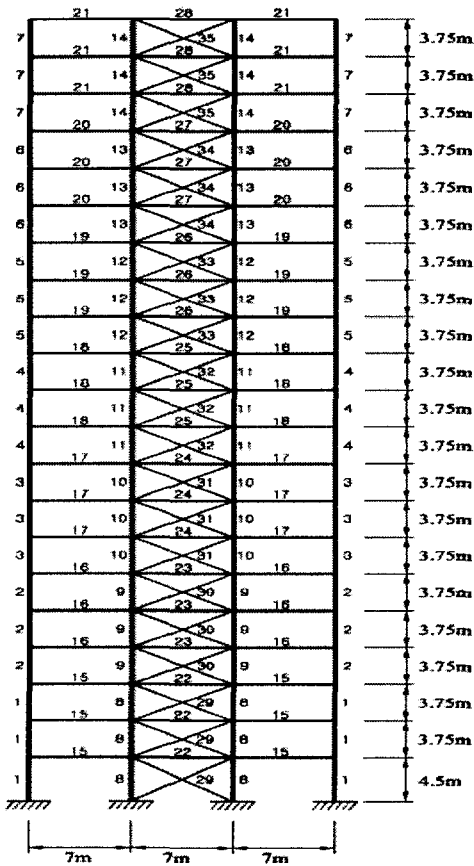


그림 3. 21층 가세골조

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

No	단면	단면적 (cm ²)	단면2차모멘트 (cm ⁴)	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

HPGAs의 효율성을 평가하기 위하여 21층 가세골조 구조물의 최소 중량 설계에 적용하여 그 효율성을 평가 하였다. Master 프로세서 한 개와 Slave 프로세서 한 개로 구성되는 모델을 기본으로 하여 Slave 프로세

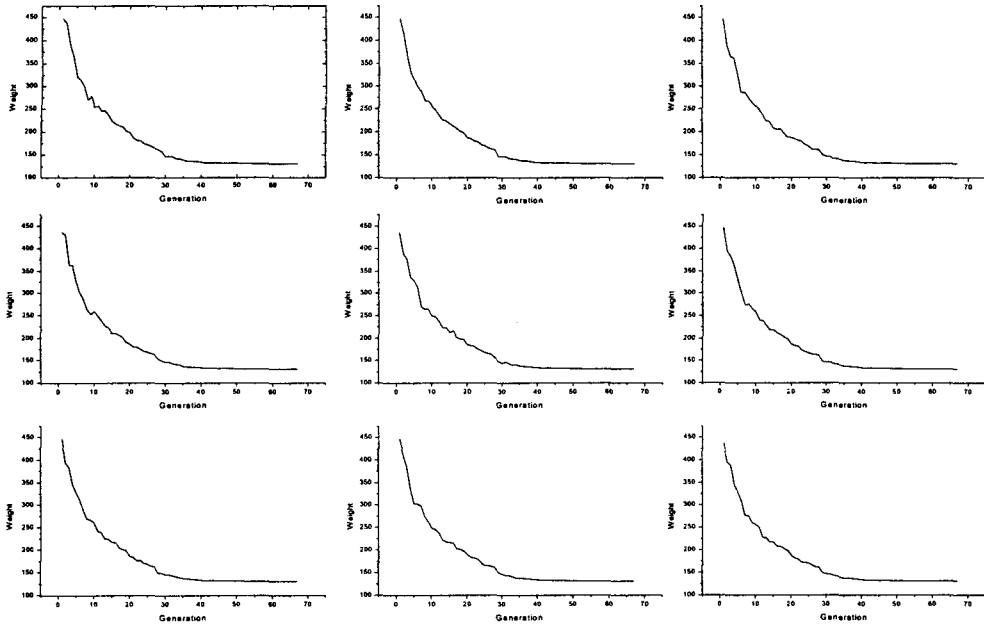


그림 4. Slave 프로세서가 8개 일때의 각 프로세서에서의 수렴곡선

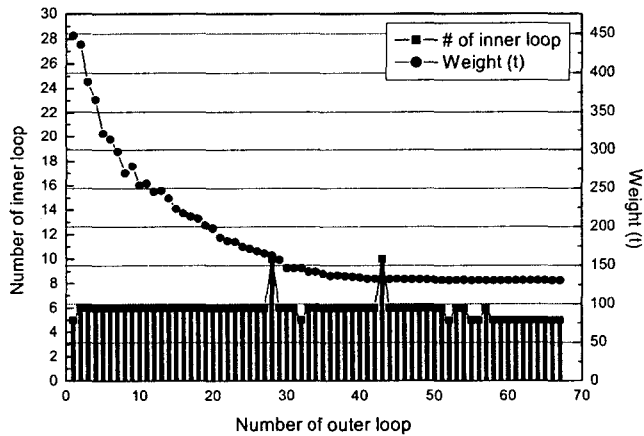


그림 5. Master 프로세서에서의 수렴곡선 및 내부 루프의 횟수

서를 2, 4, 8개로 증가시키며 그 성능을 평가하였다. 그림 3의 예제는 88노드를 가지며 189개의 부재로 구성되어 있으며 189개의 부재는 3개 층 단위로 그룹핑 하여 35개의 그룹으로 구분된다. 하중조건으로는 수직하중으로 고정하중과 활하중을 각각 3.29 kN/m^2 , 1.26 kN/m^2 로 작용시키고 횡력은 KS규준에 의거 풍력을 산정하여 각 층 당 절점에 작용시켰다. 하중조합은 횡력과 자중 고정하중 활하중이 동시에 작용할 때 최대 층변위와 층간변위에 대한 허용한계를 구조물의 층높이의 1/400로 제한하였으며 대한 건축학회 강구조 계산

기준(허용 응력 설계법)에 의하여 부재에 대한 강도를 평가하였다. 사용강재의 탄성계수는 $2.0 \times 10^8 \text{ kN/m}^2$ 단위중량은 77 kN/m^3 로 하였다. 각 부재가 선택할 수 있는 단면은 기둥과 가새에 광폭(wide shape) 계열을 보는 세폭(narrow shape) 계열을 각각 16개씩 사용하였다. 그림 3과 같이 1층은 4.5m, 2층 이상은 3.75m의 높이를 갖으며 스패ンは 7m로 동일하다. 1개의 Master 프로세서와 1개의 Slave 프로세서로 구성된 기본 모델은 32개의 개체군을 사용하며 Slave 프로세서의 개수가 증가함에 따라 개체군의 수는 초기 32개를 Slave 프로세서의 개수로 나눈 수의 개체가 사용된다.

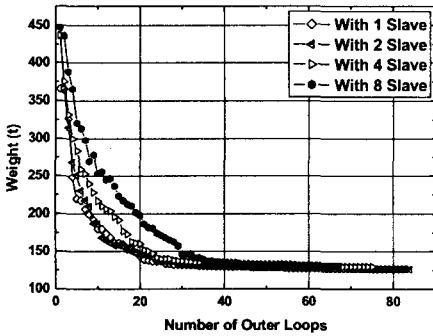


그림 6. Master 프로세서의 수렴곡선

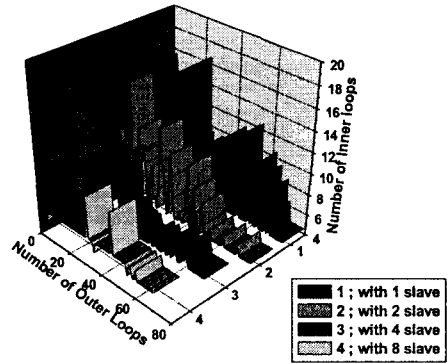


그림 7. 내부루프의 횟수

그림 4는 Slave 프로세서가 8개일 경우의 각 프로세서의 수렴곡선을 나타낸 그래프이다. 그림 4의 첫 번째 그래프는 그림 5의 격은선 그래프와 동일 한 그래프이다. 그림 5는 각 외부루프 종료시까지의 내부루프의 횟수와 전체 수렴을 나타낸 그래프이다. 그림 4에서 나타난 것처럼 각 Slave 프로세서에서의 수렴 양상이 초·중반 서로 다른 양상을 보이며 감소하나 중반 이후에는 서로 수렴 양상이 비슷하다. 이는 초기에는 서로 다른 형태의 진화과정을 거쳐 수렴에 이른다라는 의미로 해석 할 수 있으며 이는 Slave 프로세서의 수가 증가함에 따라 다양한 탐색을 하고 있다는 것을 의미한다. 그림 6은 Slave 프로세서의 개수의 변화에 따른 각 모델의 수렴양상을 나타낸 그래프이다. 그림 6에서 Slave 프로세서의 수가 감소 할 수록 초기 수렴속도가 증가하는 양상을 보인다. 그림 7은 각 세대별 내부루프의 횟수를 나타낸 그래프이다. Slave 프로세서가 적을수록 종국에도 진화가 계속 발생하므로 그림 6과 같은 진화 양상이 나타남을 알 수 있다. 그림 6 및 그림 7은 Slave 프로세서수가 작고 많은 개체군이 사용될 수록 유전알고리즘의 랜덤한 경향이 감소하여 안정적인 수렴 양상을 보인다는 것을 의미한다. 하지만 이러한 초기 단계의 진화에 있어서는 안정된 경향을 보이나, 종국에는 수렴속도가 오히려 감소하는 경향을 보인다. Slave 프로세서 수가 증가하고 적은 개체군을 운용하는 모델의 경우 초기에는 랜덤한 경향성이 강하여 수렴속도가 저하되나 종국에서는 진화 초기의 다양한 탐색을 바탕으로 보다 안정적인 수렴 경향을 보인다. 따라서 Slave 프로세서 개수가 증가함에 따라 최적해를 찾는 동안의 최종적인 수렴 횟수는 감소하는 양상을 보여주고 있다. 표 2는 연속한 5회의 실험 결과를 평균하여 나타낸 표이다. 최소중량은 3% 미만의 오차를 보이고, 알고리즘의 수행시간은 Slave 프로세서수가 증가함에 따라 선형적으로 감소하는 양상을 보이고 있다. 그림 7은 Slave 프로세서 수가 증가 하는데 따른 속도향상을 나타낸 그래프이다. Slave 프로세서 수가 증가함에 따라 선형적인 속도향상의 효과를 보이는 것은 알고리즘이 효율적으로 병렬화 되었다는 것을 의미한다.

표 2. 최적값 및 수행시간

Model	1 Slave	2 Slave	4 Slave	8 Slave
Weight(t)	127.31	126.38	129.89	130.67
Time(sec)	180.78	85.56	46.99	23.43

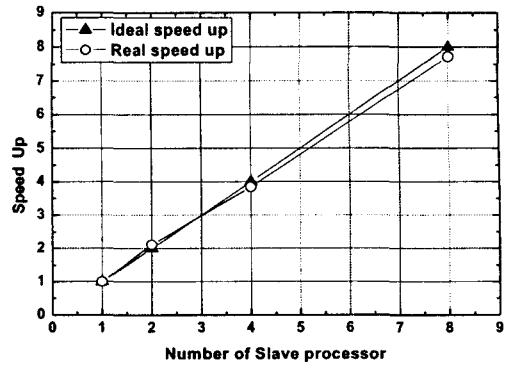


그림 7. Speed Up

6. 결론

본 연구에서 구조 최적화를 위해 개발한 HPGAs를 21층 평면 가새골조에 적용한 결과 다음과 같은 결론을 얻었다.

- 1) 적은 수의 개체를 운용함에도 안정적인 최적해를 얻을 수 있는 현상으로 보아 각 Master 프로세서와 Slave 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 확인할 수 있었다.
- 3) Slave 프로세서가 증가함에 따라 최적해의 변화 없이 선형적인 알고리즘의 수행시간 감소를 통해 효율적인 병렬화가 이루어 졌음을 알 수 있다.

참고문헌

1. Goldberg, D. E. (1989), "Genetic Algorithms in search, Optimization and machine learning", Addison-Wesley.
2. Hyo Seon Park, Chan Won Sung (2002) "Optimization of steel structures using distributed simulated annealing algorithm on a cluster of personal computers", Pergamon, Computers & Structures, 80, pp1305-1316.
3. D.T.Pham and D.Karaboga (2000) "Intelligent Optimisation Techniques", Springer
4. Kamal C. Sarma (2001), "Bilevel Parallel Genetic Algorithms for Optimization of Large Steel Structural", Computer-Aided Civil and Infrastructure Engineering, Vol. 16 NO. 5, pp. 295-304
5. K. Krishnakumar (1989), "Micro-genetic Algorithms for Stationary and Non-stationary Function Optimization", SPIE, Intelligent Control and Adaptive Systems, Vol.1196, pp.282-296.
6. De Jong, K. A. (1975) "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Doctoral Dissertation, The University of Michigan, Ann Arbor, Michigan.
7. 진장규,(2002), "유전알고리즘과 그 응용", 교우사.