

유전자 알고리즘을 사용한 분산 처리에 의한 다목적 위성 구조체의 최적화

윤진환, 임종빈, 박정선
한국항공대학교 항공우주 및 기계공학부
경기도 고양시 화전동 한국항공대학교

Optimization of Komsat II Structure Using Genetic Algorithm in Parallel Computation Environment

Jin-Hwan Yoon, Jong-bin Im, and Jung-Sun Park
E-mail : jungsun@mail.hangkong.ac.kr
Tel : (02) 300-0283 FAX : (02) 3158-3189

요 약

컴퓨터 네트워킹 기술의 발달에 힘입어 분산처리를 이용한 기법이 복잡한 구조물의 최적설계에 널리 사용되고 있다. 최적설계시 구조물이 복잡하고 설계 변수가 많아질수록 설계 변수간의 교호작용이 복잡해지고 국부최적해가 많아지는 특성이 있다. 최근의 최적 설계는 이러한 문제점을 해결하고자 다양한 전역 최적화 기법을 도입하여 적용하고 있다. 본 연구에서는 진화이론을 바탕으로 한 유전자 알고리즘과 실험계획법을 바탕으로 한 반응표면법에 분산처리 기법을 도입하여 인공위성 추진 모듈의 최적화에 적용시켰다. 그 결과 유전자 알고리즘이 조금 더 좋은 최적값을 보였으며 해석시간은 반응표면법을 적용 시켰을 경우가 훨씬 짧았다. 병렬처리 기법을 이용한 위성구조체의 최적설계에 있어 유전자 알고리즘은 해의 전역성에서 반응표면법은 시간의 효율성에서 각각 장점을 보였다.

I. 서 론

최근의 구조최적설계는 그 모델이 점차 복잡해지고 설계변수가 많아지고 있다. 이러한 문제를 단일 프로세서를 이용한 컴퓨터로 해결하기 위해서는 너무나 많은 계산 시간이 요구된다. 따라서 최근에는 최적설계 기법에 발전된 네트워크 기술을 도입한 분산처리를 이용한 최적화가 널리 이루어

지고 있다. 많은 설계변수로 인하여 최적화 과정에서 발생하는 또 다른 문제점은 전역최적해로의 수렴성 여부이다. 도함수를 이용하는 일반적인 최적화 알고리즘으로는 설계 변수가 많아짐에 따라 급격히 증가하는 교호작용에 대한 정보를 파악하여 대처 할 수 없으며 국부최적해로부터 전역최적해로의 이행도 어렵다.

유전자 알고리즘은 생물의 진화론을 기본 개념

으로 하여 개발된 전역 최적화 알고리즘으로써 뛰어난 전역 최적해의 도출가능이라는 장점에도 불구하고 최적화 수행까지 과도한 반복해석을 요구한다는 단점이 있다[1,2].

반응표면법은 준전역성을 가진 최적화 알고리즘으로써 실험계획법에 의한 통계적 경험 설계와 회귀모델 생성기법에 기반을 두어 최적화 수행을 위한 반복해석을 최대한 줄일 수 있는 장점에 비하여 회귀모델 자체의 정확성에 대한 불확실성으로 인하여 정확한 해로의 도출이 어렵다는 문제를 가지고 있다[3].

본 연구에서는 유전자 알고리즘에서 단위 세대 당 반복 수행 횟수를 결정하는 개체수(popsize)의 크기에 분산처리에 사용될 컴퓨터의 대수를 맞추어 두 가지 알고리즘으로 최적화를 수행하였다. 이를 위하여 분산 유전자 알고리즘 프로그램과 분산 반응표면법 프로그램을 작성하였으며 구조해석 코드는 상용프로그램을 사용하였다. 또한 본 프로그램을 인공위성의 추진모듈에 적용하여 최적화 시켜 그 결과를 비교 분석하였다.

II. 분산처리 프로그래밍

2.1 클라이언트-서버 통신모델

컴퓨터 네트워크에서 클라이언트-서버 통신 모델은 네트워크상에서 분산된 컴퓨팅 리소스들로 하여 사용자 그룹 사이에 존재하는 공통적인 리소스를 공유할 수 있도록 하는 소프트웨어 기반 아키텍처이며, 그림. 1과 같이 클라이언트가 서버에게 서비스를 요청하면, 서버는 요청을 처리하고 그 결과를 클라이언트에게 돌려준다.

클라이언트-서버는 하나 또는 여러 개의 관련 있는 응용프로그램을 돌 또는 그 이상의 상호 협력적인 처리 환경에서 운영하는 다양한 형태의 분산 컴퓨팅을 의미한다. 즉, 서비스를 요청하는 클라이언트와 클라이언트의 요청을 처리해주는 서버와의

협동작업을 통해서 우리가 원하는 바람직한 결과를 얻을 수 있다[4,5].

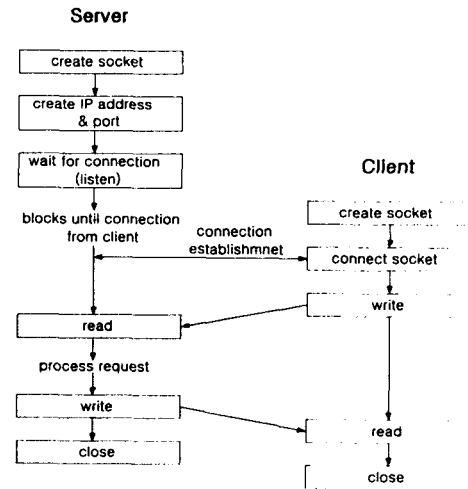


그림 1. Client-server Communication model

2.2 유전자 알고리즘의 분산처리

유전자 알고리즘은 자연계의 적자생존의 원리를 이용한 확률적인 최적화 방법이다. 최적화하고자 하는 문제의 잠재 해를 표현한 개체들의 모집단을 가지고 탐색을 시작하며 재생산(reproduction), 교배(Crossover), 변이(mutation)등의 유전 조작을 하고 매 세대에서의 적응도(fitness)를 평가하여 다음 세대에 생존할 개체들을 확률적으로 선택한다. 적응도가 좋은 모집단의 개체들은 보다 높은 확률을 갖는 유전자 조작을 하여 다음 세대의 새로운 집단을 생성하여 적응도가 점차 좋아지게 된다.

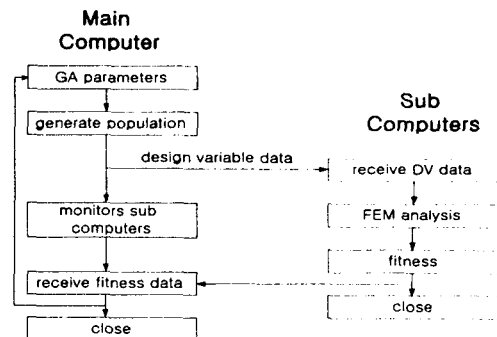


그림 2. Optimization model of genetic algorithm by network

한편, 유전자 알고리즘은 도함수를 이용한 전통적인 방법보다 전역 최적해의 접근이 우수한 장점이 있지만 많은 반복 해석을 요구하므로 개체의 수나 설계 변수가 많아지면 처리시간이 급격히 증가한다. 이를 해결하기 위해서 과거 몇 년 사이에 병렬적 유전자 알고리즘이 개발되고 있으며, 그 종류에는 전역적 병렬처리, 이주모델, 확산 모델 등이 있다[6,7]. 본 연구에서는 한 세대에서의 개체수에 따라 네트워크의 연결 수를 정하는 방법을 사용하였다.

2.3 반응표면법의 분산 처리

반응표면법은 실험이나 시뮬레이션으로부터 얻은 결과 값들을 이용하여 만든 함수로써 최적화를 시키는 방법을 말한다. 이를 위해서는 반응표면생성의 오차와 실험 횟수에 영향을 미치는 통계적인 경험 설계와 회귀모델의 생성 기법 그리고 국부영역에서의 최적점을 찾아내기 위한 일반적인 최적화 기법에 대한 기본 지식이 요구된다.

본 프로그램에서는 회귀모델로써 회귀계수(regression coefficient)들을 구하기 쉽고 큰 유연성을 가지며 이미 많은 문제에서 적용이 된 1차 회귀모델(first order regression model)과 2차회귀모델(second order regression model)을 적용시켰다.

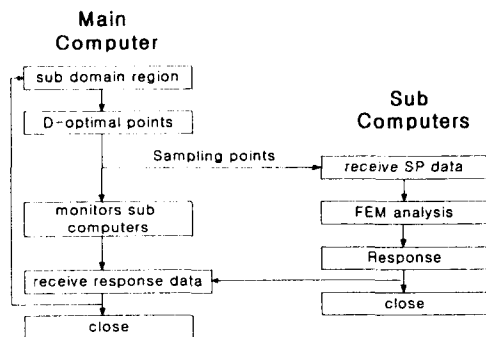


그림 3. Optimization model of response surface method by network

반응표면 생성시에 무수히 많은 추출점을 선택

하는 것이 가장 좋겠으나 실제로는 한정된 개수의 추출점을 뽑을 수밖에 없다. 반응표면을 생성하기 위하여 가장 적절한 추출점을 뽑아 내는 방법을 실험 계획법이라고 하며 본 프로그램에서는 분산과 회귀계수 등의 공분산(covariance)에 대한 정보를 가지고 있는 모멘트 매트릭스(moment matrix)의 결정계수(determinant)의 값을 최대화시키는 D-optimal 방법을 사용하였다[8,9].

반응표면법의 분산처리 네트워크 알고리즘도 유전자 알고리즘과 같으며 단지 최적화 과정 중에서 알고리즘이 필요로 하는 부분의 요청만이 다를 뿐이다. 반응표면법에서는 개체수와 같은 개념이 없어 컴퓨터의 수를 무한히 늘릴 수 있지만 유전자 알고리즘과의 비교를 위해서 개체수와 같도록 컴퓨터의 수를 제한하였다.

III. 분산처리 최적화

추진모듈은 연료탱크의 부착과 추진기를 지지하기 위한 구조물로 연료 탱크를 고정하는 링 구조물과 추진 플랫폼, 이들을 연결하는 스트러트로 구성된다[10].

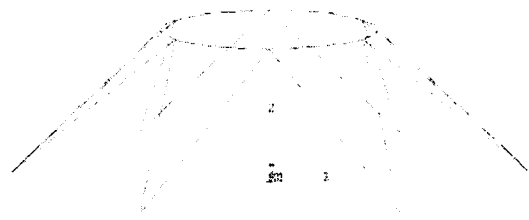


그림 4. Propulsion module

본 연구에서는 추진 모듈에서 추진 플랫폼을 제외한 링 구조물과 스트러트 구조물의 최적화를 네트워크를 사용한 분산처리 기법을 이용하여 유전자 알고리즘과 반응표면법으로 수행하였다. 추진 모듈은 축 방향 10g와 측면 X, Y, 45° 방향으로 3.5g의 하중을 받도록 하였으며 고유진동수를 고려

하여 동해석을 수행하였다.

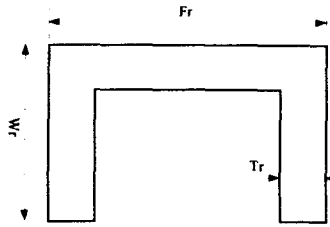


그림 5. Cross section of ring

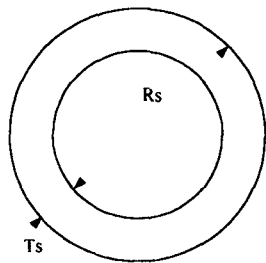


그림 6. Cross section of strut

구속조건은 추진모듈 중에서 스트러트 및 부분의 모든 자유도를 고정하였다. 추진 모듈의 설계변수는 각 구조물들의 치수들이다. 각 구조물의 단면형상에 나타난 설계변수들은 그림. 5, 그림. 6과 같으며 목적 함수 및 제한 조건은 아래와 같다.

$$\text{Minimize: } OBJ = \sum_{i=1}^{NF} W_i \quad (22)$$

Subject to :

$$\frac{\sigma_f}{56,000} - 1 \leq 0 \quad (23)$$

$$1 - \frac{f}{75} \leq 0 \quad (24)$$

$$0.6 < Fr < 1.85 \quad (25)$$

$$0.8 < Wr < 2.42 \quad (26)$$

$$0.05 < Tr < 0.15 \quad (27)$$

$$0.19 < Rs < 0.565 \quad (28)$$

$$0.025 < Ts < 0.074 \quad (29)$$

유전자 알고리즘을 적용하기 위한 초기값들로서 교배확률 0.75, 돌연변이 확률 0.01, 한 세대의 개체수 5를 사용하였고 반응표면법을 사용하기 위한 설정으로서 1차회귀 모델에서는 D-optimal 계획법을 이용하여 추출점 13개 2차 회귀 모델에서

는 추출점 42개를 선택하여 사용하였다.

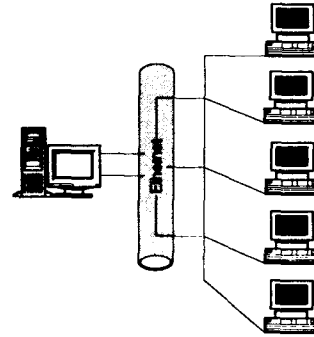


그림 7. Parallel networking

이상의 조건을 가지고 분산 처리에 적용하기 위해서 유전자 알고리즘의 개체수와 같은 5대의 컴퓨터를 사용하였으며 각각의 컴퓨터로 명령을 전달하는 프로그램은 TCP/IP환경을 이용한 비주얼 베이직으로 작성하여 사용하였다.

표 1. Optimization comparison

Components	Initial (in)	Optimum (in)	
		GA	RSM
Object function(lb)	6.610	7.729	7.740
Frequency (Hz)	64.80	75.00	75.00
Fr	1.250	1.475	1.408
Wr	1.620	1.232	1.315
Tr	0.100	0.128	0.127
Rs	0.375	0.491	0.490
Ts	0.049	0.039	0.040
Function calls		2250	391
Calculation time (sec)		30285	5247

우선 목적함수의 결과를 보면 두 가지 알고리즘 모두 본 최적화에서 가장 민감한 제한조건인 고유진동수를 만족시키면서 유전자 알고리즘으로는 7.729(lb) 반응표면법으로는 7.740(lb)로 수렴하였다. 또한 이 과정에서 유전자 알고리즘으로는 2250번의 유한요소 해석을 요청하여 30285초의 수행시간이 걸렸으며 반응표면법은 391번의 유한요소 해

석을 요청하여 5247초의 수행시간이 걸려 약 5.7배의 최적화 시간의 차이를 보였다. 이렇게 시간에서의 차이가 많이 나는 것은 반응표면법 자체가 실험 횟수를 최대한으로 줄일 수 있는 실험계획법을 기반으로 하였을 뿐만 아니라 유전자 알고리즘이 개체 수만큼의 해석을 해야지만 다음 세대로 넘어 갈 수 있는 반면에 반응표면법에서는 추출점에서의 해석을 순서에 상관없이 분산처리 시킬 수 있기 때문이다.

IV. 결 론

유전자 알고리즘과 반응표면법을 이용한 위성 구조물의 분산 최적화에서 두 방법 모두 일정한 최적값에 도달하였다. 목적함수의 값에 있어서는 유전자 알고리즘의 경우가 좀더 좋은 결과값을 도출하였으며 최적화 시간에 있어서는 반응표면법을 이용한 경우가 훨씬 빠른 수렴 성능을 보여 유전자 알고리즘보다 반응표면법의 분산처리를 이용하는 것이 훨씬 경제적임을 알 수 있었다.

두 알고리즘을 분산 처리하였을 경우와 그렇지 않을 경우에 최적값의 차이점은 없었으며 단지 최적화 시간의 단축에만 영향을 주었다. 이러한 시간의 단축은 위성전체구조물 같은 대형 구조물의 최적화에 중요한 역할을 할 수 있을 것이다.

참 고 문 헌

[1] Goldberg, D. E., *Genetic Algorithms in Search, Operation, and Machine Learning*, Addison -Wesley, 1989.
 [2] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer -verlag, 1996.
 [3] Myers, R. and Montgomery, D. C.,

Response Surface Methodology-Process an Product Optimization Using Designed Experiments, John Wiley & Sons., 1995.

[4] Comer, D. E. and Stevens, D. L., *Inernetworking with TCP/IP, Volume I, Principles, Protocols, and Archietectures*, 3rd Edition, Prentice Hall, 1999.

[5] Comer, D. E. and Stevens, D. L., *Inernetworking with TCP/IP, Volume III, Client-Server Programming and Applications, Window Soket Version*, 2nd Edition, Prentice Hall, 1997.

[6] Mühlenbein, M., Schomisch, M. and Born, J., "The Parallel Genetic Algorithms as Function Optimizer, Parallel Computing", Parallel Computing, 17, 1991, pp. 619-632.

[7] Abramson, D. and Abela, J., "A Parallel Genetic Algorithm for Solving the School Timealing Problem", Proceedings of the Fifteenth Australian Computer Science Conference (ACSC-15), 14, 1992, pp. 1-11.

[8] Broudiscou, A., Leardi, R. and Phan-Tan-Luu, R, "Genetic Algorithm as a Tool for Selection of D-optimal Design,", Chemometrics and Intelligent Laboratory Systems 35, 1996, pp. 105-116.

[9] Poland, J., Mitterer, A., Knödler, K. and Zell, A., "Geneitic Algorithm Can Improve the Construction of D-optimal Experimental Designs," Advances In Fuzzy Systems and Evolutionary Computaion, Proceedings of WSES EC, 2001, pp. 227~231.

[10] 한국항공우주연구원, 다목적 실용위성 본체설계 및 개발사업보고서, 2000.