

분산컴퓨팅 환경에서 공력 설계최적화의 효율성 연구

김 양 준¹, 정 현 주¹, 김 태 승¹, 손 창 호², 조 창 열³

A STUDY ON THE EFFICIENCY OF AERODYNAMIC DESIGN OPTIMIZATION IN DISTRIBUTED COMPUTING ENVIRONMENT

Y.-J. Kim¹, H.-J. Jung¹, T.-S. Kim¹, C.-H. Son² and C.-Y. Joh³

A research to evaluate the efficiency of design optimization was carried out for aerodynamic design optimization problem in distributed computing environment. The aerodynamic analyses which take most of computational work during design optimization were divided into several jobs and allocated to associated PC clients through network. This is not a parallel process based on domain decomposition in a single analysis rather than a simultaneous distributed-analyses using network-distributed computers.

GBOM(gradient-based optimization method), SAO(Sequential Approximate Optimization) and RSM(Response Surface Method) were implemented to perform design optimization of transonic airfoils and evaluate their efficiencies. One dimensional minimization followed by direction search involved in the GBOM was found an obstacle against improving efficiency of the design process in the present distributed computing system. The SAO was found fairly suitable for the distributed computing environment even it has a handicap of local search. The RSM is apparently the most efficient algorithm in the present distributed computing environment, but additional trial and error works needed to enhance the reliability of the approximation model deteriorate its efficiency from the practical point of view.

Key Words: 설계최적화(design optimization), 분산 컴퓨팅(distributed computing), 반응표면법(response surface method), 순차근사최적화(sequential approximate optimization), 기울기 기반 최적화법(gradient-based optimization method)

1. 서 론

전산유체역학과 수치최적화가 결합되어 구성된 공력 설계 최적화는 천음속 익형의 설계로 시작되어 최근에는 공력과 구조 등 여러 분야를 동시에 고려하여 설계하는 다분야 설계 최적화(MDO, Multidisciplinary Design Optimization)로까지 발전되어 그 연구가 활발하게 진행되고 있다[1]. 공력 설계최적화는 상당히 많은 계산시간을 필요로 하는 데, 이는 의미 있는 설계결과를 산출하기 위하여 가능한 정확한 유동해석 코드를

사용해서 많은 전산해석을 반복하여 최적화를 수행하여야 하기 때문이다. 그러므로 성공적인 설계최적화를 위해서는 효율적으로 계산시간을 감소시키는 노력이 반드시 필요하다.

계산시간을 감소시키기 위한 소프트웨어적인 접근법에는 형상함수의 사용[2], 최적화 과정의 효율화[3], 민감도 계산을 효율적으로 할 수 있는 수반(adjoint)화[4]등을 대표적으로 볼 수 있다. 또한 하드웨어적인 방법도 함께 사용할 때는, 여러 CPU를 동시 사용하는 분산·병렬처리를 통하여 공력해석 부분의 계산시간을 감소시키는 방법이 효과적으로 알려져 있다.

한편, 이와는 다른 방식으로서, 설계최적화 문제를 분절화(decomposition)하여 각각의 부 시스템 최적화 문제(subsystem optimization)를 분산된 컴퓨터에 할당하는 분산 최적화(distributed optimization) 접근법[5]도 있다. 이는 여러 부 시스템이 동시에 관여하는 MDO의 경우에 효율성 향상을 위하여 주로 사용한다.

접수일: 2006년 3월 3일, 심사완료일: 2006년 6월 9일.

1 학생회원, 울산대학교 항공우주공학과

2 정회원, 울산대학교 기계자동차공학부

3 정회원, 울산대학교 항공우주공학과

* Corresponding author E-mail: johcy@mail.ulsan.ac.kr

본 연구에서는 공력 설계최적화의 효율성 향상을 위해 분산컴퓨팅 환경을 사용하였다. 일반적으로 전산유체해석에서 영역분할(domain decomposition)방식의 병렬·분산처리를 사용하지만, 여기서는 이를 사용하지 않고 네트워크로 연결된 여러 대의 컴퓨터를 이용해 각각 단일 공력해석을 수행하도록 하는 방식을 사용하였다. 이는 병렬처리를 위해서 해석 영역을 나눌 필요가 없기 때문에 소프트웨어, 하드웨어적으로 구성하기가 훨씬 간편하며, 병렬처리 시에 발생하는 동기화 및 통신부하를 거의 없앨 수 있다는 장점이 있다.

본 연구에서는 이와 같이 구성된 분산컴퓨팅 환경에서 천음속 익형의 공력 설계최적화 문제를 사용하여 여러 최적화 알고리즘을 시험함으로써 가장 효과적으로 설계시간을 감소시킬 수 있는 알고리즘을 찾도록 하였다.

2. 설계최적화 정식화

2.1 설계변수 및 형상함수

익형의 생성은 국부 조정 능력이 뛰어난 NURBS 기반의 형상함수를 사용하였다[6]. 윗면과 아랫면에 각각 5개씩 총 10개의 조정점(control point)을 가지는 4차의 blending 함수를 사용하였다. 이 때 조정점의 y-좌표를 설계변수로 선택하고 x-좌표는 일반적으로 익형을 잘 표현할 수 있는 위치를 찾아서 고정하였다. NURBS 형상함수를 사용한 익형은 다음과 같이 표현된다.

$$Y = \sum_{i=1}^n X_i Y_i \left(\frac{x}{c} \right) \tag{1}$$

여기서, X_i 는 조정점의 y-좌표이며, Y_i 는 NURBS 형상함수이다. Fig. 1은 NURBS 형상함수의 10개의 조정점으로 생성된 익형을 보여주고 있다.

2.2 다중 블록 격자 형성 기법

시험에 사용된 설계문제는 항력과 익형의 단면적을 제약조건으로 가지는 양력최대화 문제이다.

$$\begin{aligned} & \text{Maximize } C_l(\bar{X}) \\ & \text{Subject to } C_d(\bar{X}) \leq C_{d0} \\ & \quad A(\bar{X}) \geq A_0 \end{aligned} \tag{2}$$

C_l 과 C_d 는 각각 양력계수와 항력계수이고, A 는 익형의 단면적을 의미한다. 하첨자 0은 각 계수의 초기 값을 의미하며, $C_{d0} = 0.0131$, $A_0 = 0.075$ 를 사용하였다. 최적화를 위

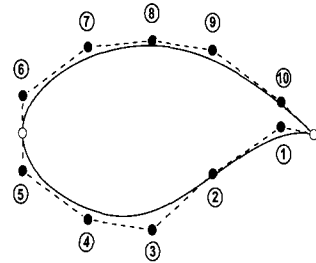


Fig. 1 Airfoil designed using NURBS shape functions

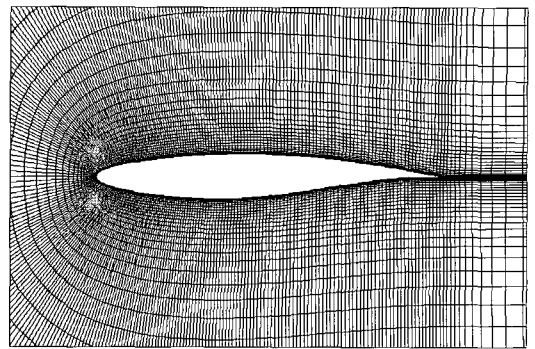


Fig. 2 Near view of computational grid(364×59)

한 설계조건으로 마하수 0.75, 받음각 0°, Reynolds수 6.5×10^6 을 사용하였다.

2.3 Navier-Stokes 유동해석

격자생성은 상용패키지인 GRIDGEN[7]을 사용하였다. bottom-up식의 격자생성 방법을 지원하는 GRIDGEN은 국부적인 형상변화에 따른 표면 격자가 자동적으로 변화하고 전체 격자시스템 또한 유연하게 조정된다. 자동화된 설계시스템을 구성하기 위해 GRIDGEN의 journal file을 사용하여 전 격자 생성과정을 batch화 하였다. Fig. 2는 공력 해석을 위한 격자계 모습이다. 원방경계가 시위길이의 20배 거리를 갖는 C-형 정렬격자이고 격자 크기는 364×59이다. 표면 첫 격자의 y^+ 는 $\theta(1)$ 로 맞추어져 있다.

익형 주위의 압축성 유동해석을 위해 cell 중심 FVM 기반의 Navier-Stokes 유동해석 패키지인 FLUENT[8]을 사용하였다. 내재적 시간적분과 공간이산화를 위하여 이차의 풍상차분법을 사용하였다. 난류모델은 1 방정식 Spalart-Allmaras 모델을 사용하였다. 공력해석과정 또한 journal file을 이용하여 batch화 하였다.

3차원 압축성점성 유동해석은 설계최적화 과정 중 형상설계나 수치최적화 과정에 비해 월등히 많은 시간이 소요된다.

따라서 효율적인 설계를 진행하기 위해서는 별도의 시간절약 기법이 필요하다. 일반적으로 최적화와 연계된 공력해석의 경우 해석시간 단축을 위하여 영역분할을 이용한 병렬처리를 하거나 유동 민감도 해석을 위한 수반(adjoint)화, 또는 가변 복잡성(variable complexity)모델[9]을 사용하기도 한다. 그러나 본 연구에서는 계산시간 단축을 위하여 이러한 방법을 사용하지 않고 대신에 네트워크로 연결된 여러 PC를 사용하여 최적화에 필요한 공력해석들을 분산처리 하도록 하였다.

2.4 수치 최적화 알고리즘

분산컴퓨팅 환경에서 최적화 알고리즘의 효율성을 평가하기 위해 GBOM(gradient-based optimization method), SAO(sequential approximate optimization), RSM(response surface method)을 각각 사용하여 비교 분석하였다. 여기서 GBOM과 SAO는 국부최적화에 바탕을 둔 기법이고 RSM은 전역최적화 기법중의 하나이다. GBOM을 사용한 설계 문제와 SAO, RSM에 의해 구성된 근사모델에 대하여 제약최적화(constrained optimization)를 수행하기 위해 범용최적화 프로그램인 DOT[10]을 사용하였다.

3. 분산컴퓨팅 환경에서의 설계최적화

3.1 네트워크 분산환경 구현

근원거리에 분산되어 있는 여러 대의 PC를 네트워크(TCP/IP protocol)를 통해 연결하여 분산 환경을 구성하였다. 이는 일반적으로 여러 대의 PC에 동종 전용 해석 패키지가 설치되어있는 대학의 CAE 실습실 특성을 고려한 것으로서 설계최적화에 응용시 추가적인 비용의 부담이 없는 장점이 있다. 본 연구에 사용한 분산환경은 기본적으로 11대의 PC가 네트워크로 연결되어 구성되었으며, 22대까지 확장 가능하도록 되어있다.

Fig. 3 은 본 연구에 사용한 설계최적화 framework architecture를 보여주고 있다[11]. 설계의 구동을 위한 1대의 server와 공력해석을 위한 10대의 client가 network로 연결되어 있다. 설계최적화를 수행하는 server에 격자생성 모듈과 형상 설계 모듈이 설치되어 있고, 각각의 client는 설계과정에 따라 요구되는 공력해석을 분산하여 동시에(simultaneously) 수행하도록 되어있다. server는 형상설계와 해당 격자생성을 순차적으로 진행하며 이 작업이 완료된 후에는 자신도 하나의 client 역할을 수행하도록 하여 총 11대의 client가 해석에 사용되도록 하였다. 각각의 모듈들을 Visual Basic script를 사용하여 통합하여 자동화된 설계최적화 시스템을 구성하였다. Table. 1은 분산 환경을 구성하고 있는 PC들의 사양을 보여주고 있다.

만일 단일 해석 job이 단위 client PC의 계산 능력을 초과

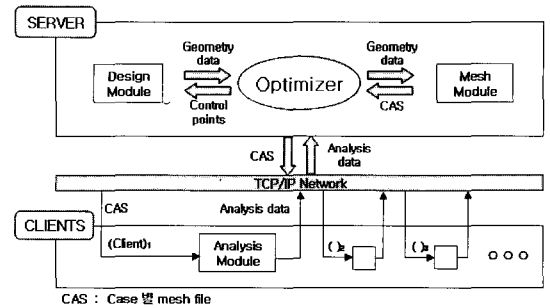


Fig. 3 Design optimization framework architecture

할 정도로 큰 최적화 문제를 다루어야 한다면 불가피하게 몇 대의 PC를 서로 연결하여 병렬처리를 통해 단일 해석 job을 수행하도록 하고, 이와 같은 client PC set을 다수로 구성하여 분산처리할 수 있을 것이다. 또는 어쩔 수 없이 효율성은 저하되었지만 일반적으로 많이 사용하고 있는 병렬처리를 처음부터 모든 해석에 적용할 수도 있을 것이다. 그러나 이런 경우는 본 연구에서와 같이 대학의 CAE 실습실 능력(Window OS 기반, 동일 상용 package 설치)을 응용하겠다는 기본 전제를 벗어남으로 여기서는 고려하지 않기로 한다.

3.2 설계최적화의 분산처리

분산처리는 최적화에 필요한 전체 공력해석 작업을 여러 개의 묶음(lot)으로 분할하고 각 lot을 매 iteration시 해당 client PC로 할당하여 처리하는 방식이다. 각 lot의 분배방법은 적용된 최적화 알고리즘 기법의 종류와 사용하는 client PC의 대수에 따라 다르다.

GBOM

전통적인 기울기 기반의 최적화 기법인 GBOM은 설계공간에 대한 방향 탐색과 ODM(one dimensional minimization)의 두 부분으로 나누어져 있는데, 본 연구에서는 방향 탐색으로 수정유용방향법(method of modified feasible direction)과 ODM으로 다항식 보간법을 사용하였다. 수정유용방향법은 방향 탐색을 위해 gradient 정보를 필요로 하는 데, 이를 계산하기 위한 설계점을 설계자가 미리 정할 수 있으므로 쉽게 공력해석

Table. 1 Specifications of network-distributed computers

	Server	Client 1~10
CPU	Pentium IV 3.0 GHz	Pentium IV 3.0 GHz
RAM	1 GB	512 MB
OS	Windows XP	Windows XP
Softwares installed	CATIA GRIDGEN DOT FLUENT	FLUENT

job을 분배할 수 있다. 그러나 ODM 과정은 매번 구한 설계결과를 평가하고 새로운 설계점을 선택하는 과정이 순차적(sequential)이므로 미리 분배할 수가 없다. 이러한 GBOM의 특성으로 인하여 본 분산처리 환경에서는 GBOM은 제한된 효과를 나타내게 된다.

SAO

Schmit 와 Farshi[12]에 의해 고안된 SAO는 전산구조해석 기반의 구조 설계최적화에 적용하였을 경우 상당한 계산비용을 줄일 수 있어 현재 구조설계 최적화에 널리 사용되고 있다. 근사식은 국부근사(local approximation) 기반으로 식 (3)과 같이 Taylor 급수 전개를 사용하여 목적함수와 제약함수를 2차의 근사식(quadratic approximation)으로 표현하였다.

$$f(\bar{X}) = f(\bar{X}_0) + \nabla f \cdot (\bar{X} - \bar{X}_0) + \frac{1}{2} (\bar{X} - \bar{X}_0)^T \cdot H \cdot (\bar{X} - \bar{X}_0) \quad (3)$$

여기서 하첨자 0은 nominal design을 의미하고, H 는 Hessian matrix이다. SAO에서 simplex 법과 선형 근사법(linear approximation)의 조합도 계산비용을 줄일 수 있는 아주 좋은 조합이다. 그러나 본 연구와 같이 비선형성이 큰 천음속 영역에서는 선형근사의 정확성이 크게 떨어지므로 이동한계(move limit)를 작게 줄 수밖에 없으며 이는 설계반복(design iteration) 횟수를 크게 증가시켜 결과적으로 2차 근사식 보다 비효율적이 된다.

본 연구와 같이 10개의 설계변수를 가질 경우에는 식 (3)의 근사모델을 구성하는 데에, 1번의 nominal analysis와 gradient를 구하기 위한 10번의 교란해석(perturbed analysis) 및 Hessian을 구하기 위한 55번의 교란해석 등 총 66번의 공력해석이 필요하다. SAO에서는 GBOM에서와 같은 순차적인 ODM 과정이 필요 없고, 총 66번의 공력해석 결과만 가지고 한 번의 설계반복(design iteration)을 진행하므로 각 설계반복 별로는 완전한 분산처리 job을 구성할 수 있다.

RSM

RSM은 전역최적화 알고리즘 중의 하나이다. 전역최적화 알고리즘은 최적화 문제의 고유한 noise를 감소시켜 전역최적해를 얻을 수 있는 장점이 있다. 또한 실험계획법(DOE, design of experiment)과 연계하여 설계점을 미리 알 수가 있으므로 분산처리환경에 아주 적합하다.

DOE에 의해 선정된 설계점에 대해서 공력해석을 수행하고 그 결과를 이용하여 C_l 과 C_d 에 대한 2차의 반응표면 모

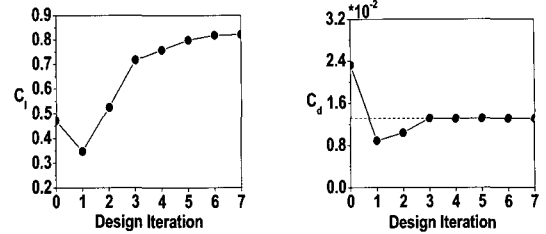


Fig. 4 Convergence history of GBOM

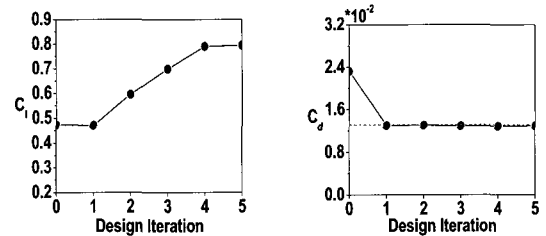


Fig. 5 Convergence history of SAO

델을 구성한다.

$$\hat{y} = c_0 + \sum_{i=1}^{10} C_j x_j + \sum_{j=k=1, j \leq k}^{10} C_{jk} x_j x_k \quad (4)$$

여기서 \hat{y} 는 근사 반응표면모델을 의미하고, C_0, C_j, C_{jk} 는 2차 다항식의 계수, x_j 는 j -번째 설계변수를 각각 나타낸다. 반응표면모델의 신뢰도는 분산분석(ANOVA, analysis of variance)을 통하여 수정결정계수(adjusted R^2)로 평가한다.

실험계획법 중 3^k 요인설계(full factorial design)를 사용하여 총 59,049개의 설계점을 먼저 생성하고, 이를 자연변수(natural variable)로 바꾸어 nominal airfoil 단면적의 $\pm 10\%$ 를 초과하는 설계점을 제외시켜서 총 58,201개의 설계점을 선정하였다. 여기에 D-optimality 조건을 적용하여 최종적으로 200개의 설계점을 선정하였다. 200개의 공력해석을 분산처리할 경우 11대의 PC(server도 포함)로 18번의 lot job을 수행하고, 추가로 2대의 client PC를 이용하여 1번의 lot job을 더 수행하여 총 19번의 lot job을 수행하였다.

4. 설계결과 및 최적화 효율성

Fig. 4와 Fig. 5는 GBOM과 SAO의 수렴과정을 각각 보여준다. RAE2822 익형을 초기형상으로 시작하여 GBOM은 7번, SAO는 5번 설계반복으로 수렴하였다.

Table. 2는 각각의 최적화 알고리즘에 대한 설계최적화 시

험 결과를 보여 주고 있다. 한 번의 공력해석 시간은 약 0.5 시간이 소요되는데 반해 형상설계 및 격자생성에 소요되는 시간은 아주 적으므로 시험결과를 요약할 때는 이를 무시하고 공력해석 횟수만으로 나타내었다. 각 최적화 알고리즘별로 소요되는 공력해석 횟수와 이를 분산처리 하였을 경우 총 analysis lot 수가 나타나 있는데, 여기서 1 analysis lot은 앞서 언급한 것과 같이 설계최적화 과정 중 11대의 client가 한 번에 수행하는 공력해석 할당량, 즉 11개의 동시 공력해석을 일컫는다. 1 analysis lot에 소요되는 시간은 1 analysis에 소요되는 시간과 동일하므로, 총 analysis lot수가 분산해석을 할 경우 실제 설계 소요시간에 해당된다.

시험 결과를 보면, 본 연구에서 구성한 분산 공력해석 기법의 분산컴퓨팅 환경과 친음속 익형의 설계문제에 대해서는 이론상으로는 RSM이 가장 효율적이다. 그러나 각 알고리즘의 특성상 실제 최적화에 필요한 추가적인 해석 또는 trial and error 작업이 항상 존재하게 되므로(이들은 적용한 설계최적화 문제의 특성 또는 설계자의 경험 등에 따라 편차가 큼), 이를 고려한다면 적어도 본 연구에 있어서는 실제적인 설계

Table. 2 Overall summarization of design optimizations in the 11PCs-distributed computing

	GBOM	SAO	RSM
Number of design iterations for convergence	7	5	1
Number of exact analyses per design iteration	11 for direction search + ≐ 10 for ODM	66	200
Number of analysis lots per design iteration	1 for direction search + ≐ 10 for ODM ¹	6	19 ²
Total number of analysis lots ³	≐ 77	30	19
Number of trial and errors or additional analyses	none	1 exact analysis per design iteration ⁴	≐ 2~3 times for improving reliability
Number of analysis lots required for practical design	≐ 77	35	≐ 38~57

- (1) sequential process이므로 1 analysis lot에 소요되는 시간과 동일
- (2) 11대 PC를 사용하여 분산해석을 할 경우
- (3) 각 알고리즘별 이론상 설계 소요시간에 관계
- (4) 다음 iteration의 move limit을 결정하기 위한 추가적인 해석
- (5) 설계문제, 설계경험 등에 따라 변함

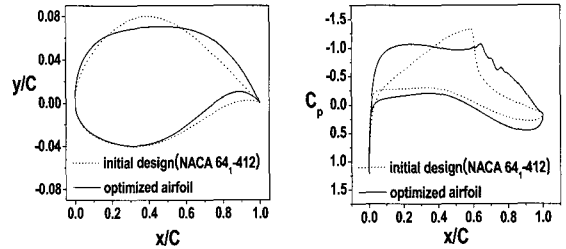


Fig. 6 Design result of GBOM

를 위해서는 SAO가 약간 더 효율적인 것으로 나타났다. 물론 최적해의 전역성에 관해서는 서로 차이가 나지만 여기서는 단지 효율성만 고려하였을 때의 결과이다.

Table. 3은 본 분산환경에 사용된 client PC의 대수를 달리 했을 때 설계시간에 대한 예측 결과를 보여준다. GBOM의 경우 분산컴퓨팅 환경에서는 단지 방향 탐색을 위한 공력해석만 분산처리가 가능하므로 본 연구와 같이 설계변수가 10개인 경우 PC의 대수가 11대보다 많더라도 효율성은 증가되지 않는다. 또한 SAO도 분산처리를 하기 어려운 순차적인 최적화 과정이지만 매번 다음 최적화를 진행하기 위해 66개의 공력해석 데이터를 필요로 하기 때문에 적어도 66대의 PC를 동원할 수 있다면 그만큼 효율성은 증가한다. RSM의 경우, 설계점을 미리 알고 최적화를 진행하기 때문에 PC의 대수를 증가시킬수록(적어도 200대까지) 단계적으로 효율성은 증가한다. 따라서 설계변수의 개수가 많고 충분히 많은 PC를 사용할 수 있다면 RSM이 가장 효율적이다.

Table. 4는 GBOM의 양력최대화 설계결과를 보여준다. 최적설계 결과 항력 제약조건을 만족하면서 양력특성이 약 74% 증가한 것을 볼 수 있다. 초기의 아음속용 재래식 익형이 최적설계 후에 초입계익형으로 변화함으로써 이와 같은 큰 특성의 변화를 가져오게 되었음을 알 수 있다. 이는 Fig 6의 형상과 압력분포에서도 확인할 수 있다.

Fig. 6은 최적설계 된 형상과 익형의 표면 압력분포를 보

Table. 3 Design efficiencies with different number of PCs involved in the present distributed computing

No. of PC's in service	Total number of analysis lots for practical design		
	GBOM	SAO	RSM
6	≐ 84	65	68~102
11	≐ 77	35	38~57
22	≐ 77	20	20~30

Table. 4 Design result of GBOM

	C_l	C_d
Initial design (NACA 64 ₁ -412)	0.4721	0.02321
Optimized airfoil	0.8227	0.01313

여준다. 양력증가를 위해 익형의 뒷전 캠버가 증가하고 익형 윗면의 충격파가 약화된 것을 볼 수 있다.

5. 결 론

설계최적화를 위하여 분산공력해석 기반의 network 분산컴퓨팅 환경을 구성하였으며, 천음속 익형 설계최적화 문제를 사용하여 본 시스템의 설계최적화 효율성을 조사하였다. 그 결과, 최적화 알고리즘에서 분산해석을 적용할 수 있는 과정이 포함된 비율과 가용 client PC의 대수 등에 따라 분산컴퓨팅 환경에서의 설계의 효율성은 각기 달라짐을 볼 수 있었다.

만일 충분히 많은 PC를 사용할 수 있다면 RSM이 가장 효율적이며, 또한 본 연구와 같이 10개의 설계변수를 가지고 2차의 근사기법을 사용했을 때에 66대보다 적은 대수의 PC를 동원한다면 SAO가 가장 효율적이다. GBOM의 경우 순차적으로 구해야하는 ODM 과정을 포함하고 있으므로 분산컴퓨팅 환경을 적용할 경우 아무리 많은 PC를 사용하더라도 효율성은 증가되지 않는다.

후 기

이 논문은 2004년 울산대학교의 연구비(2004-0203)에 의하여 연구되었으며, 관계자 여러분께 감사드립니다.

참고문헌

- [1] Sobieszczanski-Sobieski, J. and Haftka, R.T., 1997, "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments", *Structural Optimization*, 14(1), p.1-23.
- [2] Vanderplaats, G.N. and Hicks, R.M., 1976, "Numerical Airfoil Optimization Using a Reduced Number of Design Coordinates," *NASA TM X-73151*.
- [3] Joh, C.-Y., Grossman, B. and Haftka, R.T., 1993, "Design Optimization of Transonic Airfoils," *Engineering Optimization*, Vol.21, No.1, p.1-20.
- [4] Burgreen, G.W. and Baysal, O., 1994, "Three-dimensional Aerodynamic Shape Optimization of Wing Using Sensitivity Analysis," *AIAA Paper* 94-0094.
- [5] Kroo, I., 2004, "Distributed Multidisciplinary Design and Collaborative Optimization," *VKI lecture series on Optimization Methods & Tools for Multicriteria/Multidisciplinary Design*, Nov.15-19.
- [6] 문종윤, 조창열, 2000, "NURBS를 이용한 공력형상 설계 최적화," *한국항공우주학회지*, 제28권, 제8호, p.8-15.
- [7] 2003, "GRIDGEN Tutorial Workbook Version 15," *Pointwise Inc.*
- [8] 1998, "FLUENT User's Guide," *FLUENT Inc.*
- [9] Hutchison, M.G., Unger, E.R., Mason, W.H., Grossman, B., Haftka, R.T., 1993, "Aerodynamic Optimization of an HSCT Wing Using Variable-Complexity Modeling," *AIAA Paper* 93-0101.
- [10] Vanderplaats, G.N. and Hansen, S.R., 1989, "DOT Users' Manual," *VMA Engineering*.
- [11] 조창열, 이상경, 2004, "3차원 날개 공력설계를 위한 네트워크 분산 설계최적화," *한국항공우주학회지*, 제32권, 제10호, p.12-19.
- [12] Schmit, L.A. And Farshi, B., 1974, "Some Approximation Concepts for Structural Synthesis," *AIAA Journal*, Vol.12, No.5, p.692-699.