1150 대한기계학회논문집 A권, 제31권 제12호, pp. 1150~1156, 2007

<학술논문>

다분야통합최적설계 방법론의 병렬처리 성능 분석

안 문 열*· 이 세 정†

(2007년 5월 21일 접수, 2007년 10월 24일 심사완료)

Performances of Multidisciplinary Design Optimization Methodologies in Parallel Computing Environment

Moon-Youl Ahn and Se J. Lee

Key Words : Design(설계), Multidisciplinary Design Optimization(다분야통합최적설계), Optimization (최적화), Parallel Computing(병렬처리)

Abstract

Multidisciplinary design optimization methodologies play an essential role in modern engineering design which involves many inter-related disciplines. These methodologies usually require very long computing time and design tasks are hard to finish within a specified design cycle time. Parallel processing can be effectively utilized to reduce the computing time. The research on the parallel computing performance of MDO methodologies has been just begun and developing. This study investigates performances of MDF, IDF, SAND and CO among MDO methodologies in view of parallel computing. Finally, the best out of four methodologies is suggested for parallel processing purpose.

1. 서 론

최근의 설계문제는 여러 개의 분야들이 서로 연성되어 있어서 복잡한데, 이 문제를 풀기 위해서 다분야통합최적설계(Multidisciplinary Design Optimization, MDO) 방법론들이 등장하게 되었다. 하지만 다분야통합최적설계 방법은 많은 계산시간이 필요하다. 계산시간을 줄이기 위해서는 기존 네트워크로 구성된 병렬 컴퓨팅 환경을 이용한 병렬처리가 효과적이다. 즉, 병렬처리를 고려하여 효율적인 다분야통합최적설계 방법론을 선택하고 설계문제를 정식화하는 것이 필요하다.

병렬처리를 이용할 수 있는 환경을 제공하기 위한 연구도 활발하게 진행하고 있다. 공학용 그리드 컴퓨팅 환경을 제공하는 NetSolve/GridSolve[1]와 그리드 컴퓨팅 환경을 제공하는 GEODISE프로젝트[2]와 MATLAB을 이용한 분산 컴퓨팅 환경을 지원하는 MDiCE(Matlab-based Distributed Computing Environment)[3] 등이 있다. 그리고 상용 소프트웨어

기호설명

F(X) : 최적화 알고리즘의 목적함수
G(X) : 최적화 알고리즘의 부등식 제약조건
H(X) : 최적화 알고리즘의 등식 제약조건
R(X) : 분야의 오차(Residual)
Y(X) : 상태변수(state variable)
$\|x\|_p$: $\left(\sum_{i=1}^{n}|x_i|^p\right)^{\frac{1}{p}}$, 노름(norm)
ε_a : 상대오차(relative error)

* 서울시립대학교 기계정보공학과
† 책임저자, 회원, 서울시립대학교 기계정보공학과
E-mail : selee@uos.ac.kr
TEL : (02)2210-2537 FAX : (02)2248-5110

로 Mathworks사의 MATLAB의 Distributed Toolbox & Engine이 있다. 그리고 다분야통합최적설계 프레임워크 중 iSIGHT, ModelCenter, DAKOTA[4] 등이 일부 병렬처리가 가능하다.

기존의 다분야통합최적설계 문제의 계산시간을 줄이기 위한 연구로는 각 분야의 해석순서를 조절하여 다분야통합해석(Multidisciplinary Analysis, MDA)의 계산시간을 줄이는 순차분해법[5]과 병렬분해법[6] 등이 있다. 순차분해법은 다분야통합해석의 분야들 간의 반복을 발생시키는 후방연성을 최소화하는 방법이다. 이는 다분야통합해석의 해석시간을 줄여 다분야통합최적설계의 전체 수행을 줄일 수 있지만, 병렬처리에 대한 고려가 전혀 없었기 때문에 순차분해법에 따라 분해한 다분야통합최적설계 문제는 병렬 컴퓨팅 환경에 적합하지 않다. 이러한 단점을 보안하고자 병렬처리를 고려한 병렬분해법이 제시되었다. 하지만 이 방법은 다분야통합해석의 하위 시스템에 병렬처리를 적용하여 다분야통합해석의 계산시간을 줄이는데 도움이 되지만 다분야통합최적설계 문제의 전체를 고려하여 고안된 방법이 아니기 때문에 시스템 전체 계산시간을 줄일 수 있는지 검증되어 있지 않다.

본 연구에서는 다분야통합최적설계 문제를 풀 때 병렬처리와 최적화 과정 전체를 고려하여 다분야통합최적설계 방법론들의 성능을 분석, 평가하고자 한다. 우선 성능 평가를 위한 기준을 선정하여 제시하고, 선택된 예제들에 각 방법론을 적용하여 성능을 분석한다. 그리고 분석 결과를 정리하여 가장 효율적인 방법을 제시한다. 단, 기존의 네트워크 환경을 활용하고, 기존 코드들을 수정하지 않는 범위에서 coarse-grained 병렬처리, 즉 다중프로그래밍 환경에서 동시작업 프로세스들의 다중처리를 고려하기로 한다.

이 논문의 내용은 다음과 같이 구성되어 있다. 2 장에서는 병렬처리 관점에서 다분야통합최적설계 방법론들을 분석하고 3 장에서는 이러한 관점에서 본 예제문제에 다분야통합최적설계 방법론을 적용하여 그 성능을 평가한다. 4 장에서는 앞의 내용을 요약하고 병렬 컴퓨팅 환경에 적합한 다분야통합최적설계 방법론을 제시한다.

2. 병렬처리 관점에서의 MDO 방법론

본 장에서는 다분야통합최적설계 방법론들 중 MDF,[7] IDF,[8] SAND,[8] CO[9] 방법론의 병렬처리에 대한 적합성 및 가능성을 살펴보고자 한다.

2.1 Multidisciplinary Feasible(MDF)

이 방법은 다분야통합최적설계 문제를 푸는 전통적인 방법이다. Fig. 1에 나타낸 것과 같이 이 방법은 하위 시스템의 다분야통합해석과 상위 시스템의 최적화 알고리즘으로 이루어져 있다. 이 방법은 단지 하위 시스템이 다분야통합해석으로 이루어져 있다는 것만이 일반적인 최적설계 방법론과 다를 뿐 일반적인 최적설계 과정과 동일하게 취급할 수 있다.

이 방법론에 병렬처리를 적용하기 위해서 하위 시스템의 다분야통합해석의 비선형 연립방정식을 풀 때 반복 수행할 때마다 순차적으로 풀지 않고 각 분야를 분리하여 병렬처리 할 수 있다.

2.2 Individual Discipline Feasible(IDF)

이 방법은 다분야통합해석의 분야 간의 연성을 제거하여 분해하는 방법이다. 상위 시스템에서 설계변수와 상태변수(state variable)의 값을 하위 시스템에 주고 하위 시스템은 이 값을 가지고 해석을 수행한 후 계산된 상태변수를 되돌려 준다. 그러면 상위 시스템의 상태변수와 하위 시스템에서 계산한 상태변수가 값이 다르게 되는데, 이 차이를 맞추기 위하여 등식 제약(equality constraint)조건을 상위 시스템의 구속조건에 추가하게 된다.

Fig. 2에서와 같이 해석기 사이의 연성은 없어지고 MDF 방법론에서의 다분야통합해석을 각 분야로 분해할 수 있다. 분해된 해석기들은 병렬 컴퓨팅 환경에 쉽게 적용하여 처리할 수 있다.

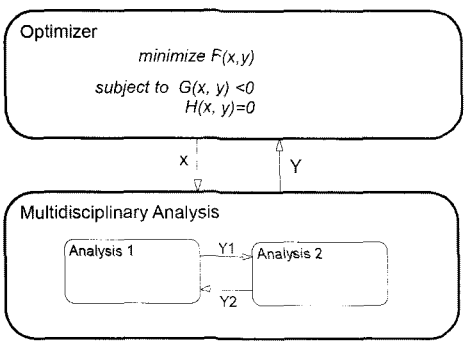

Fig. 1 Multidisciplinary feasible(MDF)

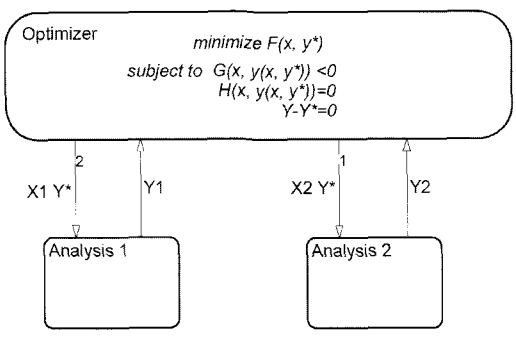

Fig. 2 Individual discipline feasible(IDF)

2.3 Simultaneous Analysis and Design(SAND)

이 방법은 하위 시스템의 오차(Residual)를 상위 시스템의 등식 제약조건으로 추가하고 상위 시스템의 설계변수에 하위 시스템의 상태변수를 추가하는 방법이다. 이 방법은 최적화 과정이 끝나기 전에 생성된 설계변수가 각 분야의 유효성(feasibility)를 만족하지 않는다는 단점을 가지고 있다.

하지만 Fig. 3에 나타낸 것처럼 연성된 분야들을 분해할 수 있고 하위시스템의 수렴과 상위 시스템의 최적화를 동시에 수행하기 때문에 다른 다분야통합최적설계 방법론에 비해 계산시간이 적다는 장점을 가지고 있다. 또한, 분해된 분야들을 병렬 컴퓨팅 환경에 적용하여 계산시간을 줄일 수 있다.

2.4 Collaborative Optimization(CO)

이 방법은 두 단계 다분야통합최적설계 방법론이다. 상위 시스템에서는 목표 값(target value, Z)을 가지고 분야 간의 적합성 조건(compatibility constraints, J)를 만족하면서 상위 시스템의 목적함수(objective, F)를 최소화하는 최적화를 실행한다. 하위 시스템에서는 하위 시스템의 설계변수(Xi)와 상태변수(Yi)와 상위 시스템의 목표 값과의 최소자승(least-square, Ji)을 하위 시스템의 목적 함수로 하고 그 목적 함수를 최소화하는 최적화를 수행한다. 이 방법은 해석기 간의 연성을 제거하여 다분야통합최적설계 문제를 푸는 방법이다.

Fig. 4에 나타낸 것처럼 하위 시스템 사이의 연성을 제거하여 분해할 수 있다. 분해된 하위 시스템은 병렬 처리할 수 있다. 하지만 이 방법은 하위 시스템의 최적화 과정에서 반복수행 과정을 포함하고 있기 때문에 상위 시스템의 반복수행의 횟수에 따라서 전체 시스템의 반복횟수가 증가한다.

3. MDO 방법론의 병렬처리 성능

본 장에서는 앞에서 말한 다분야통합최적설계 방법론의 병렬 컴퓨팅 환경에서의 성능을 평가하기 위해서 Analytic 문제[8]와 Colville 문제[10]를 예 제로 택하여 풀었다. 여기서 최적화 알고리즘은 MATLAB Optimization Toolbox에서 제공하는 최적화 기법 중 Sequential Quadratic Programming을 사용하였고 다분야통합해석의 비선형 연립방정식은 MATLAB Optimization Toolbox의 fsolve를 사용하여 풀었다. 최적화 알고리즘의 제약조건의 수렴 판정(convergence tolerance) 기준은 10^{-6}으로 하였고 목적함수의 수렴 판정 기준은 10^{-3}으로 하였다.

다분야통합최적설계 방법론의 성능을 평가하기 위하여 계산시간, 정확도, 강건성(Robustness)을 평가 기준으로 정하였다. 이 세 가지 평가기준을 구체적으로 설명하면 다음과 같다.

계산시간을 평가하기 위해 각 분야의 함수 호출 수를 첫 번째 기준으로 정하였다. 병렬처리 시 해석기의 계산시간에 비해 다른 부가적으로 소요되는 시간이 상대적으로 매우 작기 때문에 실제적인 계산시간은 대부분 해석기의 계산시간이다. 그래서 최적화 과정 중 각 분야를 호출한 수를 계산시간으로 가정하였고 각 분야의 해석시간은 모두 같다는 가정 하에 평가하였다.

각 방법론들의 정확도를 평가하기 위해서 엄밀해의 최적 설계변수와 각 방법론들에서 구한 최적 설계변수의 노름(L_2-norm)을 구하였고 최적설계 문제의 엄밀해가 존재하지 않는 경우에는 여러 초기값을 통해 구한 최적 값 중 가장 최적의 값을 엄밀해로 정하였다.

각 방법론의 강건성을 평가하기 위하여 다양한 초기 설계변수에 따라 최적 해를 구할 수 있는지 여부를 평가 기준으로 정하였다. 난수 발생기를 이용하여 초기 설계변수를 정하고 최적화를 수행한 후 구한 최적 값과 엄밀해의 최적 값의 상대오차(relative error)가 10^{-2}보다 작은 경우만을 수렴한 것으로 판정하고 수렴 횟수를 구하였다.

3.1 Analytic 문제

위에서 말한 세 가지 평가기준을 적용하여 다분야통합최적설계 방법론으로 엄밀해가 존재하는

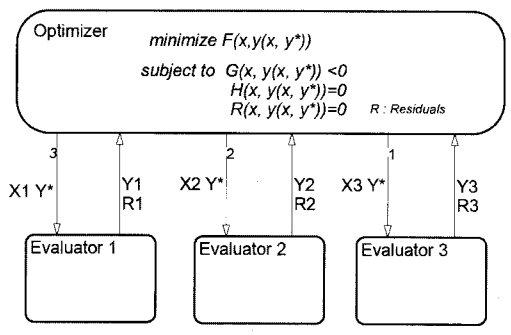

Fig. 3 Simultaneous analysis and design(SAND)

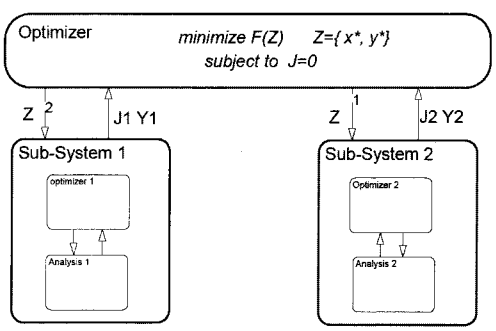

Fig. 4 Collaborative optimization(CO)

문제[8]를 풀었다.

이 최적설계문제는 두 개의 분야로 이루어져 있고 엄밀해의 최적 설계변수는 (1.9776, 0, 0)이고 그 목적함수의 최적 값은 3.18339 이다. 이 문제는 다음과 같이 정식화하였다.

$$\begin{aligned}
&\text{Minimize} && x_2^2 + x_3 + y_1 + e^{-y_2} \\
&\text{subject to} && 1 - \frac{y_1}{3.16} \le 0 \\
& && \frac{y_2}{24} - 1 \le 0 \\
& && -10 \le x_1 \le 10 \\
& && 0 \le x_2 \le 10 \\
& && 0 \le x_3 \le 10
\end{aligned}$$

Discipline 1

$$\textit{Given}: x_1, x_2, x_3, y_2$$
$$y_1 = x_1^2 + x_2 + x_3 - 0.2y_2$$

Discipline 2

$$\textit{Given}: x_1, x_3, y_1$$
$$y_2 = \sqrt{y_1} + x_1 + x_3$$

각 방법론들의 계산시간을 평가하기 위해서 최적화 과정에서의 전체 함수 호출 수를 구하여 Table 1에 나타내었다. 표시된 숫자는 병렬 처리하였을 경우와 하지 않았을 경우의 최적화가 끝났을 때의 총 함수 호출 수이다. 총 함수 호출 수에는 최적화 과정에서 필요한 도함수를 구하기 위한 함수 호출 수도 포함하였다. 병렬 처리하였을 경우의 함수 호출 수는 동시에 병렬 처리된 것들 중 함수 호출 수가 가장 큰 수로 하였다.

각 다분야통합최적설계 방법론의 최적화 과정이 끝날 때까지의 함수 호출 수는 IDF 방법론이 가장 적었다. 이러한 결과는 각 분야의 해석이 반복수행이 필요 없는 간단한 문제이기 때문이다. 하지만 최적화 과정에서 반복 수행할 때마다 각 분야의 수렴이 필요한 IDF 방법론은 각 분야의 수렴이 필요 없는 SAND 방법론보다 더 많은 계산시간을 필요로 할 것이다.

각 방법론들의 정확도를 평가하기 위해서 엄밀해의 최적 설계변수와 각 방법론에서 구한 최적 설계변수의 노름을 구하였다. Table 2는 초기 설계변수가 (1, 5, 2)일 때 각 방법론에서 구한 최적 설계변수의 정확도, 즉 노름이다.

CO 방법론의 정확도가 다른 방법론들의 정확도보다 떨어진다. CO 방법론은 상위 시스템과 하위 시스템이 모두 최적화를 실행하기 때문에 각 하위 시스템에서 발생한 오차가 상위 시스템까지 영향을 미쳐서 상위 시스템이 정확한 최적 값을 얻는데 어려움을 가진다.

Table 3에서는 각 방법론들의 강건성에 대한 평가를 위해서 난수 발생기를 사용하여 여러 개의 초기 설계변수를 만들었고 그 초기 설계변수를 가지고 수렴의 정도를 살펴보았다. 50 개의 초기 설계변수를 가지고 다분야통합최적설계 방법론을 적용하였고 엄밀해의 최적 값과 각 방법론들을 통해서 구한 최적 값의 상대오차가 10^{-2} 보다 큰 경우를 제외한 수렴 횟수를 나타내었다.

모든 방법론들이 전부 수렴하였으나 정확성을 고려하여 수렴 빈도를 보았을 때 SAND 방법론과 IDF 방법론의 강건성이 좋았다. 반면, MDF 방법론은 다른 방법론에 비해 좋지 않았다. IDF 와 SAND 방법론은 상위 시스템에서 하위 시스템의 상태변수를 상위 시스템의 최적화 설계변수로 이용되기 때문에 하위 시스템이 상위 시스템과 긴밀한 관계를 유지하여 전체 시스템의 수렴 성능이 좋은 것으로 보인다.

3.2 Colville 문제

이 예제는 1968 년에 Colville에 의해 만들어진 화학 반응장치 최적화 문제[10]이다. 본 논문에서는 일곱 개의 분야로 이루어진 문제로 정식화하였다. 여기서 IDF, SAND, CO방법론에 따라서 분해할 때 몇 개의 분야를 묶어서 하위 시스템을 다분야통합해석으로 풀 경우 계산시간이 많이 늘어날 것이다.

그래서 다분야통합해석을 제거하기 위하여 하위 시스템을 일곱 개로 분해하여 병렬처리의 효과를 극대화 시켰다.

Table 1 Numbers of function evaluations

Methodology	Parallel	Sequential
MDF	312	624
IDF	48	96
SAND	54	108
CO	2566	5086

Table 2 Accuracy of optimal design variables

Methodology	Accuracy, $\|x\|_2$
MDF	3.9973×10^{-5}
IDF	3.8884×10^{-5}
SAND	3.8884×10^{-5}
CO	1.2763×10^{-3}

Table 3 Robustness of design variable convergence

Methodology	Number of hits for $\varepsilon_a < 10^{-2}$
MDF	28
IDF	37
SAND	38
CO	34

이 문제의 설계변수와 상태변수는 아래와 같다.

설계변수	정의
X_1	Olefin Feed Rate, bpd
X_2	Isobutane Recycle Rate, bpd
X_3	Fresh Acid Addition Rate, Mbpd

상태변수	정의
Y_1	profit, $/day
Y_2	Alkylate Product Rate, bpd
Y_3	Make-Up Isobutane Rate, bpd
Y_4	Spent Acid Strength, wt%
Y_5	Motor Octane Number
Y_6	External Isobutane to Olefin Ratio
Y_7	Acid Dilution factor, ADF
Y_8	F-4 Performance No.@4.6cc Tel/gal

최적설계 문제는 아래와 같이 정식화하였다.

Minimize $-Y_1$

$Y_1=(0.063Y_2Y_5-5.04X_1-3.36Y_1-0.035X_1-10.0X_1)$

Subject to

$0 \le Y_2 \le 5000$

$0 \le Y_3 \le 2000$

$85 \le Y_4 \le 93$

$90 \le Y_5 \le 95$

$3 \le Y_6 \le 12$

$0.01 \le Y_7 \le 4$

$145 \le Y_8 \le 162$

$0 \le X_1 \le 2000$

$0 \le X_2 \le 16000$

$0 \le X_3 \le 120$

Disciplines

$Y_2 = X_2(112+13.167Y_6-0.6667Y_6^2) / 100$

$Y_3 = 1.22Y_2-X_1$

$Y_4 = 98000X_3 / (Y_2Y_7+1000X_3)$

$Y_5 = 86.35+1.098Y_6-0.038Y_6^2+0.325(Y_4-89)$

$Y_6 = (X_2+Y_3) / X_1$

$Y_7 = 35.82-0.222Y_8$

$Y_8 = -133+3Y_5$

Table 4는 계산시간을 평가하기 위해서 함수의 호출 수를 선택하여 병렬 처리하였을 경우와 하지 않았을 경우로 나누었고 최적화가 끝났을 때의 총 함수 호출 수를 나타내고 있다. 총 함수 호출 수는 최적화 과정에서 필요한 도함수를 구하기 위한 함수 호출 수도 포함하였다. 병렬 처리하였을 경우의 함수 호출 수는 병렬 처리된 것들 중 함수 호출 수가 가장 큰 수로 하였다.

각 다분야통합최적설계 방법론의 최적화 과정이 끝날 때까지의 함수 호출 수는 SAND 방법론이 가장 적었다.

이 문제는 엄밀해가 없기 때문에 정확도를 평가하기 위한 노름을 구할 때 필요한 엄밀해를 정하기 위해 여러 개의 초기 설계변수를 통해서 구한 값들을 검토하여 엄밀해를 정하였다. 그 엄밀해의 최적 설계변수는 (255.1979, 1062.3743, 56.8199)이고 최적 값은 2441.22816 으로 하였다. 초기 설계 변수는 (532.54, 1100, 120)으로 하여 최적화를 하였다.

Table 5에서 최적 설계 변수의 정확도는 엄밀해의 최적 설계변수와 각 방법론의 통하여 구한 최적 설계변수의 노름을 구하였다. MDF, IDF, SAND 방법론들의 최적 설계 변수들은 서로 비슷한 값을 갖지만 CO 방법론만 다른 값을 갖는다. 이 예제에서 CO 방법론의 정확도는 다른 방법론에 비해 좋지 않다. CO 방법론은 이전 예제에서 언급했던 것과 마찬가지로 하위 시스템의 최적화의 영향 때문에 상위 시스템의 최적화의 정확도가 떨어지게 된다.

각 방법론들의 강건성에 대한 평가를 위해서 난수 발생기를 사용하여 여러 개의 초기 설계변수를 만들었고 그 초기 설계변수를 가지고 수렴의 정도를 살펴보았다.

Table 6은 50 개의 초기 설계변수를 가지고 다분야통합최적설계 방법론을 적용하였고 엄밀해의 최적 값과 각 방법론들을 통해서 구한 최적 값의 상대오차가 10^{-2} 보다 큰 경우를 제외한 수렴 횟수를 나타내었다.

Table 4 Numbers of Function Evaluations

Methodology	Parallel	Sequential
MDF	8236	57652
IDF	209	1463
SAND	176	1232
CO	11888	67811

Table 5 Accuracy of Optimal Design Variables

Methodology	Accuracy, $\|x\|_2$
MDF	8.3099×10^{-3}
IDF	7.0862×10^{-5}
SAND	6.3819×10^{-5}
CO	1.6469×10^{-1}

Table 6 Robustness of Design Variable Convergence

Methodology	Number of hits for $\varepsilon_a < 10^{-2}$
MDF	9
IDF	49
SAND	46
CO	17

평가 결과 IDF 와 SAND 방법론의 강건성이 좋았다. 하지만 CO 와 MDF 방법론의 경우 초기값에 따라 수렴이 되지 않는 경우가 발생하였고 따라서 강건성이 좋지 않았다. IDF 와 SAND 방법론은 상위 시스템에서 하위 시스템의 상태변수를 상위 시스템의 최적화 설계변수로 이용되기 때문에 하위 시스템이 상위 시스템과 긴밀한 관계를 유지하여 전체 시스템의 수렴이 좋은 것으로 보인다. 반면, CO 와 MDF 방법론은 하위 시스템이 잘 수렴하지 않기 때문에 상위 시스템의 최적화가 제대로 수행되지 않는다.

두 가지의 예제문제의 각 방법론들의 전체 계산시간을 비교하였을 때 IDF 와 SAND 방법론이 다른 방법론에 비하여 월등히 좋은 성능을 가지고 있다. 이러한 결과는 아래와 같은 이유에 있다. 첫째, MDF 방법론의 경우 하위 시스템의 다분야통합해석의 비선형 연립방정식을 풀기 위해서 사용한 fsolve 의 경우 Trust-Region dogleg 방법을 사용하였는데 여기서 많은 계산시간이 필요하였다. 둘째, CO 방법론은 하위 시스템의 최적화 과정에서의 반복수행 횟수가 전체 계산시간을 늘어나게 하였다. 셋째, IDF 와 SAND 방법론들은 다분야통합해석을 완전히 분해되어 계산시간을 많이 필요로 하는 다분야통합해석이 필요 없기 때문에 효율적이었다.

만약, 본 논문의 예제문제와 다르게 각 분야가 반복수행을 통한 수렴이 필요하더라도 IDF 방법론은 각 분야가 잘 수렴된다면 전체 함수 호출 수는 늘어나지 않을 것이다. 반면, SAND 방법론은 각 분야의 수렴이 최적화 과정과 동시에 이루어지기 때문에 전체 함수 호출 수는 늘어나게 되겠지만 반복수행 할 때마다 각 분야의 수렴이 필요 없기 때문에 전체 계산시간은 많이 늘어나지 않을 것이다.

위와 같은 평가를 통해서 시스템 전체의 계산시간을 고려하였을 때 SAND 방법론이 가장 뛰어난 것으로 보인다. 정확도와 강건성의 성능을 포함하여 각 방법론을 비교하여 보아도 SAND 방법론이 가장 효율적인 것으로 평가되었다.

4. 결 론

본 연구에서는 병렬처리를 고려하여 다분야통합최적설계 방법론들 중 MDF, IDF, SAND, CO 방법론의 성능을 평가하였다. 평가한 결과를 바탕으로 다분야통합최적설계 방법론의 선택에 대한 기준을 제시하였다.

병렬 컴퓨팅 환경에서 다분야통합최적설계 방법론을 적용하여 예제 문제를 풀었을 때 IDF 와 SAND 방법론의 계산 시간이 가장 적었다. 각 분야의 수렴하기 위한 계산을 포함하여 계산시간을 본다면 IDF 방법론의 계산시간보다 SAND 방법론의 계산시간이 적을 것이다. 정확도를 보면 CO 방법론을 제외한 다른 방법론들은 정확도가 좋았다. 마지막으로 초기 설계변수에 따른 강건성을 살펴보면 IDF 와 SAND 방법론이 다른 방법론보다 우수하다.

이러한 성능에 대한 평가를 통하여 SAND 방법론이 병렬처리를 고려할 때 본 논문에서 비교한 방법론들 중 가장 좋다는 것을 알 수 있다.

향후 하위 시스템의 의한 영향을 평가하기 위한 연구와 연성을 제거하기 위해 발생한 상위 시스템의 설계변수와 제약조건 수의 증가에 의한 영향에 대한 연구가 필요할 것으로 보인다.

후 기

이 논문은 2006 년도 서울시립대학교 학술연구조성비에 의하여 연구되었음.

참고문헌

(1) YarKhan, A., Seymour, K., Sagi, K., Shi, Z., Dongarra, J., 2006, "Recent Developments in GridSolve," *International Journal of High Performance Computing Applications (Special Issue: Scheduling for Large-Scale Heterogeneous Platforms)*, Robert, Y eds. Sage Science Press, Vol. 20.

(2) Eres, M.H., Pound, G.E., Jiao, Z., Wason, J.L., Xu, F., Keane,A.J., Cox, S.J., 2004, "Implementation and utilisation of a Grid-enabled problem solving environment in Matlab," *Future Generation Computer Systems(in Press)*.

(3) http://www.fh-kaernten.at/mdice/.

(4) Eldred, M. S., Giunta, A. A., and Bart, G. B. W., 2005, *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis, Version 3.3+ Users Manual*, Sandia National Laboratories.

(5) Roger, J. L., 1996, "DeMAID/GA an Enhanced Design Manager's Aid for Intelligent Decomposition," *AIAA paper*, NASA Langley Research Center.

(6) Park Hyung-Wook, Kim Sung-Chan, Kim Min-Soo and Choi Dong-Hoon, 2001, "Decomposition Based Parallel Processing Technique for Efficient Collaborative Optimization," *Trans. Of the KSME(A)*, Vol. 25, No. 5, pp. 883~890.

(7) Cramer, E. J., Dennis, J., Frank, P. D., Lewis, R. M. and Shubin, G. R., 1994, "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, pp. 754~776.

(8) Tedford, N. P. and Martins, J. R. R. A., 2006, "On

the Common Structure of MDO Problems: A Comparison of Architectures," *AIAA-2006-7080, 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia.

(9) Alexandrov, N. M., Lewis, R. M., 2000, "Analytical and Computational Aspects of Collaborative Optimization," NASA TM 2000-210104.

(10) Colville, A. R., 1968, "A Comparative Study on Nonlinear Programming Codes," IBM New York Science Center Report No. 320-2949, Test Problem #8(pg. 32), IBM Corporation, Philadelphia Scientific Center, Philadelphia, PA.