수학예제를 이용한 다분야통합최적설계 방법론의 비교

이상일 *(한양대 대학원 기계설계학과), 박경진(한양대 기계정보경영공학부)

Comparison of MDO Methodologies With Mathematical Examples

S. I. Yi(Mechanical Design. Dept. HYU), G. J. Park(Mechanical Eng. Dept. HYU)

ABSTRACT

Recently engineering systems problems become quite large and complicated. For those problems, design requirements are fairly complex. It is not easy to design such systems by considering only one discipline. Therefore, we need a design methodology that can consider various disciplines. Multidisciplinary Design Optimization (MDO) is an emerging optimization method to include multiple disciplines. So far, about seven MDO methodologies have been proposed for MDO. They are Multidisciplinary Feasible (MDF), Individual Feasible (IDF), All-at-Once (AAO), Concurrent Subspace Optimization (CSSO), Collaborative Optimization (CO), Bi-Level Integrated System Synthesis (BLISS) and Multidisciplinary Optimization Based on Independent Subspaces (MDOIS). In this research, the performances of the methods are evaluated and compared. Practical engineering problems may not be appropriate for fairness. Therefore, mathematical problems are developed for the comparison. Conditions for fair comparison are defined and the mathematical problems are defined based on the conditions. All the methods are coded and the performances of the methods are compared qualitatively as well as quantitatively.

Key Words : Multidisciplinary Design Optimization(다분야통합최적설계), MDF, IDF, AAO, CSSO, BLISS, CO, MDOIS

1. 서론

최근에 고려하는 공학시스템은 거대하고 복잡하게 되고 있으며, 설계 요구 조건은 더욱 까다로와지고 있다. 설계자들은 기존의 설계 방법에 비해 복잡한 문제에 대한 새로운 방법을 찾게 되었으며, 이러한 방법 중의 하나가 다분야통합최적설계(multidisciplinary design optimization, MDO)이다[5,22]. 다분야통합최적설계는 기존 최적설계의 범주에 속한다. 일반적으로 기존 최적설계 적용 사례의 특징은 한 분야(discipline)에서만 이뤄졌다는 점이다. 그러나 실제 공학에서 발생하는 복잡한 문제는 한 분야만을 고려하여 해결하기가 쉽지 않다. 다분야통합최적설계는 이를 극복하기 위해서 다양한 분야를 고려한 설계방법론이다.

현재까지 다분야통합최적설계 방법론은 대략 7가지가 나와 있는데 이를 간략히 살펴보면 다음과 같다. 단단계(Single level) 최적화 방법론으로써 각 영역이 독립적으로 해석을 수행하는 것으로는 MDF, IDF [4,10], AAO 등이 있다. 이러한 세가지 방법론은 각각의 분야에 설계의 권한이 없으며, 단지 해석만 하게 되는 특징이 있다. 따라서 위에 언급한 세 방법을 기본으로 각각의 영역에 독자적인 설계 권한을 주는 좀더 현실적인 다단계(multi-level) 최적화 방법론들이 뒤를 이었다. 이러한 방법들에는 Sobieszczanski-Sobieski 가 제안한 Concurrent Subspace Optimization(CSSO)[3,16,20,25], Kroo 가 제안한 Collaborative Optimization(CO)[8,26], Sobieszczanski-Sobieski 가 제안한 Bi-Level Integrated System Synthesis(BLISS)[23,24], 신문균과 박경진이 제안한 독립적 하부 시스템에 의한 다분야통합최적설계(Multidisciplinary Optimization Based on Independent Subspaces, MDOIS)[2]가 있다.

위의 방법론들은 아직도 연성이 약하거나 소규모 문제만을 완벽하게 해결할 수 있으며 실제적인 대규모 문제에 대해서는 가능성만을 제시한다. 연성이 강한 대규모의 문제를 수학적으로 완벽하게 풀 수 있는 방법은 아직은 존재하지 않는 것으로

보인다. 따라서 위에서 소개한 다분야통합최적설계 방법 중 어떤 것도 최종적이라고 말할 수는 없다. 단지 이들은 현재까지 개발된 다분야통합최적설계방법이므로 앞으로 더 나은 방법들의 개발이 이어져야 할 것이다. 이를 위해서는 이제까지 제시된 여러 가지 방법론들의 적용방법에 대한 효율성과 각각의 방법론의 비교에 관한 연구가 필요하다. 기존에도 다분야통합최적설계 방법론의 효율성 비교에 관한 연구는 많이 이루어져왔다[6,9,12,13,15,17]. 기존의 다분야통합최적설계 방법론 비교에 관한 연구는 예제의 선정에 있어서 다분야통합최적설계가 풀어야 할 일반적인 정식화 정의를 바탕으로 하여 이루어 지지 못하였다. 사용된 예제의 적절성이나 비교 결과의 부분에 있어 정량적·정성적인 비교의 부분이 부족하였다. 또한 일부 방법론만을 대상으로 하여 비교하였다. 이같은 기존의 연구의 문제점을 보완하기 위하여 다분야통합최적설계가 풀어야 할 일반화된 형태를 정의하고, 이제까지 발표된 거의 대부분의 대표적인 방법론 7 가지를 모두 비교할 수 있는 수학예제를 제안한다. 이 수학예제를 각 방법론들에 대하여 정식화하고 풀어본 결과를 정량적·정성적으로 비교하여 실제공학문제 적용 시의 유용성을 검토한다.

2. 다분야통합최적설계

다분야통합최적설계는 여러 개의 영역(discipline)이 해석단계 또는 설계단계에 있어 연성(coupling)이 존재하는 시스템에 대해 설계를 하는 방법이라고 할 수 있다[5,22]. 여기서 말하는 연성이란 일반적으로 어떤 대상과 다른 대상이 서로 연결되어 있는 상태를 의미한다. 그것이 직접적이든 간접적이든 관련이 있으므로, 서로 연결된 상태에 따라서 한쪽의 변화가 다시 자기에게 돌아오는 순환적인 연결이 있을 수 있고, 그런 변화가 다시 돌아오지 않을 수도 경우도 있다. 이에 비해서 연성이 없는 상태는 서로 독립적으로 나뉘어져 있으며 아무런 관련이 없는 상태를 말한다. 즉 한 분야의 변화가 다른 분야에 영향이 없는 경우이다. 이러한 연성의 의미는 공학 문제에서 많이 쓰고 있다. 특히 다분야통합최적설계는 이를 대단히 중요하게 사용하고 있다. 일반적으로는 주로 해석단계에서 연성이 있는 문제를 어떻게 풀어갈 것인가에 관심이 있는데, 한편으로는 설계단계의 연성을 효과적으로 다루려는 노력도 있다.

해석단계의 연성관계에서 해석이 종료가 되는 상태, 즉 더 이상의 연성변수의 변화가 없는 상태를 MF(Multidisciplinary feasibility)라 한다. 특히, 다분야통합최적설계에서는 이러한 과정을 시스템해석(system analysis)이라 한다.

그리고 연성변수와 관계 없이 각 해석이 완료된 상태를 IF(Individual feasibility)라 한다. 이와 같은 과정은 다분야통합최적설계의 각 방법론의 특징에 따라 매 함수호출 시 MF 를 만족시키면서 진행할 수도 있고, 함수호출 시에는 IF 를 만족시키고 설계 종료 점에서 MF 를 만족시킬 수도 있다. 또, 방법론에 따라서는 함수호출 시에는 MF, IF 를 모두 만족시키지 못하고 설계 종료 점에서 두 가지를 만족시키는 방법론도 있다.

설계변수, 목적함수, 제한조건의 영역간의 공유여부에 따라 다분야통합최적설계를 분류하기도 한다. 두 영역이 있을 때 다분야통합최적설계의 관점에서의 정식화는 다음과 같다.

$$
\begin{array}{ll}
\text{Find} & \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_c \\
\text{to minimize} & \left\{ \begin{array}{l} f_1(\mathbf{b}_1, \mathbf{b}_c, \mathbf{z}_1^{nc}, \mathbf{z}_1^{c}, \mathbf{z}_2^{c}) \\ f_2(\mathbf{b}_2, \mathbf{b}_c, \mathbf{z}_2^{nc}, \mathbf{z}_2^{c}, \mathbf{z}_1^{c}) \\ f_c(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_c, \mathbf{z}_1^{nc}, \mathbf{z}_1^{c}, \mathbf{z}_2^{nc}, \mathbf{z}_2^{c}) \end{array} \right\} \\
\text{subject to} & \mathbf{g}_1(\mathbf{b}_1, \mathbf{b}_c, \mathbf{z}_1^{nc}, \mathbf{z}_1^{c}, \mathbf{z}_2^{c}) \le 0 \\
 & \mathbf{g}_2(\mathbf{b}_2, \mathbf{b}_c, \mathbf{z}_2^{nc}, \mathbf{z}_2^{c}, \mathbf{z}_1^{c}) \le 0 \\
 & \mathbf{g}_c(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_c, \mathbf{z}_1^{nc}, \mathbf{z}_1^{c}, \mathbf{z}_2^{nc}, \mathbf{z}_2^{c}) \le 0 \\
 & \mathbf{z}_1 = \mathbf{h}_1(\mathbf{b}_1, \mathbf{b}_c, \mathbf{z}_2^{c}) \\
 & \mathbf{z}_2 = \mathbf{h}_2(\mathbf{b}_2, \mathbf{b}_c, \mathbf{z}_1^{c})
\end{array}
\qquad (1)
$$

식 (1)에서 $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_c$는 각각 설계영역 1 과 설계영역 2 의 지역변수, 공통변수를 나타낸다. $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{g}_c$ 는 제한조건으로써 각 설계영역의 지역제한조건과 공통제한조건을 나타낸다. $\mathbf{f}$ 는 목적함수의 벡터로 지역목적함수와 공통목적함수를 포함한다. $\mathbf{h}$ 는 각 영역의 해석기를 나타내게 되며, $\mathbf{z}$ 는 각 영역의 해석기에서 나온 결과인 상태변수이다. 이 때 아래첨자는 영역의 번호를 나타내며, 윗 첨자는 연성관계를 나타낸다. c 는 연성, nc 는 비연성을 나타낸다. 예를 들어 $\mathbf{z}_i^c$ 는 i 번째 영역의 연성변수를 나타낸다. 여기서 f_1, f_2, f_c 은 설계하고자 하는 대상으로 각 설계영역의 지역 목적함수 및 공통 목적함수이다. 위의 정식화는 다분야통합최적설계가 풀어야 하는 일반적인 문제 형태이다. 이와 같은 다분야통합최적설계가 풀어야 하는 일반적인 형태를 바탕으로 방법론 비교를 위한 수학예제를 제안한다.

3. 수학예제의 개발

2.1 수학예제가 가져야 할 조건

기존의 다분야통합최적설계 방법론들은 정식화 과정에서 서로 다른 특징이 있다. 이러한 특징은

실제 문제를 푸는 경우의 효율성과 연관이 되어있다. 이러한 방법론의 효율성을 파악하기 위해서는 각 방법론의 비교가 필요하다. 하지만 실제문제를 이용한 방법론의 비교는 시간이 너무 오래 걸리고 문제의 복잡성으로 인해 비교가 어렵다. 실제문제를 모사할 수 있는 수학예제를 통해 비교하는 것이 필요하다. 실제 공학문제를 풀 경우 반응표면법 (response surface method, RSM)과 같은 방법을 이용하여 실제문제를 명시적인(Explicit) 수학식의 형태로 근사화 하여 풀 때가 많다. 따라서 수학예제를 통한 방법론의 비교는 타당하다.

기존의 수학예제를 이용한 연구에서는 다분야통합최적설계가 풀어야 하는 일반적인 형태를 정의하지 못했고, 이를 통해서 상호 비교를 위한 수학예제를 만들지 못했다. 또, 각 영역간의 연성이 작고, 문제의 목적함수와 제한조건의 함수는 상태변수보다는 주로 설계변수의 함수로 이루어져 있는 등 실제의 문제와는 거리가 있는 예제들이었다. 앞서 식 (1)을 통해서 다분야통합최적설계가 풀어야 하는 일반적인 형태를 정의하였다. 이 정의를 바탕으로, 각 방법론의 비교를 위해 수학예제가 지녀야 할 조건을 만족시키는 예제를 제안하고자 한다. 이 때 수학예제가 가져야 할 조건은 다음과 같다.

1. 상태변수를 찾는 해석이 각 영역에 존재하고, 각 영역간에 존재하는 연성을 해결하기 위한 시스템 해석이 존재하여야 한다.
2. 일반적으로 상태변수의 수는 설계변수의 수보다 많아야 한다.
3. 지역설계변수와 공통설계변수가 존재하여야 한다.
4. 일부 방법론의 경우 공통목적함수와 공통제한조건을 고려할 수 없기 때문에 방법론의 비교를 위해서 이를 배제한다.

2.2 문제의 선정

설계를 진행함에 있어서 서로 연성관계가 없는 문제는 드물 것이다. 그러나 각각의 해석 및 설계 영역에 많은 영향을 주는 설계변수는 민감도해석이나 설계자의 직관에 따라 지역설계변수와 공통설계변수 등으로 나눌 수 있다. 이러한 설계변수의 존재 형태에 따라서 어느 다분야통합최적설계 방법론이 더 유리한지를 알아본다.

식 (1)에서 정의한 다분야통합최적설계의 일반적인 형태와 3.1 절에서 정의한 다분야통합최적설계의 방법론 비교를 위한 수학예제의 조건을 만족하는 수학예제를 다음과 같이 제시한다. 이 예제는 많은 수의 설계변수와 상태변수를 갖는 문제를 설계변수의 형태에 따라 다음의 세가지 경우의 예제를 제시한다.

1. 지역설계변수만 존재하는 문제 =>수학예제 1
2. 공통설계변수만 존재하는 문제 =>수학예제 2
3. 지역설계변수와 공통설계변수 모두 존재하는 문제 =>수학예제 3

위와 같은 세가지 예제는 정식화는 같고, 상태변수의 형태만 다르도록 하여 같은 방법론에서도 설계변수의 형태에 따른 효율성을 알 수 있도록 하였다. 목적함수는 지역목적함수의 벡터형태로 단단계 최적화 방법론에서는 두 목적함수의 합의 형태로 다목적함수를 고려할 수 있도록 하였다. 또한 두 영역은 한 쪽 영역에 치우치지 않도록 유사한 형태의 함수 식으로 이루어져 있다. 목적함수와 상태변수는 볼록(convex)한 함수형태를 지니도록 하였다. 이 예제의 정식화는 다음의 식 (2)과 같다

$$
\begin{aligned}
&\text{Find} \quad b_{c\_1}, b_{c\_2}, b_{c\_3} \\
&\text{to minimize} \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \sum (z_{1\_i}^{nc} - 0.5)^2 \\ \sum (z_{2\_i}^{nc} - 0.5)^2 \end{bmatrix} \\
&\text{subject to} \quad g_1 = \sum z_{1\_i}^{nc} \le 0.0 \\
&\qquad\qquad\quad g_2 = \sum z_{2\_i}^{nc} \le 0.0 \\
&z_{1\_i}^{c} = \alpha(b_{1\_j} - 2)^2 + \beta(b_{c\_k} - 1)^2 - \frac{1}{j+1} z_{2\_i}^{c} (b_{1\_l})^{\alpha} (b_{1\_m})^{\beta} \\
&z_{1\_i}^{nc} = \alpha(b_{1\_j} - 2)^2 + \beta(b_{c\_k} - 1)^2 - \frac{1}{j+2} z_{1\_i}^{c} (b_{1\_l'})^{\alpha} (b_{1\_m'})^{\beta} \\
&z_{2\_i}^{c} = \alpha(b_{1\_j} - 3)^2 + \beta(b_{c\_k} - 2)^2 - \frac{1}{j+2} z_{1\_i}^{c} (b_{2\_l})^{\alpha} (b_{2\_m})^{\beta} \\
&z_{2\_i}^{nc} = \alpha(b_{1\_j} - 3)^2 + \beta(b_{c\_k} - 3)^2 - \frac{1}{j+3} z_{1\_i}^{c} (b_{2\_l'})^{\alpha} (b_{2\_m'})^{\beta}
\end{aligned}
\tag{2}
$$

식 (2)의 정식화에서 기호는 식 (1)에서 설명한 바와 같다. $\mathbf{z}_{i\_j}^{c}$ 기호는 는 i 번째 영역의 j 번째 연성변수, $\mathbf{z}_{i\_j}^{nc}$ 기호는 는 i 번째 영역의 j 번째 비연성변수를 나타낸다. b_{i_j} 는 i 번째 영역의 j 번째 지역설계 변수, b_{c_j} 는 j 번째 공통설계변수이다.

Table 1 Index table for the mathematical examples 1

i	j	k	l	l'	m	m'
1	1	1	2	3	2	3
2	2	2	3	1	3	1
3	3	3	1	2	2	1
4	1	2	2	3	3	1
5	2	3	3	1	1	2
6	3	1	1	2	3	2
7	1	3	2	3	1	2
8	2	1	3	1	2	3
9	3	2	1	2	1	3

Table 2 Index table for the mathematical examples 2

Number of example	α	β
Ex. 1 (local only)	1	0
Ex. 2 (common only)	0	1
Ex. 3 (local + common)	1	1

식 (2)의 정식화에 사용 된 계수는 Table 1 과 같다. 각 영역당 9 개의 연성변수와 9 개의 비연성변수를 가지고 있는 문제이기 때문에 계수 i 는 1 부터 9 사이의 수이고, 계수 i 에 따른 다른 계수의 값은 표에 나타난 바와 같다. 앞서 정의한 세가지 수학예제는 계수 α 와 β 에 따라 결정된다. Table 2 에 표시된 계수 α 와 β 의 값은 지역설계변수와 공통설계변수의 존재여부를 나타낸다. ? 은 존재함을 의미하고, ? 는 존재하지 않음을 뜻하게 된다.

3. 결과 및 비교

3 장에서 제시한 예제들을 통해 다분야통합최적설계 방법론들을 비교한다. 함수호출을 바탕으로 하여 정량적으로 비교하고, 문제의 정식화 과정에서 필요한 정보들을 이용하여 정성적으로 비교한다. 이 때 사용하는 함수호출 횟수는 각 영역에서 사용하는 함수호출 횟수를 모두 더하여 총 함수호출 횟수를 구하였다. 이는 최적화 과정에서 각 영역의 해석기가 사용하는 함수호출의 횟수와 각 영역의 평형을 위해서 사용하게 되는 시스템해석의 함수호출 횟수를 모두 더하여 사용한다. 이 때 전역민감도해석(GSE)와 최적민감도해석(OSA)는 해석적(analytic)으로 계산한 것으로 간주하여 총 함수호출 횟수를 구하였다. 최적설계 도구는 공통적으로 DOT 5.4[27]버전을 사용하였다. Table 3~5 은 식 (2) 정식화의 세가지 예제를 각 방법론에 대하여 정식화하고 풀어본 결과를 정리한 것이다. 각 각의 방법론의 특징에 따라 설계변수와 제한조건이 추가되었고, 동일한 방법을 이용하여 총 함수호출 수를 구하였다. 또 실제문제에서는 수치적인 최적화의 시간보다는 정식화의 과정에서 근사화의 과정이나 함수호출에 사용되는 시간이 지배적이라고 간주하였다. 따라서 CPU 사용시간을 통한 비교보다는 함수호출 횟수만을 비교의 대상으로 포함시켰다.

3.1 수학예제 1 의 결과

지역설계변수만 존재 할 경우의 결과는 Table 3 과 같다. 방법론의 성격에 따라 정식화의 과정에서 설계변수와 제한조건이 수가 변화하였다. 또 이에 따라 각 방법론은 보조변수의 개수만큼 등제한조건이 추가되었으며, CO 의 경우에는 하부시스템 레벨의 목적함수 두 개가 시스템레벨의 등제한조건으로 추가되었다. 이 등제한조건을 DOT 를 이용하기 위해서 그 두 배의 부등제한조건을 사용하여 등제한조건을 고려하였다.

Table 3 Results of example 1

Method	DV	IC	EC	Function call	cycles (iterations)
MDF	6	2	0	11192	6
IDF	22	2	18	1378	22
AAO	42	2	36	2098	42
CSSO	10	2	2	7587	29
BLISS	6	2	0	4734	20
CO	21	2	2	165814	138
MDOIS	6	2	0	4599	29

Table 4 Results of example 2

Method	DV	IC	EC	Function call	cycles (iterations)
MDF	3	2	0	4372	3
IDF	19	2	18	742	19
AAO	39	2	36	1850	39
CSSO	7	2	2	5031	36
BLISS	Not applicable				
CO	45	2	2	Fail	
MDOIS	3	2	0	1166	12

Table 5 Results of example 3

Method	DV	IC	EC	Function call	cycles (iterations)
MDF	9	2	0	21324	9
IDF	25	2	18	1878	25
AAO	45	2	36	2978	45
CSSO	13	2	2	15599	44
BLISS	9	2	0	3460	22
CO	51	2	2	Fail	
MDOIS	9	2	0	6085	14

목적함수는 각각의 방법론 모두가 비슷한 곳에서 수렴하였다. 대부분의 방법론은 제한조건을 잘 만족시키면서 최종 목적함수는 비슷한 값으로 수렴하였다. 그러나 CO 의 방법론은 다른 방법론들과는 다르게 더 좋은 목적함수로 수렴하였다. 이는 더 좋은 해를 찾은 것이 아니라 하부시스템레벨에서는 제한조건을 잘 만족하는 것처럼 보였으나 시스템레벨과 하부시스템레벨 사이의 적합조건을 만족시키지 못하여 이와 같은 현상을 보였다. 이는 각 영역간의 평형조건을 만족시키지 못한 것을 의미한다.

함수호출 횟수 면에서는 IDF 가 가장 좋은 성능을 보임을 알 수 있다. 각 영역이 설계의 권한을 갖는 다단계 최적화 방법론의 경우에는 MDOIS 와 BLISS 가 함수호출 횟수 면에서 좋은 성능을 보임을 알 수 있다.

3.2 수학예제 2 의 결과

공통설계만 존재 시의 결과는 Table 4 에 보이는 바와 같다. 이 예제의 경우에도 앞의 예제 1 의 결과와 같이 각각의 방법론들은 제한조건을 잘 만족시키면서 최종 목적함수의 값의 면에서 비슷한 성능을 보였다. 하지만 CSSO 방법론의 경우, 각 설계영역에서는 제한조건을 만족시켰으나, 시스템레벨에서 시스템해석을 수행하면 실제로 제한조건을 만족시키지 못할 수 있다. 이 경우가 그런 경우로써 이와 같은 현상을 방지하기 위해서는 설계자가 각 영역에서 다른 영역의 제한조건 고려 시에 정식화 과정에서 계수들의 적절한 조절이 필요하다.

이 예제의 경우 BLISS 는 이 예제의 경우에 하부시스템레벨을 구성할 수 없기 때문에 정식화하지 않았다. 함수호출 횟수 면에서는 IDF 가 가장 좋은 성능을 보였으며, 각 영역이 설계권한을 갖는 다단계 최적화 방법론의 경우에는 MDOIS 가 가장 좋은 성능을 보였으며, CO 방법론의 경우에는 수렴에 실패하였다

3.3 수학예제 3 의 결과

지역설계변수와 공통설계 모두 존재 시의 결과는 Table 5 와 같다. 이 예제의 경우에 최종 목적함수의 값은 모두 비슷한 곳에서 수렴하여 목적함수 수렴도의 성능은 모두 비슷한 결과를 보였다. 그러나 CSSO 방법론의 경우에는 앞의 두 번째 경우와 같은 이유로 인하여 다른 곳에서 수렴을 하였다. 이 경우에는 제한조건을 과도하게 만족하였다. 함수호출 횟수의 면에서는 IDF 가 가장 좋은 성능을 보였으며, 각 영역이 설계권한을 갖는 다단계 최적화 방법론의 경우에는 BLISS 가 가장 좋은 성능을 보였다. CO 방법론의 경우에는 두 번째 예제에서와 같이 수렴에 실패하였다.

세 가지 경우에 대하여 종합해 보면 모두 IDF 방법론이 함수호출 횟수 면에서 가장 좋은 성능을 보임을 알 수 있고, 다단계 최적화 방법론에 대해서는 MDOIS 와 BLISS 가 함수호출 횟수 면에서 좋은 성능을 보임을 알 수 있었다. CO 방법론의 경우에는 많은 함수호출 횟수를 보이거나 수렴성이 나쁨을 알 수 있다.

3.4 종합

각 방법론을 정식화 하는 과정에서 추가적으로 요구되는 정보들이 있다. 이 때의 정보란 식 (1)의 정식화를 각 방법론에 따라 정식화 하고 푸는 과정에서 새롭게 추가되는 설계변수나 제한조건 등과 전역민감도나 최적민감도 등의 민감도정보 등이다. 이러한 정보들은 실제 문제를 풀 때에 있어서의 어려움으로 작용하게 된다. 설계변수나 등제한조건 등의 추 가는 최적화 과정에서 최적해를 찾는 데에 있어서의 어려움으로 작용할 수 있으며, 전역민감도 방 정식의 구성이나 최적민감도 정보를 얻을 경우에는 최적화를 한 번 더 수행해야 할 수도 있는 등 실제 문제를 풀 경우에는 이 과정을 위한 함수호출이 추가될 수도 있다.

Fig.1 은 함수호출 결과와 각 방법론을 정식화하는 과정에서의 복잡성의 관계를 도표로 나타낸 것이다. 세로축은 함수호출 횟수로써 앞의 세 가지 경우의 예제를 바탕으로 하여 위로 갈수록 많은 함수호출 횟수를 사용하는 방법론을 배치하였다. 가로축은 앞서 설명한 정식화의 과정에서 추가적으로 요구되는 정보량에 관한 것이다. 이 두 가지를 하나의 도표에 함께 표시함으로 써 정식화의 간편함과 함께 함수호출 횟수가 적은 방법론을 찾기 위함이다.

Fig.1 의 도표에서 원점과 가까운 방법론이 함수호출 횟수도 적고, 정식화하여 풀기도 쉬운 방법론이라고 할 수 있다. 이 도표를 통해서 비교해 보았을 때, IDF 방법론이 함수호출 횟수 면에서나 정식화의 과정에서 추가되는 정보량 면에서 가장 좋은 방법론이라고 할 수 있다. 하지만 IDF 방법론의 경우에는 각 영역이 설계의 권한을 갖지 못하는 단단계 최적화 방법론이다. 이와 같은 단점을 보완하여 각 영역이 설계의 권한을 갖게 되는 다단계 최적화 방법론 중에는 MDOIS 와 BLISS 가 함수호출 면에서 비슷한 성능을 보였으 나 정식화의 간편성으로 인해 MDOIS 방법론이 더 사용하기 편한 방법론이라고 할 수 있다.

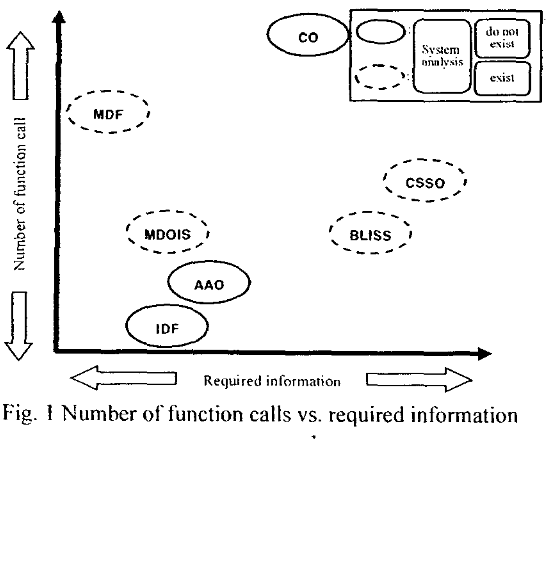

Fig. 1 Number of function calls vs. required information

5. 결론

다분야 통합최적설계가 풀어야 하는 일반적인 문제의 형태를 정의하고, 기존의 다분야통합최적설계 방법론을 비교를 위해 수학예제가 가져야 할 조건을 제시하였다. 이를 바탕으로 하여 각각의 방법론을 비교할 수 있는 설계변수의 존재 형태에 따른 세 가지 수학예제를 제시하였다. 이 예제들을 기존의 각 방법론에 대하여 정식화 하고 풀어본 결과,. 다음과 같은 결론을 얻을 수 있었다.

1. 시스템해석이 가능한 경우에는 MDOIS 의 사용을 추천한다.
2. 시스템해석이 어려운 경우에는 IDF 의 사용을 추천한다.
3. CO 의 경우에는 많은 함수호출 횟수와 낮은 수렴성으로 인해 다분야통합최적설계 방법론으로는 부적합하다고 할 수 있다.
4. 최근에 연구가 이루어 지고 있는 BLISS 방법론의 경우에는 고가의 전역민감도와 최적민감도 정보를 사용해야 한다.

참고문헌

1. 이상일, 2005, 수학예제를 이용한 다분야통합최적설계 방법론의 비교,? 한양대학교 대학원 석사학위 청구논문.
2. 신문균, 박경진, 2004, 독립적 하부 시스템에 의한 다분야 통합 최적설계,? 대한기계학회논문집 A 권 Vol. 28, No. 2, pp. 191-117.
3. 박창규, 2001, 자동미분을 이용한 민감도기반 분리시스템동시최적화기법의 개선, 대한기계학회논문집 A 권 Vol. 25, No. 2, pp. 182-191.
4. 이현섭, 2004, 개선된 이점대각이차근사화 기법을 사용한 순차적 IDF 기법, 한양대학교 대학원 석사학위청구논문.
5. Balling RJ and Sobieszcznski-Sobieski J, 1996, 밢ptimization of Coupled Systems: A Critical Overview of Approach,? AIAA Journal Vol. 34, No. 1, pp. 6-17.
6. Balling RJ and Wilkinson CA, 1996, 밚xecution Of Multidisciplinary Design Optimization Approaches On Common Test Problems,? AIAA Paper AIAA-96-4033.
7. Barthelemy JF and Sobieszczanski-Sobieski J, 1983, 밢ptimum Sensitivity Derivatives of Objectives Functions in Nonlinear Programming,? AIAA Journal Vol. 21, No. 6, pp. 913~915.
8. Braun RD, 1996, 밃ollaborative Optimization: An Architecture for Large-Scale Distributed Design,? Ph.D. Dissertation, Stanford University.
9. Chen S, Zhang F and Khalid M, 2002, 밚valuation Of Three Decomposition MDO Algorithms,? ICAS2002CONGRESS, pp.113.1-10.
10. Cramer EJ, Dennis JE Jr., Frank PD, Lewis RM and Shubin GR, 1993, 밣roblem Formulation for Multidisciplinary Optimization,? Center for Research on Parallel Computation Rice Univ., CRPC-TR93334.
11. Haftka RT, 1985, �்imultaneous Analysis and Design,? AIAA Journal Vol. 23, No. 7, pp. 1099-1103.
12. Hulme KF and Bloebaum CL, 1998, 밃 Comparison Of Solution Strategies For Simulation-based Multidisciplinary Design Optimization,? AIAA Paper AIAA-98-4977.
13. Hulme KF Bloebaum CL, 2000, 밃 Simulation-based Comparison Of Multidisciplinary Design Optimization Solution Strategies Using CASCADE,? Structure and Multidisciplinary Optimization Vol. 19, pp. 17-35.
14. Hajela P, Bloebaum CL and Sobieszcznski-Sobieski J, 1990, 밃pplication of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis,? Journal of Aircraft Vol. 27, No.12, pp. 1002-1010.
15. Kodiyalam S and Sobieszczanski-Sobieski J, 2001, " Multidisciplinary Design Optimization - some formal methods, framework requirements, and application to vehicle design,? International journal of vehicle design Vol. 25, No. 1, pp. 3-22
16. Renaud JE and Gabriele GA, 1994, 밃pproximation in Nonhierarchic System Optimization,? AIAA Journal, Vol. 32, No. 1, pp. 198-205.
17. Salas AO and Townsend C, 1998, 밊ramework Requirements For MDO Application Development,? AIAA Paper AIAA-98-4740.
18. Sobieszczanski-Sobieski J, 1982, 밃 Linear Decomposition Method for Large Optimization Problems-Blueprint for Development,? NASA TM 83248.
19. Sobieszczanski-Sobieski J, Barthelemy JF and Riley KM, 1982, 밶ensitivity of Optimum Solutions to Problem Parameters,? AIAA Journal Vol. 20, No. 9, pp. 1291-1299.
20. Sobieszczanski-Sobieski J, 1988, 밢ptimization by Decomposition: A Step from Hierarchic to Non Hierarchic Systems,? NASA CP 3031.
21. Sobieszczanski-Sobieski J, 1990, 밢n the Sensitivity of Complex, Internally Coupled Systems,? AIAA Journal Vol. 28, No. 1, pp. 153-160.
22. Sobieszcznski-Sobieski J and Haftka RT, 1996, 밠ultidisciplinary Aerospace Design Optimization:Survey of Recent Developments,? AIAA Paper AIAA-96-0711.
23. Sobieszczanski-Sobieski J, Agte J and Sandusky R Jr. 1998, 밆i-Level Integrated System Synthesis,? Proceedings of AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization AIAA-98-4916.
24. Sobieszcznski-Sobieski J, D. Altus T, Phillips M and Sandusky R, 2002, 밆i-level System Synthesis (BLISS) For Concurrent And Distributed Processing,? AIAA Paper AIAA-2002-5409
25. Tappeta RV, Nagendra S, Renaud JE and Badhrinath, 1998, 밃oncurrent Sub-Space Optimization(CSSO) MDO Algorithms in iSIGHT:Validation and Testing,?
26. Tappeta RV and Renaud JE, 1997, 밠ultiobjective Collaborative Optimization,? Journal of Mechanical Design, Vol. 119, pp.403-411.
27. DOT User뭩Manual Version 5.4, 2004, Vanderplaats Research & Development, Inc