

비균일 수렴허용오차 방법을 이용한 분지한계법 개선에 관한 연구

정상진* · Xi Chen** · 최경현*** · 최동훈****†

* 한양대학교 대학원 기계공학과, ** Eaton Corporation, *** 한양대학교 산업공학과,
**** 한양대학교 최적설계신기술연구센터

A Non-Uniform Convergence Tolerance Scheme for Enhancing the Branch-and-Bound Method

Sangjin Jung*, Xi Chen**, Gyunghyun Choi*** and Dong-Hoon Choi****†

* Graduate School of Mechanical Engineering, Hanyang Univ.,

** Eaton Corporation,

*** Dept. of Industrial Engineering, Hanyang Univ.,

**** The Center of Innovative Design Optimization Technology (iDOT), Hanyang Univ.

(Received January 27, 2011; Revised January 9, 2012; Accepted January 17, 2012)

Key Words : Discrete Optimization(이산최적화), Mixed-Discrete Nonlinear Programming(혼합이산비선형계획법), Branch-and-Bound Method(분지한계법), Convergence Tolerance(수렴허용오차), Branching Order(분지순서)

초록: 혼합이산비선형계획법(mixed-discrete nonlinear programming) 문제의 최적화를 위한 대표적인 기법 중에 하나인 분지한계법(branch-and-bound method)은 다른 기법에 비해 강건하지만 분지한계법 내부의 각 노드마다 연속 최적화를 수행해야 하기 때문에 많은 함수 계산이 요구되는 것으로 알려져 있다. 이러한 분지한계법의 단점을 극복하기 위하여 크게 두 가지 연구를 수행하였다. 먼저, 분지한계법의 각 노드마다 동일한 수렴허용오차를 설정해주던 기존의 방법을 대체할 수 있는 비균일 수렴허용오차 방법을 제안하였다. 또한 분지한계법에 적용할 수 있는 5 가지 분지순서 방법 중에서 분지한계법의 성능을 가장 극대화할 수 있는 분지순서 방법을 제시하였다. 수렴허용오차 방법과 분지순서 방법들을 각각 선택하여 분지한계법에 적용한 후 7 개의 수학예제와 4 개의 공학예제에 대하여 테스트를 수행한 결과, 제안된 비균일 수렴허용오차 방법과 5 가지 분지순서 방법 중 최소간격차이법을 분지한계법에 함께 적용할 경우 분지한계법의 성능이 가장 극대화 됨을 확인할 수 있었다.

Abstract: In order to improve the efficiency of the branch-and-bound method for mixed-discrete nonlinear programming, a non-uniform convergence tolerance scheme is proposed for the continuous subproblem optimizations. The suggested scheme assigns the convergence tolerances for each continuous subproblem optimization according to the maximum constraint violation obtained from the first iteration of each subproblem optimization in order to reduce the total number of function evaluations needed to reach the discrete optimal solution. The proposed tolerance scheme is integrated with five branching order options. The comparative performance test results using the ten combinations of the five branching orders and two convergence tolerance schemes show that the suggested non-uniform convergence tolerance scheme is obviously superior to the uniform one. The results also show that the branching order option using the minimum clearance difference method performed best among the five branching order options. Therefore, we recommend using the “minimum clearance difference method” for branching and the “non-uniform convergence tolerance scheme” for solving discrete optimization problems.

1. 서론

실제적인 공학 제품의 설계문제에는 연속설계변수(continuous design variable) 뿐만 아니라 규격화된 값들 중에서 설계변수 값이 선택되어야 하는 이산

설계변수(discrete design variable)가 종종 존재한다. 이처럼 연속설계변수 및 이산설계변수가 함께 존재하거나 또는 이산설계변수만 존재하는 설계문제의 최적해를 얻기 위하여 혼합이산비선형계획법(mixed-discrete nonlinear programming, MDNLP) 이 개발되었다.

대표적인 MDNLP 기법에는 유전 알고리즘(genetic algorithm, GA),^(1,2) 입자군집최적화(particle swarm optimization,

† Corresponding Author, dhchoi@hanyang.ac.kr

© 2012 The Korean Society of Mechanical Engineers

PSO),^(3,4) 벌칙함수 기법(penalty function approach),⁽⁵⁾ 순차적 선형근사 기법(sequential linearization approach),⁽⁶⁻⁸⁾ 분지한계법(branch-and-bound method, BBM)⁽⁹⁻¹⁶⁾ 등이 있다. 먼저 GA 와 PSO 는 함수의 민감도 정보가 필요 없기 때문에 설계문제의 특성을 잘 모르더라도 쉽게 적용이 가능한 기법으로, 설계 영역을 넓게 탐색하기 때문에 전역 최적해(global optimum)를 찾을 확률이 높다. 그러나 다른 MDNLP 기법에 비해 매우 많은 함수 계산이 요구되고 수렴속도가 느리기 때문에, 해석 시간이 오래 걸리는 실제적인 설계문제의 최적화에는 적합하지 않다. 벌칙함수 기법은 이산설계변수에 대한 벌칙함수를 구성한 후 최적화를 수행하는 기법이다. 이 기법은 벌칙매개변수(penalty parameter)를 결정하는 일이 매우 까다롭기 때문에, 만약 벌칙매개변수를 잘못 결정할 경우 이산최적해(discrete optimum)를 얻지 못하는 경우가 종종 발생한다.⁽⁴⁾ 순차적 선형근사 기법은 설계문제를 선형문제로 근사화한 후 근사최적해를 구하는 과정을 순차적으로 반복하는 기법이다. 이 기법은 다른 MDNLP 기법에 비해 효율성이 뛰어나지만, 설계문제의 비선형성이 심할 경우 이산최적해를 잘 얻지 못하는 것으로 알려져 있다.⁽¹⁷⁾ 분지한계법은 주어진 설계영역을 더 좁은 범위의 영역으로 분할한 각 노드(node)에서 비선형계획법(nonlinear programming, NLP) 알고리즘을 이용하여 최적화를 수행하는 기법이다. 분지한계법은 볼록(convex) 설계문제에 대해 수렴성이 보장되며 다른 기법에 비해 강건하기 때문에^(17,18) 가장 널리 이용되고 있는 기법 중에 하나이다. 하지만 분지한계법은 분지 및 탐색 전략에 따라 알고리즘의 성능이 크게 달라지며, 각 노드마다 연속최적화를 수행해야 하기 때문에 많은 함수 계산이 요구되는 단점이 존재한다.

본 연구에서는 분지한계법의 효율성을 향상시키기 위하여 크게 두 가지 방법을 제안하였다. 먼저, 분지한계법의 각 노드마다 동일한 수렴허용오차(convergence tolerance)를 설정해주던 기존의 방법을 대신할 수 있는 비균일 수렴허용오차 방법(non-uniform convergence tolerance scheme)을 제안한다. 또한 기존에 알려져 있는 다양한 분지순서 방법(branching order method)간의 비교를 통해 분지한계법의 성능을 가장 극대화시키는 분지순서 방법을 제시하고자 한다.

본 논문의 2 장에서는 분지한계법의 소개, 제안된 비균일 수렴허용오차 방법, 분지순서 방법에 대해서 차례로 기술하고자 한다. 3 장에서는 7 개의

수학예제와 4 개의 공학예제를 이용하여 제안된 방법의 성능을 검증하고자 한다. 그리고 4 장에서는 결론을 맺을 것이다.

2. 분지한계법의 개선 방법

2.1 분지한계법

MDNLP 문제는 일반적으로 식 (1)-(5)와 같이 정식화 할 수 있다.

$$\text{Find } \mathbf{x} \quad (1)$$

$$\text{to minimize } f(\mathbf{x}) \quad (2)$$

$$\text{subject to}$$

$$g_j(\mathbf{x}) \leq 0, \quad j=1, \dots, m \quad (3)$$

$$x_i \in D_i, D_i = (d_{i1}, d_{i2}, \dots, d_{ip}, \dots, d_{iq_i}), \quad i=1, \dots, n_d \quad (4)$$

$$x_i^L \leq x_i \leq x_i^U, \quad i = (n_d + 1), \dots, n \quad (5)$$

여기서 \mathbf{x} 는 연속설계변수와 이산설계변수들을 포함하는 설계변수 벡터이며, f 는 목적함수, g_j 는 j 번째 구속조건을 의미한다. x_i^L 과 x_i^U 는 i 번째 연속설계변수에 대한 하한치와 상한치를 각각 나타낸다. m 과 n 은 각각 구속조건수의 개수와 설계변수의 개수를 나타내며 n_d 는 이산설계변수의 개수를 의미한다. D_i 는 i 번째 설계변수가 가질 수 있는 이산 값들의 집합, q_i 는 i 번째 설계변수가 가질 수 있는 이산 값의 개수이며, d_{ip} 는 i 번째 설계변수의 p 번째 이산 값을 나타낸다.

분지한계법은 이러한 MDNLP 문제를 최적화할 수 있는 대표적인 기법 중에 하나이다. 1960 년 선형계획법(linear programming)을 위해 현재의 분지한계법과 유사한 형태의 알고리즘이 Land 와 Doig⁽⁹⁾에 의해 처음 제안되었고, 이후 비선형 문제를 풀 수 있는 분지한계법이 1965 년 Dakin⁽¹⁰⁾에 의해 개발되었다. 이후 분지한계법의 성능을 개선하기 위한 연구가 Gupta,⁽¹¹⁾ Hajela,⁽¹²⁾ Tseng,⁽¹⁵⁾ Leyffer⁽¹⁶⁾ 등에 의해 수행되어져 왔다. 분지한계법은 분지(branching) 과정을 통해 주어진 설계영역을 더 작은 설계영역으로 분할하고, 분할된 설계영역에서 현재까지 찾은 해보다 더 개선된 해를 찾을 수 없다면 그 설계영역을 삭제하여 탐색 공간을 줄여나가는 방법이다. 분지한계법은 나무(tree) 구조와 같이 현재 노드 밑에 자식노드들이 계속 생성되는 방식으로 진행된다. 분지한계법의 자세한 절차는 아래와 같다.

Step 1: NLP 기법을 이용하여 첫 번째 노드의 최적화를 수행한다.

Step 2: 첫 번째 노드에서 이산최적해를 얻었거나 해를 얻지 못했을 경우, 분지한계법을 종료한다.

다. 그렇지 않으면, Step 3 으로 진행한다.

- Step 3: 분지순서 방법을 이용하여 상하한 범위를 새로 설정해야 할 설계변수를 선정하고, 설계변수의 범위가 새로 설정된 자식노드 2 개를 생성한다.
- Step 4: 두 자식노드에서 각각 최적화를 수행한다.
- Step 5: 두 자식노드에서 얻어진 해를 각각 체크하여 두 노드의 상태를 결정한다.
- Step 6: 노드탐색 방법을 이용하여 분지를 수행할 노드를 새로 선정하고, Step 3 으로 진행한다. 만약 더 이상 분지를 수행해야 할 노드가 없다면, 분지한계법을 종료한다.

Step 5 에서 각 노드에서는 최적화를 통해 얻어진 해를 체크하여 그 노드에서 분지를 더 진행할 것인지 아니면 멈출 것인지 결정해야 한다. 노드의 상태는 크게 세 가지로 분류된다. 첫째, 노드에서 최적화를 수행하여 얻은 해가 기존의 후보이산최적해(incumbent)보다 목적함수 값이 더 좋으나 이산해(discrete solution)는 아닐 경우, 이 노드에서 분지를 수행한다. 이 노드들은 분지를 위해 활성화(active)되어 있으며, 이러한 노드에서 분지를 수행하면 기존의 후보이산최적해보다 더 좋은 이산최적해를 자식노드에서 얻을 가능성이 높아진다. 둘째, 노드에서 최적화를 수행하여 얻은 이산해가 기존의 후보이산최적해보다 목적함수 값이 좋을 경우, 이 해는 새로운 후보이산최적해가 된다. 이 노드에서는 더 이상 분지가 진행되지 않는다. 셋째, 노드에서 최적화를 수행하여 얻어진 해가 후보이산최적해보다 목적함수 값이 나쁘거나 또는 해를 얻지 못했을 경우가 존재한다. 이러한 노드에서는 더 이상 분지가 진행되지 않는다.

Step 3 에서는 분지순서 방법을 이용하여 상하한 범위를 새로 설정할 설계변수를 선정하고, 설계변수 범위가 새로 설정된 자식노드 2 개를 생성해야 한다. 이때 어떤 설계변수를 선정하느냐에 따라 분지한계법의 효율성이 크게 달라진다. 본 연구에서는 다양한 예제를 통해 분지한계법의 성능을 가장 극대화할 수 있는 분지순서 방법이 무엇인지 분석해 볼 것이다.

한편, Step 4 에서 볼 수 있듯이 분지한계법에서는 각 노드마다 최적화를 수행하기 때문에 다른 MDNLP 기법에 비해 효율성이 떨어지는 단점이 존재한다. 본 연구에서는 각 노드마다 수렴조건을 다르게 설정하여 분지한계법의 전체적인 효율성을 높이는 비균일 수렴허용오차 방법을 제안하였다. 2.2 절에서는 제안된 비균일 수렴허용오차 방법에 대해 자세히 설명하고자 하며, 2.3 절에서는 대표적

인 5 가지 분지순서 방법에 대해 알아볼 것이다.

2.2 비균일 수렴허용오차 방법

분지한계법의 각 노드마다 NLP 기법을 이용하여 연속최적화(continuous optimization)를 수행하므로, NLP 기법의 성능은 분지한계법의 전체적인 효율성 및 정확성에 큰 영향을 미친다. NLP 기법 중에서 강건하고 효율적인 방법으로 알려진 일반화된 환산경사도법(generalized reduced gradient method)⁽¹¹⁾이나 순차이차계획법(sequential quadratic programming, SQP)^(15,16) 등이 각 노드의 최적화를 위해 주로 이용되어 왔다. 본 연구에서는 Jung 등⁽¹⁹⁾이 개발한 SQP 를 이용하여 각 노드의 최적화를 수행하였다.

각 노드에서 최적화를 수행할 때 가장 어려운 점 중에 하나는 NLP 기법의 종료 시점을 판단하는 일이다. 즉, NLP 기법의 수렴허용오차를 어떻게 설정해두느냐에 따라 최적화의 효율성 및 정확성에 큰 차이가 발생할 수 있다. 또한 각 노드에서의 최적화의 효율성과 정확성은 분지한계법 전체의 효율성 및 정확성에 큰 영향을 미친다. 그러나 기존의 분지한계법에서는 이러한 특성을 고려하지 않고 모든 노드에 동일한 수렴허용오차가 적용되었기 때문에, 불필요한 함수 계산으로 인해 분지한계법의 효율성이 나빠지는 경우가 많았다. 이를 개선하기 위하여 본 연구에서는 새로운 비균일 수렴허용오차 방법을 제안하고자 한다.

본 연구에서는 크게 두 가지 수렴조건에 대한 수렴허용오차를 각 노드마다 다르게 설정해 주었다. 첫 번째 수렴조건은 해의 구속조건 위배 여부에 관한 것이다. 식 (3)에서 볼 수 있듯이 구속조건이 0 보다 크면 수학적으로 위배되었음을 의미한다. 그러나 컴퓨터를 이용한 계산 시 항상 내부적인 오차가 존재하며, 시뮬레이션 모델에서 계산된 성능치는 실제 모델의 성능치에 대한 근사값이다. 따라서 구속조건 위배 여부를 판단할 때 작은 허용오차를 부여한다. 이 수렴허용오차를 t_c 로 정의하고자 하며, t_c 는 $3.0e-3$ (t_{c0})으로 주로 설정된다.^(20,21) 즉, 구속조건은 식 (6)과 같이 정의되며, 일반적으로 g_j 는 알고리즘 내에서 g_j 의 초기값 등으로 정규화(normalization)된다.

$$g_j(\mathbf{x}) \leq t_c, \quad j=1, \dots, m \quad (6)$$

두 번째 수렴조건은 이전 설계점에서의 목적함수 값과 현재 설계점에서의 목적함수 값의 상대적인 차이가 수렴허용오차 t_{RF} 보다 작거나 같아야 한다는 것이다. t_{RF} 는 주로 $1.0e-3$ (t_{RF0})으로 설정된

다.^(20,21) 이를 수식으로 나타내면 식 (7)과 같다. 만약 이 수렴조건을 만족할 경우 최적해가 얻어지고 SQP는 종료된다.

$$\frac{|f(\mathbf{x}^k) - f(\mathbf{x}^{k-1})|}{|f(\mathbf{x}^{k-1})|} \leq t_{RF} \quad (7)$$

여기서 k 는 SQP 알고리즘 내부의 반복횟수 (iteration number)를 나타낸다.

기존의 분지한계법에서는 모든 노드에 동일하게 $t_C = t_{C0}$, $t_{RF} = t_{RF0}$ 로 설정해주는 균일 수렴허용오차 방법이 이용되었다. 그러나 모든 노드에 동일한 수렴허용오차를 적용할 경우, 각 노드마다 설계변수의 범위만 단지 다른 유사한 최적화 문제를 풀기 위해 너무 큰 수치적 부담이 요구된다.

본 연구에서는 분지한계법의 효율성을 높이기 위해서 Table 1과 같이 각 노드의 수렴허용오차를 다르게 설정해주는 비균일 수렴허용오차 방법을

제안한다. Table 1에서 볼 수 있듯이 이 방법에서는 g_{max} 의 크기에 따라 t_C , t_{RF} 값이 상대적으로 조정된다. g_{max} 는 SQP의 첫 번째 반복회(iteration)가 끝난 후 업데이트된 설계점에서의 모든 구속조건 값 중에서 가장 큰 값을 나타낸다. 이를 수식으로 나타내면 다음과 같다.

$$g_{max} = \max \{g_1^1, g_2^1, \dots, g_j^1, \dots, g_m^1\} \quad (8)$$

Table 1 t_C and t_{RF} values in the non-uniform convergence tolerance scheme

g_{max}	t_C	t_{RF}
$1 \leq g_{max}$	α	$\alpha \frac{t_{RF0}}{t_{C0}}$
$\frac{t_{C0}}{\alpha} \leq g_{max} < 1$	αg_{max}	$\alpha g_{max} \frac{t_{RF0}}{t_{C0}}$
$g_{max} < \frac{t_{C0}}{\alpha}$	t_{C0}	t_{RF0}

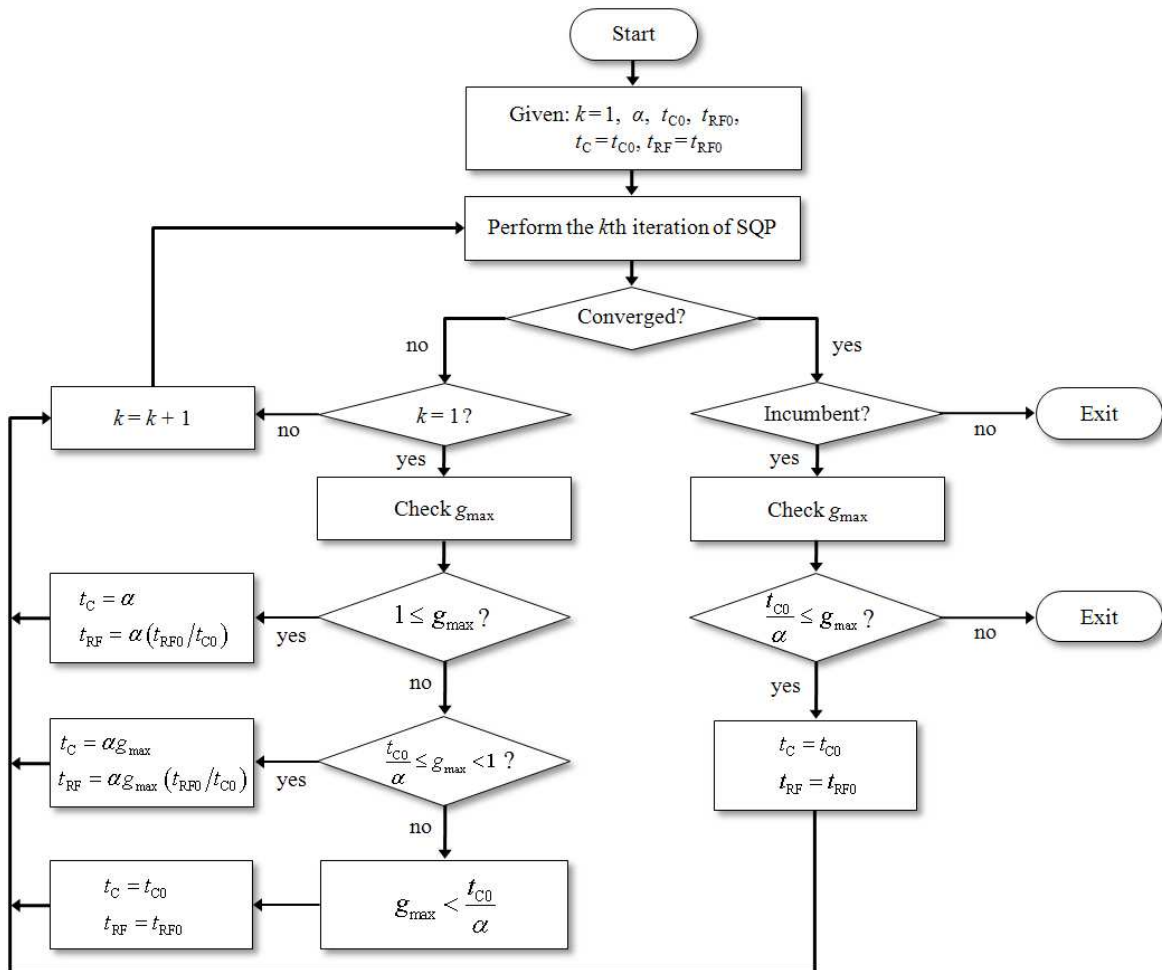


Fig. 1 Non-uniform convergence tolerance scheme for each node

여기서 g_j^1 은 SQP 의 첫 번째 반복회가 끝난 후 얻어진 현재 설계점에서 j 번째 구속조건의 정규화된 값이다.

Fig. 1 은 분지한계법의 한 노드에 비균일 수렴허용오차 방법이 적용된 최적화 절차를 나타낸다. 먼저 SQP 의 첫 번째 반복회를 수행한 후 g_{\max} 값을 체크한다. g_{\max} 가 1 보다 클 경우 t_C, t_{RF} 를 각각 $t_C = \alpha, t_{RF} = \alpha(t_{RF0}/t_{C0})$ 로 설정한다. g_{\max} 가 t_{C0}/α 보다 같거나 크고 1 보다 작을 경우, t_C, t_{RF} 를 각각 $t_C = \alpha g_{\max}, t_{RF} = \alpha g_{\max}(t_{RF0}/t_{C0})$ 로 조정해준다. 만약 g_{\max} 가 t_{C0}/α 보다 작다면, t_C, t_{RF} 는 초기에 설정해 준 t_{C0}, t_{RF0} 로 그대로 유지한다. 본 연구에서 α 는 0.3 으로 설정하였다. 임의의 값인 0.3 은 여러 시험문제를 통해 얻은 학습적인 값이다. t_C, t_{RF} 가 결정된 후 반복회수 k 를 업데이트하고 SQP 의 다음 반복회를 수행한다. 단, 앞서 언급한 절차만 이용할 경우 분지한계법에서 부정확한 이산최적해가 얻어질 가능성이 존재한다. 이를 방지하기 위하여 아래와 같은 조건을 추가하였다. Fig. 1 에서 볼 수 있듯이 만약 새로운 후보이산최적해가 얻어졌을 때 g_{\max} 가 t_{C0}/α 보다 크거나 같을 경우, t_C, t_{RF} 를 각각 $t_C = t_{C0}, t_{RF} = t_{RF0}$ 로 재설정하여 SQP 가 종료된 시점부터 다시 알고리즘을 진행한다. 이 방법을 추가함으로써 기존의 균일 수렴허용오차 방법을 이용한 분지한계법과 동일한 정확성의 후보이산최적해를 노드에서 얻을 수 있었다.

2.2 분지순서 방법

부모노드에서 이산최적해를 얻지 못하고 분지를 할 때, 부모노드에서 이산 값을 가지지 못했던 이산설계변수의 상하한 범위가 각각 재조정되면서 자식노드 두 개가 생성된다. 이때 부모노드에서 이산 값을 얻지 못한 이산설계변수가 여러 개일 경우, 범위를 재조정할 한 개의 이산설계변수만을 택해야만 한다. 이때 어떤 설계변수를 택하느냐에 따라 분지한계법의 효율성이 크게 달라진다. Tseng⁽¹⁵⁾은 상하한 범위를 재조정할 이산설계변수를 결정할 수 있는 5 가지의 분지순서(branching order) 방법을 제안하였다. 그러나 5 가지 방법 중 어떤 방법이 분지한계법의 성능을 가장 극대화시키는지에 대한 연구가 부족한 상태이다. 3 장에서는 다양한 예제에 Tseng 이 제안한 분지순서 방법들을 적용하여 이 중 가장 효과적인 방법이 무엇인지 분석해 보고자 한다. 5 가지의 분지순서 방법은 다음과 같다.

• 최소간격법(minimum clearance method)

이 방법은 현재 설계변수와 가장 가까운 이산값과 현재 설계변수 값 사이의 거리가 가장 가까운 설계변수를 분지를 할 설계변수로 택하는 방법이다. 이를 수식으로 나타내면 아래와 같다.

$$C_i^{\min} = \min(x_i^C - x_i^{D-}, x_i^{D+} - x_i^C) \quad (9)$$

여기서 x_i^C 는 i 번째 설계변수의 현재 값, x_i^{D-} 는 x_i^C 보다 작으면서 가장 가까운 이산 값, x_i^{D+} 는 x_i^C 보다 크면서 가장 가까운 이산 값을 각각 나타낸다. 그리고 C_i^{\min} 은 각 설계변수의 최소간격(minimum clearance)를 나타내며, 최소간격법에서는 C_i^{\min} 이 가장 작은 이산 설계변수를 분지를 수행할 설계변수로 선정한다.

• 최대간격법(maximum clearance method)

x_i^C 와 x_i^{D-} 사이의 간격과 x_i^C 와 x_i^{D+} 사이의 간격 중에서 큰 값을 최대간격(maximum clearance) C_i^{\max} 라 하며, 아래와 같이 계산된다.

$$C_i^{\max} = \max(x_i^C - x_i^{D-}, x_i^{D+} - x_i^C) \quad (10)$$

이 방법에서는 C_i^{\max} 가 가장 큰 이산설계변수를 분지를 수행할 설계변수로 선정한다.

• 최소간격차이법(minimum clearance difference method)

이 방법은 앞에서 구한 C_i^{\max} 와 C_i^{\min} 의 차이가 가장 작은 설계변수를 분지를 수행할 설계변수로 택하는 방법이다. C_i^{\max} 와 C_i^{\min} 의 차이를 수식으로 나타내면 아래와 같다.

$$CD_i = C_i^{\max} - C_i^{\min} \quad (11)$$

• 최대간격차이법(maximum clearance difference method)

이 방법은 식 (11)의 CD_i 가 가장 큰 설계변수를 분지를 수행할 설계변수로 택하는 방법이다.

• 최대비용차이법(maximum cost difference method)

이 방법에서는 설계변수의 현재 값에서 x_i^C 만 x_i^{D-} 로 바꾼 후 계산한 목적함수 값 f_i^L 과 x_i^C 를 x_i^{D+} 로 바꾼 후 계산한 목적함수 값 f_i^U 의 차이가 가장 큰 설계변수를 분지를 수행할 설계변수로 택한다. 이를 수식으로 나타내면 아래와 같다.

$$f_i^L = f(x_1^C, x_2^C, \dots, x_{i-1}^C, x_i^{D-}, x_{i+1}^C, \dots, x_n^C) \quad (12)$$

$$f_i^U = f(x_1^C, x_2^C, \dots, x_{i-1}^C, x_i^{D+}, x_{i+1}^C, \dots, x_n^C) \quad (13)$$

$$CostD_i = |f_i^L - f_i^U| \quad (14)$$

그러나, 분지를 수행할 각 노드마다 $CostD_i$ 를 계산하기 위해서는 너무 많은 함수 계산이 추가로 요구된다. 따라서 최대비용차이법에서는 식 (14) 대신 목적함수 f 를 선형 근사화하여 $CostD_i$ 를 계산하는 식 (15) 를 이용한다.

Table 2 Test problems

No.	Problem type	n	n_d	Design variable type	Size of discrete sets	Ref.
1	Mathematical problem	2	2	x_1, x_2 : discrete (integer)	4*7	(22)
2	Mathematical problem	6	6	x_1, x_2, x_4, x_5, x_6 : discrete (integer)	5*4*3*9*3*3	(23)
2	Mathematical problem	3	3	x_1, x_2, x_3 : discrete (integer)	11*11*11	(6)
4	Mathematical problem	2	2	x_1, x_2 : discrete	6*5	(17)
5	Mathematical problem	2	2	x_1, x_2 : discrete	6*5	(24)
6	Mathematical problem	2	1	x_1 : discrete (integer) x_2 : continuous	5	(7)
7	Mathematical problem	7	3	x_1, x_2, x_3 : discrete (integer) x_4, x_5, x_6, x_7 : continuous	5*6*6	(25)
8	Three bar truss design	3	3	x_1, x_2, x_3 : discrete	34*34*34	(26)
9	Spring design	3	3	x_1, x_2 : discrete x_3 : discrete (integer)	16*17*14	(27)
10	Gear reducer design	7	7	$x_1, x_2, x_4, x_5, x_6, x_7$: discrete x_3 : discrete (integer)	11*11*12*11*11*101*51	(28)
11	Pressure vessel design	4	2	x_1, x_2 : discrete x_3, x_4 : continuous	114*114	(29)

$$CostD_i = \left| \frac{\partial f(x_i^C)}{\partial x_i} (x_i^{D+} - x_i^{D-}) \right| \quad (15)$$

즉, 식 (15)의 $CostD_i$ 값이 가장 큰 설계변수를 분지를 수행할 설계변수로 선정한다.

지금까지 설명한 5 가지 분지순서 방법들을 도식화하면 Fig. 2 와 같다. Fig. 2 에서 볼 수 있듯이 선택된 방법에 따라 분지를 수행할 설계변수가 달라지는 것을 알 수 있다. 3 장에서는 5 가지 분지순서 방법 중 분지한계법의 성능을 가장 극대화시키는 방법이 무엇인지 알아보려고 한다.

3. 테스트 결과

본 연구에서는 제안된 방법의 성능을 검증하기 위하여 설계변수 및 함수의 형태가 각기 다른 7 개의 수학예제와 4 개의 공학예제를 이용하였다. Table 2 에는 각 예제의 종류, 설계변수 개수, 이산 설계변수 개수, 설계변수의 종류, 이산 집합의 크기, 참고문헌이 정리되어 있으며, Table 3 에는 설계변수의 범위 및 허용 가능한 이산 값의 집합, 목적함수, 구속조건이 정리되어 있다.

먼저 문제 1-3 은 모든 설계변수가 정수변수인 문제이다. 문제 1 과 2 는 목적함수와 구속조건이 모두 선형이며, 문제 2 의 크기가 문제 1 보다 더 크다. 문제 3 은 목적함수가 이차함수, 구속조건은

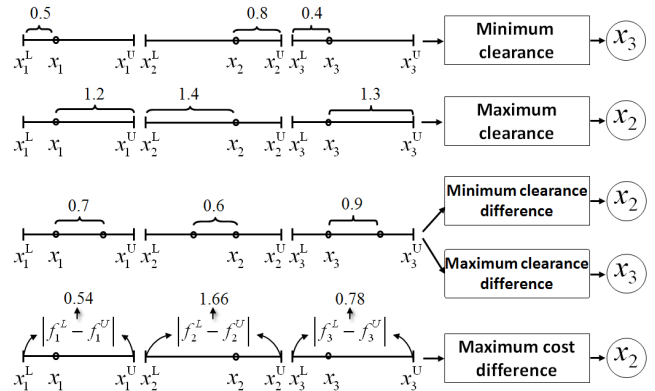


Fig. 2 Branching order methods

선형이다. 문제 4, 5 에서는 설계변수가 가질 수 있는 이산 값들이 모두 실수이다. 문제 6, 7 은 일부 설계변수는 정수이고 나머지는 연속변수인 혼합이산문제이며, 목적함수와 구속조건은 모두 다항함수이다. 문제 8-11 은 모두 공학예제이며, 이 중에서 문제 8-10 은 순수이산문제, 문제 11 은 혼합이산문제이다. 문제 8-11 은 문제 1-7 보다 비선형성이 심한 함수들로 이루어져 있다. 각 예제에 대한 더 자세한 설명은 Table 2 의 참고문헌들을 참조하기 바란다.

제안된 방법의 성능 검증 및 각 분지순서 방법의 비교를 위하여 본 연구에서는 균일 및 비균일 수렴허용오차 방법과 5 가지 분지 순서 방법들을 분지한계법에 각각 적용하여 테스트를 수행하였다.

Table 3 Discrete sets, objective function and constraints of each test problem

No.	Set of discrete values and variable bounds	Objective function	Constraints
1	$x_1 \in \{0, 1, 2, 3\}$ $x_2 \in \{0, 1, 2, 3, 4, 5, 6\}$	$f = -20x_1 - 10x_2$	$g_1 = -20x_1 - 10x_2 + 75 \leq 0$ $g_2 = 12x_1 + 7x_2 - 55 \leq 0$ $g_3 = 25x_1 + 10x_2 - 90 \leq 0$
2	$x_1 \in \{0, 1, 2, 3, 4\}$ $x_2 \in \{0, 1, 2, 3\}$ $x_3 \in \{0, 1, 2\}$ $x_4 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ $x_5 \in \{0, 1, 2\}$ $x_6 \in \{0, 1, 2\}$	$f = 3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6$	$g_1 = -110x_1 - 205x_2 - 160x_3 - 160x_4 - 420x_5 - 260x_6 + 2000 \leq 0$ $g_2 = -4x_1 - 32x_2 - 13x_3 - 8x_4 - 4x_5 - 14x_6 + 55 \leq 0$ $g_3 = -2x_1 - 12x_2 - 54x_3 - 285x_4 - 22x_5 - 80x_6 + 800 \leq 0$
2	$x_1 \in \{-2, -1, 0, \dots, 8\}$ $x_2 \in \{1, 2, 3, \dots, 11\}$ $x_3 \in \{-2, -1, 0, \dots, 8\}$	$f = 7x_1^2 + 6x_2^2 + 8x_3^2 - 6x_1x_3 + 4x_2x_3 - 15.8x_1 - 93.2x_2 - 63x_3 + 500$	$g_1 = 142x_1 + 172x_2 + 118x_3 \leq 1992$ $g_2 = 98x_1 + 114x_2 + 44x_3 \leq 1162$ $g_3 = 40x_1 + 72x_2 + 34x_3 \leq 703$
4	$x_1 \in \{0.3, 0.7, 0.9, 1.2, 1.5, 1.8\}$ $x_2 \in \{0.4, 0.8, 1.1, 1.4, 1.6\}$	$f = x_1 + x_2$	$g_1 = \frac{1}{x_1} + \frac{1}{x_2} - 2 \leq 0$
5	$x_1 \in \{0.3, 0.7, 0.9, 1.2, 1.5, 1.8\}$ $x_2 \in \{0.4, 0.8, 1.1, 1.4, 1.6\}$	$f = x_1^2 + x_2^2$	$g_1 = \frac{1}{x_1} + \frac{1}{x_2} - 2 \leq 0$
6	$x_1 \in \{1, 2, 3, 4, 5\}$ $0 \leq x_2 \leq 5$	$f = x_1^2 - 8x_2$	$g_1 = -8.63x_1 + x_2^3 \leq 0$
7	$x_1 \in \{1, 2, 3, 4, 5\}$ $x_2 \in \{0, 1, 2, 3, 4, 5\}$ $x_3 \in \{0, 1, 2, 3, 4, 5\}$ $-1 \leq x_i \leq 5, i = 4, 5, 6, 7$	$f = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$	$g_1 = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 127$ $g_2 = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 282$ $g_3 = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 196$ $g_4 = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$
8	$x_1, x_2, x_3 \in \{1.60, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0\}$	f : nonlinear function	g_1, g_2, \dots, g_5 : nonlinear functions
9	$x_1 \in \{0.05, 0.06, 0.07, \dots, 0.20\}$ $x_2 \in \{0.25, 0.3125, 0.375, \dots, 1.25\}$ $x_3 \in \{2, 3, 4, \dots, 15\}$	f : nonlinear function	g_1, g_2, g_3 : nonlinear functions g_4 : linear function
10	$x_1 \in \{2.6, 2.7, 2.8, \dots, 3.6\}$ $x_2 \in \{0.70, 0.71, 0.72, \dots, 0.80\}$ $x_3 \in \{17, 18, 19, \dots, 28\}$ $x_4 \in \{7.3, 7.4, 7.5, \dots, 8.3\}$ $x_5 \in \{7.3, 7.4, 7.5, \dots, 8.3\}$ $x_6 \in \{2.90, 2.91, 2.92, \dots, 3.90\}$ $x_7 \in \{5.00, 5.01, 5.02, \dots, 5.50\}$	f : nonlinear function	g_1, g_2, \dots, g_{11} : nonlinear functions
11	$x_1 \in \{0.0625, 0.125, 0.1875, \dots, 6.1875\}$ $x_2 \in \{0.0625, 0.125, 0.1875, \dots, 6.1875\}$ $10 \leq x_3 \leq 200$ $10 \leq x_4 \leq 200$	f : nonlinear function	g_1, g_2, g_4 : linear functions g_3 : nonlinear function

Table 4 Optimum solutions and numbers of function evaluations of BBM using the uniform convergence tolerance scheme

No.	NF_1 (B1)	NF_2 (B2)	NF_3 (B3)	NF_4 (B4)	NF_5 (B5)	Optimum solution
1	113	66	66	113	66	B1, B4 : $\mathbf{x}^* = (2, 4)^T, f(\mathbf{x}^*) = -80.0$ B2, B3, B5: $\mathbf{x}^* = (1, 6)^T, f(\mathbf{x}^*) = -80.0$
2	373	620	354	620	746	B1, B2, B3, B4, B5: $\mathbf{x}^* = (4, 0, 0, 5, 2, 0)^T, f(\mathbf{x}^*) = 97.0$
3	230	179	259	191	191	B1, B2, B3, B4, B5: $\mathbf{x}^* = (2, 7, 3)^T, f(\mathbf{x}^*) = 69.0$
4	137	137	139	137	139	B1, B2, B3, B4, B5: $\mathbf{x}^* = (1.5, 0.8)^T, f(\mathbf{x}^*) = 2.3$
5	124	128	124	128	124	B1, B2, B3, B4, B5: $\mathbf{x}^* = (1.2, 1.1)^T, f(\mathbf{x}^*) = 2.65$
6	132	132	132	132	132	B1, B2, B3, B4, B5: $\mathbf{x}^* = (2, 2.58)^T, f(\mathbf{x}^*) = -16.68$
7	180	261	261	180	180	B1, B4, B5: $\mathbf{x}^* = (2, 2, 0, 4.31, -0.67, 1.14, 1.38)^T, f(\mathbf{x}^*) = 683.12$ B2, B3 : $\mathbf{x}^* = (2, 2, 0, 4.31, -0.63, 1.13, 1.46)^T, f(\mathbf{x}^*) = 682.97$
8	47	47	47	47	47	B1, B2, B3, B4, B5: $\mathbf{x}^* = (1.6, 4.97, 1.6)^T, f(\mathbf{x}^*) = 9.50$
9	311	738	314	738	640	B1, B2, B3, B4, B5: $\mathbf{x}^* = (0.05, 0.3125, 15)^T, f(\mathbf{x}^*) = 0.0133$
10	137	135	137	135	137	B1, B2, B3, B4, B5: $\mathbf{x}^* = (3.5, 0.7, 17, 7.3, 7.8, 3.35, 5.29)^T, f(\mathbf{x}^*) = 2998.27$
11	205	205	87	205	87	B1, B2, B3, B4, B5: $\mathbf{x}^* = (0.8125, 0.4375, 42.10, 176.64)^T, f(\mathbf{x}^*) = 6059.71$

Table 5 Optimum solutions and numbers of function evaluations of BBM using the non-uniform convergence tolerance scheme

No.	NF_6 (B1)	NF_7 (B2)	NF_8 (B3)	NF_9 (B4)	NF_{10} (B5)	Optimum solution
1	109	66	66	109	66	B1, B4 : $\mathbf{x}^* = (2, 4)^T, f(\mathbf{x}^*) = -80.0$ B2, B3, B5: $\mathbf{x}^* = (1, 6)^T, f(\mathbf{x}^*) = -80.0$
2	373	576	328	576	662	B1, B2, B3, B4, B5: $\mathbf{x}^* = (4, 0, 0, 5, 2, 0)^T, f(\mathbf{x}^*) = 97.0$
3	201	170	182	179	179	B1, B2, B3, B4, B5: $\mathbf{x}^* = (2, 7, 3)^T, f(\mathbf{x}^*) = 69.0$
4	96	92	85	92	85	B1, B2, B3, B4, B5: $\mathbf{x}^* = (1.5, 0.8)^T, f(\mathbf{x}^*) = 2.3$
5	89	108	89	108	89	B1, B2, B3, B4, B5: $\mathbf{x}^* = (1.2, 1.1)^T, f(\mathbf{x}^*) = 2.65$
6	115	115	115	115	115	B1, B2, B3, B4, B5: $\mathbf{x}^* = (2, 2.58)^T, f(\mathbf{x}^*) = -16.68$
7	171	253	253	171	171	B1, B4, B5: $\mathbf{x}^* = (2, 2, 0, 4.30, -0.56, 1.12, 1.34)^T, f(\mathbf{x}^*) = 683.25$ B2, B3 : $\mathbf{x}^* = (2, 2, 0, 4.31, -0.63, 1.13, 1.46)^T, f(\mathbf{x}^*) = 682.97$
8	41	41	41	41	41	B1, B2, B3, B4, B5: $\mathbf{x}^* = (1.6, 4.97, 1.6)^T, f(\mathbf{x}^*) = 9.50$
9	119	492	94	492	137	B1, B2, B3, B4, B5: $\mathbf{x}^* = (0.05, 0.3125, 15)^T, f(\mathbf{x}^*) = 0.0133$
10	107	107	107	107	107	B1, B2, B3, B4, B5: $\mathbf{x}^* = (3.5, 0.7, 17, 7.3, 7.8, 3.35, 5.29)^T, f(\mathbf{x}^*) = 2998.27$
11	151	151	82	151	82	B1, B2, B3, B4, B5: $\mathbf{x}^* = (0.8125, 0.4375, 42.10, 176.64)^T, f(\mathbf{x}^*) = 6059.71$

먼저 Table 4 는 균일 수렴허용오차 방법과 5 가지 분지 순서 방법을 분지한계법에 적용하여 얻은 이산최적해 및 함수계산횟수(number of function evaluations, NF) 결과이며, Table 5 는 비균일 수렴허용오차 방법과 5 가지 분지순서 방법을 적용하여

얻은 결과이다. Table 4, 5 에서 B1-B5 는 선택된 분지순서 방법을 각각 나타낸다.

- B1: 최소간격법(minimum clearance method)
- B2: 최대간격법(maximum clearance method)

- B3: 최소간격차이법(minimum clearance difference method)
- B4: 최대간격차이법(maximum clearance difference method)
- B5: 최대비용차이법(maximum cost difference method)

Table 4, 5 와 같이 얻어진 결과를 정확성 및 효율성 측면에서 분석해 보고자 한다. 먼저 균일 수렴허용오차 방법을 이용한 분지한계법의 경우 문제 7 을 제외한 모든 예제에서 이산최적해로 수렴하였다. 문제 7 에서 분지순서 방법으로 B2, B3 을 이용했을 경우 이산최적해로 수렴하였으나, B1, B4, B5 를 이용한 분지한계법에서는 더 나쁜 해로 수렴하였다. 본 연구에서 제안한 비균일 수렴허용오차 방법을 이용한 분지한계법의 경우, 모든 예제에서 균일 수렴허용오차 방법을 이용하는 기존의 분지한계법과 동일한 이산최적해로 수렴하였다. 문제 7 의 경우 기존의 분지한계법과 마찬가지로 분지순서 방법으로 B2 와 B3 를 이용했을 때만 이산최적해로 수렴하였다. 즉, 제안된 방법을 이용한 분지한계법이 기존의 분지한계법과 동일한 정확성을 보이는 것을 확인할 수 있었다.

본 연구에서는 분지한계법의 효율성 비교를 위하여 식 (16)을 이용한 비교 방법을 이용하였다. 본 연구에서 이용한 11 가지 예제들은 설계변수의 개수 및 범위, 목적함수와 구속조건의 특성 등이 각기 다르다. 따라서 Table 4, 5 의 NF 값들을 상대적으로 비교하기 위해서는 식 (16)을 이용한 정규화가 필요하다.

$$\overline{NF}_i = \frac{NF_i}{\max(NF_1, NF_2, NF_3, \dots, NF_9, NF_{10})} \quad (16)$$

예를 들어 문제 3 에서 식 (16)을 이용하여 NF_8 의 정규화된 값을 계산하면 다음과 같다.

$$\begin{aligned} \overline{NF}_4 &= \frac{NF_8}{\max(NF_1, NF_2, NF_3, \dots, NF_9, NF_{10})} \\ &= \frac{182}{\max(230, 179, 259, \dots, 179, 179)} = \frac{182}{259} = 0.70 \end{aligned} \quad (17)$$

식 (16)을 이용하여 Table 4, 5 의 NF 값들을 모두 정규화하면 Table 6 과 같다. 만약 Table 6 에서 값이 1.00 일 경우에는 해당 문제에 적용된 방법의 효율성이 가장 나쁨을 의미하며, 값이 작을수록 효율성이 좋음을 나타낸다. Table 6 에서 볼 수 있듯이 동일한 분지순서 방법이 선택된 상태에서 기존의 균일 수렴허용오차 방법보다 제안된 비균일 수렴허용오차 방법을 분지한계법에 적용할 경우, 모든 예제에서 효율성이 크게 향상된 결과를 도출할 수 있었다. 한편, 5 가지 분지순서 방법 중에서는 최소간격차이법(B3)을 이용할 경우 분지한계법의 효율성이 평균적으로 가장 우수하였다. 함께 적용된 수렴허용오차 방법이 다르더라도 최소간격차이법을 적용한 분지한계법의 성능이 상대적으로 가장 우수하였다. 최소간격차이법은 현재 설계점과 그

Table 6 Scaled numbers of function evaluations of the original BBM and the enhanced BBM

No.	BBM using the uniform convergence tolerance scheme					BBM using the non-uniform convergence tolerance scheme				
	\overline{NF}_1 (B1)	\overline{NF}_2 (B2)	\overline{NF}_3 (B3)	\overline{NF}_4 (B4)	\overline{NF}_5 (B5)	\overline{NF}_6 (B1)	\overline{NF}_7 (B2)	\overline{NF}_8 (B3)	\overline{NF}_9 (B4)	\overline{NF}_{10} (B5)
1	1.00	0.58	0.58	1.00	0.58	0.96	0.58	0.58	0.96	0.58
2	0.50	0.83	0.47	0.83	1.00	0.50	0.77	0.44	0.77	0.89
3	0.89	0.69	1.00	0.74	0.74	0.78	0.66	0.70	0.69	0.69
4	0.99	0.99	1.00	0.99	1.00	0.69	0.66	0.61	0.66	0.61
5	0.97	1.00	0.97	1.00	0.97	0.70	0.84	0.70	0.84	0.70
6	1.00	1.00	1.00	1.00	1.00	0.87	0.87	0.87	0.87	0.87
7*	-	-	-	-	-	-	-	-	-	-
8	1.00	1.00	1.00	1.00	1.00	0.87	0.87	0.87	0.87	0.87
9	0.42	1.00	0.43	1.00	0.87	0.16	0.67	0.13	0.67	0.19
10	1.00	0.99	1.00	0.99	1.00	0.78	0.78	0.78	0.78	0.78
11	1.00	1.00	0.42	1.00	0.42	0.74	0.74	0.40	0.74	0.40
Average	0.88	0.91	0.79	0.95	0.86	0.70	0.74	0.61	0.79	0.66

주: * 표시된 문제 7 은 분지순서 방법에 따라 수렴된 이산최적해가 다르기 때문에 효율성 비교에서 제외하였음.

설계점 근방의 이산 값과의 거리가 가장 먼 이산 설계변수를 분지를 수행할 설계변수로 택하는 방법이다. 즉, 분지를 위해 선택된 설계변수의 현재 값이 근방의 이산 값들과의 거리가 멀 경우, 그 근방의 이산 값들이 최종 이산최적해가 될 가능성은 낮다. 그러므로 이러한 설계변수의 범위를 먼저 재조정하여 분지를 할 경우 이산최적해가 존재할 범위를 빨리 찾을 가능성이 높아진다. 이러한 이유로 인해 최소간격차이법이 다른 방법에 비해 빠른 수렴속도를 보이는 것으로 분석된다.

Table 6 에서 볼 수 있듯이 비균일 수렴허용오차 방법과 분지순서 방법 B3(최소간격차이법)의 조합으로 얻어진 \overline{NF}_8 이 평균 값 0.61 로 가장 효율성이 뛰어났다. 이는 효율성이 가장 나쁜 조합인 \overline{NF}_4 의 평균보다는 36% 더 성능이 좋으며, 같은 최소간격차이법을 이용했으나 기존의 균일 수렴허용오차 방법이 적용된 \overline{NF}_3 보다는 23% 효율성이 향상된 결과이다.

지금까지 Table 4, 5 의 결과를 정확성과 효율성 측면에서 분석한 결과, 본 연구에서 제안된 비균일 수렴허용오차 방법과 분지순서 방법 중 최소간격차이법을 분지한계법에 적용할 때 모든 예제에서 정확하면서도 가장 효율적으로 이산최적해를 도출할 수 있었다.

4. 결론

MDNLP 기법 중에서 가장 널리 이용되고 있는 기법 중에 하나인 분지한계법은 다른 MDNLP 기법에 비해 강건하지만 각 노드마다 부문제 최적화를 수행하기 때문에 많은 함수 계산이 요구되는 단점이 존재한다. 이러한 분지한계법의 단점을 극복하기 위하여 본 연구에서는 각 노드마다 수렴허용오차를 다르게 설정해주는 비균일 수렴허용오차 방법을 제안하였다. 또한 분지한계법에서 분지를 수행할 때 상하한 범위를 재설정 할 설계변수를 선정하기 위해 개발된 5 가지 분지순서 방법 중 분지한계법의 성능을 가장 극대화할 수 있는 분지순서 방법이 무엇인지 알아보기 위한 연구를 수행하였다.

제안된 수렴허용오차 방법의 성능 검증 및 각 분지순서 방법의 비교를 위하여 본 연구에서는 7 개의 수학예제와 4 개의 공학예제를 이용하였다. 테스트를 수행한 결과, 비균일 수렴허용오차 방법을 이용하는 개선된 분지한계법이 균일 수렴허용오차를 이용하는 기존의 분지한계법과 모든 예제

에서 동일한 정확성을 보였다. 또한 제안된 방법을 이용한 분지한계법은 모든 예제에서 기존의 분지한계법보다 효율성이 뛰어남을 확인할 수 있었다. 5 가지 분지순서 방법 중에서는 최소간격차이법을 이용할 경우 분지한계법의 정확성 및 효율성이 가장 뛰어났다. 즉, 비균일 수렴허용오차 방법과 최소간격차이법을 분지한계법에 함께 적용한 경우, 가장 결과가 안 좋은 조합인 균일 수렴허용오차 방법 및 최대간격차이법을 이용하는 분지한계법에 비해 평균 36% 효율성이 더 좋았으며, 같은 최소간격차이법을 이용했으나 기존의 균일 수렴허용오차 방법이 함께 적용된 분지한계법 보다는 23% 정도 효율성이 향상된 결과를 얻을 수 있었다. 이러한 결과들을 통해 본 연구에서 제안된 비균일 수렴허용오차 방법과 5 가지 분지순서 방법 중 최소간격차이법을 함께 분지한계법에 적용할 경우 분지한계법의 성능이 가장 극대화됨을 확인할 수 있었다.

후 기

본 연구는 2011 년도 교육과학기술부의 재원에 의한 한국연구재단의 지원(No. 2011-0016701) 및 교육과학기술부의 “터보펌프 볼 베어링의 시제품 개발” 사업의 지원을 받아 수행되었습니다.

참고문헌

- (1) Rajeev, S. and Krishnamoorthy, C. S., 1992, "Discrete Optimization of Structures Using Genetic Algorithms," *Journal of Structural Engineering*, Vol. 118, No. 5, pp. 1233~1250.
- (2) Lin, C.-Y. and Hajela, P., 1992, "Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables," *Engineering Optimization*, Vol. 19, No. 4, pp. 309~327.
- (3) He, S., Prempain, E. and Wu, Q. H., 2004, "An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems," *Engineering Optimization*, Vol. 36, No. 5, pp. 585~605.
- (4) Kitayama, S., Arakawa, M. and Yamazaki, K., 2006, "Penalty Function Approach for the Mixed Discrete Nonlinear Problems by Particle Swarm Optimization," *Structural and Multidisciplinary Optimization*, Vol. 32, No. 3, pp. 191~202.
- (5) Shin, D. K., Gürdal, Z. and Griffin, O. H., 1990, "A Penalty Approach for Nonlinear Optimization with Discrete Design Variables," *Engineering Optimization*, Vol. 16, No. 1, pp. 29~42.
- (6) Loh, H. T. and Papalambros, P. Y., 1991, "A

- Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems," *Journal of Mechanical Design, ASME*, Vol. 113, No. 3, pp. 325~334.
- (7) Loh, H. T. and Papalambros, P. Y., 1991, "Computational Implementation and Tests of a Sequential Linearization Algorithm for Mixed-Discrete Nonlinear Design Optimization," *Journal of Mechanical Design, ASME*, Vol. 113, No. 3, pp. 335~345.
- (8) Bremicker, M., Papalambros, P. Y. and Loh, H. T., 1990, "Solution of Mixed-Discrete Structural Optimization Problems with a New Sequential Linearization Algorithm," *Computers and Structures*, Vol. 37, No. 4, pp. 451~461.
- (9) Land, A. H. and Doig, A. G., 1960, "An Automatic Method of Solving Discrete Programming Problems," *Econometrica*, Vol. 28, No. 3, pp. 497~520.
- (10) Dakin, R. J., 1965, "A Tree-Search Algorithm for Mixed Integer Programming Problems," *The Computer Journal*, Vol. 8, No. 3, pp. 250~255.
- (11) Gupta, O. K. and Ravindran, A., 1983, "Nonlinear Integer Programming and Discrete Optimization," *Journal of Mechanisms, Transmissions and Automation in Design, ASME*, Vol. 105, No. 2, pp. 160~164.
- (12) Hajela, P. and Shih, C. J., 1989, "Optimal design of Laminated Composites Using a Modified Mixed Integer and Discrete Programming Algorithm," *Computers and Structures*, Vol. 32, No. 1, pp. 213~221.
- (13) Sandgren, E., 1990, "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *Journal of Mechanical Design, ASME*, Vol. 112, No. 2, pp. 223~229.
- (14) Arora, J. S., Huang, M. W. and Hsieh, C. C., 1994, "Methods for Optimization of Nonlinear Problems with Discrete Variables: A Review," *Structural Optimization*, Vol. 8, No. 2-3, pp. 69~85.
- (15) Tseng, C. H., Wang, L. W. and Ling, S. F., 1995, "Enhancing Branch-and-Bound Method for Structural Optimization," *Journal of structural engineering*, N.Y., Vol. 121, No. 5, pp. 831~837.
- (16) Leyffer, S., 2001, "Integrating SQP and Branch-and-Bound for Mixed Integer Nonlinear Programming," *Computational Optimization and Applications*, Vol. 18, No. 3, pp. 295~309.
- (17) Thanedar, P. B. and Vanderplaats, G. N., 1995, "Survey of Discrete Variable Optimization for Structural Design," *Journal of Structural Engineering*, Vol. 121, No. 2, pp. 301~305.
- (18) Huang, M. W. and Arora, J. S., 1997, "Optimal Design with Discrete Variables: Some Numerical Experiments," *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 1, pp. 165~188.
- (19) Jung, S., Choi, D. H. and Choi, G., "A sequential Quadratic Programming with an Approximate Hessian Matrix Update Using an Enhanced Two-Point Diagonal Quadratic Approximation," *13th AIAA/ISSMO Multi-Disciplinary Analysis and Optimization Conference*, Fort Worth, Texas, USA.
- (20) 1999, *DOT user's manual, version 5.0*, Vanderplaats Research & Development, Inc.
- (21) 2011, *PLAnO user's manual, version 3.3*, PIDOTECH, Inc.
- (22) Arora, J. S., 2004, *Introduction to Optimum Design*, 2nd Edition, Elsevier/Academic Press, pp. 517~518.
- (23) Chvatal, V., 1983, *Linear Programming*, Freeman, W. H., New York.
- (24) Salajegheh, E. and Vanderplaats, G. N., 1993, "Optimum Design of Trusses with Discrete Sizing and Shape Variables," *Structural Optimization*, Vol. 6, No. 2, pp. 79~85.
- (25) Arora, J. S., 2004, *Introduction to optimum design*, 2nd Edition, Elsevier/Academic Press, pp. 529.
- (26) Chen, X., 2009, *An Enhanced Branch-and-Bound Method for Discrete Optimization*, Master's Thesis, Hanyang University, pp. 36~38.
- (27) Arora, J. S., 2004, *Introduction to Optimum Design*, 2nd Edition, Elsevier/Academic Press, pp. 528.
- (28) Rao, S. S. and Xiong, Y., 2005, "A hybrid Genetic Algorithm for Mixed-Discrete Design Optimization," *Journal of Mechanical Design, Transactions of the ASME*, Vol. 127, No. 6, pp. 1100~1112.
- (29) Nema, S., Goulermas, J., Sparrow, G. and Cook, P., 2008, "A Hybrid Particle Swarm Branch-and-Bound (HPB) Optimizer for Mixed Discrete Nonlinear Programming," *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*, Vol. 38, No. 6, pp. 1411~1424.