

## 타부탐색을 이용한 이산설계공간에서의 구조물의 최적설계

이 권 희<sup>†</sup>, 주 원 식<sup>\*</sup>

(2002년 10월 17일 접수, 2003년 3월 18일 심사완료)

### Structural Optimization Using Tabu Search in Discrete Design Space

Kwon-Hee Lee and Won-Sik Joo

**Key Words :** Tabu Search(타부탐색), Discrete Design Space(이산설계공간), Structural Optimization(구조최적설계), Global Optimum(전역적 최적해)

#### Abstract

Structural optimization has been carried out in continuous or discrete design space. Methods for continuous design have been well developed though they are finding the local optima. On the contrary, the existing methods for discrete design are extremely expensive in computational cost or not robust. In this research, an algorithm using tabu search is developed for the discrete structural designs. The tabu list and the neighbor function of the Tabu concepts are introduced to the algorithm. It defines the number of steps, the maximum number for random searches and the stop criteria. A tabu search is known as the heuristic approach while genetic algorithm and simulated annealing algorithm are attributed to the stochastic approach. It is shown that an algorithm using the tabu search with random moves has an advantage of discrete design. Furthermore, the suggested method finds the reliable optimum for the discrete design problems. The existing tabu search methods are reviewed. Subsequently, the suggested method is explained. The mathematical problems and structural design problems are investigated to show the validity of the proposed method. The results of the structural designs are compared with those from a genetic algorithm and an orthogonal array design.

#### 1. 서론

수치적 최적설계 기법은 현재 하드웨어 및 소프트웨어의 발전과 더불어 현장에 널리 적용되고 있다. 특히, 구조설계는 최적설계의 적용이 가장 활발하게 이루어지는 분야 중의 하나이다. 이러한 구조최적설계는 종종 이산설계변수(discrete design variable)를 취급해야 한다. 즉, 구조설계 시 부재의 크기가 규격화되어 있거나, 특정한 값 중에서 선정되어야 한다면, 이는 조합최적화(combinatorial optimization)문제로 볼 수 있다.

이산설계공간(discrete design space)에서의 최적설계는 연속설계공간에서 최적해가 갖는 쿤-터커 조건(Kuhn-Tucker conditions)등의 이론적 정리가 성립

되지 않는다. 현재까지 이산설계를 위한 여러가지 방법들이 제안되고 있다.<sup>(1,2)</sup> 현장에서 주로 사용되는 방법은 연속설계공간에서 최적해를 구한 후, 그 해 보다 한 단계 높은 값을 선정하는 것이다. 그러나 이 방법은 고유진동수 등의 제한조건을 위배할 수 있으며 중량의 증가를 가져올 수 있다.<sup>(2)</sup>

Arora 등<sup>(3)</sup>은 대화형 설계최적화를 통해 설계변수를 고정시키고자 하였지만 설계자동화 개념이 없어지고 설계자의 주관이 너무 작용한다는 단점이 있다. Salajegheh 등<sup>(4)</sup>은 분단 탐색법(branch and bound method)을 정수계획법을 위해 전개하였고, Rajeev 등<sup>(5)</sup>은 유전자 알고리즘(genetic algorithms)을 이용하여 이산설계변수를 처리하였다. 최근에는 컴퓨터의 처리속도 증가에 따라 유전자 알고리즘이 많이 사용되고 있다. 그러나 유전자 알고리즘이 갖고 있는 통계적 특성 상, 함수계산횟수가 지나치게 많게 되어 구조설계 시 적용에 한계가 있다.

직교배열표를 연속적으로 이용하는 방법<sup>(6,7)</sup>은

<sup>†</sup> 책임저자, 회원, 동아대학교 기계산업시스템공학부  
E-mail : leekh@donga.ac.kr

TEL : (051)200-7638 FAX : (051)200-7656

<sup>\*</sup> 회원, 동아대학교 기계산업시스템공학부

유전자 알고리즘이 갖고 있는 설계 시간을 단축할 수 있는 장점이 있다. 그러나 초기값에 따른 최적해의 강건성이 보장이 되지 않는다. 본 논문에서는 유전자 알고리즘이 갖고 있는 알고리즘의 강건성 및 전역적 최적해를 찾을 수 있는 장점과 직교배열표를 이용한 방법이 갖고 있는 함수계산의 단축효과를 동시에 얻을 수 있는 방법을 제시하였다.

타부탐색(tabu search)의 근간은 1960년대 Hansen에 의해 이루어 졌으며, 1970년대 Glover<sup>(8)</sup>에 의해 현재의 형태로 제시되고 있다. 본래 타부탐색은 경영과학(operations research)분야에서 작업자 스케줄링, 기계 스케줄링, 전화통신 경로배정 문제 등의 현실적인 문제를 해결할 수 있는 알고리즘으로 널리 이용되고 있다. 유전자 알고리즘이나 시뮬레이티드 어닐링(simulated annealing) 방법이 통계적 방법에 속한다면 타부탐색은 발견론적(heuristic) 방법이라고 할 수 있다.<sup>(9)</sup>

타부탐색이 구조설계에 처음 적용된 시기는 1992년으로써, Hu<sup>(10)</sup>는 비록 국부적 해에 빠지는 한계를 갖는 알고리즘을 제시했지만 타부탐색의 장점을 구조설계에 반영하였다. 그 이후, 1995년에 Dhingra 등<sup>(11)</sup>은 이산 및 연속 설계변수를 갖는 구조설계 문제에 적용할 수 있는 타부탐색 방법을 개발하였다. 앞의 Hu에 비해 타부탐색의 개념에 보다 충실한 알고리즘을 제시하였다. 그러나 그는 자동화된 멈춤조건(stop criteria)을 제공하지 않고 임의의 반복횟수를 수행한 후, 최적해를 선정하는 방법을 선택하였다. 그리고 2001년, Machado 등<sup>(12)</sup>은 타부목록(tabu list)을 제거하고 멈춤조건을 정의한 자동화한 알고리즘을 제시하였다. 그러나 구조설계 시 타부목록은 함수계산 감소를 위해 필요한 개념이다. 국내에서는, 양보석 등<sup>(13)</sup>이 제진판 설계에, 조지현 등<sup>(14)</sup>이 칩의 배치 설계에 각각 타부탐색을 적용한 예가 있다.

본 연구에서는 이산설계변수를 갖는 구조설계를 위한 최적설계 알고리즘을 타부탐색이 가지는 특징을 이용하여 개발하였다. 초기의 설계점을 중심으로 이웃함수(neighborhood function)들을 정의하고, 이들로부터 반복적으로 최적해를 산출한다. 이때 구해진 최적해에 해당되는 이웃함수를 타부목록에 추가한다. 여기서 함수계산의 효율성을 위해 주어진 이웃함수들로부터 최적해를 산출하기 위한 무작위 탐색횟수를 제한한다. 그리고 이웃함수의 수인 스텝(step)수도 일정하게 고정하였다.

제시된 타부탐색을 이용한 알고리즘이 자동화된 수렴조건을 갖기 위한 조건을 정의하였다. 이는 이산설계 시, 경험적으로 결정되었다. 그리고 구

조최적설계시 부여되는 제한조건을 처리하기 위한 특성함수를 정의하였다. 이 함수는 순차적비제약 최소화기법(SUMT)에서 정의되는 가목적함수(pseudo-objective function), 구조설계시 유전자 알고리즘에서 정의되는 벌칙함수(penalty function)를 포함하는 목적함수(fitness function), 직교배열표를 이용한 이산설계 방법<sup>(2,7)</sup> 등에서 사용되는 특성함수와 동일한 개념이다.

수치적 성능을 검토하기 위하여 제시된 알고리즘을 연속설계변수를 갖는 수학적 문제와 이산설계변수를 갖는 3 부재 트러스, 10 부재 트러스, 전기자동차 프레임의 구조 설계문제에 각각 적용하였다. 구조설계 문제의 예에서는 유전자 알고리즘과 직교배열표를 이용한 알고리즘<sup>(2,7)</sup>의 결과들과 비교하였다.

## 2. 타부탐색(Tabu Search) 알고리즘 소개

### 2.1 타부탐색<sup>(8,9)</sup>

Glover는 타부탐색의 주요 개념을 다음과 같이 설명하고 있다. 다음과 같은 최적화 문제를 생각해 보자.

$$\text{Minimize } c(x): x \in X \text{ in } R^n \quad (1)$$

여기서  $c(x)$ 는 목적함수,  $x$ 는 설계변수 벡터로써 이산설계변수 또는 연속설계변수 모두 가능하다.  $X$ 는 유용해(feasible solution)로써 제한조건이 없는 함수의 경우 설계변수의 부가제한조건(side constraint)을 만족하는 설계점들의 초집합(superset)이다. 제한조건이 있는 문제의 경우에는 모든 제한조건들을 만족하는 설계점들의 초집합이다. 그리고  $R^n$ 은  $n$  차원의 실수공간을 의미한다.

식 (1)의 최적해는 임의의 설계점에서 다른 설계점으로서의 연속적인 이동에 의해 얻어진다.  $X$ 의 부분집합  $X(s)$ 에 대하여 이동  $s$ 를 다음과 같은 사상(mapping)으로 정의한다.

$$s: X(s) \rightarrow X \quad (2)$$

즉  $S(x)$  및  $X(s)$ 는 다음과 같이 표시할 수 있다.

$$S(x) = \{s \in S: x \in X(s)\} \quad (3)$$

$$X(s) = \{x \in X: s \in S(x)\} \quad (4)$$

여기서 집합  $S(x)$ 를 이웃함수(neighborhood function)라고 한다.

Glover는 가장 단순한 형태의 타부탐색을 Fig. 1과 같이 제안하고 있다. 여기서  $T$ 는 타부목록,  $k_{max}$ 는 최대반복수,  $x^*$ 는 최적해, OPTIMUM은 이웃함수중에 최적해를 찾는 함수이다. 타부탐색에서 타부목록에 포함된 이웃함수는 탐색시 제외된다.

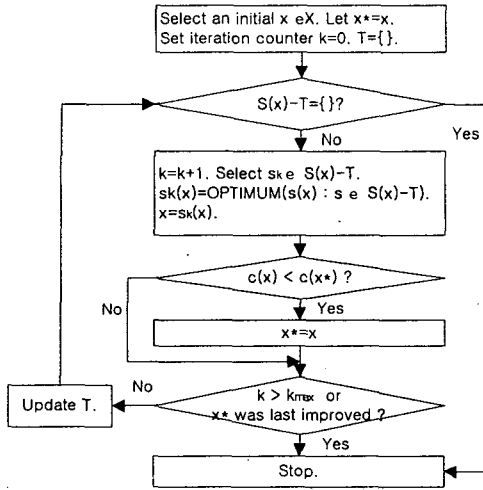


Fig. 1 Simple tabu search

2.2 Hu의 알고리즘<sup>(10)</sup>

Hu는 전역적 최소를 찾기 위한 제한조건이 있는 최적화 문제를 해결할 수 있는 타부탐색법을 제안하였다. 그는 무작위 탐색(random search)이 가지는 비효율성을 해결하기 위해 무작위 이동(random moves)을 하는 타부탐색의 알고리즘을 제시한 것이다. 식 (5)-(8)과 같은 정식화를 생각해 보자.

$$\begin{aligned} \text{Minimize } & f(x) & (5) \\ \text{Subject to } & F_i(x) = 0 \quad (i=1, \dots, N) & (6) \\ & G_j(x) \leq 0 \quad (j=1, \dots, M) & (7) \\ & a \leq x_k \leq b \quad (k=1, \dots, n) & (8) \end{aligned}$$

여기서  $f(x)$ 는 목적함수,  $F_i(x)$ 는  $i$  번째 등제한조건 함수,  $G_j(x)$ 는  $j$  번째 부등제한조건함수,  $a, b$ 는 설계변수의 하한값과 상한값을 표시한다.

위의 식 (6)-(8)을 만족하는 유용해  $y$ 는 다음과 같이 표시할 수 있다.

$$y = S(x, s, H) \quad (9)$$

여기서  $H = [h_1, h_2, \dots, h_r]^T$ 는 스텝벡터로써 다음과 같이 정의 된다.

$$h_1 = b - a, h_2 = h_1/10.0, \dots, h_r = h_{r-1}/10.0 \quad (10)$$

여기서  $r$ 은 스텝의 수이다. 현재점  $x$ 의 한 성분  $x_k$ 에 대하여 임의의 스텝  $h_i$ 에서 그 이웃은 다음과 같이 표시된다.

$$N_i(x_k, h_i) = \{y_k : |x_k - y_k| \leq h_i\} \quad (11)$$

여기서  $y_k$ 는  $y$  벡터의  $x_k$ 에 대응하는 한 성분이다.

Hu가 제안한 알고리즘을 정리하면 Fig. 2와 같이 표시할 수 있다.

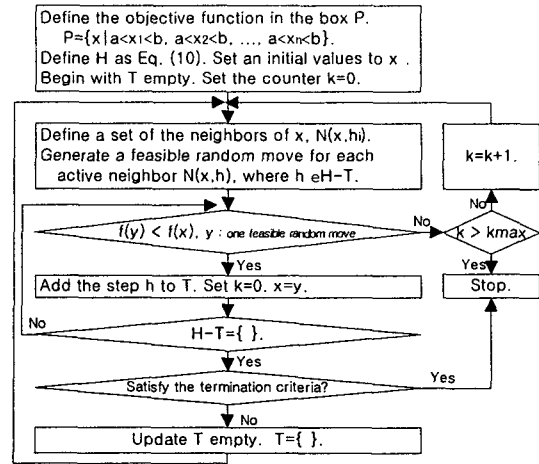


Fig. 2 Tabu search with random moves

그러나 Hu가 제안한 타부탐색법은 설계점  $y$ 의 함수값이 항상  $x$ 에 의한 함수값보다 작은 경우에만 설계점이 변경되므로 국부최적해에 빠질 가능성이 존재한다. 그리고  $n$ 차원에 대한 설계변수의 무작위 탐색은 수렴을 어렵게 만든다. 또한 각 설계변수에 대한 하한값, 상한값을 동일하게 고정시키고 있다.

2.3 Dhingra의 알고리즘<sup>(11)</sup>

Dhingra 등은 Hu가 제안한 방법을 기초로 본래의 타부탐색 개념에 충실한 알고리즘을 제시하였다. 제한조건이 없는 연속설계변수인 경우의 문제를 위한 타부탐색을 Fig. 3에 표시하였다. Fig. 3에서  $N_{max}$ 란 전체 반복횟수에 대한 제한값이다. Hu가 제안한 방법과 가장 큰 차이점은 이웃들에서 산출된 최적값인  $x_{po}$ 가 전체최적해인  $x_g$ 에 해당하는 함수값보다 클 경우에도 그 다음 단계의 설계점  $x_c$ 로 치환된다는 점이다. 이것은 타부탐색이 국부최적해에 빠지는 것을 막아주는 역할을 한다. 그리고 고유의 타부탐색이 갖고 있는 열망기준(aspiration criteria)을 알고리즘에 포함시켰다.

열망기준은 타부제약이 오히려 좋은 해를 얻는데 장애가 될 수 있다는 판단 아래, 타부조건을 무시할 수 있는 새로운 조건을 설정한 것이다. Dhingra가 제안한 타부탐색법은 타부탐색의 개념에 충실하고 있지만 구조설계를 위한 멈춤조건이 포함되어 있지 않다. 그리고 그가 제안한 열망기준은 오히려 함수계산을 증가시키는 단점을 보이고 있다.

3. 타부탐색을 이용한 이산설계알고리즘

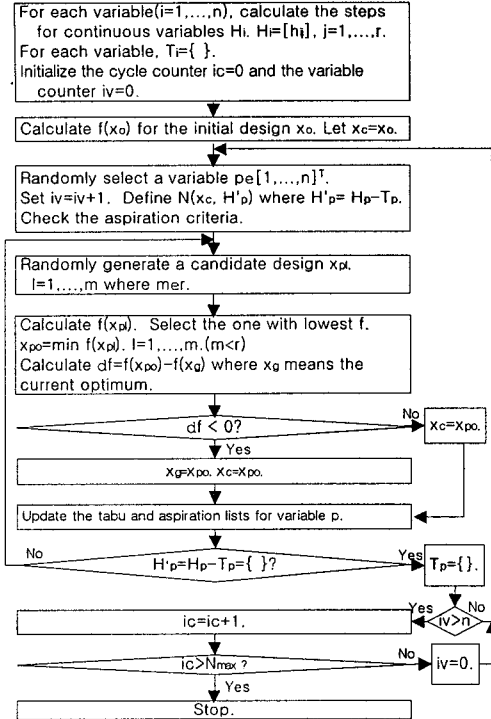


Fig. 3 Tabu search suggested by Dhingra

기존의 구조설계를 위한 타부탐색 알고리즘들은 구조설계의 적용성에 관심을 갖고 개발한 반면, 함수계산횟수의 감소에는 한계가 있다. 본 논문에서는 이산설계변수를 갖는 구조설계 시 함수계산의 감소를 위한 알고리즘을 다음과 같이 제시하였다.

3.1 스텝의 정의와 이웃의 생성

기존의 연구자와 동일하게 스텝은 정의되며 n 개의 설계변수를 처리하기 위해 행렬 **H**로 정의하였다. **H<sub>u</sub>**는 식 (8)과 같이 모든 설계변수의 한계값을 동일하게 보고 있지만 이것은 일반적이지 않다. 따라서 각 설계변수의 한계값을 다르다고 가정하면, 스텝행렬 **H**는 다음과 같이 정의된다.

$$\mathbf{H} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \dots \ \mathbf{H}_n] \quad (12)$$

여기서 k 열의 스텝벡터 **H<sub>k</sub>**라고 하면, **H<sub>k</sub>**=[h<sub>k1</sub>, h<sub>k2</sub>, ..., h<sub>kr</sub>]<sup>T</sup>이다. 그리고 k 번째 설계변수의 하한값과 상한값을 각각 a<sub>k</sub>, b<sub>k</sub>라고 하면 벡터의 성분은 식 (10)과 동일하게 다음과 같이 정의할 수 있다.

$$h_{k1}=b_k-a_k, \ h_{k2}=h_{k1}/10.0, \ \dots, \ h_{kr}=h_{k(r-1)}/10.0 \quad (13)$$

본 논문에서는 수학적 문제의 경우에는 스텝의 수 r을 4, 3으로 하였고 구조설계 문제의 경우에는 3으로 고정하였다.

각 설계변수의 스텝이 결정되면 각 스텝에 대한 k 번째 설계변수 x<sub>k</sub>의 이웃은 다음과 같이 된다.

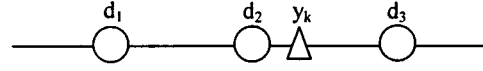


Fig. 4 Assignment of discrete values

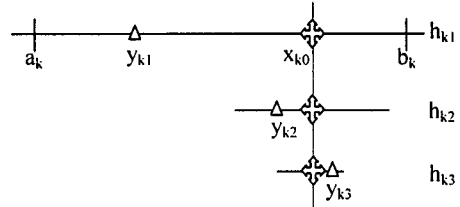


Fig. 5 Steps and neighbors of x<sub>k</sub>

$$N_{ki}(x_k, h_{ki}) = \{y_k : |x_k - y_k| \leq h_{ki}\} \quad (14)$$

$$, i = 1, \dots, r, \quad k = 1, \dots, n$$

여기서 y<sub>k</sub>는 무작위로 결정되며 다음과 같은 수식으로 표현할 수 있다.

$$y_k = x_k + R \cdot h_{ki} \quad (15)$$

$$-1 \leq R \leq 1(\text{random num.})$$

여기서 R은 무작위로 결정되는 값이다. 그러나 y<sub>k</sub>가 x<sub>k</sub>의 a<sub>k</sub> 및 b<sub>k</sub>의 범위를 벗어나면, 벗어 나는 쪽의 한계값을 갖도록 한다.

이산설계변수인 경우에는 발생한 임의의 이웃값에서 가장 가까운 이산값으로 대체한다. 설계변수 x<sub>k</sub>의 y<sub>k</sub>가 Fig. 4와 같이 무작위로 결정되었다면 y<sub>k</sub>가 d<sub>2</sub>의 값을 갖게 한다. 즉 x<sub>k</sub>가 d<sub>1</sub>, d<sub>2</sub>, ..., d<sub>z</sub>의 z 개의 이산값을 갖는다면 이산값은 다음식과 같이 대체된다.

$$y_k = \text{nearest}(y_k, (d_1, d_2, \dots, d_z)) \quad (16)$$

3.2 이웃들에서의 탐색횟수

**H<sub>u</sub>**는 최적화문제를 풀기 위한 타부탐색 알고리즘을 제안하면서 Fig. 2의 k가 k<sub>max</sub>가 초과되면 해 찾는 과정을 중지시켰다. 따라서 k<sub>max</sub>는 임의의 큰 수가 될 것이며 이 과정에서 함수계산이 많아진다. 설계변수 x<sub>k</sub>의 임의의 설계점 x<sub>k0</sub>에 대한 스텝 및 이웃들을 Fig. 5에 표시하였다. 여기서 y<sub>k1</sub>, y<sub>k2</sub>, y<sub>k3</sub>는 각 스텝에서 무작위로 결정된 x<sub>k</sub>의 값이다. 따라서 식 (14)로 결정되는 이웃은 다음과 같은 포함 관계가 있다.

$$N_{k1} \supset N_{k2} \supset \dots \supset N_{kr} \quad (17)$$

**H<sub>u</sub>**의 알고리즘은 Fig. 5의 세번째 이웃에서 최소가 산출됐을 경우, 이 보다 작은 함수값을 찾기 위해 첫번째 스텝에서 많은 함수계산을 할 가능성이 있다. 그리고 심지어는 k<sub>max</sub>를 초과하여 알고리즘을 중단시킬 수도 있다. 본 논문에서는 이것

을 방지하기 위하여 일정한 수 만큼을 탐색하여 더 좋은 값을 찾지 못 해도 타부목록에 추가하도록 하였다. 예제풀이 시,  $k_{max}=3$  으로 설정하였다. 이는 경험적인 수치이다.

3.3 제한조건함수의 처리

식 (5)-(8)로 정의되는 최적설계 정식화 문제는 부등제한조건 및 등제한조건함수를 포함하고 있다.  $H_u$  가 이용한 유용해를 판별하는 방법은 임의의 설계점에 대해, 식 (6)-(7)을 계산하고 이것이 만족되는 설계점을 찾는 것이다. 그러나 이것은 불필요한 함수계산을 하게 만드는 원시적 방법이다.

이 방법 대신에 기존의 통계적 방법 및 직교배열표 설계 방법에서 사용되는 것과 동일한 의미의 특성함수를 부등제한조건만을 고려하여 다음과 같이 정의하였다.

$$\psi(x) = f(x) + w \cdot \sum_{j=1}^M \text{Max}[0, G_j(x)] \quad (18)$$

여기서  $w$  는 임의의 큰 수이다.

3.4 멈춤조건

타부탐색에 있어서 종료조건은 함수계산횟수에 큰 영향을 미치는 조건이다. 즉 수렴조건을 너무 완화시키면 국부 최적해에 빠지기 쉽고, 과도하게 엄격히 설정하면 함수계산횟수가 급격히 증가한다. Machado 등<sup>(12)</sup>은 경험적으로 멈춤조건을 다음과 같이 부여하였다.

$$|f_c^* - f_{c-q}^*| \leq \epsilon \quad (q=1,2,\dots,n_q) \quad (19)$$

$$f_c^* - f_{opt} \leq \epsilon \quad (20)$$

여기서  $f_c^*$  는  $c$  번째 순환에서 찾은 최소해이며  $f_{opt}$  는 지금까지의 최소해,  $\epsilon$  는 아주 작은 수이다. 그리고  $n_q$  는 규정된 정수로서 6, 7, 8 등을 추천하고 있다.

본 논문에서는 위의 멈춤조건을 이용하였는데  $n_q$  를 연속설계변수인 경우에는 3, 이산설계변수인 경우에는 2로 설정하였다. 제시된 방법에서 해의 정확도에 영향을 미치는 파라미터들은 스텝 수  $r$ , 최대탐색수  $k_{max}$ ,  $n_q$ ,  $\epsilon$  등이 있다.

지금까지 내용을 기초로, 이산설계변수를 위한 알고리즘의 순서도는 Fig. 6 과 같이 표시할 수 있다. 연속설계변수 문제의 최적화문제에 제시된 방법을 사용할 경우, Fig. 6 에서 무작위 탐색한 값을 이산값으로 대치시키는 과정만을 생략하면 된다.

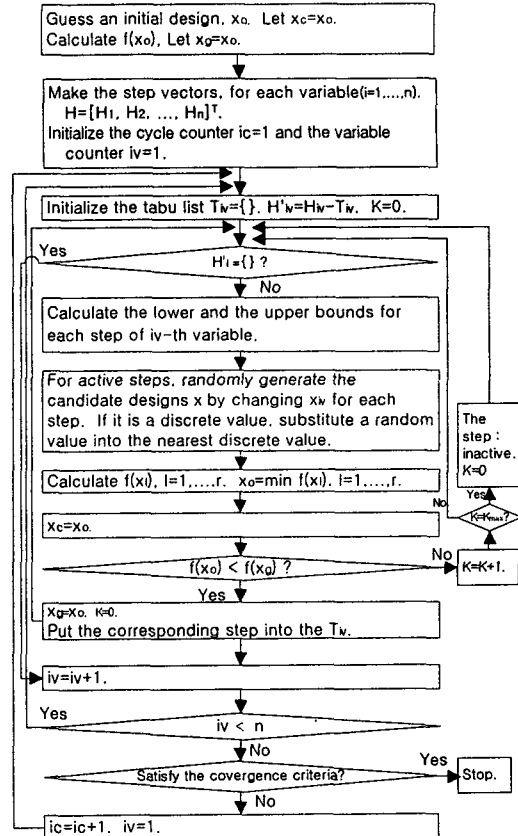


Fig. 6 Tabu search algorithm for a discrete design

4. 적용예제 및 검토

4.1 수학적문제

4.1.1 함수 #1<sup>(9,12)</sup>

다음과 같이 2 개의 연속설계변수를 갖는 함수를 생각해 보자.

$$f(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \quad (21)$$

$$-3 \leq x_1 \leq 3, \quad -2 \leq x_2 \leq 2$$

이 함수의 함수 등고선을 Fig. 7 에 표시하였는데, 그림에서 \*표시는 국부최적해를 표시하고 있다. 국부최적해 및 그 함수값은 Table 1 과 같다. 이 문제는 6 개의 국부최적해와 2 개의 전역적최적해를 갖고 있다.

본 논문에서 제안한 알고리즘을 이용하고, 초기치를 변경하여 구한 결과를 Table 2 에 표시하였다. 식 (19)-(20)의 수렴조건 상수  $\epsilon$  는 수학적 문제 #1 에 대해 0.0001 로 설정하였다. 여기서 반복횟수는 Fig. 6 의  $ic$  횟수를 의미한다.

Table 1 Local optima of function #1

No.	$(x_1, x_2)$	Function value
1	0.0898, -0.7126	-1.0300
2	-0.0898, 0.7126	-1.0300
3	1.7036, -0.7960	-0.2154
4	-1.7036, 0.7960	-0.2154
5	1.6071, 0.5687	2.1042
6	-1.6071, -0.5687	2.1042

Table 2 Tabu search results of function #1

Initial value $x_1, x_2$	No. of iterations (no. of evaluations)	$(x_1^*, x_2^*)$	$f(x^*)$
-3.0, -2.0	5(211)	-0.0894, 0.7166	-1.032
-3.0, 2.0	5(157)	-0.0967, 0.7056	-1.031
3.0, -2.0	5(211)	-0.0894, 0.7166	-1.032
3.0, 2.0	4(157)	-0.0966, 0.7056	-1.031
0.0, 0.0	4(164)	-0.1004, 0.7120	-1.031
a)	600	-0.0898, 0.7127	-1.032
b)	630	-0.0898, 0.7126	-1.032

a) By Hu b) By Machado

Table 3 Local optima of function #2

No.	$(x_1, x_2)$	Function value
1	0.0, -1.0	3.0
2	-0.6, -0.4	30.0
3	1.8, 0.2	84.0
4	1.2, 0.8	840.0

Table 4 Tabu search results of function #2

Initial value $x_1, x_2$	No. of iterations (no. of evaluations)	$(x_1^*, x_2^*)$	$f(x^*)$
-2.0, -2.0	11(290)	-0.00374, -1.0007	3.003
-2.0, 2.0	99(3442)	0.00989, -0.99203	3.035
2.0, -2.0	99(3442)	0.00989, -0.99203	3.035
2.0, 2.0	428(15221)	-0.00112, -1.0010	3.000
0.0, 0.0	104(3634)	0.00448, -0.99482	3.012
a)	38775	-0.09956, -1.0	3.000

a) By Machado

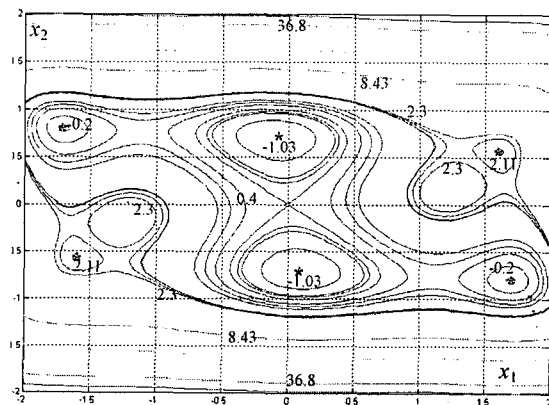


Fig. 7 Contour of function #1

초기값을 최외곽 경계점으로 설정한 경우, 반복수는 4~5 회를 보이고 있으며, 함수계산은 157~211 회가 필요하였다.

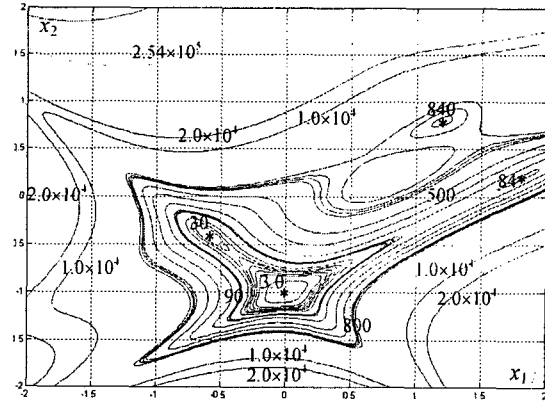


Fig. 8 Contour of function #2

Table 2 에서 초기치에 a), b)로 표시한 것은 각각 Hu 및 Machado 가 제안한 방법으로 해를 구한 것인데, 초기치 및 수렴조건이 명시되어 있지 않아 그 결과만을 수록하였다. 수렴조건이 서로 틀리기 때문에 본 논문에서 제시한 방법과 두 방법을 직접 비교하기는 어렵다. 그러나 반복횟수를 비교해보면, 본 논문에서 제시한 방법이 효율적으로 양질의 해를 구할 수 있음을 간접적으로 보여주고 있다.

4.1.2 함수 #2<sup>(12)</sup>

다음 식과 같이 2 개의 연속설계변수를 갖는 함수가 있다.

$$f(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

$$-2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2 \quad (22)$$

이 함수는 4 개의 국부최적해를 갖고 있는데, 이 함수의 등고선을 Fig. 8 에 표시하였다. 그림에서 국부최적해를 \*로 나타내었고 각 값을 Table 3 에 수록하였다.

Table 4 는 초기치를 변경해 가면서 최적해를 산출한 것이며 Machado 가 제안한 방법으로 구한 해와 비교하였다. 함수 #1 의 예제와 마찬가지로 기존의 방법이 정해에 가깝지만, 본 논문에서 제시한 방법에 의한 결과는 함수계산횟수가 급격히 감소하면서 양질의 해에 접근하고 있음을 알 수 있다. 수렴조건 상수  $\epsilon$ 는 수학적 문제 #2 에 대해 0.01 로 설정하였다.

수학함수 #2의 문제가 #1의 문제에 비해 반복횟수 및 함수계산이 많은 이유는 #2의 함수가 설계변수의 유용영역에서 비선형성이 매우 크기 때문이라고 판단된다.

4.2 구조설계문제

4.2.1.3 부재 트러스

Fig. 9 와 같은 3 부재 트러스<sup>(15)</sup>가 3 개의 다중하중조건 하에 있을 때 다음과 같은 최적화문제를 고려해 보자. 이 문제에 대한 재료상수 및 허용값은 참고문헌 (15)에 수록되어 있다.

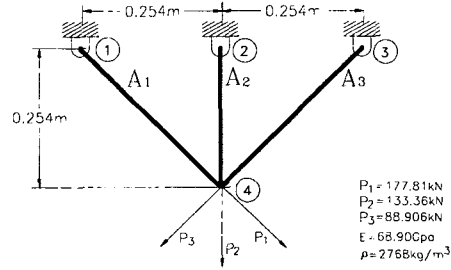


Fig. 9 Three-bar truss

**find** design variables  $A_1, A_2, A_3$   
**to minimize** mass (23)

$$|\sigma_i| \leq \sigma_{all} \quad (i=1, 2, 3)$$

**subject to**  $x$ -displacement of node 4  $\leq u_{all}$   
 $y$ -displacement of node 4  $\leq v_{all}$

$$6.452 \times 10^{-5} m^2 \leq A_i \leq 6.452 \times 10^{-2} m^2 \quad (i=1,3)$$

**candidate values**  $A_1, A_2, A_3 \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100\} \times (6.4516 \times 10^{-4}) m^2$

Table 5 Results of three-bar truss design

Initial $\times(6.4516 \times 10^{-4}) m^2$			Optimum $\times(6.4516 \times 10^{-4}) m^2$			No. of evaluations
$A_1$	$A_2$	$A_3$	$A_1$	$A_2$	$A_3$	
50	50	50	10	5	5	227
5	5	5	10	5	5	208
100	100	100	10	5	5	227

여기서  $\sigma_i, \sigma_{all}$  는  $i$  부재에 작용하는 응력과 허용응력을,  $u_{all}, v_{all}$  는  $x, y$  방향의 허용변위이다. 초기치에 따른 총합수계산 및 그 결과를 Table 5 에 표시하였다. 구조설계문제에 대하여, 수렴조건 상수  $\epsilon$  는 0.1 로 설정하였다. 최적해에서의 질량은 11.89kg 이다. 이 결과는 설계변수의  $20^3$  의 각 조합에 대하여 결과를 검토한 결과 전역적 최적해임을 알 수 있었다. Table 5 에서 볼 수 있듯이 초기값에 관계없이 일정한 해를 찾고 있다.

Fig. 10(a)-(b)는 각각 함수계산횟수와 식 (18) 로 정의 되는 특성함수값, 그리고 반복횟수와 특성함수값을 표시하고 있다. Fig. 10(a)에서 급격히 증가하는 부분은 무작위탐색 점이 불용영역에 존재하는 경우이다. Fig. 10(b)에서는  $ic$  가 증가함에 따라 해가 수렴하고 있음을 보여준다.

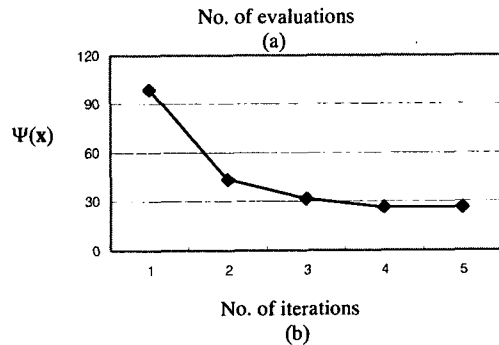
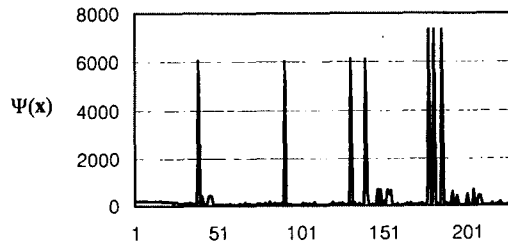


Fig. 10 Variations of characteristic function(three-bar truss)

4.2.2 10 부재 트러스

Fig. 11 과 같이 10 부재 트러스 구조물의 단면적을 결정하는 문제이다. 물성치는 탄성계수( $E$ ) 68.9GPa, 포아송비( $\nu$ ) 0.33, 밀도( $\rho$ ) 2770  $kg/m^3$  을 사용하였고, 문제의 정식화는 다음과 같다.<sup>(7)</sup>

**find** design variables  $A_i$   
**to minimize** mass (24)

**subject to**  $-172.4MPa \leq \sigma_i \leq 172.4MPa$

**candidate values**

$$A_i (\times 10^{-6} m^2) \in \{60, 970, 1940, 3230, 4000, 4650, 5420, 6450, 7290, 8130\} \quad (i=1, 2, \dots, 10)$$

**initial values** :  $[A_1, A_2, \dots, A_{10}] = [7290, 7290, 7290, 7290, 4650, 4650, 4650, 4650, 4650, 4650] (\times 10^{-6} m^2)$

이 설계 문제는  $10^{10}$  의 조합에 대하여 검토가 되어야 하는 조합최적화 문제이지만, 이산설계법을 이용하면 함수계산횟수를 감소시킬 수 있다. Table 6 은 제시된 방법, 직교배열표를 연속적으로 이용한 방법, 유전알고리즘<sup>(16)</sup>을 이용한 해를 각각 비교한 것이다. 이때 유전알고리즘의 조건은 참고문헌 (7)에 수록되어 있다.

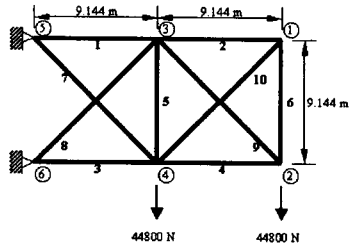


Fig. 11 Ten-bar truss

Table 6 Results of ten-bar truss design

D.V. ( $\times 10^{-6} \text{m}^2$ )	L <sub>27</sub> iteration	Genetic Algorithm	Tabu
A <sub>1</sub>	5420	5420	5420
A <sub>2</sub>	970	970	970
A <sub>3</sub>	5420	970	5420
A <sub>4</sub>	3230	5420	1940
A <sub>5</sub>	4650	4650	970
A <sub>6</sub>	970	60	970
A <sub>7</sub>	4000	970	4000
A <sub>8</sub>	4000	4650	4000
A <sub>9</sub>	4650	4650	3230
A <sub>10</sub>	970	60	1940
Mass (kg)	<u>996.8</u>	<u>813.0</u>	<u>869.2</u>
No. of evaluations	<u>532</u>	<u>5369</u>	<u>646</u>

본 논문에서 제시한 방법은 유전자 알고리즘에 비해 약 7%의 값이 크게 나오고 있지만 함수계산횟수는 약 1/8 배로 감소하였다. 이 결과는 4 회의 반복 만에 수렴한 것이다.

4.2.3 전기자동차프레임

Fig. 12 와 같이 전기자동차의 스페이스 프레임의 제 1 차 고유진동수를 고려한 최적설계 문제이다. 설계변수로는 구조물의 부재인 사각형 중공빔의 두께를  $t_1, t_2, t_3, t_4, t_5, t_6$  으로 6 개로 설정하였다. 나머지 단면의  $B_i$  와  $H_i$  는 일정하게 고정하였다. 최적화를 위한 정식화는 다음과 같다.<sup>(7)</sup>

**find** design variables  $t_1, t_2, t_3, t_4, t_5, t_6$   
**to minimize** mass  
**subject to** 1st natural frequency  $f_1 \geq 29\text{Hz}$  (25)  
**candidate values**  
 $t_1 (\times 10^{-3}\text{m}) \in \{4.2, 4.5, 4.7, 5.0, 5.2, 5.5, 5.7, 6.0, 6.2, 6.5, 6.7, 7.0\}$   
 $t_2, t_3, t_4, t_5, t_6 (\times 10^{-3}\text{m}) \in \{3.2, 3.5, 3.7, 4.0, 4.2, 4.5, 4.7, 5.0, 5.2, 5.5, 5.7, 6.0\}$   
**initial values** :  $[t_1, t_2, t_3, t_4, t_5, t_6] = [6.7, 5.7, 5.7, 5.7, 5.7, 5.7] (\times 10^{-3}\text{m})$

Table 7 은 제시된 방법, 직교배열표를 연속적으로 이용한 방법, 유전자 알고리즘을 이용한 해를 각

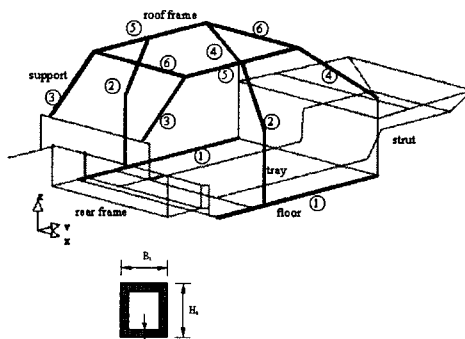


Fig. 12 Electrical vehicle frames

Table 7 Results of space frame design

D.V. ( $\times 10^{-3}\text{m}$ )	L <sub>18</sub> iteration	Genetic Algorithm	Tabu
t <sub>1</sub>	5.0	5.2	5.0
t <sub>2</sub>	4.2	3.7	4.0
t <sub>3</sub>	3.2	4.7	3.5
t <sub>4</sub>	3.7	3.2	3.5
t <sub>5</sub>	4.0	3.5	3.2
t <sub>6</sub>	4.5	5.2	3.5
Mass (kg)	<u>171.78</u>	<u>171.68</u>	<u>169.9</u>
No. of evaluations	<u>418</u>	<u>2901</u>	<u>451</u>

각 비교한 것이다.<sup>(7)</sup> 이 문제에서는 타부탐색에 의한 해가 가장 우수하게 산출되었으며 함수계산횟수도 직교배열표를 이용한 경우와 큰 차이를 보이고 있지 않다.

5. 결론 및 향후과제

본 논문에서는 타부탐색의 개념을 이용하여 구조설계 시, 효율적인 함수계산횟수를 가지고 이산설계 해를 산출할 수 있는 알고리즘을 개발하였다

(1) 제시된 알고리즘은 연속설계변수를 갖는 문제에 대해서 기존의 타부탐색법과 비교하면, 해의 정확도는 떨어지는 반면 함수계산횟수가 감소함을 알 수 있었다. 그리고 초기값에 관계 없이 전역적 최적해를 찾고 있다.

(2) 구조물의 이산설계 시, 제시된 알고리즘은 직교배열표를 반복사용한 경우보다는 약 10~20%의 함수계산이 더 많아지는 반면, 우수한 최적해를 찾고 있다. 직교배열표를 반복 사용하였을 때는 초기값에 따라서 그 근방의 최적해가 달라질 수 있다.

제시된 방법을 유전자 알고리즘과 비교하면, 10 부재 트러스는 유전자 알고리즘이 우수하게, 전기자동차 스페이스 프레임 문제는 제시된 방법이 우수하게 산출되고 있다. 이는 이산 설계변수의 배



치간격 정도에 따라 의존된다. 반면 함수계산횟수 측면에서 보면 타부탐색법이 우수하다고 볼 수 있다.

(3) 타부탐색은 설계변수의 영역이 넓어지거나 설계변수가 증가하게 되면 함수계산 횟수가 급격히 늘어난다. 그러나 본 논문의 예제 검토 결과 이산설계변수의 문제에서는 타 방법보다 비교적 우수한 결과를 보이고 있다.

(4) 타부탐색은 다른 통계적 방법과 함께 사용할 수 있는 알고리즘이다. 본 논문에서 제시한 방법은 시뮬레이티드 어닐링(simulated annealing) 등의 방법과 연계가 가능할 것이다.

## 후 기

이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2002-003-D00032).

## 참고문헌

- (1) Arora, J. S. and Haug, M. W., 1994, "Methods for Optimization of Nonlinear Problems with Discrete Variables: A Review," *Structural Optimization*, Vol. 8, pp. 69~85.
- (2) Lee, K. H. and Park, G. J., 2002, "Robust Optimization in Discrete Design Space for Constrained Problems," *AIAA J.*, Vol. 40, No. 4, pp. 774~780.
- (3) Park, G. J. and Arora, J. S., 1987, "Role of Database Management in Design Optimization System," *Journal of Aircraft*, Vol. 24, No. 11, pp. 745~750.
- (4) Salajegheh, E. and Vanderplaats, G.N., 1993, "Optimum Design of Structures with Discrete Sizing and Shape Variables," *Structural Optimization*, Vol. 6, No. 2, pp. 79~85.
- (5) Rajeev, S., and Krishnamoorthy, C. S., 1992, "Discrete Optimization of Structures Using Genetic Algorithms," *Journal of Structural Engineering, ASCE*, Vol. 118, pp. 1233~1250.
- (6) Ku, J. K., Rao, S. S. and Chen, L., 1998, "Taguchi-aided Search Method for Design Optimization of Engineering Systems," *Engineering Optimization*, Vol. 30, pp. 1~23.
- (7) Lee, J. W., Park, J. S., Lee, K. H. and Park, G. J., 2001, "Development of an Optimization Algorithm Using Orthogonal Arrays in Discrete Design Space," *Transaction of the KSME (A)*, Vol. 25, No. 6, pp. 1621~1626.
- (8) Glover, F., 1989, "Tabu Search-Part I," *ORSA Journal of Computing*, Vol. 1, No. 3, pp. 190~206.
- (9) Kim Y. K., Yoon, B. S. and Lee, S. B., 1997, *Meta-Heuristic-Genetic Algorithm Simulated Annealing Tabu Search*, Young-Ji.
- (10) Hu, N., 1992, "Tabu Search Method with Random Moves for Globally Optimal Design," *Int. J. for Numerical Methods in Eng.*, Vol. 35, pp. 1055~1067.
- (11) Dhingra, A. K. and Bennage, W. A., 1995, "Discrete and Continuous Variable Structural Optimization Using Tabu Search," *Engineering Optimization*, Vol. 24, pp. 177~196.
- (12) Machado, J. M., Shiyu, Y., Ho, S. L. and Peihong, N., 2001, "A Common Tabu Search Algorithm for the Global Optimization of Engineering Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, pp. 3501~3510.
- (13) Yang, B. S., Choi, B. K., Jeon, S. B. and Yu, Y. H., 1998, "Optimum Design of Damping Plate Using Combined Optimization Algorithm by Genetic Algorithm and Random Tabu Search Method," *Transaction of the KSME (A)*, Vol. 22, No. 7, pp. 1258~1266.
- (14) Cho, J. H. and Youn, S. K., 1999, "A Study of Layout Design of Chips on PCB Considering Temperature Distribution," *Transaction of the KSME (A)*, Vol. 23, No. 10, pp. 1637~1645.
- (15) Arora, J.S., *Introduction to Optimum Design*, McGraw- Hill Book Company, New York, 1989.
- (16) Engineous Software, Inc., 2000, *iSIGHT Designer's Guide*, Ver. 5.0.