

퍼지 simplex search 알고리즘을 이용한 동력 스크류 효율의 최적설계

Optimum Design of Power Screw Efficiency by Fuzzy Simplex Search Algorithm

현 창 현* 이 병 기**
Hyun, Chang-Hun Lee, Byeong-Ki

Abstract

The Nelder-Mead simplex algorithm has been one of the most widely used methods for the nonlinear unconstrained optimization, since 1965. Recently, the new algorithm, (so-called the Fuzzy Simplex Algorithm), with fuzzy logic controllers for the expansion, reflection and contraction process of this algorithm has been proposed. In this paper, this new algorithm is developed. And, the formulation for the optimum design of the power screw's efficiency is made. And then, the developed fuzzy simplex algorithm as well as the original one is applied to this optimum design problem. The Fuzzy simplex algorithm results in a faster convergence in this problem, as reported in other study, too.

키워드 : 심플렉스 알고리즘, 동력스크류

Keywords : Nelder-Mead simplex algorithm, Fuzzy simplex algorithm, Power screw efficiency

1. 서 론

퍼지논리[1]는 언어로 애매하게 기술된 지식을 수학적으로 취급할 수 있도록 하는 수단이다. 이 수단이 제어공학에는 널리 받아들여져서, 제어대상 의 수학적 모델을 구하기 어려워 기존의 제어 기 법으로는 다루기 곤란한 제어분야에 성공적으로 많이 응용되어 왔다[2]. 반면, 최적화 분야에는 그 다지 활용되지 못하여서, 알려진 연구로는 Rao[3], Diaz[4], Wood et al.[5], Wood 와 Antonsson[6], Thurston 와 Carnahan[7], Carnahan et al.[8], Vadde et al.[9], Ekel et al.[10] 그리고 Ohkubo와 Dissanayake[11]등이 있을 정도이다. 그 내용들은

최적화 문제 속의 목적함수, 제약조건 또는 설계변 수에 포함된 애매한 내용을 기술하기 위한 목적으 로 퍼지논리 적용을 시도한 것, 그리고 하나의 최 적화 문제 속에 두 개 이상의 목적함수가 있는 경 우 이를 효과적으로 결합하기 위한 방법으로 퍼지 논리를 사용한 것들이다.

수치 최적화 알고리즘 속에 나타나는 대다수의 매개변수들은 경험적으로 정해진다. 그리고, 그 값 들은 대부분 고정시켜 사용한다. 이렇게 고정시켜 놓으면 일반적으로 그 효율성은 해결하고자 하는 최적화 문제에 따라 좋을 수도 있고 나쁠 수도 있 다. 즉, 목적함수 또는 제약조건 또는 초기시작점 의 값을 변경시키면, 그 고정된 매개변수 값의 효 율성 역시 변동하여 최적 값으로의 수렴속도 및 정밀도등이 달라진다. 이렇게 그 값의 의미가 희박 하게 되는 일이 흔하다. 이러한 경우 퍼지 논리를 사용하여 매개변수들을 상황에 따라 좀 더 융통성 있게 변화시킨다면 원하는 답을 효율적으로 정확

* 강원대학교 기계메카트로닉스공학부 교수

** 강원대학교 대학원 기계메카트로닉스공학과 석사과정

하게 얻을 수 있다. 이런 착상을 채택한 연구로 다음의 것들이 있다. Grignon and Fadel [12] 은 퍼지알고리즘을 적용하여, 2 점 지수법에서 이동 경계점을 변경시키는 방법을 제안하였고, Mulkey and Rao[13]는 순차 선형계획법(sequential linear programming) 알고리즘 속의 매개변수들을 조정하기 위하여 퍼지알고리즘을 적용하는 법을 제시하였다. 그리고 Trabia et al [14]의 연구에서는 Nelder-Mead simplex 알고리즘 속에 나타나는 매개변수들을 고정된 값이 아니라 상황에 따라 적응하여 변경된 값을 가지도록 그 기능을 제공함으로써 좀 더 효율적(예: 함수계산 횟수를 줄임)으로 탐색할 수 있도록 하고자 하였다.

본 연구에서는 이 simplex search 알고리즘과 관련된 문제를 다룰 것이므로 좀 더 구체적으로 이를 살펴 보기로 한다. 이 알고리즘은 최적화 문제 속의 함수들이 복잡한 비선형 또는 불연속인 경우도 다룰 수 있고, 단순하기 때문에 현재까지도 애용되고 있다. Spendley et al. [15]이 처음 제안하고, Nelder-Mead [16] 에 의해서 개량되었으며, Rekalitis et al. [17], Rao [18] 등 여러 연구자들에 의해서 다루어져 왔다.

이 알고리즘의 내용을 살펴 볼 때, 가장 높은 함수 값을 투영 시켜서 새로운 점을 찾는 작업에서 확장 또는 축소량을 정하는 기준이 명확히 제시되어 있기는 하다. 하지만 그 속에 각 꼭지점에서의 함수 값들의 상대적인 크고 작음이 고려되어 있지 않다. 우선 확장해야 될 경우를 좀 더 살펴 본다면, 가장 높은 함수 값에 대응하는 simplex 꼭지점과 나머지 꼭지점들의 좌표 값을 이용해서 산출된 중심까지의 길이를 2 배로 확장한다. 각 꼭지점에서의 함수 값들간의 상대적 크기에 대한 고려 없이 무조건 그렇다. 만약 이 함수 값들간의 상대적인 크기를 고려해서 확장의 정도를 조절하면 조금 더 합리적일 것이다. 즉, 현재 simplex 꼭지점들의 함수 값 중에서 가장 작은 함수 값이, 가장 큰 함수 값을 투영하여 생성된 새로운 꼭지점에서의 함수 값보다도 훨씬 크면 확장은 훨씬 더 크게 하는 것이 좋다고 말할 수 있다. 한편 조금 크면 확장은 적당히 중간 정도로 하는 것이 좋다고 말할 수 있다. 그런데 이렇게 애매하게 표현되기 때문에 입력에 해당되는 전제 부와 출력에 해당되는 결론 부 사이에 관한 관계에 대해 정량적 기술이 곤란하다. 하지만, 이러한 애매한 입출력 정보를 처리 할 수 있는 퍼지논리 제어기를 제작하여 이용하면, 그 애매한 표현도 수학적으로 다룰 수 있게 된다. 이상의 내용은 축소, 투영의 경우도 같다.

본 연구에서는 우선 Trabia 와 Xiao Bin Lu[14]이 제안한 퍼지 논리 제어기를 채용하여 성능을 개선시킨 simplex search 알고리즘을 개발한다. 그리

고, 이를 사용하여 아래에 그 연구 필요성이 기술되는 동력 스크류의 효율의 최적설계에 적용해 보고자 한다.

동력 스크류는 기계의 회전운동을 직선운동으로 변환 또는 역 변환하는 기구로서 공작기계, 로봇 등 많은 기계에 채용되고 있으며 그 설계 법이 대학교과서에 수록될 정도로 기본적인 기구이다. 그 설계내용 중 효율계산 관련 사항은 특히 중요하여 그 계산식이 제공되고 있다[20]. 하지만, 주어진 설계조건 상황이 복잡한 경우 그 조건을 만족시키면서 가장 높은 효율 값을 찾아내려면 그것만으로는 안 된다. 최적화 알고리즘을 사용하여야 만 한다. 이러한 연구가 놀랍게도 아직까지 발표된 것이 없다. 이 문제에는 설계변수가 불연속인 것들이 포함되어 있어서 미분법을 이용하는 최적화 알고리즘은 사용하기 곤란하다. 이런 문제에 적합한 simplex search 알고리즘, 그것도 퍼지이론을 응용하여 성능이 개선된 simplex search 알고리즘을 사용하여 효율의 최적설계를 시도한 예가 없음은 물론이다. 최적화문제의 목적함수 또는 설계변수들이 불연속인 경우는 범위가 너무 넓어서 본 연구에서는 모두 연속인 경우로 국한한다. 이것 때문에 연구결과가 실용성이 좀 떨어질 수도 있겠으나 퍼지이론의 최적설계 응용에 미력하나마 일조하고, 좀 더 좋은 기계들이 만들어 질 수 있기 위해서는 바람으로 이러한 연구를 시도 해 본다.

2. 퍼지 simplex algorithm

기존 simplex 알고리즘[16]의 1 회전 수행중 중요 내용은 다음과 같이 요약된다. 독립변수 개수가 n 이라고 하자. 그러면 초기 시작점이 주어지고 맨 처음의 simplex 생성용 매개변수 값 (a)이 주어지면 regular simplex가 생성된다. 이것의 각 꼭지점에서 함수 값 계산을 하고, 크기 순으로 다음과 같이 정렬한다.

$$f_1 \leq f_2 \leq \dots \leq f_n \leq f_{n+1}$$

단, $f_i = f(\mathbf{x}_i)$

그 다음 \mathbf{x}_{n+1} 점을 제외하고 나머지 점들에 대한 중심 값 $\bar{\mathbf{x}}$ 를 구한다. \mathbf{x}_{n+1} 과 $\bar{\mathbf{x}}$ 를 잇는 직선 상에서 \mathbf{x}_{n+1} 의 투영된 점(reflection point)인 \mathbf{x}_r 을 다음 식으로 계산한다.

$$\mathbf{x}_r = \bar{\mathbf{x}} + \rho(\bar{\mathbf{x}} - \mathbf{x}_{n+1}) \quad (1)$$

여기서 $\rho =$ 초기 투영계수

다음에 $f_r = f(\mathbf{x}_r)$ 을 계산한다. 이 값이 f_1, f_n, f_{n+1} 사이 어느 위치에 속하느냐에 따라 확장 축소과정을 결정한다.

확장 과정에서는 다음의 식을 이용하여 확장 점 \mathbf{x}_e 를 계산한다.

$$\mathbf{x}_e = \bar{\mathbf{x}} + \xi(\mathbf{x}_r - \bar{\mathbf{x}}) \quad (2)$$

여기서 $\xi =$ 확장계수

축소과정은 외부 축소과정과 내부 축소과정으로 나뉜다. 외부 축소 점 \mathbf{x}_c 는 다음 식으로 계산한다.

$$\mathbf{x}_c = \bar{\mathbf{x}} + \gamma(\mathbf{x}_r - \bar{\mathbf{x}}) \quad (3)$$

여기서 $\gamma =$ 축소계수

내부 축소 점은 다음 식으로 계산한다.

$$\mathbf{x}_{cc} = \bar{\mathbf{x}} - \gamma(\bar{\mathbf{x}} - \mathbf{x}_{n+1}) \quad (4)$$

2.1 퍼지 simplex algorithm 에서 투영과정

퍼지 simplex 알고리즘에서의 투영은, 위의 내용 중 식 (1)에서 $\bar{\mathbf{x}}$ 대신 다음의 식으로 계산 $\bar{\mathbf{x}}_{cf}$ 를 사용한다[14].

$$\bar{\mathbf{x}}_{cf} = (1 - \theta_c) \bar{\mathbf{x}} + \theta_c \mathbf{x}_1 \quad (5)$$

여기서 θ_c 는 퍼지논리 제어기에 y_h 가 입력되면 출력되는 값이다. y_h 는 다음과 같이 정의된다.

$$y_h = \left| \frac{f(\mathbf{x}_{n+1}) - f(\mathbf{x}_1)}{f(\mathbf{x}_{n+1})} \right| \quad (6)$$

단, $f(\mathbf{x}_{n+1}) =$ simplex 의 가장 높은 함수 값

$f(\mathbf{x}_1) =$ simplex 의 가장 낮은 함수 값

그 의미는 어떤 simplex 의 가장 큰 함수 값과 가장 작은 함수 값 사이의 차이 즉 상대적인 높낮이 척도가 된다. 이 값으로 현재의 simplex가 얼마만큼 기울어 졌나 짐작하고 꼭지점들의 중심 $\bar{\mathbf{x}}$ 를 함수 값이 가장 낮은 꼭지점 \mathbf{x}_1 쪽으로 적당량 이동시키려는 것이다. 그 양은 제어기의 출력 θ_c 로 정해진다. 그러면 함수 값이 가장 높은 꼭지점 \mathbf{x}_{n+1} 과 이동되어 새로이 생성된 중심

$\bar{\mathbf{x}}_{cf}$ 를 서로 연결하는 직선이 좀 더 낮은 곳으로 향하여 진행되고, 결국 좀 더 신속히 목적함수의 최소 값에 수렴하게 된다.

2.2 투영에 쓰이는 퍼지논리 제어기 특성

입력 :

퍼지 제어기에서의 자료의 처리는 퍼지 집합론을 기반으로 행해지므로 식(6)으로 주어지는 수치적인 입력 y_h 값들의 퍼지화가 필요하다. 즉, y_h 를 퍼지 제어기 내에서 사용하려면, 그것을 0과 1 사이의 어떤 값, 즉 퍼지 값으로 환산해 주어야 한다. 그러기 위해선 그 소속의 정도를 정해주는 소속함수를 제공해 주어야 한다. 이를 위해 사용된 소속함수는 가우스 곡선 형태의 것 식(7)을 사용하였다.

$$\mu(z, \sigma, c) = e^{-z^2 / 2\sigma^2} \quad (7)$$

y_h 의 경우에는 3개의 영역 즉, Z_{yr} : 0.0 ~ 0.5, M_{yr} : 0.5 ~ 0.92, B_{yr} : 0.92 ~ 1.0으로 퍼지분할 하였다. 각 분할영역에 해당되는 소속함수의 구체적 형태는 다음과 같다.

$$Z_{yr} : \mu = e^{-(y_h)^2 / (2 \cdot 0.2097^2)}$$

$$M_{yr} : \mu = e^{-(y_h - 0.7)^2 / (2 \cdot 0.0466^2)}$$

$$B_{yr} : \mu = e^{-(y_h - 1)^2 / (2 \cdot 0.0465^2)}$$

퍼지 투영 제어규칙:

위의 입력 y_h 를 다음의 제어규칙으로 처리한다.

1. y_h 가 Z_{yr} 에 소속되면 θ_c 값은 Z_r 에 소속된다
2. y_h 가 M_{yr} 에 소속되면 θ_c 값은 M_r 에 소속된다
3. y_h 가 B_{yr} 에 소속되면 θ_c 값은 B_r 에 소속된다

출력:

출력 θ_c 은 위의 제어규칙에 의해 입력 y_h 를 변환 처리하여 산출된다. 마찬가지로 소속 함수 형태를 식(7)과 같이 사용하였다. θ_c 의 범위는 0에서 0.05사이로 정하였고 3 영역으로 퍼지분할 하였다. 각 영역마다의 소속함수의 구체적 형태는 다음과 같다.

$$Z_r : \mu = e^{-(\theta_c)^2 / (2 \cdot 0.09893^2)}$$

$$M_r : \mu = e^{-(\theta_c - 0.035)^2 / (2 \cdot 0.002329^2)}$$

$$B_r : \mu = e^{-(\theta_c - 0.05)^2 / (2 \cdot 0.002329^2)}$$

이상의 식(7)로 부터 얻어진 입출력 소속함수의 위치 c 와 크기 σ 는 수많은 실험을 통하여 적당하다고 결론지어진 값들이다.

입력 y_h 와 θ_c 의 관계를 비 퍼지화 된 수의 값으로 나타내고 이들의 관계를 그려보면 Fig. 1과 같다.

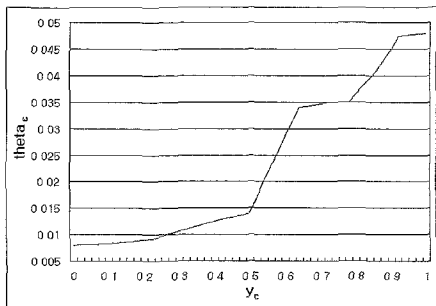


Fig. 1. Relation between the input and the output of the fuzzy controller (Reflect)

2.3 퍼지simplex algorithm에서의 확장과 축소과정

Nelder-Mead는 수많은 실험을 통하여, 기존 simplex algorithm의 확장계수(식(2)의 ξ) $\theta_{ex} = 2.0$, 축소계수(식(3)과(4)의 γ) $\theta_{co} = -0.5$ 값이 가장 좋은 측에 속한다고 결론을 내리고 이 값을 고정시켜 놓고 최적화문제의 종류에 관계없이 사용하였다. 그런데, 어떤 문제의 경우는 이것이 수렴의 신속성에 별로 도움이 안 된다. 본 연구에서는 문제의 상황에 적용하여 변할 수 있게 하였다. 즉, 두 개의 퍼지논리 제어를 설치하여 상황에 따라 입력변수 값이 주어지면 그에 대응하여 출력으로 $\theta_{ex} = 0 \sim 2.0$ $\theta_{co} = 0 \sim -0.5$ 사이의 어떤 값을 취할 수 있도록 하였다.

2.4 확장용 퍼지논리 제어기 특성

입력:

입력은

$$y_{new} = \left| \frac{f(\mathbf{x}_{n+1}) - f(\mathbf{x}_r)}{f(\mathbf{x}_{n+1})} \right| \quad (8)$$

$$Dy_{new} = \frac{f(\mathbf{x}_r(i)) - f(\mathbf{x}_r(i-1))}{f(\mathbf{x}_r(i))} \quad (9)$$

단, $f(\mathbf{x}_{n+1})$ =simplex 의 가장 큰 함수 값

$f(\mathbf{x}_r)$ =simplex 의 가장 큰 점이 투영된 점에서의 함수 값

i = 반복수

으로 하였다. (8)식은 투영되기 전 simplex 에서 계산된 가장 높은 함수 값과 투영된 후 확장 또는 축소중 어느 것으로 해야하는 지에 관한 조정이 이루어진 그 결과물로서 새로이 정해진 점에서의 함수 값을 빼고, 그 결과를 투영되기 전 simplex

에서 계산된 가장 큰 함수 값으로 나는 것이다. (9)식은 연속된 사이클에서 채워진 새로운 점들의 진행정도를 측정하기 위하여 정의 한 것이다. 그 소속 함수들의 형태는 다음과 같이 정의하였다.

(8)식에 대하여:

$$Z_{ye} : \mu = e^{-\frac{(y_{new})^2}{(2 \cdot 0.0186^2)}} \quad (0 \leq y_{new} < 0.1)$$

$$M_{ye} : \mu = e^{-\frac{(y_{new}-0.1)^2}{(2 \cdot 0.0232^2)}}$$

$$\mu = 1 \quad (0.1 \leq y_{new} < 0.8)$$

$$\mu = e^{-\frac{(y_{new}-0.8)^2}{(2 \cdot 0.0465^2)}}$$

$$B_{ye} : \mu = e^{-\frac{(y_{new}-1)^2}{(2 \cdot 0.0493^2)}} \quad (0.8 \leq y_{new} < 1.0)$$

(9)식 에 대하여:

$$N : \mu = e^{-\frac{(Dy_{new}+0.01)^2}{(2 \cdot 0.000465^2)}}$$

$$\mu = 1 \quad (-0.01 \leq Dy_{new} < -0.008)$$

$$Z : \mu = e^{-\frac{(Dy_{new}+0.008)^2}{(2 \cdot 0.000465^2)}}$$

$$\mu = 1 \quad (-0.008 \leq Dy_{new} < 0.0075)$$

$$\mu = e^{-\frac{(Dy_{new}-0.0075)^2}{(2 \cdot 0.000966^2)}}$$

$$P : \mu = e^{-\frac{(Dy_{new}-0.01)^2}{(2 \cdot 0.000931^2)}}$$

$$\mu = 1 \quad (0.0075 \leq Dy_{new} < 0.01)$$

퍼지 확장 제어규칙:

위의 입력을 다음의 제어규칙으로 처리한다.

1. y_{new} 가 Z_{ye} 이고 Dy_{new} 가 N이면, θ_{ex} 는 PL
2. y_{new} 가 Z_{ye} 이고 Dy_{new} 가 Z이면, θ_{ex} 는 PM
3. y_{new} 가 Z_{ye} 이고 Dy_{new} 가 P이면, θ_{ex} 는 PM
4. y_{new} 가 M_{ye} 이고 Dy_{new} 가 N이면, θ_{ex} 는 PL
5. y_{new} 가 M_{ye} 이고 Dy_{new} 가 Z이면, θ_{ex} 는 PL
6. y_{new} 가 M_{ye} 이고 Dy_{new} 가 P이면, θ_{ex} 는 PL
7. y_{new} 가 B_{ye} 이고 Dy_{new} 가 N이면, θ_{ex} 는 PXL
8. y_{new} 가 B_{ye} 이고 Dy_{new} 가 Z이면, θ_{ex} 는 PXL
9. y_{new} 가 B_{ye} 이고 Dy_{new} 가 P이면, θ_{ex} 는 PL

출력:

입력변수 값을 확장제어기에 주어진 규칙으로 처리한 출력은 $\theta_{ex} = 0 \sim 2.0$ 사이의 어떤 값을 취한다. 소속함수의 형태는 다음과 같이 정의되었다.

$$PM : \mu = e^{-\frac{(\theta_{co}-1.8)^2}{(2 \cdot 0.00931^2)}} \quad (1.8 \leq \theta_{co} < 1.85)$$

$$PL : \mu = e^{-\frac{(\theta_{co}-1.85)^2}{(2 \cdot 0.00931^2)}}$$

$$\mu = 1 \quad (1.85 \leq \theta_{co} < 2.15)$$

$$\mu = e^{-\frac{(\theta_{co}-2.15)^2}{(2 \cdot 0.0295^2)}}$$

$$PXL : \mu = e^{-\frac{(\theta_{co}-2.2)^2}{(2 \cdot 0.0206^2)}} \quad (2.15 \leq \theta_{co} < 2.2)$$

입력과 출력의 관계는 Fig 2 에 보였다.

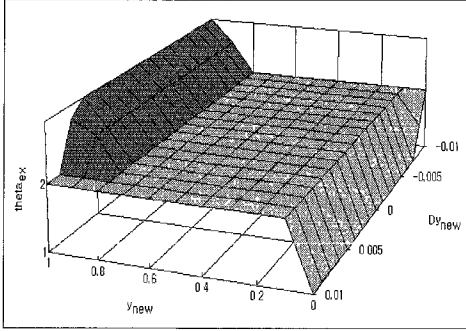


Fig. 2. Relation between the input and the output of the fuzzy controller (Expand)

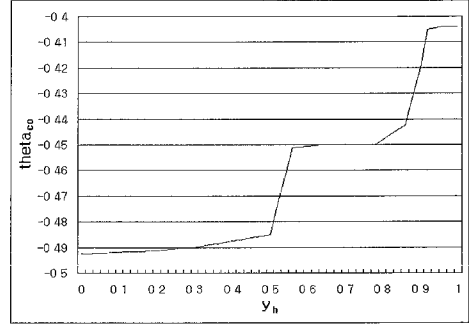


Fig. 3. Relation between the input and the output of the fuzzy controller (Contract)

2.5 축소용 퍼지논리 제어기 특성

입력:

제어기 입력은 (6)과 같은 식을 사용한다. 소속함수의 형태는 다음과 같이 정의되었다. 투영 제어기에서 사용된 것과는 다르다.

$$\begin{aligned} Z_{yc} : \mu &= e^{-(y_h)^2/(2 \cdot 0.2097^2)} & (0 \leq y_h < 0.5) \\ M_{yc} : \mu &= e^{-(y_h - 0.7)^2/(2 \cdot 0.0466^2)} & (0.5 \leq y_h < 0.9) \\ B_{yc} : \mu &= e^{-(y_h - 1)^2/(2 \cdot 0.0465^2)} & (0.9 \leq y_h < 1) \end{aligned}$$

퍼지 축소 제어규칙:

위의 입력을 다음의 제어규칙으로 처리한다.

1. y_h 가 Z_{yc} 이면, θ_{co} 는 NB
2. y_h 가 M_{yc} 이면, θ_{co} 는 NM
3. y_h 가 B_{yc} 이면, θ_{co} 는 NS

출력:

출력 θ_{co} 의 소속함수는 다음과 같이 정의되었다. $\theta_{co} = 0 - -0.5$ 사이의 어떤 값을 출력으로 취할 수 있도록 하였다.

$$\begin{aligned} NB : \mu &= e^{-(\theta_{co} + 0.5)^2/(2 \cdot 0.009893^2)} & (-0.5 \leq \theta_{co} < -0.48) \\ NM : \mu &= e^{-(\theta_{co} + 0.45)^2/(2 \cdot 0.01477^2)} & (-0.48 \leq \theta_{co} < -0.41) \\ NS : \mu &= e^{-(\theta_{co} + 0.4)^2/(2 \cdot 0.00465^2)} & (-0.41 \leq \theta_{co} < -0.4) \end{aligned}$$

입력과 출력의 관계는 Fig. 3 와 같다

2.6 수렴 판정 조건

본 연구에 사용된 수렴판정조건은 다음과 같다. 여기서 n 은 변수의 개수, ϵ 는 허용오차를 의미한다.

$$\sqrt{\frac{\sum_{i=0}^n \left(f(x_i) - \frac{\sum_{i=0}^n f(x_i)}{n+1} \right)^2}{n+1}} < \epsilon \quad (10)$$

2.7 퍼지 simplex algorithm 순서도

이상의 내용을 Fig. 4.에 순서도로 정리해 놓았다.

3. 동력 스크류에의 적용

3.1 하중방향과 반대로 스크류를 전진시킬 때 동력 스크류의 효율

효율 계산식은 다음과 같다[18]

$$\begin{aligned} e &= \frac{1 - \mu \tan \lambda}{1 + \mu \cot \lambda} \\ &= \frac{1 - \mu \tan \lambda}{\mu + \tan \lambda} \times \tan \lambda \end{aligned} \quad (11)$$

단, λ = 리이드각

μ = 스크류와 너트사이의 마찰계수

그리고 하중을 받는 스크류가 파손을 면하려면, 비틀림 에너지 이론을 만족되어야 한다. 즉, 다음의 식이 성립되어야 한다

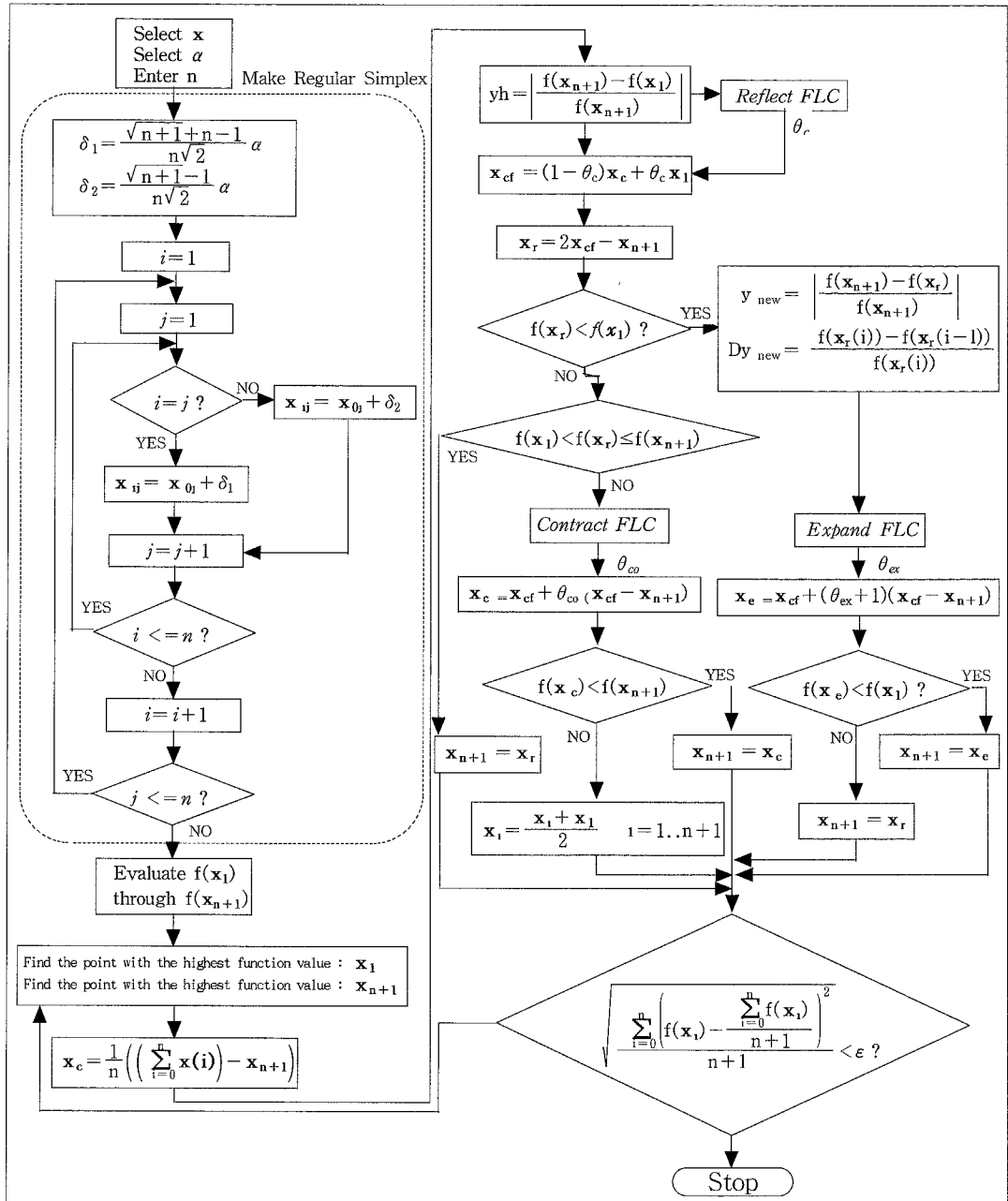


Fig.4. Flowchart for the fuzzy simplex algorithm

$$(\sigma_x^2 + 3\tau_{xy}^2)^{1/2} \leq \frac{S_y}{n} \quad (12)$$

이들 관계를 정리하면 식(11)과 (12)은 다음처럼 표현된다.

$$e = \frac{(\frac{\mu}{\pi} + \frac{1}{2})L^2 - dL}{\frac{1}{2}L^2 + (-\frac{\pi\mu}{2} - d)L - \pi\mu d} \quad (13)$$

$$(\sigma_x^2 + 3\tau_{xy}^2)^{1/2}$$

$$\begin{aligned}
 &= \frac{4}{\pi} \sqrt{\left(\frac{F}{d_r^2}\right)^2 + 48\left(\frac{T}{d_r^3}\right)^2} \\
 &= \frac{4F}{\pi(d-L)^2} \times \\
 &\quad \sqrt{1 + \frac{48}{(d-L)^2} \left[\left(\frac{d-L}{2} - \frac{L}{4}\right) \frac{\pi\mu(d-\frac{L}{2})+L}{\pi(d-\frac{L}{2})-\mu L} \right]^2} \\
 &\leq \frac{S_y}{n} \quad (14)
 \end{aligned}$$

단, σ_x = 범선 응력, F = 작용하중,
 A_r = 이뿌리직경으로 계산한 단면적
 τ_{xy} = 전단응력, T = 작용 토크, r = 반경
 d = 호칭직경, d_r = 이뿌리 직경, d_m = 퍼지직경
 p = 퍼지, L = 리이드, S_y = 재료의 인장강도
 n = 안전계수

$$\sigma_x = \frac{F}{A_r}, \quad A_r = \frac{\pi d_r^2}{4}, \quad \tau_{xy} = \frac{16T}{\pi d_r^3}$$

$$T = \frac{Fd_m}{2} \left[\frac{\pi\mu d_m + L}{\pi d_m - \mu L} \right], \quad d_r = d - L$$

$$d_m = d - \frac{L}{2}, \quad p = L, \quad \tan \lambda = \frac{L}{\pi d_m}$$

3.2 최적화 문제의 정식화

본 연구에서는 동력 스크류의 효율을 나타내는 식(11)을 설계조건 식(12)하에서 최대화시키는 문제로 한정하고, 설계변수인 호칭직경 d 와 리이드 L 은 원래 불연속의 이산 값으로 주어진다. 그러나 서론에서 언급한 것처럼 본 연구에서는 모두 연속으로 가정한다.

그 범위는

$$0.25 \leq d \leq 5.0 \text{ (in)}, \quad 0.0625 \leq L \leq 1.0 \text{ (in)}$$

이다. 그 외 μ, n, F, S_y 그리고 π 는 설계시 미리 주어지는 상수 값들이다. 구속조건을 나타내는 식(12)으로부터 다음이 성립한다

$$\begin{aligned}
 &(\sigma_x^2 + 3\tau_{xy}^2)^{1/2} \\
 &= \frac{4F}{\pi(d-L)^2} \times \\
 &\quad \sqrt{1 + \frac{48}{(d-L)^2} \left[\left(\frac{d-L}{2} - \frac{L}{4}\right) \frac{\pi\mu(d-\frac{L}{2})+L}{\pi(d-\frac{L}{2})-\mu L} \right]^2} \\
 &\quad - \frac{S_y}{n} \leq 0 \quad (15)
 \end{aligned}$$

이런 조건을 만족시키면서 다음의 효율 e 를 최대화시키는 즉, 함수 $f(d, L)$ 를 최소화시키는 설계변수 d, L 을 찾아야 한다.

$$\begin{aligned}
 f(d, L) &= -e \\
 &= (-1) \frac{\left(\frac{\mu}{\pi} + \frac{1}{2}\right)L^2 - dL}{\frac{1}{2}L^2 + \left(\frac{\pi\mu}{2} - d\right)L - \pi\mu d} \quad (16)
 \end{aligned}$$

3.3 최적화 방법

위의 정식화된 내용은 목적함수와 구속조건을 포함하고 있어서 기존 및 퍼지 simplex 알고리즘으로 처리하려면 구속조건을 목적함수로 편입시켜 구속조건이 없는 형태로 만든 다음 사용해야 한다. 여기서는 벌점함수(penalty function method) 방법을 사용한다. 원래의 목적함수가

$$f = f(\mathbf{x})$$

이며 등가 조건과 부등 조건이

$$h_i(\mathbf{x}) = 0 \quad ; \quad i = 1 \text{ to } p$$

$$g_i(\mathbf{x}) \leq 0 \quad ; \quad i = 1 \text{ to } m$$

이면, 다음의 새로운 목적함수만으로 변형된다.

$$\begin{aligned}
 \phi(\mathbf{x}, r) &= f(\mathbf{x}) + P(\mathbf{h}(\mathbf{x}), \mathbf{g}(\mathbf{x}), r) \\
 &= f(\mathbf{x}) + r \cdot \left\{ \sum_{i=1}^p [h_i(\mathbf{x})]^2 + \sum_{i=1}^m [g_i^+(\mathbf{x})]^2 \right\} \\
 &\quad (\text{단, } g_i^+(\mathbf{x}) = \max(0, g_i(\mathbf{x})), \quad r = 10^{20})
 \end{aligned}$$

벌점함수(penalty function) $P(\mathbf{h}(\mathbf{x}), \mathbf{g}(\mathbf{x}), r)$ 로는, 중괄호 속처럼, 가장 보편적인 이차 손실함수(quadratic loss function)[19]를 사용하였다. 이차 손실함수의 벌점 부과 방식은, 부등 조건과 항등 조건이 만족되는 경우 g_i 와 h_i 는 0이 되어 P 의 값이 0이 되게 하고, 만족하지 않는 경우에는 g_i 와 h_i 에는 양의 값을 주어 P 의 값이 큰 수를 가지게 하는 것으로 이루어진다.

그리하면, 본 연구의 최적화 문제는 다음과 같은 형태로 변형된다.

$$\begin{aligned}
 &\text{Minimize } \{ \phi(d, L) \} \\
 &= f(d, L) + P(\mathbf{g}(d, L), r) \\
 &= (-1) \frac{\left(\frac{\mu}{\pi} + \frac{1}{2}\right)L^2 - dL}{\frac{1}{2}L^2 + \left(\frac{\pi\mu}{2} - d\right)L - \pi\mu d} \\
 &\quad + r(g_1^2 + g_2^2 + g_3^2 + g_4^2 + g_5^2 + g_6^2 + g_7^2)
 \end{aligned}$$

여기서

$$g_1(d, L) = \frac{4F}{\pi(d-L)^2} \times \sqrt{1 + \frac{48}{(d-L)^2} \left(\left(\frac{d}{2} - \frac{L}{4} \right) \frac{\pi\mu(d - \frac{L}{2}) + L}{\pi(d - \frac{L}{2}) - \mu L} \right)^2} - \frac{S_y}{n} \quad (17)$$

$$g_2(d, L) = -d + 0.25 \quad (18)$$

$$g_3(d, L) = d - 5.0 \quad (19)$$

$$g_4(d, L) = -L + 0.0625 \quad (20)$$

$$g_5(d, L) = L - 1.0 \quad (21)$$

$$g_6(d, L) = -d \quad (22)$$

$$g_7(d, L) = -L \quad (23)$$

$$r = 10^{20}$$

그리고, d 와 L 는

$$0.25 \leq d \leq 5.0 \text{ (in)}, \quad 0.0625 \leq L \leq 1.0 \text{ (in)}$$

이다. 그 외 μ , n , F , S_y , 그리고 π 는 설계 시 미리 주어지는 상수 값들이다. 각 문자와 상수 값은 4 절에 주어진 것과 동일하다.

4. 계산 결과 및 고찰

기존의 simplex 알고리즘과 퍼지 simplex 알고리즘으로 실행시킨 결과는 Table 1과 같다. 설계변수 d 와 L 은 원래 불연속의 이산 값으로 주어진다. 그러나, 서론에서 언급한 것처럼 본 연구에서는 모두 연속으로 국한하였다. 그 범위는

$$0.25 \leq d \leq 5.0 \text{ (in)}, \quad 0.0625 \leq L \leq 1.0 \text{ (in)}$$

이다.

그 외 상수 값들은 다음과 같다.

$$\mu = 0.15 \quad n = 1 \quad F = 1000 \text{ lb}$$

$$S_y = 81000 \text{ psi} \quad \pi = 3.141592$$

수렴 판정조건식(10)은 2.6절에서 설명한 바와 같이 simplex의 중심점으로부터 각 꼭지점까지 거리의 평균을 의미한다. 수렴 판정조건의 허용오차는 $2.5e-9$ 이 만족되도록 하였다.

Table 1을 보면, 탐색 후 얻은 최대의 효율은 67.3%이고 그때의 최적설계변수는 $d = 5 \text{ (in)}$ $L = 1 \text{ (in)}$ 임을 알 수 있다.

기존 알고리즘의 경우는 2, 7, 8 번을 제외하고 최적 설계변수를 찾았음을 알 수 있다. 이 3 가지의 경우에 있어서 효율의 최대값은 최적해와 거의 가깝지만, 최적 설계변수의 d 값이 3(in)로 근접하고 있다. 설계변수의 최적값 $d=5 \text{ (in)}$ 와는 아주 다르다. 2번의 경우 $L=0.1$ 을 $L=0.11$ 로 바꾸서 출발시키면 제대로 최적 설계변수 값에 도착하였다. 출발점이 조금 밖에 차이가 없는 데도 종착역으로의 수렴에 지대한 차이를 보이고 있다. 그 이유는 더

Table 1. The Running Results of Nelder-Mead and Fuzzy Simplex

No.	Starting Point		Optimum Design Value				Max. Efficiency		Function Eval.		Iterations	
	d	L	Nelder and Mead		Fuzzy		Nelder and Mead	Fuzzy	Nelder and Mead	Fuzzy	Nelder and Mead	Fuzzy
			d	L	d	L						
1	0.7	0.7	5.00000	1.00000	5.00000	1.00000	0.67259	0.67259	190	171	96	72
2	1.0	0.1	3.10453	1.00000	5.00000	1.00000	0.66734	0.67259	125	156	61	69
3	1.0	0.5	5.00000	1.00000	4.99978	1.00000	0.67259	0.67259	192	178	96	71
4	2.0	0.2	4.99999	1.00000	4.99994	1.00000	0.67259	0.67259	173	166	85	69
5	2.0	0.7	5.00000	0.99853	5.00000	1.00000	0.67229	0.67259	158	172	79	72
6	3.0	0.3	4.99998	1.00000	5.00000	0.99981	0.67259	0.67255	147	126	73	47
7	3.0	1.0	3.00000	1.00000	4.99911	1.00000	0.66682	0.67259	209	211	104	88
8	4.0	0.4	2.79533	1.00000	4.99999	1.00000	0.66566	0.67259	140	178	69	81
9	4.0	0.9	5.00000	1.00000	4.81384	1.00000	0.67259	0.67228	175	189	88	80
10	5.0	0.5	4.99980	1.00000	5.00000	1.00000	0.67259	0.67259	438	157	219	67
Mean Value							0.67076	0.67255	195	170	97	72

깊이 연구해 보아야 하겠다. 그리고, 7번의 경우는 $L=1.0$ 즉 경계선상의 한 점인데 도착하는 최적 설계변수 값은 변함이 없었다. 함수계산 횟수를 보면 분명 수많은 simplex를 형성했는데 다른 곳으로 움직이지 않는다. L 의 경계점에서 시작하면 그 주위에서 멈춘다[17]. 자세한 관찰은 다음 연구로 미룬다. 8번의 경우도 2 번과 비슷하게 수렴한다.

퍼지 simplex 알고리즘의 경우는 9번을 제외하고 동일한 최적 값을 찾았음을 알 수 있다 함수계산 평균횟수 및 평균반복 횟수는, 기존 알고리즘으로 계산한 것보다 퍼지 쪽이 작다. 이러한 상황에서, 다음과 같이 결론지을 수 있다.

Nelder-Mead simplex 알고리즘을 사용한 것보다 퍼지 simplex 알고리즘 수행속도가 빠르며 시작점의 위치에 덜 민감하게 최적 값을 찾아낼 수 있다. 그래서, 기존 알고리즘보다 신뢰성과 효율이 높음[14]이 다시 입증되었다.

5. 결 론

기존의 Nelder-Mead simplex 알고리즘에 퍼지 논리 제어를 채용하여 확장, 투영, 축소에 대응하는 매개변수 값을 출력시키는 퍼지논리 simplex 알고리즘을 개발하고, 이를 동력 스크류의 효율 극대화 문제에 적용하는 과정에서 다음의 결론을 얻었다.

- (1) 주어진 설계조건 하에서 동력 스크류의 효율 극대화에 관해 발표된 연구는 놀랍게도 현재까지 없었는데, 본 연구에서 그 문제를 정식화하여 제시하였다. 그 형태는 고도의 비선형 불연속 함수가 포함된 구속조건을 가진 최적화 문제임을 알 수 있었다.
- (2) 퍼지논리 제어를 채용한 simplex 알고리즘의 국내개발을 처음으로 이루었다. 이것을 동력 스크류 효율 극대화 문제에 적용해 본 결과 기존의 Nelder-Mead simplex 알고리즘을 사용한 것보다 수행속도가 빨랐다. 그리고, 시작점의 위치에 덜 민감하게 최적 값을 찾아낼 수 있었다. 그래서, 기존 알고리즘보다 신뢰성과 효율이 높음[14]이 다시 확인되었다.

참고문헌

- [1] 채석, 오영석, *퍼지이론과 제어*, 청문각, 2001
- [2] 이광형 · 오길복, *퍼지이론 및 응용 I · II*, 홍릉과학출판사, 1991

- [3] Rao, S., "Description and Optimum Design of Fuzzy Mechanical Systems," *ASME J. Mech, Transm., Autom. Des.*, Vol. 108, pp. 1-7, 1986
- [4] Diaz, A., "Goal Aggregation in Design Optimization," *Eng. Optimiz*, Vol. 13, pp. 257-273, 1988
- [5] Wood, K, Antonsson, E., and Beck, J., "Representing Imprecision in Engineering Design : Comparing Fuzzy and Probability Calculus," *Res. Eng. Des.*, Vol. 1, pp. 187-203, 1990
- [6] Wood, K., and Antonsson, E., "Modeling Imprecision and Uncertainty in Preliminary Engineering Design," *Mech Mach Theory*, Vol. 25, No. 3, pp.305-324, 1990
- [7] Thruston, D., and Carnahan, J., "Fuzzy Ratings and Utility Analysis in Preliminary Design Evaluation of Multiple Attributes," *ASME J. Mech. Des.*, Vol. 114, pp.648-658, 1992
- [8] Carnahan, J., Thurston, D., and Liu T., "Fuzzy Ratings for Multiattribute Design Decision - Making," *ASME J. Mech. Des.*, Vol. 116, pp.511-521, 1994
- [9] Vadde, S., Allen, J., and Mistree, F., "Compromise Decision Support Problems for Hierarchical Design Involving Uncertainty," *Comput. Struct.*, Vol. 52, No. 4, pp.645-658, 1994
- [10] Ekel, P., Pedrycz, W., and Schizinger, R., "A General Approach to Solving a Wide Class of Fuzzy Optimization Problems," *Fuzzy Sets Syst.*, Vol. 97, pp.49-66, 1998
- [11] Ohkubo, S., and Dissanayake, P., "Multicriteria Fuzzy Optimization of Structural Systems," *Int. J. Numer. Methods in Eng.*, Vol. 45, pp.195-214, 1999
- [12] Grignon, P., and Fadel, G., *Fuzzy Move Limit Evaluation in Structural Optimization*, AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Florida.
- [13] Mulkay, E., and Rao, S., *Fuzzy Heuristics for Sequential Linear Programming*, Proceedings of the 1997 ASME Design Engineering Technical Conferences, 1997
- [14] Trabia, M,B, and Xiao Bin Lu, "A Fuzzy Adaptive Simplex Search Optimization Algorithm", *ASME, J. Mech. Design*, June, Vol. 123, pp.216-225, 2001

- [15] Spendley, W., Hext, G., and Hirmsworth, F., "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation," *Technometrics*, Vol. 4, pp.441, 1962
- [16] Nelder, J., and Mead, R., "A Simplex Method for Function Minimization," *Comput. J.*, Vol. 7, pp. 308-313, 1965
- [17] Rekalitis, G., Ravindaran, A., and Ragsdell, K., *Engineering Optimization : Methods and Applications*. Wiley - Interscience, New York, 1983
- [18] Rao, S., *Engineering Optimization : Theory and Practice*, Wiley - Interscience, New York, 1996
- [19] Arora, J.S., *Introduction to Optimum Design*, McGraw - Hill International Editions, 1989
- [20] Joseph Edward Shigley and Charles R. Mischke., *Mechanical Engineering Design - Fifth Edition*, McGraw - Hill International Editions, 1989