

통계적 방법(Stochastic method) 에 의한 전기기기의 최적설계

□ 서장호, 곽상엽, 정현교 / 서울대학교 전기공학부

서론

과거에는 전기기기의 사양이 주어질 경우, 원하는 특성을 최적화하는 설계를 하기 위해서 설계자의 지식, 경험 및 자료를 근거로 하여 경험적으로 설계하는 방식을 택하였다. 이는 기기에 관련된 각종 이론, 실험 및 계산 자료 등 방대한 양의 데이터를 이용하여 설계하는 방식으로서, 그 기기에 요구되는 특성이 그다지 많지 않을 경우에는 이와 같은 방법만으로도 충분히 최적에 가까운 설계가 가능하다[1-5]. 하지만, 이러한 방법은 풍부한 경험과 방대한 데이터가 뒷받침되어야 한다. 최근 새로운 응용 분야가 생기고 특수한 형태의 전기기기가 등장함에 따라 기존의 경험에 기반한 방법으로는 모든 특성을 만족하는 설계점을 찾기 어려울 뿐만 아니라 설계가 되었다고 하더라도 그 설계가 최적인지를 증명하기는 불가능하다고 할 수 있다.

최근 컴퓨터의 발달에 힘입어 유전알고리즘(GA), 진화전략 알고리즘(ES), 신경회로망 등 확률론적 최적화 방법의 연구가 활발하다. 그런데, 이와 같은 일반적인 확률론적 최적화 알고리즘을 단일 목적함수를 갖는 전기기기 최적화 문제들에 단순히 적용하면 전역 최적점(global optimal point) 외에 가끔 목적함수의 크기가 거의 같은 다른 국소 최적점들(local optimal points)에 수렴하는 현상이 발생한다. 이것은, 전역 최적점과 다른 위치에 거의 비슷한 목적함수 값을 가지는 점이 존

재한다는 것을 뜻한다. 즉, 실제 전기기기 최적설계에서는 전역 최적점뿐만 아니라 전역 최적점과 다른 곳에 위치하는 국소 최적점들도 반드시 고려해야 한다.

이렇게 여러 국소 최적점까지도 찾는 최적화를 multimodal 최적화라고 한다[6]. 전역최적점만 찾는 특성을 가지는 일반적인 확률론적 최적화 알고리즘으로는 여러 국소 최적점을 찾을 수 없다. 이를 위해 유전 알고리즘에 생태계의 니체(niche)개념을 도입하여 탐색공간에서 여러 해를 병렬적으로 찾을 수 있게 하는 니칭 유전 알고리즘이 제안되고 있으며 해집단의 변이나 교배 등에 대한 연구들도 활발하다[8-12].

본 기고에서는 multimodal 최적화를 위한 유전알고리즘과 패턴 탐색 기법(pattern search method)이 결합된 Auto-Tuning 니칭 유전 알고리즘(Auto-Tuning Niching GA)에 대한 내용을 소개한다. 또한, 최신 확률론적인 기법의 하나인 PSO(Particle Swarm Optimization)를 이용하여 실제 전기기기에서의 multimodal 최적화 문제를 다루고자 한다.

Auto-Tuning Niching GA

유전 알고리즘은 생명체의 자연 진화 법칙을 일반 최적화 문제에 적용한 방법으로 설계 영역에 다수의 설계점을 분포 시켜 목적 함수 값과 제약 조건의 위반 정도에 따라 각 설계점에 적합도(fitness)을 부여한다.

적합도가 클수록 다음 단계인 교배와 변종의 과정에 참여할 확률을 크게 하여 적합도가 좋은 설계점에 비슷한 형질을 가진 설계점들이 다음 단계에 보다 많이 형성되어 계산이 진행 될수록 전체 설계점들은 좋은 방향으로 탐색이 진행되게 한다[13].

그러나 multimodal 최적화 문제에 일반적인 유전 알고리즘을 적용하면 다른 피크에 해당하는 스키마타 (schemata) 사이의 경쟁을 제한할 수 없으므로 genetic drift 현상에 의해 하나의 전역 최적점에만 수렴하게 되므로 유전 알고리즘은 multimodal 최적화 성능이 결여 되어 있다.

이러한 유전 알고리즘의 특성에 생태계의 '니체 (niche)' 개념을 도입하여 탐색공간에서 여러 해를 병렬적으로 찾을 수 있도록 제안된 방법을 통틀어 니칭 유전 알고리즘이라 한다. 니체란, 자연 생태계에서는 비슷한 모습을 가진 개체들의 집합인 종(specie)들이 각각의 자기 영역을 말하며 최적화 문제에 있어서는 multimodal 한 탐색공간 내에서의 각 피크 점들이 가지는 주어진 영역(niche radius)를 뜻한다.

기존의 니칭 알고리즘에서는 함수의 모양을 개략적으로 알지 못하면 니체 반경을 정하기가 어렵고, 또한 같은 변수 공간 내에서는 일정(uniform)하게 주어진다. 따라서 이를 피크간의 거리가 서로 다른 응용분야에서는 적용하기 어렵다. 하지만, Auto-Tuning 니칭 알고리즘은 결정론적 방법인 패턴 탐색 기법을 이용하여 니체 반경이 알고리즘 내에서 자동으로 정해지며 (auto-tuning), 찾아진 피크마다 각기 다른 니체 반경을 가지므로 어떠한 문제에도 적용할 수 있는 범용성을 가진다. Auto-Tuning 니칭 알고리즘의 상세한 구현 과정은 다음과 같다.

Step 1. 초기화

여기서는 초기 개체집단의 크기 N 을 결정하고 N 개 만큼의 개체를 랜덤하게 발생시킨다. 초기 개체 집단의 크기는 변수의 개수(NV)에 따라 $10 \cdot NV$ 개로 한다. 예를 들어, 변수의 개수가 3일 경우 초기 개체집단의 크기는 30이 된다.

Step 2. 개체들과 엘리트 개체와의 중복 여부 판단

N 개의 개체들이 각각 모든 엘리트집합에 저장되어 있는 개체의 니체 반경 내에 있는가를 판단한다. 만일 엘리트 개체와 중복된다면 그 개체의 적합도는 0이 된다. 초기 엘리트집합의 크기(NE)는 0이므로 초기에는 모든 개체가 생존하게 된다.

Step 3. 결정론적 알고리즘에 의한 피크 탐색

앞의 Step 2에서 생존한 개체는 결정론적 알고리즘을 통해서 피크를 탐색하게 된다. 이때 최대한 함수 호출을 줄이면서 정확한 피크를 탐색할 수 있는 알고리즘을 선택하는 것이 중요하다.

Step 4. 엘리트 집합 및 니체 반경의 결정

Step 3에서 찾아진 피크들이 엘리트 집합에 들어가게 된다. 이때 엘리트 집합에 미리 저장된 피크와 중복된 피크는 제외된다. 패턴 탐색법에 의해 옮겨지기 전의 위치에서 옮겨진 피크까지 이동한 거리를 계산하여 각 피크의 니체 반경을 결정하게 된다. 새로운 피크가 기존 피크와 동일한 경우에는 이동한 거리가 먼 경우를 택하여 니체 반경으로 한다. 이때 이동한 거리는 변수마다 따로 계산된다. 따라서 피크는 변수마다 다른 니체 반경을 가지게 되며 니체 반경은 세대가 반복되면서 증가하는 형태를 가지게 된다.

Step 5. 자식세대의 크기 결정 및 선택

자식 세대의 크기는 Step 4에서 결정된 엘리트 집합 NE 개의 $2 \cdot NE$ 로 결정된다. 즉, 자식세대는 엘리트 개체 NE 개와 나머지 개체 중 적합도가 높은 NE 개로 구성된다.

Step 6. 재생산(reproduction)

교배(crossover)와 변이(mutation)를 통해서 자식세대를 재구성하게 된다.

Step 7. 종료

적절한 종료 조건을 만족하면 알고리즘을 종료하고, 만족하지 않으면 Step 2로 돌아간다.

종료 조건은 평가 함수와 같이 피크 개수를 알고 있는 경우에는 모든 피크를 찾았을 때로 하였고, 피크의 개수를 알지 못하는 경우는 100번의 세대가 반복되어도 새로운 피크를 찾지 못할 때를 종료 조건으로 하였다.

그림 1에 Auto-Tuning 니칭 알고리즘의 흐름도를 나타내었으며 그림 2에 알고리즘의 구현 원리를 그림으로 설명하였다. 어떤 세대에서 개체 6개가 그림 2(a)와 같이 분포하였다고 가정하자. 이때 각 개체들에 패턴 탐색 알고리즘을 적용하여 피크를 탐색하게 되면 개체 p1-p3까지는 그림 2(b)의 P1으로 수렴하게 되고 p4-p6까지는 P2로 수렴하게 된다. 이때, P1과 P2는 이 전 세대에 중복된 피크점이 없으면 엘리트 집합에 포함되게 된다. 또한

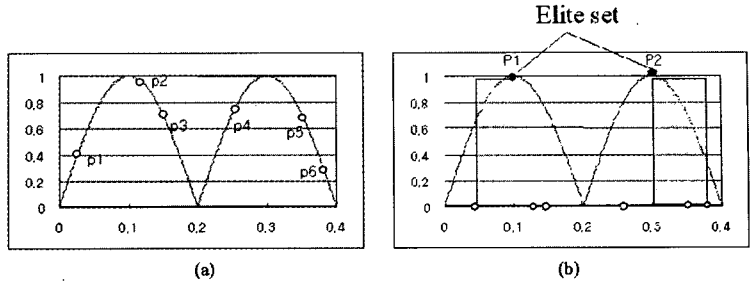


그림 2 Auto-Tuning 니칭 유전 알고리즘의 피크 탐색과정 및 니체 반경 결정 과정

P1과 P2에 대한 니체 반경은 가장 멀리 이동한 점과의 거리로서 결정된다. 즉, P1의 니체 반경은 가장 멀리 떨어진 p1과의 거리가 되며 P2의 니체 반경은 p6와의 거리로서 나타난다. 그림 2(b)에서는 사각형의 가로 길이가 니체 반경이 된다.

Auto-Tuning 니칭 유전 알고리즘을 이용한 최적화 예제

평가 함수

평가 함수는 식 (1)과 같으며 이 함수는 각 변수에 대해 피크간의 거리가 일정하다는 특징을 갖는다.

$$f_1(x_1, x_2) = \sin^6(5\pi x_1) \times \sin^6(5\pi x_2) \quad (0 \leq x_1, x_2 \leq 1) \quad (1)$$

총 피크수는 25개이고 이때 함수의 모양은 그림 3와 같다.

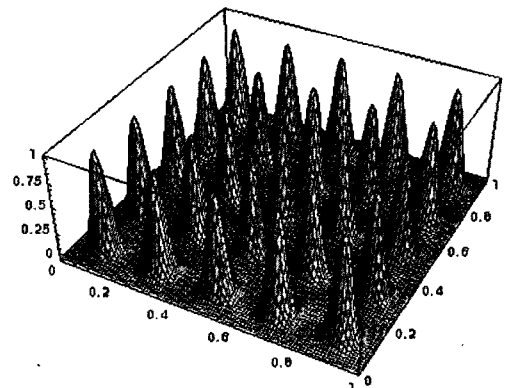


그림 3 평가 함수

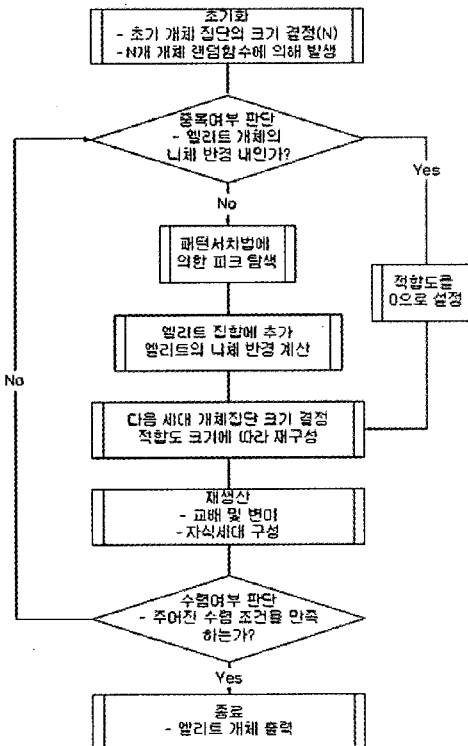


그림 1 Auto-Tuning 니칭 유전 알고리즘의 흐름도

최적화 결과

그림 4에서는 평가함수의 세대별 피크의 분포와 니체 반경의 변화를 나타내었다. 사각형 점으로 표시된 것이 해의 위치가 되며 십자가 모양의 막대가 해로 각 변수의 니체 반경을 나타내고 있다.

그림 4에서 1세대(a)에서는 15개의 피크를 찾았으며 2세대(b)에서는 20개의 피크를 찾았다. 그림 4(c)에서는 3세대와 4세대에서 찾은 피크가 같으므로 피크 분포는 같으나 피크별 니체 반경이 피크점 (0.5, 0.5), (0.5, 0.1), (0.7, 0.5)에서 다르게 나타남을 알 수 있다. (더 큰 니체 반경이 4세대의 것이다.) 그림 4(d)에서는 피크를 모두 찾았을 때를 보여주는데 이때 니체 반경이 대부분의 변수 영역을 차지한다는 것을 알 수 있다.

Multi-grouped PSO

경험적 탐색 방법의 하나로써 J. Kennedy 와 R. Eberhart에 의해 제안된 PSO(Particle Swarm Optimization)는 종래의 벌이나 물고기와 같은 그룹의 움직임에 관한 연구로부터 유도되었다[14]. 즉, 무리의 개체들이 가지고 있는 정보를 (서로) 공유해가며 목표 지점을 찾아가는 과정을 최적화 알고리즘으로 응용한

것이다. PSO는 알고리즘이 간단하고 전역 최적화를 위한 높은 탐색 능력을 가지고 있기 때문에 여러 최적화 문제에 많이 적용되고 있다[15]-[17]. PSO에서 그룹 내의 각 개체가 해석 영역에서 최적해를 찾아가는 과정은 두 가지 요소에 의해 영향을 받는다. 하나는, 각 개체 별로 현재까지의 탐색 중 최고의 적합도를 가질 때의 위치 정보 요소(pbest)이고, 다른 하나는, 그룹 내의 pbest들 중에서 가장 좋은 값을 저장하는 그룹 위치 정보 요소(gbest)이다. 일반적인 PSO는 전역 최적해를 찾는 데 있어 매우 강력한 성능을 가지고 있다. 하지만 PSO는 모든 개체들이 결국에는 한 지점으로만 수렴하게 되는 근본적인 한계를 가지고 있기 때문에, 순수 PSO만으로는 multimodal 최적화를 하는 것은 매우 어려운 일이다. 따라서 그룹화 된 각각의 PSO로 해석 영역의 여러 국소최적점을 찾게 하는 새로운 알고리즘인 MGPSO(Multi grouped PSO)를 제시한다.

MGPSO에서 각각의 PSO가 서로 같은 해를 찾는 오류를 막기 위해서 gbest에 영역의 개념을 도입하였다. MGPSO의 구현 과정은 다음과 같다.

Step 1. 기본 조건 설정

MGPSO의 첫 번째 단계에서는 해석 영역의 최소 및 최대 범위를 지정한다. 그룹의 개수는 해석 영역에서 찾고 싶은 해의 개수로 정한다. 또한, 각 그룹의 해가 겹치는 오류를 방지할 수 있도록 그룹별 gbest의 영역의 크기를 지정한다.

Step 2. 개체의 위치와 속도 초기화

각 개체의 위치 벡터와 속도 벡터를 초기화 하고 현재의 위치 벡터를 pbest로 정한다. 각 그룹별 gbest는 그룹 내 개체들의 pbest들 중에서 가장 좋은 것을 선정한다.

Step 3. 속도 업데이트

i번째 그룹의 j번째 개체의 속도는 다음과 같은 방정식에 의해서 업데이트된다.

$$V_{ij}^{k+1} = wV_{ij}^k + C_1 \varphi_1 (pbest_{ij}^k - X_{ij}^k) + C_2 \varphi_2 (gbest_{ij}^k - X_{ij}^k) + C_3 \varphi_3 (X_{ij}^k - gbest_{ij}^k) \quad (2)$$

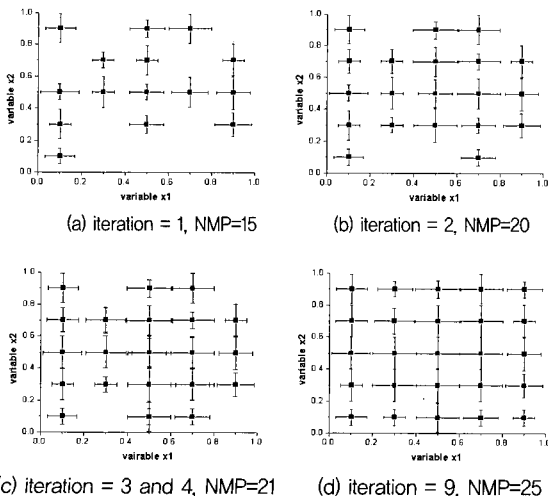


그림 4 세대별 피크 분포 및 니체 반경의 변화 양상

여기서 V_{ij}^k 와 X_{ij}^k 는 k번째 반복일 때 i번째 그룹의 j 번째 개체의 속도와 위치이다. w 는 반복횟수에 따라 0.9 에서 0.4로 선형적으로 감소하는 관성계수이며 ϕ_1 와 ϕ_2 는 0 과 1사이에서 발생하는 랜덤 변수이다. C_1 와 C_2 는 pbest와 gbest쪽으로 향하게 하는 가속계수이다. C_3 는 MGPSO에서 새롭게 추가된 반발계수로서 현재의 개체가 다른 그룹의 영역을 침범하면 그 영역을 벗어나도록 돕는다. 이 반발 속도 성분이 발생하는 간단한 예제를 그림 5에서 보여준다. 여기서 그룹은 2개로 가정하였다.

Step 4. 위치 업데이트

각 개체들은 다음과 같은 수식을 이용하여 위치를 수정한다.

$$X_{ij}^{k+1} = X_{ij}^k + V_{ij}^k \Delta t \quad (3)$$

여기서, Δt 는 시간 간격으로 일반적으로 '1' 이 사용된다.

Step 5. pbest업데이트

만약 한 개체의 현재 위치가 해석 영역 내에 있고 다른 gbest들의 영역을 침범하지 않았으면 그 위치에 대해서 적합도를 평가한다. 계산된 적합도가 지금까지 가지고 있는 최고의 적합도보다 좋은 값이면 현재 위치를 pbest로 저장한다.

Step 6. gbest업데이트

일반적인 PSO에서 gbest는 그룹내의 개체들 중에서 가장 좋은 pbest로 대체된다. 하지만 이러한 전략이 multi-modal 최적화에 적용된다면 다른 그룹의 gbest와 겹칠 수 있다. 따라서 이러한 해의 중복을 피하기 위해서는 각 그룹의 gbest들은 다른 그룹의 gbest들과 적당한 거리를 두고 떨어져 있어야 한다. 게다가 PSO의 빠른 수렴 특성을 유지하기 위해서는 각 그룹의 gbest는 높은 적합도를 가지는 pbest들 중에서 선택되어야 한다. 이와 같은 두 가지 조건을 만족하기 위한 방법은 아래와 같다.

- (A)pbest의 적합도로 각 그룹의 개체들을 정렬한다.
- (B)각 그룹에서 가장 좋은 성능을 보이는 pbest를 선택하고 그것을 temporary-gbest 라고 정한다.
- (C)temporary-gbest들 중에서 가장 좋은 적합도를 가지는 temporary-gbest를 선택한 후 그것을 basis-gbest라고 정한다.
- (D)다른 그룹의 temporary-gbest들이 basis-gbest의 영역에 놓여 있는지 확인한다. 만약 temporary-gbest가 그 영역에 놓여있다면, 그 temporary-gbest가 속해있는 그룹에서 두 번째로 좋은 위치인 second-pbest를 그룹의 temporary-gbest로 선정한다. 이때 또 다시 basis-gbest의 영역에 놓이게 되면 반복 수행을 거쳐서 temporary-gbest가 그 영역 밖에 놓이도록 한다. 더 이상의 temporary-gbest가 basis-gbest의 영역에 없다면 basis-gbest를 해당 그룹의 gbest로 선정한다.
- (E)위의 (D)에서 선정된 gbest들은 제외하고 gbest의 개수가 Step 1에서 정의된 그룹의 개수와 같아질 때까지 (C), (D)의 단계를 반복한다.

그림 6에서는 두 개의 그룹으로 이루어진 MGPSO에서 위의 과정에 대한 간단한 예를 보여준다. 여기서 A1은 basis-gbest이고 B1, B2는 각각 그룹 2의 temporary-gbest, second-best pbest이다. Step 6을 수행하면, A1은 그룹1의 gbest가 되고 B1은 A1의 반경 안에 있기 때문에 B2가 그룹2의 temporary-gbest가 된다.

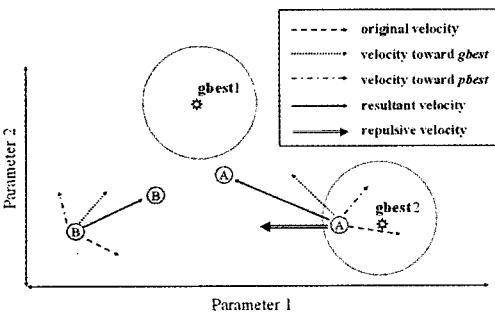


그림 5 반발 속도의 발생을 보여주는 예

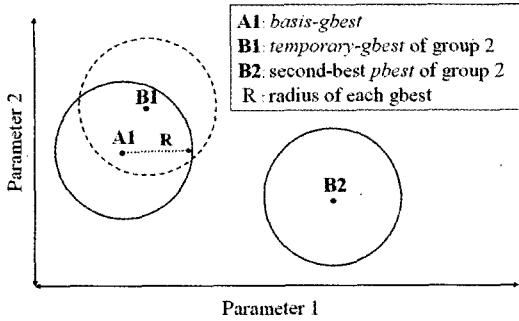


그림 6 gbest의 결정을 보여주는 예

Step 7. 반복과 수렴조건 검사

위의 단계들을 그룹 내에 있는 모든 개체에 대해서 수행을 한 후에 Step 3에서 Step 6까지 반복한다. 모든 개체들이 속한 그룹의 gbest로 수렴하면 반복을 멈춘다.

MGPSO를 이용한 최적화 예제

시범함수

제안된 알고리즘의 타당성을 검증하기 위해서 다음과 같은 2차원 multimodal 함수를 사용하였다.

$$f = \sum_{k=1}^{N_p} \frac{b_k}{1 + \frac{(x-x_{pk})^2 + (y-y_{pk})^2}{a_k}}, \quad -60 < x, y < +60 \quad (4)$$

여기서 N_p 는 피크의 개수이다. a_k 와 b_k 는 k 번째 피크의 폭과 높이이며 x_{pk} 와 y_{pk} 는 k 번째 피크의 좌표이다. 그림 7은 크기와 폭의 값이 서로 다른 10개의 피크를 가진 시범함수를 보여준다.

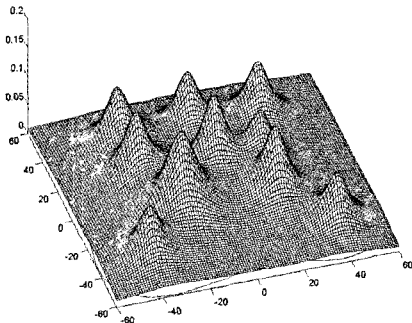


그림 7 MGPSO의 시범함수

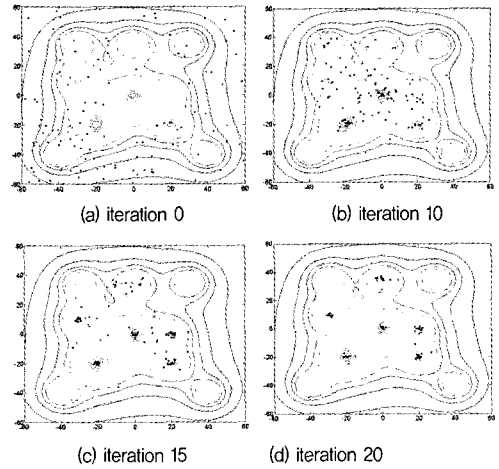


그림 8 MGPSO의 테스트 함수에 적용한 결과 ($\mu=6, \lambda=20$).

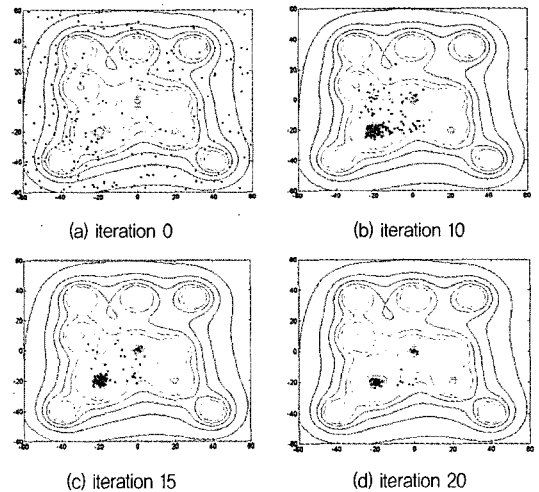


그림 9 MGPSO의 테스트 함수에 적용한 결과 ($\mu=6, \lambda=20, R_0=0$)

최적화 결과

MGPSO를 테스트 함수에 적용한 결과가 그림 8,9에 나타내었다. 찾고 싶은 해를 6개로 한정하였기 때문에 그룹의 개수를 6개로 결정했다. 각 그룹별 개체 수는 20개로 정하였다. 그림 8는 제안된 MGPSO를 이용한 결과로서 반복횟수에 따라서 각 그룹이 해석영역 안에 있는 해를 찾는 것을 보여준다. 그림 9는 그룹의 영역을 없앤 MGPSO의 결과로서 그룹의 영역이 없기 때문에 서로 다른 해를 찾지 못하고 중복된 해를 찾게 된다

는 것을 나타낸다.

MGPSO를 전기기기 최적화에 적용한 사례

해석 모델

MGPSO 알고리즘을 4극 18슬롯을 가지는 영구자석 매입형 전동기(Interior Permanent Magnet Motor) 설계에 적용하였다. 이 전동기의 사양은 표1에 나타내었다. 이 전동기에 대한 형상은 그림 10에서 보여준다.

목적 함수 및 설계 변수의 선정

전동기의 고효율화에 있어서 가장 문제되는 부분이 열 발생이고 이는 전동기의 효율과 직결되는 문제이다. 따라서 효율을 높이면 손실이 줄어 열 발생을 줄일 수 있고 이는 전동기의 신뢰성을 높일 수 있으므로 최적화 목적함수로서 전동기의 효율을 채택하였다. 전동기의 출력이 일정하다고 가정한다면 목적함수는 오직 손실에만 관련된다. 일반적으로 소형 영구자석 전동기에서는 표유부하손, 회전자 손실 그리고 인버터에 의한 고조파손실이 매우 작기 때문에 이들을 무시할 경우 목적함수는 고정자 동손, 철손, 그리고 기계손으로 구성되며 이를 설계 파라미터의 함수로 나타낼 수 있

다. 본 연구에서는 정격속도, 정격부하에서의 효율을 최대로 하는 것을 목적으로 하기 때문에 기계손을 일정한 값으로 가정한다.

만약 잔류 자속 밀도, 슬롯수, 공극 길이, 고정자 외경, 축방향 길이 그리고 출력을 일정하다고 가정하면 고정자 턴수(N_s), 고정자 내경(r_s), 자극 각도(2α), 자석 두께(l_m)가 설계변수가 된다.

위의 변수들이 만족해야 할 제약 조건은 다음과 같다.

- i. 전동기의 일정 출력(속도, 토크 일정)
- ii. 자석 크기의 제약 (자석이 회전자에 매입되기 위한 조건)
- iii. 일정한 치 자속 밀도와 요크 자속 밀도
- iv. 고정자 전류 밀도의 제약
- v. 자석의 감자 한계 조건

최종 후보해의 선정

표 2는 MGPSO 알고리즘을 이용한 최적 설계 결과로 나온 3개의 후보해를 보여준다. 여기서 단일 최적화로 고려될 수 없는 제작의 편의성, 설계자의 선호도, 역기 전력을 고려하여 최종 후보해를 선정하였다. 여기서 후보해1을 최적 설계변수로 선정했는데, 그것은 높은 효율과 작은 역기전력 낮은 권선 온도를 가졌기 때문이다.

결론 및 향후 전망

이상으로 본 기고에서는 기존의 niching GA에 패턴 탐색방법을 결합한 Auto-tuning niching GA와 PSO의 장점을 살린 MGPSO에 대해서 알아보았다. 알고리즘

표1 전동기 사양

Rated power	300W
Rated RPM	3750rpm
Stator outer radius	50mm
Axial length	100mm
Stacking length	50mm
Magnet	Nd-Fe-B
Time rating	5 minutes

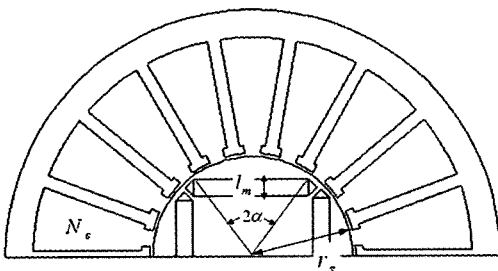


그림 10 IPMSM의 단면도

표 2 MGPSO 알고리즘을 이용한 최종 후보해

Solutions	Case 1	Case 2	Case 3
Efficiency [%]	69.13	69.14	69.05
Back-EMF [V]	11.04	13.45	15.01
N_s	54	66	72
r_s [mm]	11.53	11.50	11.70
l_m [mm]	2.50	2.41	2.71
2α [°]	68.2	68.75	66.9
Winding Temperature [°C]	129.5	131.4	132.6

의 타당성을 알아보기 위해서 시범함수 및 실제 기기 설계에 적용한 결과 Auto-Tuning niching GA 및 MGPSO는 실제 기기 최적화에 적합한 알고리즘이라는 것을 알 수 있었다.

최근에는 개체들의 군집행동을 최적화에 적용하려는 사례(Particle Swarm Optimization, Ant colony Optimization)가 나오고 있으며 이에 따른 새로운 연산자 및 패러다임이 소개되고 있다. 또한 기존의 다른 알고리즘과의 결합을 통한 혼합형 알고리즘으로 좀더 좋은 성능을 얻으려는 노력도 시도되고 있다. 따라서 향후에는 새로운 최적화 알고리즘을 전기기기 최적 설계에 적용하거나 기존 알고리즘과의 결합으로 좀더 우수한 성능을 낼 수 있는 최적화가 많이 연구될 것으로 보인다.

[참고문헌]

- [1] T.J.E.Miller & A.Hughes, "Comparative Design & Performance Analysis of Air-Cored & Iron Cored Synchronous Machines," Proceedings of IEE, Vol.124, No.2, 1977, pp.127-132.
- [2] A.Levran & E.Levi, "Design of Polyphase Motors with PM Excitation," IEEE Trans. on Magnetics, Vol.20, No.3, 1984, pp.507-515.
- [3] G.R.Slemon, "On the Design of High Performance PM Motors," Proceedings of 1992 IEEE IAS Conf., pp. 279-285.
- [4] B.C.Kuo et al, "Computer-aided Design of Brushless PM Motors," Proceedings of Incremental Motion Control Systems & Devices, 1986, pp.77-84.
- [5] P.Viarouge et al, "Design & Construction of a Brushless Permanent Magnet Servo Motor for Direct-Drive Application," IEEE Trans. on Ind. Appl., Vol.23, No.3, 1987, pp.526-531.
- [6] B. Sareni, L. Krahenbuhl and A. Nicolas, "Niching Genetic Algorithms for Optimization in Electromagnetics", The 11th COMPUMAG' 97, 563-564, Rio de Janeiro, 1997.
- [7] S.Ramberger and S.Russenschuck, " Genetic Algorithms with Niching for Conceptual Design Studies," The 11th COMPUMAG' 97, 509-5510, Rio de Janeiro, 1997.
- [8] 이철균, Niching 유전알고리즘을 이용한 전기자동차용 유도 전동기의 최적 설계, 박사학위논문, 서울대학교, 이철균, 1998
- [9] Jelasity, M.; Dombi, J., "GAS, an approach to a solution of the niche radius problem," First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications(GALESIA.), pp. 424-429, 1995
- [10] B. L. Miller, M. J. Shaw, "Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization," IEEE International Conference on Evolutionary Computation, pp. 786-791, 1996.
- [11] Sareni, B., Krahenbuhl, L., "Fitness sharing and niching methods revisited," IEEE Transactions on Evolutionary Computation, vol. 2, Issue 3, pp. 97-106, Sept. 1998.
- [12] Fonseca, C.M., Fleming, P.J., "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 28, Issue: 1, Jan., 1998
- [13] 문병로, "유전 알고리즘", 두양사, 2003.
- [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. 4, pp. 1942-1948, 1995.
- [15] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 211-224, 2004.
- [16] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," IEEE Trans. Antennas Propag., vol. 52, no. 2, pp. 397-407, 2004.
- [17] S. Naka, T. Genji, T. Yura and Y.Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," IEEE Trans. Power Syst., vol. 18, no. 1, pp. 60-68, 2003.