

# 효율적인 점진적 변수 선택 기반 근사 최적화 기법

[글] 손석호<sup>1\*</sup>, 이용빈<sup>1</sup>, 류동흠<sup>2</sup>, 최동훈<sup>1,2</sup> (주)피도텍<sup>1</sup>, 한양대학교<sup>2</sup> seokho.son@pidotech.com

## 1. 서 론

CAE (Computer aided engineering)는 컴퓨터를 이용한 해석, 분석 등의 전반적인 과정을 의미하며, 제품의 기획/설계 및 제조 단계에서 컴퓨터를 이용한 모의실험(Simulation)을 활용함으로써 설계 및 테스트 기간을 대폭 감소시킬 수 있는 기술이다. 그러나 보다 더 정밀하고 정확도 높은 모의실험 모델이 요구됨에 따라 CAE 해석도 많은 비용을 소비하게 되었다 [1]. 이러한 문제 때문에 CAE를 활용하더라도 산업제품 상세설계 단계에서 수많은 해석을 요구하는 기존 최적 설계 기법을 사용하는 것은 현실적으로 어렵다 [2].

2000년대에 들어오면서 산업제품 상세설계 기간을 단축하기 위해 CAE 해석 모델 대신 해석 시간 및 비용이 적게 드는 근사모델(Approximate model)을 사용하게 되었다 [3]. 또한, Polynomial Regression (PR), RBF (Radial Basis Function), Kriging 등 다양한 근사모델을 활용한 근사최적설계 기법(Approximate optimization; AO)들이 개발되었다 [4,5]. 근사모델을 구성하기 위해서는 실험점(각 실험점들은 1회 CAE 해석이 요구됨)이 필요하며, 이러한 실험점 개수는 일반적으로 설계인자의 개수(number of design variables; NDV)에 의존한다. 그러므로 근사최적설계 기법의 경우에도 설계인자가 많은 설계문제의 최적해를 도출해내기 위해서는 많은 CAE 해석을 요구한다. 또한 근사최적설계기법보다 더 정확한 해를 찾기 위한 순차적 근사최적설계 기법(Sequential approximate optimization; SAO)은 매 iteration마다 최소 필요 실험점 개수만큼 CAE 해석을 요구하기 때문에 일반적으로 근사최적설계기법보다 더 많은 CAE 해석을 요구한다.

따라서, 본 논문에서는 산업제품의 상세설계 단계에서 단지 20~30회 미만의 CAE 해석 안에 최적 설계안을 도출해내는 효율적인 점진적 변수선택 기반 근사 최적화 기법(Efficient progressive screening-based approximate optimization, 이하 EPSAO)을 제안

한다. 본 논문에서 제안한 EPSAO는 구속조건이 없는 설계 문제에만 적용할 수 있으며, 구속조건이 있는 설계 문제에 대한 적용은 향후 연구 과제로 남아있다.

다음 장에서는 EPSAO에 대해 상세히 설명을 하며, 3장에서는 잘 알려진 최적화 문제(28개의 수학적 예제)에 적용하여 제안한 기법의 성능을 평가한다. 마지막으로 4장에서는 제안한 기법에 대해 고찰하도록 한다.

## 2. 제안한 기법 상세 설명

### 2.1 EPSAO의 기본 개념 및 순서도

기존 근사최적설계 기법 및 순차적 근사최적설계 기법은 전역 최적해(Global optimum)를 찾는 것을 목표로 한다. 그러므로 정확한 전역 최적해를 탐색하기 위해 많은 실험 정보들을 필요로 하며, 그만큼 많은 CAE 해석이 요구된다. 그러나 제안한 EPSAO는 산업제품의 상세설계 단계의 시간/비용을 대폭 감소시키고, 기존 설계보다 향상된 해를 찾는 것을 목표로 개발되었기 때문에 설계인자의 개수와 상관없이 20~30회 미만의 CAE 해석만으로 향상된 해를 도출할 수 있다.

그림 1은 EPSAO의 순서도이며, 다음 설명은 각 단계별 EPSAO 알고리즘 순서이다.

- (1) 적응형 초박 초기점 생성: 정의된 설계영역 안에서 초기 실험점을 기반으로 새로운 추가 실험점을 생성
- (2) 주요변수 판별: 각 최적설계 반복 단계(iteration) 별 주요 변수 판별
- (3) 다항 회귀모델 생성: 판별된 주요변수들을 기반으로 다항 회귀모델(PR model) 생성
- (4) 지능형 설계영역 설정: 근사최적설계 수행을 위한 지능형 설계영역(Intelligence Move Limit; IML) 설정
- (5) 근사최적설계 수행: 개선된 CMA-ES (Covariance matrix adaption-evolution strategy)를 이용한 근사 최적설계 수행
- (6) 수렴 판단: 수렴 조건을 만족할 때까지 반복적으로 단계 2-5를 수행

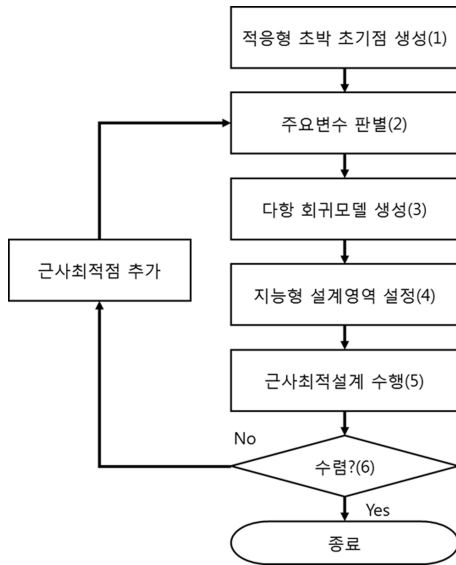


그림 1. EPSAO의 순서도

### 2.2 적응형 초박 초기점 생성

초기 단계에서는 정의된 설계 영역 내에서 주어진 초기 실험점(Initial point, 초기 실험점은 실제 산업에서 사용되는 적정 설계안으로 가정함)과 설계 영역의 크기를 동시에 고려하여 최적의 거리를 계산하는 알고리즘에 따라 새로운 실험점을 추가한다. 새롭게 추가되는 실험점은 그림 2와 같이 위치하게 되며, 이 실험점을 대각 실험점(Diagonal point)이라 명명한다. 이 과정은 설계인자의 개수와 상관없이 동일하며, 단지 2개의 실험점만으로 알고리즘의 초기단계를 수행하게 된다. 이러한 전략은 기존 근사최적설계 기법이나 순차적 근사최적설계 기법의 초기단계에서 설계인자의 개수에 따라 CAE 해석 회수가 증가하는 문제를 피할 수 있다.

### 2.3 주요 변수 판별

EPSAO의 현재 반복 단계에서 주어진 실험점으로 변수 독립적으로나 변수 상호적으로 성능지수에 미치는 영향력을 판단하여 주요 변수를 판별하게 된다. 이때, 주요 변수는 회귀계수(coefficient of regression)의 크기를 비교하여 판단하게 되며, 선택되는 주요 변수의 개수는 현재 반복 단계 수와 동일하다. 그러나 변수의 중요도 판별을 위해 생성하는 완전일차다항식(Full linear polynomial model)을 구성하기 위해서 일반적인 최소자승법(Least square method) [6]을

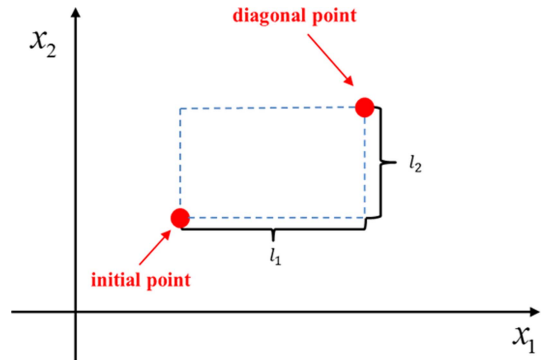


그림 2. 적응형 초박 초기점 생성 방법

사용하면 기존의 설계인자의 개수 이외에 하나의 추가 실험점이 필요하게 된다. 따라서, 적은 실험점으로도 완전일차다항식을 구성할 수 있는 “Moore-Penrose generalized inverse”를 사용하였다 [7].

### 2.4 다항 회귀모델 구성

주요 변수가 판별 되면 선정된 주요 변수만 사용하여 다항 회귀모델을 구성하게 된다. 이로 인해 초기 실험점 주위에서 빠른 해의 향상을 기대할 수 있다. 또한, 초기 단계부터 시작하여 반복되는 최적화 단계에서 실험점의 개수가 달라지기 때문에 각 단계의 실험점의 개수에 따라 다항 회귀항의 개수를 선택할 수 있도록 하여 모델을 구성하도록 하였다.

### 2.5 지능형 설계영역 설정

주요 변수를 기반으로 다항 회귀 모델을 구성 한 후 근사최적설계의 각 반복 단계마다 지능적으로 설계영역을 변화시켜줌으로써 정확성(해의 향상을 나타내는 지표)과 효율성을 동시에 향상을 시켰다. 그림 3은 지능형 설계영역의 크기를 결정하는 방법을 설명한 것이다.

그림 3(a)에서 보면 이전 단계에서의 가장 우수한 해인  $x_k^0$ 의 응답값이  $f(x_k^0)$  이 현재 단계에서의 가장 우수한 근사 최적해  $\hat{x}_k^*$ 의 응답값  $f(\hat{x}_k^*)$ 보다 작거나 동일하게 되면, 이전 단계의 설계영역보다 설계영역을 줄이게 된다.

반대로 그림3(b)와 같이 이전 단계의 응답값  $f(x_k^0)$  이 현재 단계의 응답값  $f(\hat{x}_k^*)$  보다 크거나 동일하면 현재 단계의 근사 최적해  $\hat{x}_k^*$ 를 중심으로 설계영역을 확장하게 된다.

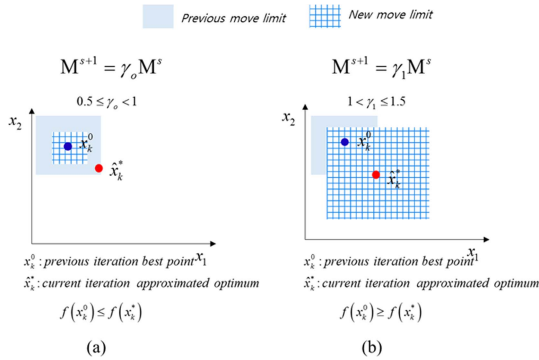


그림 3. 지능형 설계영역 설정

### 2.6 개선된 CMA-ES를 이용한 근사 최적설계 수행

CMA-ES(Covariance Matrix Adaption-Evolution Strategy)는 유전자 알고리즘의 일종으로 설계 전 영역의 탐색능력이 향상시킨 전역 최적화 기법이다. CMA-ES가 시작되면 탐색 분포(search distribution)의 파라미터인 평균(mean)을 임의로 선정하여 초기 개체를 생성하고, 생성된 개체의 적합도를 평가합니다.

이런 최적화 과정 중에 새로운 개체가 생성되는 확률분포(probability distribution function; PDF)가 설계 영역을 벗어나게 된다. 설계 영역을 벗어난 개체는 최적화 과정 중에 사용할 수 없는 개체로써 이로 인하여 함수 해석의 비용을 가중 시킨다.

이런 문제를 해결하기 위해서 개선된 CMA-ES에서는 갱신된 확률분포로 생성된 개체가 설계영역을 벗어난 경우 이를 설계 영역의 경계로 이동시키고, 해당 확률변수를 이동된 개체의 위치에 해당하는 값으로 재 계산하여 확률변수가 현재 개체의 위치 정보를 가지도록 한다. 따라서, 설계 영역 내에서만 개체를 생성하므로 CMA-ES의 효율성을 증대시키게 된다.

### 2.7 수렴 조건

근사최적설계 수행 후 수렴 조건을 만족하면 알고리즘을 멈추고, 만족하지 않는다면 근사 최적점을 주어진 실험점 집합에 포함 시키고 주요 변수 판별부터 다시 수행한다. 이때 수렴 조건은 다음과 같다.

- 현재 반복 단계에서 가장 우수한 성능인자를 가지는 실험점의 설계인자 값의 절대 변화가 연속 3번의 반복 단계 동안 일어나지 않는다면, 수렴 조건 만족
- 현재 반복 단계에서 가장 우수한 성능지수 값의 절대 변화가 연속 3번의 반복 단계 동안 일어나지 않는다면, 수렴 조건 만족

- 현재 반복 단계 에서 가장 우수한 성능지수 값의 상대 변화가 연속 3번의 반복 단계 동안 일어나지 않는다면, 수렴 조건 만족

## 3. 개선된 CMA-ES의 성능평가

### 3.1 성능 평가 지표

개선된 CMA-ES의 성능을 평가하기 위해서 다음과 같이 3가지 성능 평가 지표를 사용하였다.

- 정확성(accuracy): 실제 해(exact solution)대비 각 기법으로 도출한 최적해와의 오차
- 효율성(efficiency): 수렴하기 위해 수행한 총 함수 해석 횟수
- 강건성(robustness): 각 기법의 수렴 여부

### 3.2 성능 평가 결과

기존의 CMA-ES 대비하여 개선된 CMA-ES의 성능을 평가하기 위하여 전체설계 영역 대비 가용영역(feasible region)과 목적함수의 형태에 따라 문제를 분류하여 전역최적화 기법의 성능평가에 주로 사용되는 수학적 문제에 적용하였다 [8].

정확성의 성능지표는 그림 4에서와 같이 실제 해와 각 기법에서 구한 최적해의 오차(error)를 계산하였으며, 각 문제마다 20번 반복 수행한 평균오차(mean

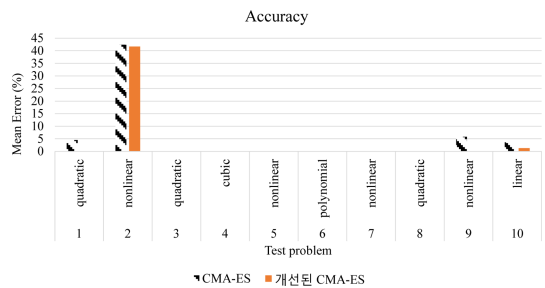


그림 4. 개선된 CMA-ES의 정확성 평가

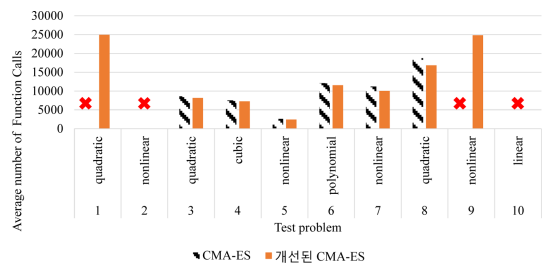


그림 5. 개선된 CMA-ES의 효율성 평가

error)로 비교하였다.

결과를 확인하여 보면 3-8번까지의 문제의 경우는 두 기법 모두 오차가 거의 없는 것을 확인할 수 있었지만, 1, 2, 9, 10번 문제의 경우는 기존의 CMA-ES의 정확도가 낮은 것을 확인할 수 있었다. 총 10개의 문제의 평균오차는 기존의 CMA-ES는 5.63%이며, 개선된 CMA-ES는 4.3%로 개선된 CMA-ES가 우수함을 확인할 수 있었다.

두 기법의 효율성은 20번의 반복 수행에 대한 평균 함수 해석 횟수(average number of function calls)로 비교하였다.

그림 5을 확인하여 보면 강건성과 효율성을 동시에 확인할 수 있다. 총 10개의 문제 중에서 1번과 9번의 문제의 경우는 기존의 CMA-ES가 전역 최적해를 찾지 못하였으며, 2번과 10번의 문제의 경우는 두 기법 모두 전역 최적해를 찾지 못하였다. 따라서, 강건성도 개선된 CMA-ES가 더 우수함을 확인할 수 있다.

마지막으로 본 논문에서 적용한 예제에 대한 효율성을 비교하여 보면, 모든 경우에서 개선된 CMA-ES가 전역 최적해를 찾기 위한 평균 함수 해석 횟수가 더 적은 것을 알 수 있으며 효율성이 우수함을 확인할 수 있다.

#### 4. EPSAO의 성능평가

제안한 EPSAO의 성능을 평가하기 위해 상용 최적 설계 소프트웨어인 Hyperstudy에 탑재되어 있는 최적화 기법 중 EPSAO와 특성이 비슷한 SORS (Second Order Response Surface), SRSM (Scalable response surface method)와 성능 비교를 해보았다.

##### 4.1 성능 평가 지표

EPSAO의 성능을 평가하기 위해 CMA-ES와 마찬가지로 3가지 성능 평가 지표를 사용했으며, 그 성능 평가 지표는 다음과 같다.

- 정확성: 수학적 예제의 실제 해와 각 기법이 찾은 해의 차이
- 효율성: 각 기법이 해를 찾기 위해 수행한 함수 해석 회수
- 강건성: 각 기법의 수렴 여부

##### 4.2 성능 평가 예제

EPSAO의 성능을 평가하기 위해 크게 두 분류의 예제에 적용해 보았다. 첫 번째 적용한 문제는 그림 6(a)에서 볼 수 있듯이 계곡 형태(valley shaped)이거나 그

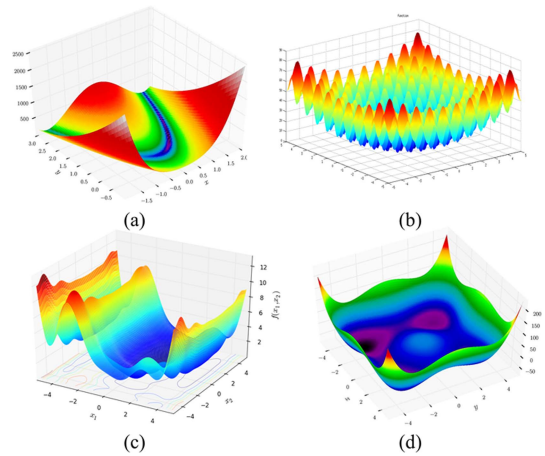


그림 6. 수학적 문제의 함수 형태

림 6(b), (c)와 같이 국부 최적해(local optimum)가 많이 존재하여 전역 최적화(global optimum)를 찾기 어려운 문제이다. 또한, 그림 6(d)와 같이 전 설계 영역에서는 평평한 형태(plated type)이지만, 설계 영역의 경계지점에서만 급격하게 변하는 특성을 갖는 수학적 문제들 [9-12]이다.

두 번째 문제는 최적화 기법의 테스트 문제로 잘 알려진 Schittkowski문제[13]로써 매우 비선형이 높은 문제들이다. 적용한 수학적 문제들은 설계인자의 개수를 변경할 수 있는 문제로서 각 문제마다 설계인자의 개수를 2, 4, 6, 8, 10으로 하여 총 20 (4\*5)개의 문제를 풀어 보았다. Schittkowski 문제는 설계인자의 개수가 2~10으로 구성된 8개의 문제이다.

##### 4.3 성능 평가 결과

그림 7과 9은 수학적 문제들과 Schittkowski문제들에 적용한 각 기법들의 성능 평가 결과이며, 각 성능 평가 지표에 대해 평균을 내어 1로 정규화 하였다. 각 지표의 값이 1에 가까울수록 성능이 우수하다.

그림 7을 보면 강건성은 모든 기법에서 동일한 성능을 가졌다(모든 기법이 20개의 수학적 문제의 해를 도출해냄). 그러나 정확도와 효율성에서는 성능차이를 보였다. 총 20개의 수학적 문제에 대하여 정확도의 경우에 EPSAO가 SORS보다 약20%, SRSM보다 약 10% 성능이 우수한 것을 확인할 수 있다.

효율성의 경우에는 EPSAO와 SORS는 성능이 비슷했지만 SRSM은 다른 두 기법에 비해 50%의 성능이 저하 됨을 확인할 수 있다. 그림 8은 수학적 예제 중

### Mathematical Problems

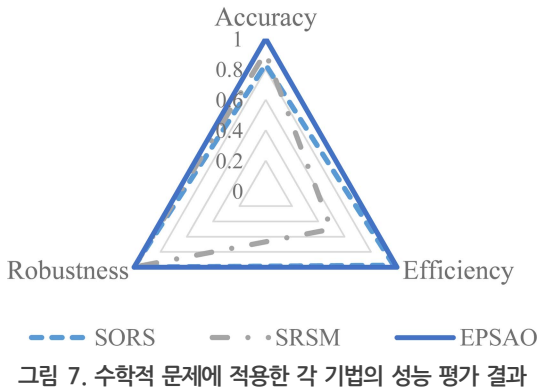


그림 7. 수학적 문제에 적용한 각 기법의 성능 평가 결과

### Schittkowski Problems

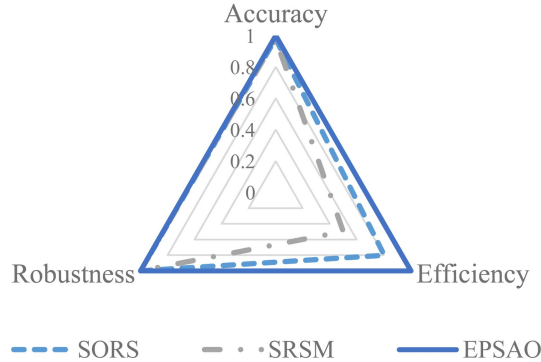


그림 9. Schittkowski 문제에 적용한 각 기법의 성능 평가 결과

### Efficiency - Mathematical (Levy)

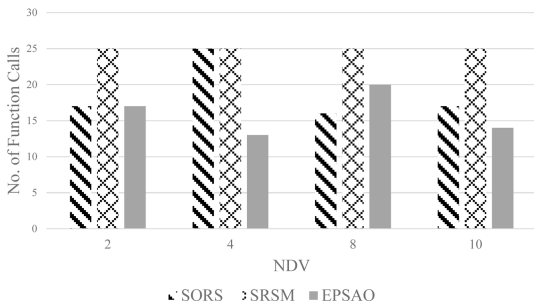


그림 8. Levy 함수에 대한 함수 해석횟수 비교

### Efficiency - Schittkowski Prob.

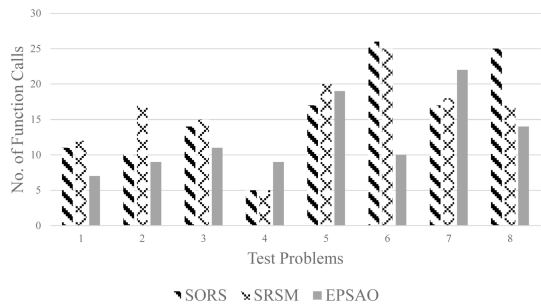


그림 10. Schittkowski 문제에 적용한 각 기법의 함수 해석횟수

에서 대표적으로 Levy function에 대한 각 기법 별 함수 해석횟수를 나타낸 그림이다. SORS와 EPSAO는 효율성이 우수한 것을 확인할 수 있지만, SRSM의 경우는 효율성이 우수하지 못함을 확인할 수 있다.

그림 9를 보면 적용한 예제에서는 모든 기법이 같은 정확도와 강건성을 보였다. 그러나 효율성의 경우, EPSAO가 SORS보다 약 20%, SRSM보다 약 50% 우수한 성능을 보였다. 그림 10에서는 Schittkowski 적용 예제의 각 기법 별 함수 해석횟수를 나타낸 그림으로 문제의 특성마다 전역 최적해를 찾을 때까지의 함수 해석횟수가 차이가 존재하지만, 본 예제에서는 EPSAO의 효율성이 가장 우수함을 확인할 수 있다.

결과를 종합하여 보면 본 논문에서 수학적 예제들과 Schittkowski 문제를 적용한 결과, 본 연구를 통하여 개발한 EPSAO가 SORS와 SRSM에 비하여 우수한 정확성과 강건성을 보장하면서 효율성이 매우 우수한 것을 확인할 수 있었다.

## 5. 결 론

본 논문에서는 산업제품의 상세설계 단계에서 설계 인자의 개수와 상관없이 20~30회 미만의 CAE해석만으로 향상된 해를 도출할 수 있는 EPSAO를 제안했다.

EPSAO는 적응형 초반 초기점 생성 알고리즘, 주요 변수 판별 및 다항 회귀모델 생성 알고리즘, 지능형 설계영역 관리 알고리즘과 전역 최적해를 찾아주는 CMA-ES로 구성 되어 있다. 따라서, 제안한 EPSAO는 기존 근사최적설계 기법이나 순차적 근사최적설계 기법의 초기단계에서 설계인자의 개수에 따라 CAE 해석 회수가 증가하는 문제를 피하고, 효율적으로 개선된 해를 찾을 수 있게 된다.

본 논문에서는 전역 최적해를 찾아주는 개선된 CMA-ES 기법과 EPSAO에 다양한 예제를 적용하여 정확성, 효율성, 강건성을 비교하였다.

총 10개 문제에서 평균오차가 기존의 CMA-ES는 5.63%, 개선된 CMA-ES는 4.3%로 개선된 CMA-ES가 더 우수한 정확성을 보였다. 또한, 개선된 CMA-

ES는 적용한 모든 예제에서 우수한 강건성을 보장하면서 더 적은 평균 함수 해석 횟수로 전역 최적해를 찾음으로써 기존의 CMA-ES보다 더 우수한 효율성을 보였다.

본 논문에서 제안한 EPSAO의 성능은 상용 최적화 소프트웨어인 SORS와 SRSM의 결과와 비교 평가하였다. 결과를 확인하여 보면, 수학적 예제에서는 EPSAO의 정확도가 SORS와 SRSM보다 각각 20%, 10% 우수한 것을 확인할 수 있었다. 효율성의 경우에는 SORS와 유사한 성능을 보이며, SRSM비하여 매우 우수한 성능을 보인다. Schittkowski 문제에서는 정확도와 강건성이 모든 기법이 유사하였지만, EPSAO의 효율성이 SORS와 SRSM보다 각각 20%, 50% 우수한 것을 확인할 수 있었다.

결론적으로 본 논문에서 개발한 EPSAO를 다양한 예제에 적용하여 본 결과, SORS와 SRSM에 비하여 우수한 정확성과 강건성을 보장하면서 효율성이 매우 우수한 것을 확인할 수 있다.

제안한 EPSAO는 현재 구속조건이 없는 문제에만 적용이 가능하다. 그러므로 향후 연구과제로 구속조건 문제를 처리할 수 있는 EPSAO를 개발하고자 한다.

## 감사의글

본 연구는 산업통상자원부 엔지니어링 기술진흥사업 엔지니어링SW 기술개발 기반구축 (N051700001) 과제의 지원으로 수행되었습니다.

## 참고문헌

1. Gu, L. "A Comparison of Polynomial based Regression Models in Vehicle Safety Analysis" ASME Design Engineering Technical Conferences-Design Automation Conference, ASME, Pittsburgh, PA, Paper No. DETC2001/DAC-21063, 2001.
2. G. N. Vanderplaats, "Numerical Optimization Techniques for Engineering Design", McGraw-Hill Companies, 1984.
3. S. Son and D. Choi, "The Effects of Scale Factor and Correction on the Multi-fidelity Model", Journal of Mechanical Science and Technology, 30(5), 2075-2081, 2016.
4. S. Son, B. Choi, W. Jin, Y. Lee, C. Kim, and D. Choi, "Wing Design Optimization for a Long-Endurance UAV using FSI Analysis and the Kriging Method", International Journal of Aeronautical and Space Sciences, 17(3), 423-431, 2016.
5. S. Son, S. Lee, and D. Choi, "Experiment-Based Design Optimization of a Washing Machine Liquid Balancer for Vibration Reduction", International Journal of Precision Engineering and Manufacturing, 13(8), 1433-1438, 2012.
6. S. J. Miller, "The Method of Least Squares", Mathematics Department Brown University Providence, RI 02912.
7. C. R. Rao and S. K Mitra, "Generalized Inverse of Matrices and Its Applications", Wiley, 1971.
8. J. J. Liang, T. P. Runarsson, E. M. Montes, M. Clerc, P. N. Suganthan, C. A. Colleo, and K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization", Technical Report, September 18, 2006.
9. L. C. W. Dixon and G. P. Szego, "The Global Optimization Problem: An Introduction" Towards global optimization, 2, 1-15, 1978.
10. M. Locatelli, "A Note on the Griewank Test Function" Journal of Global Optimization, 25(2), 169-174, 2003.
11. C. Audet, P. Hansen, and F. Messine, "Extremal Problems for Convex Polygons", Journal of Global Optimization, 38(2), 163-179, 2007.
12. Global Optimization Test Functions Index. Retrieved June 2013, from [http://infinity77.net/global\\_optimization/test\\_functions.html#test-functions-index](http://infinity77.net/global_optimization/test_functions.html#test-functions-index)
13. K. Schittkowski and W. Hock, "Test Examples for Nonlinear Programming Codes", Lecture notes in Economics and Mathematical Systems, Springer, 187, 1981.