

전역 최적화 문제의 효율적인 해결을 위한 근사최적화 기법

이세정[†]

서울시립대학교 기계정보공학과

An Efficient Heuristic Algorithm of Surrogate-Based Optimization for Global Optimal Design Problems

Se Jung Lee[†]

Department of Mechanical and Information Engineering, University of Seoul

Received 8 June 2012; received in revised form 5 September 2012; accepted 6 September 2012

ABSTRACT

Most engineering design problems require analyses or simulations to evaluate objective functions. However, a single simulation can take many hours or even days to finish for many real world problems. As a result, design optimization becomes impossible since they require hundreds or thousands of simulation evaluations. The surrogate-based optimization (SBO) strategy became a remedy for such computationally expensive analyses and simulations. A surrogate-based optimization strategy has been developed in this study in order to improve global optimization performance. The strategy is a heuristic algorithm and it exploits not only multiple surrogates, but also multiple optimizers. Multiple optimizations of multiple surrogate models yield multiple candidate design points of optima. During the sequential sampling process, the algorithm ranks candidate design points, selects the points as many as specified, and builds the improved surrogate model. Various mathematical functions with different numbers of design variables are chosen to compare the proposed method with the other most recent algorithm, MSEGO. The proposed method shows superior performance to the other method.

Key words: Approximation, Design of Experiments (DOE), Global optimization, Meta-model, Sequential sampling, Simulation, Surrogate, Surrogate-based optimization (SBO)

1. 서 론

대부분의 공학 설계 문제들은 목적 함수나 구속 조건 함수를 계산하기 위하여 해석 또는 시뮬레이션을 하게 된다. 그런데 실제 설계 문제에서는 시뮬레이션을 컴퓨터에서 한 번 수행하는데 수십 분, 수십 시간 또는 수 일이 걸리는 경우가 많이

있다. 이러한 시뮬레이션이 포함된 최적설계 문제는 경우에 따라 수 백, 수 천 번의 시뮬레이션 계산을 요구하게 되므로 실질적으로 최적화가 불가능한 지경에 이르기도 한다. 그러므로 근사최적화(Surrogate-Based Optimization, SBO) 방법이 이를 해결할 수 있는 대안으로 각광받고 있으며 현재에도 다양한 새로운 기법들이 개발되고 있다. 근사 모델(Surrogate, Meta-model, Response Surface Model, Approximation Model 등 다양한 이름으로 불림)이란 관심 있는 출력 또는 응답을 직접 계산

[†]Corresponding Author, selee@uos.ac.kr
©2012 Society of CAD/CAM Engineers

하지 않고, 그 모델을 만들어 간접적으로 계산하는 공학적 방법이다. 즉 오랜 시간이 소요되는 시뮬레이션을 대체하는 모델로서, 산출되는 출력의 경향을 빠르게 계산할 수 있도록 고안된 수학적 모델이다. 공학 해석의 경우 시뮬레이션 코드의 내부가 알려져 있지 않고 입력과 출력만으로 근사모델을 만들어야 하는 경우가 대부분이다. 이러한 모델링을 블랙 박스 모델링이라고도 하는데, 이를 가능하게 해 주는 근사모델들을 본 논문에서 다룬다.

근사최적화는 근사모델을 이용하여 최적화를 수행하는 방법론을 칭한다. 이는 설계 영역 근사화와는 구별되는 개념으로 설계 영역 근사화에서는 근사 함수의 정밀도, 즉 얼마나 정확하게 출력을 예측할 수 있는가가 관심사이지만, 근사최적화에서는 얼마나 빨리 최적 점을 찾는 지가 관심사이다. 달리 표현하면 근사 함수의 정밀도가 높다고 하여 반드시 최적 점을 빨리 찾는 것은 아닐 수 있다는 것이다. 그러므로 설계 영역 근사화에서는 시스템의 전체 영역에서 시뮬레이션의 출력의 경향을 잘 표현하도록 근사모델을 구축한다. 이러한 근사모델은 시스템의 전역적 경향을 쉽게 파악할 수 있는 도구를 제공하며, 최적화는 그 후처리 과정으로 포함될 수도 있다. 그러나 이러한 방법은 근사모델의 개선 과정이 없으므로 후처리 과정에서 찾은 최적 점을 검증할 수가 없다.

근사최적화의 기본 개념은 가능한 적은 수의 시뮬레이션 계산을 하여 근사모델을 개선시키고, 점차적으로 최적 값을 찾아 가는 방법이다. 근사최적화에 대하여 개요 및 구체적인 관련 주제에 대한 논의는 Simpson^[1]이나 Queipo^[2]를 참조하고, 최근 진전된 연구 결과 및 연구 경향에 대해서는 Forrester^[3]와 Jin^[4]의 논문을 참고하면 개괄적인 방향을 잡을 수 있다.

전형적인 근사최적화 절차는 Fig. 1과 같다. 우선 초기 실험 점(설계 점 또는 샘플)을 정하여 근사모델을 만들고, 이 근사모델의 최적화를 수행한 후, 새로운 실험 점을 추가하여 근사모델을 다시 만드는 작업을 반복한다. 근사최적화에서 중요한 논제는 크게 두 가지로 요약할 수 있는데, 실험 점의 선택과 근사모델의 구축이다. 현재까지 대부분의 연구자들이 관심을 가지고 집중했던 분야가 이 두 분야이다. 한편 저자는 위 두 가지 주제에 한 가지 주제를 추가하고자 하는데, 바로 최적화 기술의 적용 방식이다. 이는 본 논문의 핵심 개념이

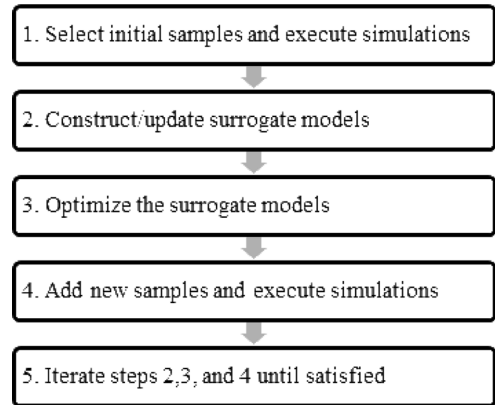


Fig. 1 A typical procedure of SBO

며, 이에 대하여는 다음 장에서 자세하게 다루기로 한다.

설계 영역 근사화와 근사최적화 분야에서 어떠한 근사모델이 좋은 모델인가에 대하여 지난 십여 년 간 많은 연구가 이루어졌다. 2001년 Jin^[5]로부터 Kleijnen^[6]과 Wang^[7]은 각각 2005, 2007년 까지 근사화 방법론에 대한 다양한 제안들을 정리하여 분석하였으며, 최근에는 Reischel^[8]과 Betteghoret^[9], Li^[10] 등이 적용 분야는 다르지만 근사모델의 활용에 대한 개관을 잘 보여주고 있다. 한편 근사최적화 분야의 최적화에 사용된 근사모델의 장, 단점에 대한 논의도 다양한 문헌들^[3,7,11-13]에서 보고되었다. 이들 문헌에서 비교에 많이 사용되는 근사모델은 다항식 회귀(Polynomial Regression, PR), 크리깅(Kriging, KRG), Support Vector Machine(SVM), Radial Basis Function(RBF) 근사화 등이었다. 대부분의 경우, 원래 함수의 경향을 미리 알지 못하므로 어떤 근사모델이 정확할 지 알 수가 없으며, 특정 근사모델의 정확도를 예측하는 방법에 있어서도 공감대가 있지는 않다. 그래서 이들 문헌들을 요약하여 보면, 하나의 특정 근사모델이 좋다는 쪽으로 수렴한 것이 아니라 각기 경우에 따라 다른 근사모델이 우수하였다는 결론을 얻었다. 즉 어떤 하나의 특정 모델이 항상 우수한 근사모델이 아니라는 결과를 얻은 셈이다. 또한 최적화와 관련 지어서는 근사모델의 품질이 좋다고 하여 꼭 최적화가 잘 수행된다는 보장도 없다는 점을 배웠다.

이후 많은 연구자들이 여러 개의 근사모델을 함께 사용하는 방법을 연구하기 시작하였다. 이 분야의 연구는 두 가지 방향으로 발전하였는데, 하

나는 다수의 근사모델의 평균을 취하는 방법이 다^[14-17]. 이들은 다양한 근사모델을 활용하여 함수 값 예측 시 평균을 만들었는데, 다른 용어로는 근사모델의 가중 평균법이라고도 부른다. 이 방법은 이후 가중치를 정하는 다양한 방법들이 개발되었다. 이들 문헌들의 결론은 정확도가 우수한 근사모델이 항상 가장 좋은 설계로 이어지지 않았고, 다만 최적화 해법의 강건 성을 높일 수는 있었다고 보고하였다. 최근에 Gu^[18]는 전역 최적화에서 해를 빨리 찾을 수 있도록 자동으로 근사모델을 선택하는 방법을 제안하였는데, 선택과 더불어 새로운 실험 점을 탐색하는 과정을 수행하도록 제안하였다.

또 다른 하나의 방법은 다수의 근사모델을 직접 활용하는 것인데, 본 논문에서는 이를 다중 근사모델이라 칭하겠다. Muller^[19]와 Viana^[20,21]는 다중 근사모델을 이용한 최적화 방법을 제안하였다. 이들은 가장 우수한 근사모델을 선택하여 최적화를 진행하는 방법, 근사모델의 가중평균을 이용하는 방법과 자신들이 제안한 방법을 비교, 분석하였다.

앞에 소개한 대부분의 기존 연구들의 특징은 최고의 근사모델이던, 다중 근사모델이던 각각의 근사모델들을 특정 기준에 의하여 평가하고, 그 결과 취사 선택, 또는 가중 평균을 취하는 방법을 사용하였다. 결국 이 특정 기준이 무엇이나에 따라 근사모델의 품질이 결정된다고 볼 수 있다. 가장 널리 문헌에서 사용되는 기준이 실험 점의 추가 없이 근사모델의 정확도를 예측할 수 있는 교차 유효성 검사(Cross-validation, CV)이다. 이 방법에 의하여 각 근사모델의 오차를 계산하고 비교하여 선택 또는 평균하는 방법이 널리 사용되었다.

다만 Viana^[21]는 다중 근사모델을 직접 이용하는 방법과 가중평균 방법을 비교하였는데, 사용한 최적화 기법은 크리깅을 기반으로 하는 Efficient Global Algorithm(EGO)에 다른 근사모델을 보조로 활용하였으며, 결과적으로 다중 근사모델의 직접적 이용이 가중평균 방법보다 우수한 최적화 성능을 보였다. 즉 다중 근사모델을 이용하여 모델을 선택 또는 평균을 하지 않고, 단지 근사최적화 과정 중 새로운 실험 점을 선택할 때 활용하여 최적화 효율을 더 높일 수 있음을 보였으며 따라서 최적화에서는 다중 근사모델을 사용할 것을 추천하였다. 이 방법이 근사모델의 정확도를 추측하는 것보다 개념적으로 강건한 방법이 될 수 있는 가

능성을 보였는데, 본 연구는 이 개념을 토대로 하여 추가 실험 점을 선정하는 새로운 기법을 제안하고자 한다. 따라서 근사최적화의 여러 주제들 중 근사모델을 작성하여 최적화를 진행하여 가는 과정에 집중하여 논문을 기술하겠다.

근사최적화에 관련된 또 하나의 주제는 실험 점의 선정 방법이다. 근사모델의 정확도는 시물레이션 또는 실험을 수반하는 실험 점의 개수와 위치에 따라 달라진다. 실험 점을 선택하는 행위를 실험계획(Design of Experiment, DOE)이라고 하며, 실제 실험과 달리 컴퓨터 시물레이션의 경우처럼 같은 입력에 대하여 반복적으로 같은 출력이 나오는 경우에 DACE(Design and Analysis of Computer Experiments)로 구분하여 칭하기도 하는데, 본 논문에서는 컴퓨터 시물레이션을 대상으로 하고 있으나 편의상 널리 사용되는 용어인 실험계획으로 구분 없이 부르기로 한다. Fig. 1에 근사최적화 절차에서는 실험계획이 1, 4단계에 필요하다. 이를 각각 초기 실험계획과 추가 실험계획으로 부르기로 한다. 추가 실험계획은 초기 실험계획과 무관하게 실행할 수는 없다. 왜냐하면 초기 실험 점들의 특성을 무시할 수가 없기 때문이다. 그 특성이란 근사모델의 품질을 최상으로 유지하기 위하여 실험 점들의 위치가 지켜야 하는 규칙들이다. 예를 들면 두 개 이상의 실험 점이 겹치는 경우 근사화가 불가능한 모델도 있다. 또한 시물레이션의 전체 경향을 잘 대표하려면 실험 점들이 전체 설계 영역에 걸쳐 골고루 분포하는 게 유리하다.

근사화를 위한 실험계획법은 전통적으로 Monte Carlo Simulation(MCS)와 Latin Hypercube Design (LHD)가 많이 사용되었다. 두 방법을 비교해 보면 LHD가 더 효율적이라는 결론이었다^[22]. 원래 LHD는 일 차원 실험 점에 대해서만 균일 성(Uniformity)를 보장하기 때문에 다차원에 대하여 균일 성을 확보하기 위하여는 실험계획에서 보안을 하여야 한다. 예를 들면 Johnson 등^[23]이 제안한 minimax 또는 maximin 거리 개념을 도입한 수정된 LHD, 직교 배열을 이용한 개선책^[24] 등이다. 이외에도 다양한 최적화 또는 준(Quasi) 최적화를 활용하기도 하는데, Cioppa and Lucas^[25]는 계산 시간이 많이 걸리기는 하지만 차원 수가 많은 경우에도 균일 성을 확보하는 법을 제안하였고, 가장 최근 논문은 Deutsch and Deutsch^[26]가 발표한 것으로 차원 수가 아주 많은 경우에 적용할 수 있는 새로운 기

법을 개발하여 MCS 및 기존 LHD방법들과 비교하였고, Zhu 등^[27]은 maximin LHD를 개선하여 공간 충전(Space-filling) 특성도 충족하도록 하였다. 한편 Crombecq 등^[28]은 실험 점이 만족해야 하는 다양한 특성들을 최적화를 통하여 조합하는 새로운 실험계획법을 제안하였는데, MATLAB의 lhsdesign이 반복횟수가 늘어나면 내부 최적화에 걸리는 계산 시간이 너무 길어져 비효율적이라고 언급하였다.

MCS도 발전하여 준 MCS 또는 준 랜덤 수열(Quasi Random Sequence, QRS)이라고 불리기도 하는데, LHD에 비하면 연구가 덜 활발한 편이었다. 그러나 최근에 Crombecq^[29]이 QRS가 최적화된 LHD보다 좋은 품질의 근사화를 할 수 있다는 보고를 하였다. 또한 Singhee 등^[30]도 통계적 회로 분석 분야에 적용한 결과 QRS가 LHD보다 우수함을 보고하였다. Azoulay^[31]는 최근 QRS 방법을 프로그램화 하기 위한 가이드를 제시하였다. 이러한 연구들을 종합하여 보면, 근사화를 위한 실험계획도 근사모델을 선택하는 방법과 마찬가지로 적용 분야나 근사모델의 종류 등에 따라 어느 특정 방법이 언제나 우수하다고 결론을 낼 수는 없다고 하겠다. 본 연구에서는 초기 및 추가 실험계획에서 QRS를 사용한 경우만을 다루기로 한다. 그 이유는 최적화된 LHD는 실험 점을 만드는데 필요한 컴퓨터 계산 시간이 QRS와는 비교할 수 없을 정도로 길고, 실제 실험 점의 품질은 QRS가 LHD보다 최소한 나쁘지는 않다는 위의 인용 논문들의 결과를 확인하였기 때문이다.

근사최적화에 관련된 중요한 주제들에 대하여 논의를 하였으므로, 다음 장부터 본 연구에서 제안하는 다중 근사모델을 이용한 다중 근사최적화의 개념을 설명하겠다. 이어서 제안한 방법과 가장 최근에 발표된 다른 문헌에서 개발한 방법을 비교하기 위하여 수치 예제들을 소개하고, 수치 실험 방법을 설명하겠다. 마지막에 수치 실험 결과의 분석과 더불어 연구의 결론을 맺도록 하겠다.

2. 다중 근사모델을 이용한 다중 근사최적화

근사최적화의 개론적 소개는 Wang and Shan^[7]에 잘 정리되어 있다. 최근에는 Kazemi 등^[13]은 제한조건이 있는 근사최적화 문제에 대한 해결법을

제안하였고, Luo 등^[32]은 EGO 알고리즘에 근사 모델링 방법을 내장시켜 전역 최적화하는 시도를 하였다. 또한 Yahyale 등^[33]은 국부적 특성을 활용하는 근사최적화 방법을 제안하였다. 최근 경향은 특정 문제를 풀기 위하여 근사화와 최적화를 함께 조합하여 알고리즘화 하고 있다. 그 이유는 두 기법이 융합되면서 내부 파라미터를 효과적으로 조절할 수 있기 때문이다. Kuo 등^[34]은 Particle Swarm Optimization(PSO)를 근사최적화에 접목시켰고, Liu 등^[35]은 진화 알고리즘을 도입하고, 근사화는 자동으로 파라미터를 결정해 주는 Taylor크리깅^[36]을 도입하였다. 그러나 하나의 특정 방법이 모든 문제에 좋은 결과를 줄 수도 없겠지만, 근사화와 최적화 기법들이 방대하게 존재하므로 이들의 모든 조합을 시험해 볼 수는 없을 것이다. 따라서 저자는 둘을 융합하는 것이 아니라 기존의 코드들을 활용하여 조합하되, 특정 방법에 국한되지 않도록 새로운 근사최적화 전략을 제안하고자 한다. 본 연구에서 최적화 문제는 모두 최소화 문제로 변환하여 사용하였으며, 최적화는 최소화를 의미한다.

본 연구에서 제안하는 근사 최적화 방법의 핵심은 다중 근사모델을 이용한 순차적 실험계획과 다중 최적화를 통한 추가 실험 점의 선택 방법이다. 근사최적화에 다중 근사모델을 사용하는 개념은 EGO와 같이 가장 널리 사용되는 최적화 방법에 크리깅을 이용하여 한 개의 실험 점만을 추가하였기에 한꺼번에 여러 개의 실험 점을 추가할 수 있도록 고안되었다. Viana 등^[20,21]의 논문을 보면 다중 근사모델을 사용하여 목적함수를 근사화해서 기존의 교차 유효성 검사(CV), 기대향상 값(Expected Improvement, EI), 또는 향상 확률(Probability of Improvement, PI) 등의 성능 지표를 이용하여 정밀도가 떨어진다고 진단되는 근사모델은 삭제하고 상대적으로 정밀한 근사모델에서 예측되는 실험 점을 복수로 추가하는 방법을 제안한 것이다. 그러나 저자의 제안은 어차피 이러한 성능지표들이 모두 예측 값이라는 한계가 있으므로, 반복되는 최적화 과정 중의 현 상태에서 판단할 수 있는 지표는 오로지 근사 최적 값에 해당하는 실험 점이라는 점이다. 따라서 반복과정 중 추가 실험 점은 근사최적화의 최적 점이 되고, 다중 근사모델을 사용하므로 여러 개의 실험 점이 추가할 수 있는 예비 후보가 된다. 이 기법의 또 다른 장점은 CV, EI, PI의 계산에 요하는 추가부담이 없다는 것

이다. 다중 근사모델을 이용한 EGO(Multiple Surrogate EGO, MSEGO)와 비교해 본 결과 계산 시간이 상당히 단축되는 것을 볼 수 있었다.

순차적으로 실험 점을 선택 할 때는, 후보 실험 점 중에서 이전 사이클(cycle 또는 iteration)에서의 최적 값보다 향상되지 않은 후보들은 제외시킨다. 알고리즘 초기에 매 사이클마다 추가할 실험 점 개수를 정하는데, 만일 현 사이클에서 남은 후보의 수가 추가 실험 점 개수보다 적으면, 탐색하지 않았던 설계 영역에서 실험 점을 추가한다. 이 기법은 초기 실험계획을 어떤 기법을 사용하였는지에 따라 매우 복잡한 계산이 수반될 수도 있다. 예를 들면 OLHD 같은 경우가 이에 해당된다. 저자는 본 연구에서 QRS를 사용한다고 하였는데 이 추가 실험계획 과정에서 이 선택의 장점이 확인해 지는데, 거의 계산이 없이 추가 실험 점이 선택된다. 사용한 QRS는 Sobol 수열이며, 대표적인 불일치가 작은(Low-Discrepancy) 수열이며 전체 실험 점 수에 제한이 없다.

근사최적화를 진행하다 보면, 최적화에 실패하는 경우가 생긴다. 이러한 사실은 하나의 근사모델에 대하여 하나의 최적화 기법을 사용한 경우는 어려운 상황에 직면하게 됨을 의미한다. 해결책은 해당 사이클에서 영역 탐색용(Exploration) 실험 점만 추가하거나, 아니면 매우 어려운 복구 과정을 개발하여야 한다. 즉 알고리즘의 강건 성 면에서 취약하다. 앞에 소개한 다중 근사모델을 사용하더라도 목적함수의 복잡도에 따라 최적화 기법이 적합하지 않은 경우가 생길 수도 있는데, 이를 미리 알 수가 없다는 점이 어려운 문제이다. 따라서 저자는 최적화 기법도 복수로 사용하여 강건 성을 확보하였다. 이미 근사모델이 만들어져 있으면 최적화를 복수로 하는 것이 계산상 큰 부담이 되지 않으며 병렬처리도 쉽게 구현할 수 있다. 그래서 전역 최적화 문제를 풀고 있으나, 경우에 따라 국부 최적 점이 유용한 경우가 있으므로 국부 최적화 기법도 포함시킬 수 있다. 본 연구에서 사용된 최적화 기법은 PSO, Simulated Annealing(SA), 진화 알고리즘, MATLAB의 fmincon의 네 종류이다. 예를 들면 네 가지의 최적화 기법과 네 가지의 근사모델을 이용한다면 총 열 여섯 개의 후보 실험 점을 만들 수 있다. 여기서 네 개만을 선택하기로 하였다면, 우선 이전 사이클보다 열등한 최적 점을 제시한 후보는 제외시키고, 중복된 후보인 경

우 단일화하고, 나머지 후보의 개수가 넷 이상이면 최적 값의 우열에 따라 네 점을 선택하고, 넷 미만이면 모두 선택하고 모자라는 수만큼 QRS로 추가한다. 전술한 다중 근사모델과 다중 최적화 기법을 활용한 추가 샘플링 방법을 알고리즘으로 기술하면 다음과 같다.

Sequential Sampling Algorithm

- (1) Set n_s = number of surrogates,
 n_o = number of optimizers,
 m = number of samples to add per cycle,
 X = initial samples at the current cycle,
 Y = simulation output at X ,
 $Y = \min(Y)$
- (2) Optimize the n_s surrogates with the n_o optimizers respectively to yield $(n_s \times n_o) \hat{Y}^*$
- (3) Remove the samples already contained in the initial samples and unify the duplicates
- (4) Remove the samples not shown any progress compared to the previous minimum
- (5) Let m_r = number of remaining samples out of $n_s \times n_o$ samples,
 if $m_r \geq m$, select m samples with smallest \hat{Y}^*
 if $m_r < m$, add all m_r samples and $(m - m_r)$ QRS samples
- (6) Perform m simulations at the samples chosen and update X and Y

위 알고리즘은 1장 Fig. 1의 4 단계를 구체적으로 구현한 것이며 이로써 전체 근사 최적화 프레임워크가 완성되었으며, 이를 Multiple-Surrogate-Multiple-Optimizer(MSMO) 방법으로 부르기로 한다.

3. 수치 실험 환경

제안한 알고리즘의 성능 평가를 위하여 구속 조건이 없는 전역 최적화 함수들을 문헌에서 선택하였다. 함수들은 모두 최소화하는 문제로 설정되었으며, 다만 예제들 중에서 설계 영역이 대칭이며 전역 최적 점이 영역의 중간인 경우, 최적화 방법에 따라서 우연히 해를 쉽게 찾는 경우가 발생할 수 있으므로 설계 영역을 고의로 비대칭으로 수정

하여 이러한 일이 생길 가능성을 배제하였다. 함수의 선택 기준은 기존 문헌에서 검증되어 있는 함수 중, 최적화 난이도, 설계변수의 개수, 다중 최적 점의 수 등의 다양성을 추구하였다. 선택한 함수들과 그 설계 영역, 이미 알려져 있는 전역 최적해를 다음에 정리하였다.

Branin function (branin)

$$f(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$$

$$-5 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 15$$

$$f^* = 0.3979, \quad \mathbf{x}^* = \{ \pi, 2.275 \}$$

Sasena function (sasena)

$$f(\mathbf{x}) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(1 - x_2)^2 + 7\sin(0.5x_1)\sin(0.7x_1x_2)$$

$$0 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 10$$

$$f^* = -1.4565, \quad \mathbf{x}^* = \{2.504, 2.578\}$$

Rosenbrock's valley function (rosen)

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$$

$$-2 \leq x_i \leq 2 \text{ for } i = 1, \dots, n$$

$$f^* = 0, \quad x_i^* = 1, \text{ for } i = 1, \dots, n$$

Schwefel's function (schwe)

$$f(\mathbf{x}) = \sum_{i=1}^n [-x_i \sin(\sqrt{|x_i|})]/n$$

$$0 \leq x_i \leq 500 \text{ for } i = 1, \dots, n$$

$$f^* = -418.98, \quad x_i^* = 420.97, \text{ for } i = 1, \dots, n$$

알고리즘을 개발하면서 주로 branin과 sasena 함수를 활용하여 성능을 검사하였는데, 이 두 함수는 설계변수가 두 개이므로 비교적 어렵지 않게 해를 찾을 수 있기 때문에 선택했다. branin함수는 최적화 분야에서 오래 전부터 사용하던 것으로 Jones^[37]와 Rafael^[38]에서 사용된 예를 찾아볼 수 있다. sasena함수는 비교하고자 하는 알고리즘을 개

발한 논문^[21]에서 채택한 예제 문제이다. rosen과 schwe는 설계변수의 개수를 2, 4, 8개로 변화시켜, 변수의 수가 변화할 때 최적화 기법의 성능 변화를 조사할 수 있다. rosen은 바나나 함수라고도 부르는데, 변수가 두 개인 경우, 전역 최소 점이 가늘고, 긴 평평한 계곡 내부에 존재하므로 전역 최적화가 쉽지 않은 문제이다. schwe는 많은 국부 최소 점이 존재하며 전역 최소 점은 가까운 국부 최소 점에서 멀리 떨어져 있는 함수이므로 전역 최소 점을 찾을 때 엉뚱한 점으로 수렴하기 쉬운 함수이다^[39].

비교 대상인 알고리즘은 Viana^[21]의 MSEGO이다. 가장 최근에 발표된 논문이기도 하고, 이미 다른 기법들과 비교하여 우수함을 보였던 알고리즘이다. 특히 다중 근사모델의 가중평균 결과와도 비교하여 우수성을 증명하였다. 본 연구에서 제안한 알고리즘으로 위 예제 문제들을 풀고 Viana가 공개한 MATLAB 툴 박스^[40]를 수정 없이 그대로 이용한 결과와 비교를 하였는데, MSEGO가 최적해를 주어진 정확도에 따라 구하지 않고, 일정 횟수의 사이클을 반복하였을 때 실행을 정지하는 수렴 조건을 택하였기에 모든 비교에서 일정 횟수 반복 수행 후 샘플 중 최소 값을 비교하였다. 다만 샘플링이나 최적화 기법 자체에 난수 발생기를 이용한 경우 시작하는 종자(seed)에 따라 알고리즘이 다르게 반응하여 다른 답을 산출할 수가 있다. 그래서 각 경우마다 최소한 열 번 이상 실행시킨 다음, 그 결과를 비교하여 수렴속도가 중간인 결과만을 본문에서 서로 비교하였다.

본 연구의 수치 실험을 위하여 쿼드 코어 처리기를 장착한 컴퓨터에 윈도우7환경에서 MATLAB R2011a를 사용하였다. MATLAB에서 근사모델링, 최적화, 샘플링 등을 위하여 다음에 설명하는 툴 박스들을 사용하였는데, 따로 기술하지 않은 경우, 설정해야 할 파라메타들은 기본(default) 값을 사용하였다.

전역 최적화를 위해서는 Donckels^[41]의 툴 박스를 사용하였다. 이 툴 박스는 세 종류 최적화 기법을 제공하는데, Simplex-Simulated Annealing (SSA), Particle Swarm Optimization(PSO), Shuffled Complex Evolution(SCE)이다. SSA는 모의 담금질 방법에 기초하여 비선형 Simplex 알고리즘을 조합하였다. PSO는 집단을 토대로 한 최적화 기법으로 초기 해 집단을 무작위로 생성하여 최적 해

를 탐색하는 데, 새 떼와 물고기 떼와 같은 생체 군집의 사회적 행동 양식을 바탕으로 하고 있다. 개별 해(particle)는 최적 해를 얻기 위해 다차원 공간을 옮겨 다니며 그들 자신과 그들 이웃의 경험에 대한 정보를 이용하여 최적 점으로 이동한다. SCE는 자연 선택의 진화 메커니즘을 모방하였는데, 무작위로 선택한 초기 개체를 몇 개의 집단으로 나누어 각 집단이 재생산 과정을 거치고, 다시 이합집산을 행하는 과정을 반복하여 최적 해에 이르게 하는 방법이다. 세 방법 모두 널리 사용되는 전역 최적화 방법들이므로 본 연구에서는 모두 채용하였다. MATLAB에서 제공하는 전역 최적화 툴 박스도 있는데, 본 연구에서 사용한 Donckels의 코드보다 우수하지 않아서 사용하지 않았다.

실험계획, 근사모델 작성, EGO 형식의 근사최적화 등은 Viana의 툴 박스를 사용하였다. 이는 기존의 공개되어 있었던 코드들을 기초로 하여 저자의 알고리즘을 결합하여 다양한 기능을 제공하는 종합 툴 박스로 발전되었다. 이는 OLHD를 포함하는 다양한 실험계획법, PR, KRG, SVM, RBF 등의 근사모델링, EGO, MSEGO 등과 같은 근사최적화, CV와 같은 오류 분석 등의 기능을 제공하고 있으며 일반에 공개되어 있다.

4. 수치 실험 결과

본 논문의 예제 문제에서 사용한 근사모델은 네 종류로 KRG, SVM, RBF, SHE 모델인데 모두 Viana가 공개한 툴 박스에 구현되어 있는 그대로 사용하였으며, SHE는 선형 Shepard^[42] 모델이다. 따로 설정을 하지 않으면, KRG는 전역 경향 함수로 상수를, 상관 관계 함수로 Gaussian을 사용하고, SVM은 Gaussian 함수를 사용한다. 또한 RBF는 Multiquadric 함수를 기저 함수로 사용한다. 최적화를 위해서 전역 최적화와 국부 최적화를 함께 사용하였는데, 전역 최적화 기법은 SSA, SCE, PSO의 세 가지를, 국부 최적화 기법은 fmincon을 사용하였다. 따라서 목적함수에 상관없이 네 가지 최적화 기법과 네 종류 근사모델이 사용되므로 순차 샘플링의 매 사이클마다 열 여섯 개의 후보를 만들고 그 중 적합한 실험 점을 선택하는 과정이 반복된다. 추가 실험 점의 수는 모든 함수의 최적화 과정에서 동일하게 4개로 정하였다. 2, 4, 8 등 여러 시나리오를 적용하였는데, 그 결과는 대동소이

하므로 4개인 경우만 본 논문에 실었다. 다만, 추가 실험 점을 선택할 때 최적 점과 공간 충전 실험 점을 각각 몇 개로 할 것인가도 결정해야 할 중요한 요소이다. 하지만 이는 목적함수의 특성에 좌우되는 경향이 크므로 본 연구에서는 총 4개로 정하고 우선적으로 최적 점에서 선정하되 4개를 못 채우는 수만큼 만 QRS에 의해 실험 점을 추가하였다. 초기 실험 계획에는 QRS중 Sobol수열이 사용되었으며, 추가 실험 계획 시 근사 최적 점의 수가 필요한 실험 점 수보다 작을 때 그 나머지를 만들 때에도 같은 수열을 사용하였다.

근사모델에 대한 최적화를 할 때, 수렴조건은 목적함수 및 설계변수의 상대오차를 10^{-7} 으로 정했다. 한편 근사모델의 품질을 높이기 위하여 설계변수를 0에서 1사이로 정규화 하였고, 목적함수의 값은 정규화 하지 않았다. 풀고 있는 예제 문제들은 모두 전역 최적 해를 알고 있는 문제들이므로 상대오차를

$$RE = |Y_e^* - Y^*| / (1 + |Y_e^*|)$$

의 식으로 계산하여 정확도를 비교하였다. 여기서 Y^* 는 근사 최적화 한 사이클이 끝난 다음 갱신한 함수 값 Y 중에서 최소 값이며, Y_e^* 는 미리 알고 있는 전역 최적 해이다.

근사모델을 만들 때, 모델의 특성에 따라 중복된 샘플이 들어가면 모델의 품질이 열악해 지거나 아예 모델을 못 만드는 경우도 생길 수 있다. 그래서 본 연구에서는 근사모델을 만들기 전에 실험 점간 최소 거리를 계산하여 그 거리가 10^{-11} 보다 작으면 중복된 것으로 판단하고 하나를 제거하였다. 그러나 근사모델에 대한 최적화를 통하여 추가되는 실험 점의 경우에는 최적 해의 정밀도에 연관되므로 다른 방식으로 판단하였다. 즉 기존 실험 점들과의 최소 거리를 계산한 다음, 최적화의 수렴 조건과 기존 실험 점들간의 최소거리의 1/10 중 큰 값보다 작으면 추가 실험 점에서 제외하였다.

선정한 함수들에 대하여 제안한 방법과 MSEGO를 비교하여 본다. 모든 비교에 있어서 함수 계산 횟수는 초기 실험 점의 수에 최대 사이클 수의 배를 한 값을 더한 값이다. 그래서 MSMO와 MSEGO가 같을 수 밖에 없고, 함수 계산 횟수에 대한 언급은 더 이상 하지 않겠다. 우선 설계변수가 두 개인 branin, sasena, rosen, schwe에 대하여

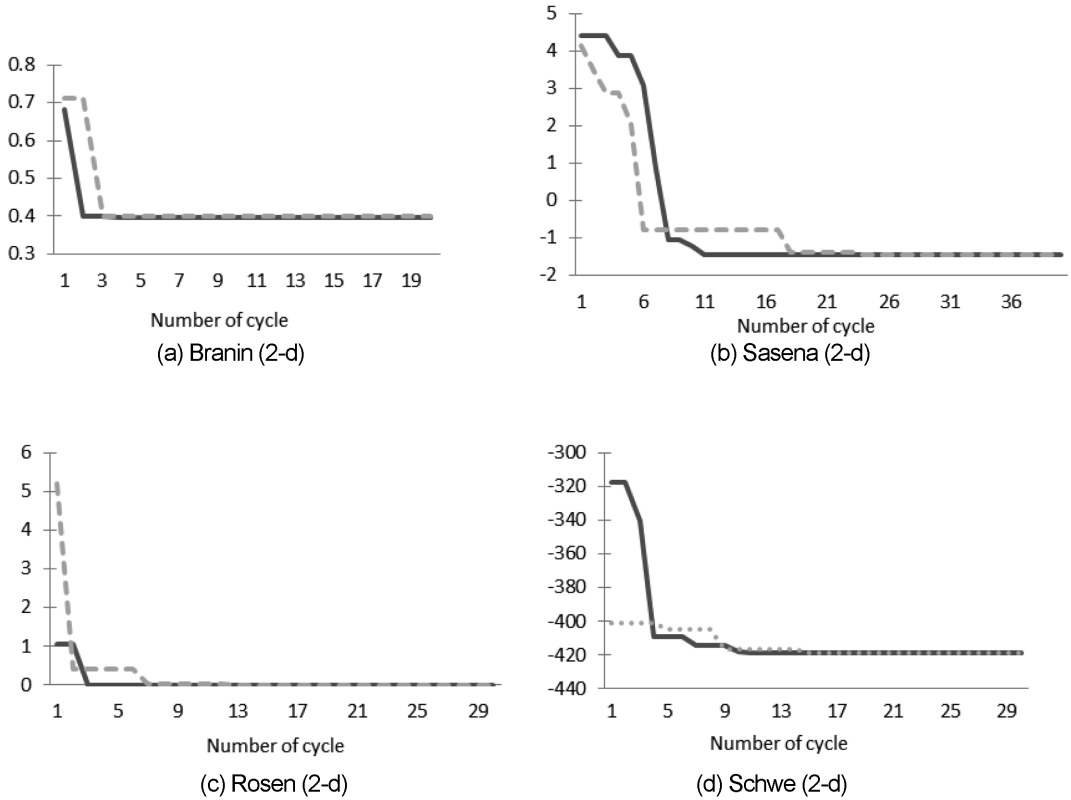


Fig. 2 Objective function histories of 2-d functions (solid line: MSMO, dotted line: MSEGO)

성능을 비교해 보자. 네 함수 모두 최적화에 실패한 경우는 없었다. Fig. 2는 순차적 근사 최적화 과정에 따라서 네 가지 목적함수 값의 변화 추이를 그래프로 표현하였다. sasena의 경우를 제외한 나머지 세 경우에서 MSMO가 더 빨리 값이 줄어들어 가는 것을 확인할 수 있다. Table 1에 각 함수 별로 초기 실험 점의 수, 최대 최적화 반복 횟수, 마지막 사이클에서의 실제 최적 해와의 상대오차를 수록하였다. 역시 sasena의 경우를 제외하고 MSMO가 우수한 결과를 보여 준다. sasena 함수는 MSEGO 개발자의 문헌에서 사용되었던 예제

문제를 가져 온 것인데, MSMO 방법이 MSEGO 보다 우수한 결과를 보여준 경우도 많았는데, 모든 수치 실험의 중간 값을 취하다 보니 MSEGO가 이 특정 함수에 대하여 우수한 결과를 보였다. 이는 다른 함수들을 활용하여 다시 확인하겠다. 다만 이 함수의 경우, 계산 시간을 비교하여 보았는데, MSMO는 28초, MSEGO는 277초 소요되었다. 여기서 두 방법의 비교를 공정하게 하기 위하여 MSMO의 병렬 처리 기능을 끄고, 순차 처리만을 하여 계산 시간을 비교하였는데, MSMO가 거의 10배 빠른 계산 속도를 보여 주었다. 이유는 순

Table 1 Numerical settings and relative errors of 2-d functions

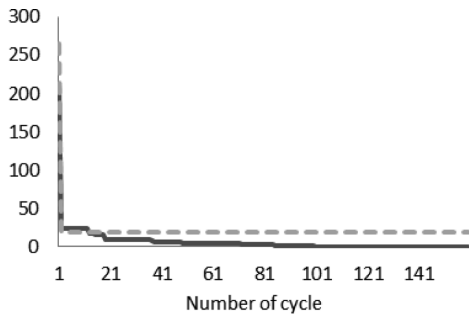
Functions	Initial number of design points	Maximum number of cycles	Relative errors	
			MSMO	MSEGO
Branin	12	20	2.86E-7	1.94E-4
Sasena	12	40	1.12E-3	4.23E-5
Rosen	12	30	3.80E-7	3.14E-4
Schwe	12	30	9.40E-7	1.17E-7

차적 실험 계획 시, EI나 PI와 같은 근사모델의 품질을 평가하기 위한 지표들의 예측 계산을 하지 않기 때문에 사료된다. 또한 Table 1에서 상대오차가 10^{-6} 보다 작은 경우는 모두 10^{-6} 으로 간주해야 할 것이다. 왜냐하면 근사 최적화 과정에서 해의 수렴 조건을 10^{-7} 으로 정하였으므로 더 작은 상대오차가 나온다는 것은 행운으로 봐야 할 것이다.

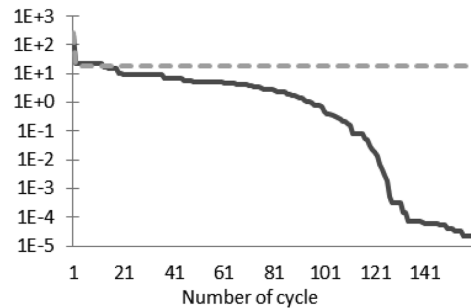
Fig. 3은 4-d rosen 함수에 대하여 초기 실험 점을 24개로 한 경우의 결과이다. MSMO는 46 사이클 만에 10^{-3} 수준의 상대오차에 도달하였으며 마지막 60사이클에서 10^{-5} 수준에 이른 반면, MSEGO는 초기 샘플에서의 목적함수 값이 더 작았음에도 불구하고 최적 해를 구하지 못하고 마지막 사이클까지 도달하였다. 마지막 사이클에서의 상대오차가 10^{-1} 수준이므로 해를 못 찾은 것으로 보아 무방할 것이다. 특히 Fig. 3(b)를 보면 MSMO는 사이클이 경과함에 따라 꾸준히 상대오차가 줄어드는 반면, MSEGO는 어느 순간부터 정체되어 있는 것을 알 수 있다. Fig. 4는 4-d schwe 함수인데, 초기 실험 점은 80개로, 90 사이클까지 최적화를 수

행하였으며, 이 결과는 rosen에서의 비교 결과와 유사하다. MSEGO는 마지막 사이클에서 상대오차가 10^{-2} 에 도달하였고, MSMO는 10^{-5} 수준에 도달하였다. 상대오차의 추이를 보면 역시 MSEGO의 줄어드는 속도가 현저하게 느린 것을 확인할 수 있다.

Fig. 5는 8-d rosen의 경우이다. 설계변수의 수가 많아지면서 점점 최적화 과정의 난이도가 증가하는데, 최대 사이클 수도 160으로 증가시켰고, 초기 샘플 수는 200개로 정하였다. 초기 실험 샘플에서의 근사 최적 해는 MOMS가 210, MSEGO가 265로 비슷한 수준이었다. 최적화가 순차적으로 진행되면서 초반에는 두 방법 모두 빨리 함수 값을 줄였는데, 상대오차를 보면 중반 이후 MSEGO의 성능이 열악함을 알 수 있다. 반면 MSMO는 안정적으로 최적 해를 찾아 가는 것을 확인했다. 마지막 사이클에서 상대오차는 MSMO는 10^{-5} , MSEGO는 10^{-1} 으로 MSEGO가 해를 찾지 못하였다. 이 경우 MSEGO를 11번 재 실행 하였는데 모두 해를 찾는 데 실패했다. 8-d rosen함수에 대하여 근사

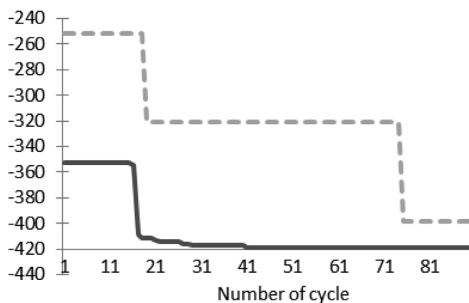


(a) Objective function histories

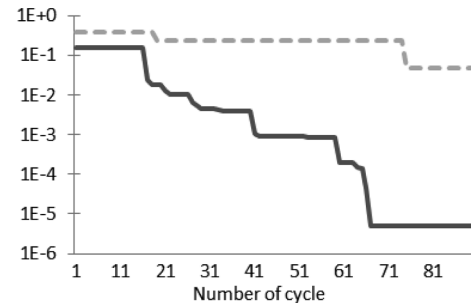


(b) Relative errors

Fig. 3 Objective function and relative error histories of 4-d rosen function



(a) Objective function histories



(b) Relative errors

Fig. 4 Objective function and relative error histories of 4-d schwe function

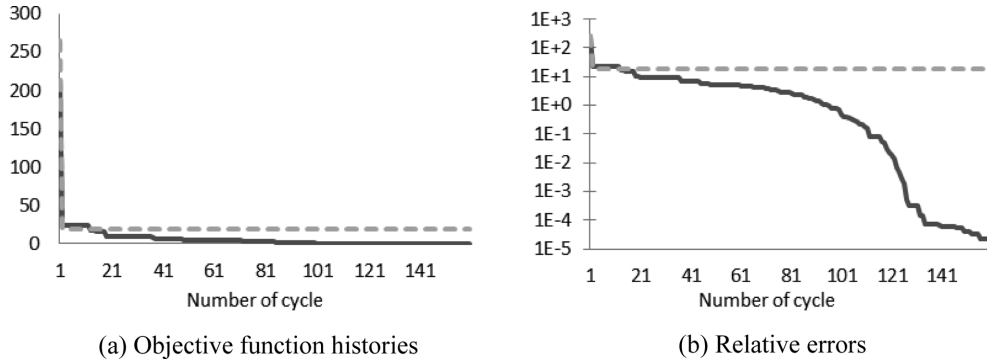


Fig. 5 Objective function and relative error histories of 8-d rosen function

최적화 방법이 직접 최적화하는 것보다 얼마나 효율적인지를 확인하기 위하여, rosen함수를 직접 최적화 하였는데, 수렴 조건을 10^{-6} 으로 했을 때, PSO 기법으로는 함수 계산 횟수가 25500번, SCE로는 11332번 정도 필요하였다. MSMO는 대략 1000번 정도에서 같은 결과를 주었다. 여기서 PSO를 적용할 때, 스왑 개수를 30개로 정하였다. schwe함수도 200 사이클 이내에 MSEGO가 해를 찾지 못하여 결과를 여기에 수록하지는 않았다.

5. 결론 및 향후 연구 방향

본 연구는 전역 최적화 문제를 해결하기 위한 근사최적화 방법을 제안하였다. 기존 연구에서와 다른 독자적인 접근을 정리해 보면 다음과 같은데, (1) 다중 근사모델의 다중 최적화를 동시에 진행하는 순차 샘플링 제안, (2) EI나 PI 등의 예측 계산을 배제하고 경험적 접근법 수행, (3) OLHD 류의 실험 계획 대신 QRS를 사용한 점이다.

기존의 단일 근사모델을 이용하는 방법이나 다중 근사모델을 가중 평균하는 방법들보다 우수함을 보였던 MSEGO 알고리즘과 본 연구에서 제안한 다중 근사모델과 다중 최적화를 이용한 근사 최적화 방법을 비교하였다. 다양한 함수들에 대하여 두 방법을 비교 분석하였는데, 제안한 방법이 우수함을 확인하였다. 특히 설계 변수가 두 개일 때는 큰 차이를 보이지는 않았지만, 설계 변수의 개수가 네 개, 여덟 개가 되면서 두 방법의 차이가 극명하게 나타났는데, MSMO가 우수한 성능을 보였다.

본 연구에서 제안한 프레임워크는 다양한 형태로 발전시킬 수 있다. 우선 보다 다양한 근사모델을

도입할 수 있는데, 사용할 근사모델의 최적의 숫자와 최적의 조합을 찾는 연구가 필요하다. 모델의 수가 많으면 많을 수록 좋겠지만 근사모델의 최적화 작업도 적지 않은 계산 시간을 필요로 하게 된다. 저자는 서로 종류가 다른 네 가지 근사모델 방법을 사용하였는데 예제 문제를 해결하는 데는 어려움이 없었다. 한편 다중 모델의 다중 최적화는 병렬로 처리하기가 용이하므로 계산 시간을 단축시킬 수 있다.

향후 연구 주제는 제안한 방법을 제한 조건이 있는 최적설계 문제에 적용하는 것이다. 아직 제한 조건이 있는 전역 최적화 문제를 효과적으로 해결할 수 있는 다양한 툴 박스를 찾지 못하였으나, 현재 활발하게 연구가 진행중인 분야이므로 조만간 가능할 것으로 생각한다. 또한 본 연구는 제안한 방법의 성능에만 국한하여 분석하였는데 최적 설계를 하는 엔지니어 입장에서 보면 원하는 정밀도 까지 해를 찾기를 원하므로 근사 최적화의 수렴 성이나 중단 조건 등에 대한 연구도 이어질 것이다.

감사의 글

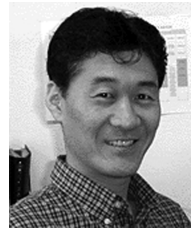
이 논문은 2011년도 서울시립대학교 교내학술 연구비에 의하여 연구되었음.

참고문헌

1. Simpson, T., Poplinski, J. and Koch, P., 2001, Metamodels for Computer-based Engineering Design: Survey and Recommendations, *Engineering with Computers*, 17, pp. 129-150.
2. Queipo, N. et al., 2005, Surrogate-based Analysis

- and Optimization, *Progress in Aerospace Sciences*, 41, pp. 1-28.
3. Forrester, A. and Keane, A., 2009, Recent Advances in Surrogate-based Optimization, *Progress in Aerospace Sciences*, 45(1), pp. 50-79.
 4. Jin, Y., 2011, Surrogate-assisted Evolutionary Computation: Recent Advances and Future Challenges, *Swarm and Evolutionary Computation*, 1, pp. 61-70.
 5. Jin, R., Chen, W. and Simpson, T., 2001, Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria, *Structural and Multidisciplinary Optimization*, 23, pp. 1-13.
 6. Kleijnen, J.P.C., et al., 2005, State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments, *INFORMS Journal on Computing*, 17(3), pp. 263-289.
 7. Wang, G.G. and Shan, S., 2007, Review of Meta-modeling Techniques in Support of Engineering Design Optimization, *Journal of Mechanical Design*, 129, pp. 370-380.
 8. Reishenthal, P.H. and Lesieutre, D.J., 2010, Statistical Benchmarking of Surrogate-Based and Other Optimization Methods Constrained by Fixed Computational Budget, 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 12-15, Orlando, Florida, pp. 1-16.
 9. Bettebghor, D., Bartoli, N., Grihon, S., Morlier, J. and Samuelides, M., 2010, Surrogate Modeling Approximation Using a Mixture of Experts Based on EM Joint Estimation, *Structural and Multidisciplinary Optimization*, 43(2), pp. 243-259.
 10. Li, Y.F., Ng, S.H., Xie, M. and Goh, T.N., 2010, A Systematic Comparison of Metamodeling Techniques for Simulation Optimization in Decision Support Systems, *Applied Soft Computing*, 10(4), pp. 1256-1272.
 11. Barton, R.R. and Meckesheimer, M., 2006, Meta-model-Based Simulation Optimization, *Handbooks in Operations Research and Management Science*, 13(18), pp. 535-574.
 12. Wojciechowski, S., 2010, A Method for Simulation Based Optimization Using Radial Basis Functions, *Optim Eng.*, 11, pp. 501-532.
 13. Kazemi, M., Wang, G., Rahnamayan, S. and Gupta, K., 2011, Metamodel-Based Optimization for Problems with Expensive Objective and Constraint Functions, *Journal of Mechanical Design*, 133, pp. 014505-1-7.
 14. Goel, T., Haftka, R.T., Shyy, W. and Queipo, N.V., 2006, Ensemble of Surrogates, *Structural and Multidisciplinary Optimization*, 33(3), pp. 199-216.
 15. Sanchez, E., Pintos, S. and Queipo, N.V., 2007, Toward an Optimal Ensemble of Kernel-based Approximations with Engineering Applications, *Structural and Multidisciplinary Optimization*, 36(3), pp. 247-261.
 16. Acar, E. and Rais-Rohani, M., 2008, Ensemble of Metamodels with Optimized Weight Factors, *Structural and Multidisciplinary Optimization*, 37(3), pp. 279-294.
 17. Zhou, X.J., Ma, Y.Z. and Li, X.F., 2011, Ensemble of Surrogates with Recursive Arithmetic Average, *Structural and Multidisciplinary Optimization*, 44(5), pp. 651-671.
 18. Gu, J., Li, G.Y. and Dong, Z., 2011, Hybrid and Adaptive Meta-model-based Global Optimization, *Engineering Optimization*, pp. 1-18.
 19. Müller J. and Piché, R., 2010, Mixture Surrogate Models Based on Dempster-Shafer Theory for Global Optimization Problems, *Journal of Global Optimization*, 51(1), pp. 79-104.
 20. Viana, F.A.C., Haftka, R.T. and Steffen, V., 2009, Multiple Surrogates: How Cross-validation Errors Can Help Us to Obtain the Best Predictor, *Structural and Multidisciplinary Optimization*, 39(4), pp. 439-457.
 21. Viana, F.A.C., Haftka, R.T. and Watson, L.T., 2012, Efficient Global Optimization Algorithm Assisted by Multiple Surrogate Techniques, *Journal of Global Optimization*, DOI 10.1007/s10898-012-9892-5.
 22. McKay, M.D., 1992, Latin Hypercube Sampling as a Tool in Uncertainty Analysis of Computer Models, *Proceedings of the 1992 Winter Simulation Conference*, pp. 557-564.
 23. Johnson, M.E., Moore, L.M. and Ylvisaker, D., 1990, Minimax and Maximin Distance Designs, *Journal of Statistical Planning and Inference*, 26, pp. 131-148.
 24. Leary, S., Bhaskar, A. and Keane, A., 2003, Optimal Orthogonal-array-based Latin Hypercubes, *Journal of Applied Statistics*, 30(5), pp. 585-598.
 25. Cioppa, T.M. and Lucas, T.W., 2007, Efficient Nearly Orthogonal and Space-filling Latin Hypercubes, *Technometrics*, 49, pp. 45-55.
 26. Deutsch, J.L. and Deutsch, C.V., 2012, Latin Hypercube Sampling with Multidimensional Uniformity, *Journal of Statistical Planning and Inference*, 142(3), pp. 763-772.
 27. Zhu, H., Liu, L., Long, T. and Peng, L., 2011, A Novel Algorithm of Maximin Latin Hypercube Design Using Successive Local Enumeration, *Engineering Optimization*, pp. 1-14.
 28. Crombecq, K., Laermans, E. and Dhaene, T., 2011, Efficient Space-filling and Non-collapsing

- Sequential Design Strategies for Simulation-based Modeling, *European Journal of Operational Research*, 214(3), pp. 683-696.
29. Crombecq, K., 2010, Generating Sequential Space-filling Designs Using Genetic Algorithms and Monte Carlo Methods, SEAL'10 Proceedings of the 8th International Conference on Simulated Evolution and Learning, pp. 80-84.
 30. Singhee, A. and Rutenbar, R.A., 2010, Why Quasi-Monte Carlo is Better Than Monte Carlo or Latin Hypercube Sampling for Statistical Circuit Analysis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11), pp. 1763-1776.
 31. Azoulay, A., 2011, Quasirandom Optimization, MS Thesis, The Florida State University, Dept. of Scientific Computing.
 32. Luo, C., Zhang, S.-L., Wang, C. and Jiang, Z., 2011, A Metamodel-assisted Evolutionary Algorithm for Expensive Optimization, *Journal of Computational and Applied Mathematics*, pp. 1-6.
 33. Yahyaie, F. and Filizadeh, S., 2011, A Surrogate-model Based Multi-modal Optimization Algorithm, *Engineering Optimization*, 43(7), pp. 779-799.
 34. Kuo, R.J. and Yang, C.Y., 2011, Simulation Optimization Using Particle Swarm Optimization Algorithm with Application to Assembly Line Design, *Applied Soft Computing*, 11(1), pp. 605-613.
 35. Liu, H. and Maghsoodloo, S., 2011, Simulation Optimization Based on Taylor Kriging and Evolutionary Algorithm, *Applied Soft Computing Journal*, 11(4), pp. 3451-3462.
 36. Liu, H., 2009, Taylor Kriging Metamodeling For Simulation Interpolation, Sensitivity Analysis And Optimization, Ph.D. Thesis, Auburn University.
 37. Jones, D.R., Schonlau, M. and Welch, W.J., 1998, Efficient Global Optimization of Expensive Black-Box Functions, *Journal of Global Optimization*, 13, pp. 455-492.
 38. Rafael, G.D.R. and Salinas, C.J.S., 2011, Empirical Study of Surrogate Models for Black Box Optimizations Obtained Using Symbolic Regression via Genetic Programming, GECCO'11, July 12-16, Dublin, Ireland. ACM 978-1-4503-0690-4/11/07.
 39. Yang, X.-S., 2010, Test Problems in Optimization, Engineering Optimization: An Introduction with Metaheuristic Applications (Eds Xin-She Yang), John Wiley & Sons.
 40. Viana, F.A.C., 2011, SURROGATES Toolbox User's Guide, Version 3.0, available at <http://sites.google.com/site/felipeacviana/surrogatestoolbox>.
 41. Donckels, Brecht, M.R., 2009, Optimal Experimental Design to Discriminate Among Rival Dynamic Mathematical Models, PhD Thesis, Dept. of Applied Mathematics, Biometrics and Process Control, Ghent University, Ghent, Belgium, MATLAB toolbox is available at <http://biomath.ugent.be/~brecht/downloads.html>.
 42. Berry, M.W. and Minser, K.S., 1999, Algorithm 798: High-dimensional Interpolation Using the Modified Shepard Method, *ACM Transactions on Mathematical Software*, 25(3), pp. 353-366.



이 세 정

1976년~1980년 서울대학교 기계공학
 1980년~1982년 한국과학기술원 기계공학 석사
 1982년~1985년 현대건설
 1986년~1989년 Pennsylvania State university 기계공학 박사
 1990년 삼성중공업
 1991년~1992년 한국생산성본부
 2000년~2001년 Georgia Institute of Technology, Aerospace Engineering, 교환교수
 2009년~2010년 Emory University, Biomedical Engineering, 교환교수
 1993년~현재 서울시립대학교 기계정보공학과 교수