

Received April 6, 2020, accepted April 17, 2020, date of publication April 27, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990455

Efficient Design Optimization of High-Performance MEMS Based on a Surrogate-Assisted Self-Adaptive Differential Evolution

MOBAYODE O. AKINSOLU¹, (Member, IEEE), BO LIU², (Senior Member, IEEE),
PAVLOS I. LAZARIDIS³, (Senior Member, IEEE),
KEYUR K. MISTRY³, (Graduate Student Member, IEEE),
MARIA EVELINA MOGNASCHI⁴, (Member, IEEE), PAOLO DI BARBA⁴, (Member, IEEE),
AND ZAHARIAS D. ZAHARIS⁵, (Senior Member, IEEE)

¹Faculty of Arts, Science and Technology, Wrexham Glyndŵr University, Wrexham LL11 2AW, U.K.

²James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.

³School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, U.K.

⁴Department of Electrical, Computer and Biomedical Engineering, The University of Pavia, 27100 Pavia, Italy

⁵Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

Corresponding author: Bo Liu (bo.liu@glasgow.ac.uk)

This work was supported by the Faculty of Arts, Science and Technology, Wrexham Glyndŵr University, U.K.

ABSTRACT High-performance microelectromechanical systems (MEMS) are playing a critical role in modern engineering systems. Due to computationally expensive numerical analysis and stringent design specifications nowadays, both the optimization efficiency and quality of design solutions become challenges for available MEMS shape optimization methods. In this paper, a new method, called self-adaptive surrogate model-assisted differential evolution for MEMS optimization (ASDEMO), is presented to address these challenges. The main innovation of ASDEMO is a hybrid differential evolution mutation strategy combination and its self-adaptive adoption mechanism, which are proposed for online surrogate model-assisted MEMS optimization. The performance of ASDEMO is demonstrated by a high-performance electro-thermo-elastic micro-actuator, a high-performance corrugated membrane micro-actuator, and a highly multimodal mathematical benchmark problem. Comparisons with state-of-the-art methods verify the advantages of ASDEMO in terms of efficiency and optimization ability.

INDEX TERMS MEMS design optimization, high-performance MEMS design, surrogate model assisted evolutionary algorithm, Gaussian process, differential evolution.

I. INTRODUCTION

Microelectromechanical systems (MEMS) with high performance requirements are widely used in modern engineering systems [1]. For example, the performance requirements of contemporary micro-actuators/micro-sensors are becoming increasingly stringent for biomedical devices and wearable robotics in recent years [2], [3]. Despite that the overall performance is also affected by supporting electronics and software, shape optimization of the MEMS is often essential.

The associate editor coordinating the review of this manuscript and approving it for publication was Sunith Bandaru¹.

Present MEMS shape optimization methods can be divided into two categories, which are: (1) local optimization informed by design expertise and (2) evolutionary algorithms (EAs), which aim to perform global optimization. In the first category, design expertise is firstly used to simplify the optimization problem by providing an initial design, narrowing down the search ranges and reducing the number of design variables based on sensitivity, [4]–[6]. This kind of method is playing an important role in modern high-performance MEMS optimization. However, the main drawbacks include a lack of generality and limited optimization ability. The design expertise is often case-specific, leading to an ad-hoc optimization process. The local optimization methods have limited search ability, and may not be able to meet the

stringent design specifications when the initial design is not sufficiently good [7].

Compared with the first category, which is more fitted for design experts, the second category, EA-driven MEMS design optimization [8]–[11], has the advantage of being free of an initial design, generality, and robustness. Particularly, EAs often have reasonably high global optimization capability for nonlinear and multimodal problems, which are suitable for high-performance MEMS optimization. At present, besides the traditional genetic algorithms, differential evolution (DE) [12] and particle swarm optimization (PSO) [13] are arguably the most widely used EAs for MEMS shape optimization [4], [8], [14]. DE and PSO show good performance for various high-performance MEMS, but it is also found that their optimization abilities still need improvement when handling complex design cases. In the computational intelligence field, various improved EAs have been proposed. Only considering DE, there are a few popular improved versions, e.g., [15]–[17]. Among them, a state-of-the-art method is Linear Population Size Reduction - Success-History Based Parameter Adaptation for Differential Evolution (L-SHADE). However, our experiments (Section IV) show that even L-SHADE may not be able to achieve the required optimization quality for the targeted problem.

Another important challenge for EA-driven MEMS shape optimization is efficiency. Computationally expensive numerical simulations (e.g., finite element analysis) are often unavoidable to obtain accurate performance estimation of MEMS devices [18]. In contrast with local optimization-based methods, which are often efficient when there exists a sufficiently good initial design, standard EAs, and their improved versions often need a large number of simulations to obtain the optimum. The optimization time, therefore, becomes very long or even prohibitive in some cases [10], while nowadays most real-world systems including MEMS often require a reduced time-to-market [19], [20]. Thus, substantial optimization efficiency improvement without compromising performance is highly desirable.

To address this problem, off-line surrogate model-based EAs (SAEAs) are introduced into MEMS shape optimization [21]. An initial random sampling (such as using Design of Experiment [22] methods) is firstly carried out. Using the sampled design variables as the input and the performance via numerical simulations as the output, a black-box surrogate model is constructed to approximate the performance of the MEMS device. The surrogate model is often constructed by statistical learning techniques. For example, Lee *et al.* [21] use artificial neural networks (ANN) to construct a black-box surrogate model for predicting the performance of new candidate designs. In the optimization process, the computationally cheap surrogate model is used to replace the expensive numerical simulation model to improve computational efficiency. Although efficient for various MEMS, high-performance MEMS requires a very accurate surrogate model to meet the stringent constraints and obtain a highly optimized objective function value. When the number

of design variables is more than a few, obtaining a sufficiently accurate surrogate model may need a tremendous number of initial samples, canceling out the saved simulation time [23].

Because the optimal region is unknown beforehand, one-shot sampling wastes a lot of samples (i.e., computationally expensive simulations) in regions far away from the optimal region. Infill sampling techniques have been introduced to address this drawback for simulation-driven design optimization, which show successful results for complex problems [24]. Rather than using a one-shot initial sampling to obtain the surrogate model, the surrogate model is updated iteratively in the optimization process. By sampling the “interested” regions based on analysis of previous samples, computationally expensive simulations are saved. When active learning or infill sampling techniques are combined with EAs, on-line SAEAs are constructed. In online SAEA, a very rough surrogate model is firstly built, and simulations are used in each iteration to explore the design space and improve the surrogate model.

Even though efficient state-of-the-art online SAEAs have been proposed [25]–[27], to the best of our knowledge, these methods are arguably not purpose-built shape optimization methods for high-performance MEMS which need to meet highly stringent specifications. Surrogate model-aware evolutionary search (SMAS) is a typical framework introduced by the authors, which premises on surrogate-assisted metaheuristics using infill sampling techniques and it outperforms several popular online SAEAs [32]. A MEMS design optimization method based on SMAS is also proposed [23]. However, our experiments (Section IV) show that although with high efficiency, the optimization ability is insufficient for the targeted problem. Many recent successful SAEAs mainly focus on handling the challenges in higher-dimensional problems [27] or a very limited computing resources [25], which are also not suitable for the targeted problem.

A new SAEA is proposed in this paper, called self-adaptive surrogate model-assisted differential evolution for MEMS optimization (ASDEMO), which aims to:

- Substantially improve the optimization efficiency of standard EAs and some modern variants for high-performance MEMS shape optimization;
- Improve the optimization ability of standard EAs and some modern variants for high-performance MEMS shape optimization;
- Be general enough to handle MEMS shape optimization with about 10 design variables or more without any initial solutions or ad-hoc process.

The remainder of the paper is organized as follows. Section II introduces the basic techniques. Section III introduces the ASDEMO algorithm, including the main innovations and the parameter settings. Section IV tests ASDEMO using two practical high-performance MEMS and a highly multimodal mathematical benchmark problem. The concluding remarks are presented in Section V.

II. BASIC TECHNIQUES

A. GAUSSIAN PROCESS MACHINE LEARNING AND PRESCREENING

The selected surrogate modeling method in ASDEMO is Gaussian process (GP). GP modeling is a theoretically sound and principled method for determining a much smaller number of free model parameters when compared to many other surrogate modeling approaches such as ANN [28], [29]. It can also provide an estimate of the model uncertainty for each predicted point, which is shown to have a large advantage in SAEAs [30]–[33]. GP has been widely used for surrogate modeling in the electromagnetic machine design optimization field [34], including MEMS. GP works as follows:

To model an unknown function $y = f(x), x \in R^d$, GP modeling assumes that $f(x)$ at any point x is a Gaussian random variable $N(\mu, \sigma^2)$, where μ and σ are two constants independent of x . For any $x, f(x)$ is a sample of $\mu + \epsilon(x)$, where $\epsilon(x) \sim N(0, \sigma^2)$. For any $x, x' \in R^d, c(x, x')$, the correlation between $\epsilon(x)$ and $\epsilon(x')$, depends on $x - x'$. More precisely,

$$c(x, x') = \exp\left(-\sum_{i=1}^d \theta_i |x_i - x'_i|^{p_i}\right), \tag{1}$$

where parameter $1 \leq p_i \leq 2$ is related to the smoothness of $f(x)$ with respect to x_i , and parameter $\theta_i > 0$ indicates the importance of x_i on $f(x)$. More details about GP modeling can be found in [35].

Given K points $x^1, \dots, x^K \in R^d$ and their f -function values y^1, \dots, y^K , then the hyper parameters $\mu, \sigma, \theta_1, \dots, \theta_d$, and p_1, \dots, p_d can be estimated by maximizing the likelihood that $f(x) = y^i$ at $x = x^i (i = 1, \dots, K)$ [36]:

$$\frac{1}{(2\pi\sigma^2)^{K/2}\sqrt{\det(C)}} \exp\left[-\frac{(y - \mu\mathbf{1})^T C^{-1}(y - \mu\mathbf{1})}{2\sigma^2}\right] \tag{2}$$

where C is a $K \times K$ matrix whose (i, j) -element is $c(x^i, x^j)$, $y = (y^1, \dots, y^K)^T$ and $\mathbf{1}$ is a K -dimensional column vector of ones.

To maximize (2), the values of μ and σ^2 must be:

$$\hat{\mu} = \frac{\mathbf{1}^T C^{-1} y}{\mathbf{1}^T C^{-1} \mathbf{1}} \tag{3}$$

and

$$\hat{\sigma}^2 = \frac{(y - \mathbf{1}\hat{\mu})^T C^{-1} (y - \mathbf{1}\hat{\mu})}{K}. \tag{4}$$

Substituting (3) and (4) into (2) eliminates the unknown parameters μ and σ from (2). As a result, the likelihood function depends only on θ_i and p_i for $i = 1, \dots, d$. (2) can then be maximized to obtain estimates of $\hat{\theta}_i$ and \hat{p}_i . In this work, the maximization of (2) is carried out by sequential quadratic programming [37]. Although the computational complexity grows cubically with the number of training data points, the computing overhead of GP modeling is often low. The reason is that most MEMS shape optimization problems have around or less than 10 design variables. A large number of training data points is therefore not needed to build a reliable surrogate

model. With optimized $\hat{\theta}_i$ and \hat{p}_i , the estimates $\hat{\mu}$ and $\hat{\sigma}^2$ can then be readily obtained from (3) and (4).

Given the hyper parameter estimates $\hat{\theta}_i, \hat{p}_i, \hat{\mu}$ and $\hat{\sigma}^2$, one can predict $y = f(x)$ at any untested point x based on the f -function values y^i at x^i for $i = 1, \dots, K$. The best linear unbiased predictor of $f(x)$ is [36], [38]:

$$\hat{f}(x) = \hat{\mu} + r^T C^{-1} (y - \mathbf{1}\hat{\mu}) \tag{5}$$

and its mean squared error is:

$$s^2(x) = \hat{\sigma}^2 \left[1 - r^T C^{-1} r + \frac{(1 - \mathbf{1}^T C^{-1} r)^2}{\mathbf{1}^T C^{-1} r}\right] \tag{6}$$

where $r = (c(x, x^1), \dots, c(x, x^K))^T$. $N(\hat{f}(x), s^2(x))$ can be regarded as a predictive distribution for $f(x)$ given the function values y^i at x^i for $i = 1, \dots, K$.

Considering the predicted value $\hat{f}(x)$ and the prediction uncertainty $s^2(x)$, several prescreening methods are proposed to predict the quality of a candidate design [31]. Lower confidence bound (LCB) is used in this paper and we consider the minimization of $f(x)$. Given the predictive distribution $N(\hat{f}(x), s^2(x))$ for $f(x)$, a LCB prescreening of $f(x)$ can be defined as [30]:

$$f_{lcb}(x) = \hat{f}(x) - \omega s(x) \tag{7}$$

where $\omega \in [0, 3]$ is a constant. The use of LCB prescreening can conduct explorative global search when using a large ω or conduct fast local search when using a small ω . Since $\hat{f}(x)$ is Gaussian distributed, according to the 3σ rule, when $\omega = 2$, the confidence level of $f_{lcb}(x)$ to be the LCB of $\hat{f}(x)$ is about 97%. The comparisons of using different ω to define LCB are detailed in [30], [31]. $\omega = 2$ is used in this paper as suggested in [31].

B. DIFFERENTIAL EVOLUTION

DE algorithm is used as the search engine in the proposed ASDEMO algorithm. DE is an effective and popular global optimization algorithm. It uses a differential operator to create new candidate solutions [12].

Suppose that P is a population. Let $x = (x_1, \dots, x_d) \in R^d$ be an individual solution in P . To generate a child solution $u = (u_1, \dots, u_d)$ for x , a donor vector is first produced by mutation:

$$v = x^{r1} + F \cdot (x^{r2} - x^{r3}) \tag{8}$$

where x^{r1}, x^{r2} and x^{r3} are three different solutions randomly selected from P . $F \in (0, 2]$ is a control parameter, often called the scaling factor [12]. The above mutation strategy is called DE/rand/1, which is one of the most widely used strategies. Besides, there are quite a few different mutation strategies, trading off exploration ability and convergence speed.

Then the following crossover operator is applied to produce the child u :

- 1 Randomly select a variable index $j_{rand} \in \{1, \dots, d\}$,

- 2 For each $j = 1$ to d , generate a uniformly distributed random number $rand$ from $(0, 1)$ and set:

$$u_j = \begin{cases} v_j, & \text{if } (rand \leq CR) | j = j_{rand} \\ x_j, & \text{otherwise} \end{cases} \quad (9)$$

where $CR \in [0, 1]$ is a constant called the crossover rate.

In ASDEMO, three other DE mutation strategies are also adopted. The first one is DE/rand-to-best/2 [39], which is shown in (10).

$$v^i = x^i + F \cdot (x^{best} - x^i) + F \cdot (x^{r1} - x^{r2}) + F \cdot (x^{r3} - x^{r4}) \quad (10)$$

where x^i is the i^{th} candidate solution in P and x^{best} is the best candidate solution in P . x^{r1} , x^{r2} , x^{r3} and x^{r4} are mutually exclusive candidate solutions randomly selected from the current population and are different from x^i and x^{best} .

Another strategy is the DE/rand/2/dir mutation strategy [38], which is shown in (11).

$$v^i = x^{r1} + \frac{F}{2} \cdot (x^{r1} - x^{r2} - x^{r3}) \quad (11)$$

where x^{r1} , x^{r2} and x^{r3} are three different solutions selected from P and x^{r1} has the smallest objective function value (considering a minimization problem).

Another mutation strategy used in ASDEMO is DE/hybrid trigonometric mutation [40]: in 95% rate, the DE/rand/1 strategy is used; while in 5% rate, (12) is used.

$$\begin{aligned} p_s &= |f(x^{r1})| + |f(x^{r2})| + |f(x^{r3})| \\ p_1 &= |f(x^{r1})|/p_s \\ p_2 &= |f(x^{r2})|/p_s \\ p_3 &= |f(x^{r3})|/p_s \\ v^i &= (x^{r1} + x^{r2} + x^{r3})/3 + (p_2 - p_1) \cdot (x^{r1} - x^{r2}) \\ &\quad + (p_3 - p_2) \cdot (x^{r2} - x^{r3}) + (p_1 - p_3) \cdot (x^{r3} - x^{r1}) \end{aligned} \quad (12)$$

III. THE ASDEMO ALGORITHM

A. CHALLENGES AND MAIN IDEAS OF ASDEMO

ASDEMO is an on-line SAEA, for which, the surrogate model keeps improving in the optimization process. Hence, the predicted good designs may be wrong especially at the beginning of the optimization when there are insufficient training data points, which may cause wrong convergence. To address this problem, some available SAEAs begin with a standard EA for a certain number of iterations (expensive simulations are used for the whole population). After a surrogate model with sufficient quality is obtained, the solution quality and the surrogate model quality are then iteratively improved in the consecutive search [41], [42]. This kind of method often has reasonably good solution quality, but the efficiency is sacrificed. In contrast, some other available SAEAs perform expensive simulations to the “optimal” solutions predicted by the existing surrogate

model, despite that its quality may not be good enough [31]. The number of expensive simulations is therefore highly reduced, but the solution quality remains a weakness. Hence, neither of them is suitable for high-performance MEMS shape optimization, requiring both high solution quality and high efficiency. As said in Section I, SMAS framework [32] is a new kind of SAEA and advantages are shown compared to the above two kinds of SAEAs [32]. However, it is found that when the expected performance of the MEMS device is high, both the solution quality and efficiency need improvement.

It is worth understanding the landscape characteristics of the targeted problem. Compared to normal MEMS shape optimization, high-performance MEMS optimization introduces stringent constraints and higher expectations of the optimal objective function value. To cope with that, both (very) high exploration and exploitation abilities are essential for the optimization algorithm. The reasons include: (1) Stringent constraints and highly optimal objective function value make the optimal region become (very) narrow and the optimization algorithm must be able to search elaborately in that narrow region (i.e., a high exploitation ability). (2) To get access to the optimal region, high exploration ability must be available to jump out of the local optima in the design landscape. More particularly, when several constraints are imposed on, their extent of difficulty to be satisfied varies and the population may be dominated by relatively easier ones among them in an early stage and the diversity may be lost to satisfy the more stringent constraints (e.g., Example 1 in Section IV). Hence, a high exploration ability is essential to maintain population diversity.

In addition, there is a critical balance between the exploration ability, the convergence speed [43] and the surrogate model quality. The higher the exploration ability (i.e., population diversity), the lower the probability of finding the correct search direction (i.e., convergence speed), and more necessary training data points are required to construct a reliable surrogate model (i.e., computationally expensive simulations). Hence, the central questions become: (1) How to build an SAEA framework (Section III B), and (2) How to design the optimization kernel (Section III C) to call for an appropriate balance among these factors?

B. THE NEW SAEA FRAMEWORK

The flow diagram of ASDEMO is shown in Figure 1. The algorithm consists of the following steps.

- Step 1:** Sample α (often a small number) designs from the design space using the Latin Hypercube sampling method [22], perform numerical simulations to all of them and let them form the initial database.
- Step 2:** If a preset stopping criterion (e.g., a maximum number of simulations) is met, output the best design from the database; otherwise go to Step 3.
- Step 3:** Select the λ best designs from the database to form a population P .

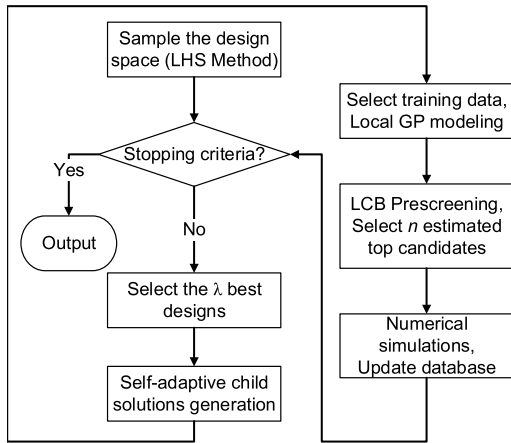


FIGURE 1. The flow diagram of ASDEMO.

- Step 4:** Apply the proposed self-adaptive DE search (Section III C) on P to generate n child populations and each population has λ child solutions.
- Step 5:** For each candidate design in each population, take the τ nearest designs (based on Euclidean distance) from the database and their performance values as the training data points to construct a local GP surrogate model. (There are $n \times \lambda$ GP models in total.)
- Step 6:** Prescreen the $n \times \lambda$ child solutions generated in Step 4 using the GP models in Step 5 and the lower confidence bound method (7). Select the top n child solutions based on the lower confidence bound values.
- Step 7:** Simulate the estimated top n child solutions from Step 6. Add them and their performance (via numerical simulation) to the database. Go back to Step 2.

It can be seen that some ideas are borrowed from our SMAS framework [23], [32], [33]. The standard EA process is not used. Instead, only the predicted top candidate designs are simulated and the current best λ candidate designs are used as the new population in each iteration. The goal is to improve the locations of the training data points. Often, the number of training data points is considered as the main factor affecting the quality of a surrogate model, but their locations are overlooked. Intuitively, using the same number of training data points located near to the points waiting to be predicted (child populations in Step 4) can obtain a better surrogate model than using those far away from them. Because the training data points are selected from candidate designs generated by the search operators, the search should generate candidate designs in expected locations (i.e., near to the child population).

To meet this requirement, in each iteration, the λ current best candidate designs construct the parent population (it is reasonable to assume that the search focuses on the promising region) and the top n candidate designs based on prescreening in the child population are selected to replace the few worst

ones in the parent population. Hence, only a few candidates at most are changed in the parent population in each iteration, so the top quality candidates in the child solutions among several consecutive iterations may be quite near to each other (they will then be simulated and are used as training data points). Therefore, the training data points describing the current promising region are much denser compared to those generated by a standard EA population updating mechanism, which may spread in different regions (many of them are far from optimal) of the design space, while there may not be sufficient training data points around the candidate solutions to be prescreened.

New candidate design generation and prediction are two other critical issues of the SAEA framework. Generating new candidate designs through a self-adaptive DE mutation method is described in the next subsection. However, correct predictions of the new candidate designs become a new challenge. SMAS builds a single surrogate model using the solutions near the child population to be predicted, while the new candidate designs for the targeted problem are expected to have diversity for exploration or serving as a local improvement for exploitation. A single global surrogate model is therefore inaccurate for the prediction. In ASDEMO, a local GP model is built for each child solution waiting to be predicted using the method in Step 5. Experiments show a sufficient prediction capability.

C. THE OPTIMIZATION KERNEL

To achieve the goals described in Section III A, the optimization kernel is designed as follows:

For each child population $i = 1, 2, \dots, n$:

- Step 1:** If the algorithm is within the learning period (the current number of iterations is smaller than a threshold L), the rate of using DE/rand-to-best/2, (10), DE/rand/1/dir, (11) and DE/hybrid trigonometric mutation strategies, (12) is $\frac{1}{3}$. Otherwise; use the rates in Step 5.
- Step 2:** Perform a roulette wheel selection [44] based on the rates to determine a DE mutation strategy and generate a child population C_i (λ child solutions).
- Step 3:** Use the local GP surrogate models in Step 5 of the SAEA framework (Section III B) to predict all the candidate designs in C_i .
- Step 4:** Compare the predicted value of each solution in C_i and the current best design (simulated value). Add the number of solutions that are better than the current best design to N_s (the number of successes of (10), (11) or (12) and add λ to N_u (the number of uses of (10), (11) or (12)). Until all the n groups of child solutions are generated.
- Step 5:** Update the rates of using DE/rand-to-best/2, (10), DE/rand/1/dir, (11) and DE/hybrid trigonometric mutation, (12) by N_s/N_u . Update the number of iterations.

Selecting appropriate DE mutation strategies is critical for the success of the optimization kernel. Some considerations are stated as follows:

- (1) The selected mutation strategies must have high exploration ability. Besides the reasons regarding stringent constraints and high expectations of the optimal objective function value (Section III-A), the SAEA framework (Section III-B) is another reason. The search focuses on the promising subregion and only the top n candidates are simulated and are used to update the surrogate model, laying emphasis more on exploitation. Thus, the exploration ability mainly comes from the mutation strategy themselves.
- (2) Given the required exploration ability, the convergence speed should be as high as possible. The reference used is DE/rand/1 mutation, which is popular and successful. The goal of the ASDEMO optimization kernel is to improve the convergence speed and keep the exploration ability of the DE/rand/1 mutation strategy.
- (3) Different DE mutation strategies have a different trade-off between the exploration ability and the convergence speed. This trade-off is expected to be similar for the selected strategies; Otherwise, because the self-adaptive mutation strategy selection method is greedy (Step 5), the strategy with the fastest convergence speed may dominate the selection in an early stage, which is harmful to preserving the diversity in the whole search process. Considering multiple constraints with various extent of difficulty to be satisfied, the above consideration is even more important.
- (4) MEMS shape optimization problems can be separable or non-separable, and ASDEMO aims to be a general method, so both conditions should be considered.
- (5) Memetic algorithms, which include local search into global search, show largely improved exploitation ability than standard EAs [45]. Different from our previous works using different stages for local and global optimization [46], [47], local search will be embedded in DE mutation operators to help satisfy stringent constraints as well as locally improve the objective function value.

Based on the above considerations, three more efficient DE mutation strategies [38]–[40] compared to DE/rand/1 are employed. A method to improve the convergence speed is to use x^{best} to guide the search. DE mutation strategies employing x^{best} show strong competitiveness for not highly multimodal and separable problems [39]. To keep the exploration ability, two pairs of individuals are used. That is why DE/rand-to-best/2, (10) is selected.

Another convergence speed improvement method is to utilize the objective function value. A typical example is the DE/rand/dir category. One pair of individuals is selected because using two pairs of individuals results in too high diversity compared to the other two selected strategies. It is shown that DE/rand/1/dir (11) can maintain the diversity

of DE/rand/1, and shows good performance for multimodal and non-separable problems [38]. This is a complement to DE/rand-to-best/2, (10).

To promote local exploitation, we include local search in DE mutation at a low rate. A typical method is DE/trigonometric mutation, (12) [40]. Using (12), the perturbations are biased towards the candidate designs providing the smallest objective function values, to perform a local search. When hybridizing with DE/rand/1 (Section 2.2), the exploration ability is kept and the exploitation ability is enhanced.

Considering different MEMS design problem landscapes, the most fitted mutation strategy is used with the largest probability based on previous experience of the targeted problem. Our experiments show two observations: (1) For different MEMS optimization problems, the rate of using a certain mutation strategy can be very different, verifying the effectiveness of the self-adaptive method. (2) The optimal design obtained by the self-adaptive method is better than using each selected mutation strategy alone in all of our test cases, which shows the complement of the selected strategies.

D. PARAMETER SETTINGS

In ASDEMO, the algorithm parameters include (1) the DE parameters: λ , F and CR , (2) surrogate modeling parameters: the number of training data points (τ) for each child solution waiting to be prescreened, the initial sample size (α), and (3) self-adaptive search parameters: the threshold of the learning period (L) and the number of child populations (n).

ASDEMO is a SMAS-based algorithm. Because the population updating scheme in SMAS emphasizes exploitation, large F and CR should be used to maintain the diversity. Empirical studies are in [48]. We suggest $F = 0.8$, $CR = 0.8$, $\lambda \in [5 \times d, 10 \times d]$. For surrogate modeling parameters, we suggest $\alpha = 5 \times d$ and $\tau = 8 \times d$ when the size of the database is larger than $8 \times d$; otherwise, the whole database is used. This is based on the empirical rules in [32], [36] for on-line surrogate modeling. Clearly, L is not sensitive and we suggest it to be within [30, 50]. For simplicity, we use $L = 30$ in all test cases. For n , $n = 3$ is used according to empirical study.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

In this section, two real-world high-performance MEMS examples (a 4 variable electro-thermo-elastic micro-actuator and a 10 variable corrugated membrane micro-actuator) and a mathematical benchmark problem (the 15 – d Ackley function [49]) are used to demonstrate the ASDEMO method. The reason why the Ackley function is selected among various benchmark problems is that it has a narrow optimal region and a highly multimodal landscape, which has common characteristics with the targeted problem, but it is arguably more complex considering the number of local optima. The forward problems of both MEMS examples are based on finite element analysis. The reference methods are PSO [13], DE [12], SMAS [23], [32], IWO [51], [52], L-SHADE [50] as

well as ASDEMO only with a single mutation strategy (i.e., the strategies in (10), (11) and (12), respectively).

The reasons to select the comparison references are as follows: DE and PSO are selected because they are the dominant methods in the MEMS shape optimization area. IWO is selected because it is a relatively new method and it is attracting much attention in shape design optimization [51]–[53]. L-SHADE is selected because it is a state-of-the-art method in the evolutionary computation domain with applications in engineering design optimization [16], [17]. Although ASDEMO is particularly proposed for high-performance MEMS shape optimization, it is interesting to investigate its optimization ability compared with L-SHADE. Standard SMAS is selected because it is a typical surrogate model-assisted EA, and it has shown considerably higher performance compared to several popular surrogate model-assisted single objective optimization techniques [32]. ASDEMO with a single mutation strategy (i.e., the strategies in (10), (11) and (12), respectively) is considered to verify the performance improvement of the proposed hybrid mutation strategy and the self-adaptive adoption mechanism.

The population/swarm/plant size and the maximum number of evaluations are problem dependent and are described in each subsection. The same population size is used for all the methods for a fair comparison. For other algorithm parameters, $F = 0.8$ and $CR = 0.8$ are used for DE for a fair comparison. For PSO, MATLAB PSO optimizer is used with MATLAB’s default algorithm parameters: 1.49 for both cognitive and social parameters, adaptive inertia weight with the boundary of [0.1, 1.1] and a minimum adaptive neighborhood size of 0.25. Based on the recommendations in [51], a nonlinear modulation index of 3 and the number of seeds increasing linearly from 0 to 5 are used for IWO. For L-SHADE, the control parameters (i.e., scaling factor and crossover rate) are adaptive in each generation and historical memory with 5 memory cells and an archive rate of 1.4 are used following [50], [54]. The parameters for SMAS, ASDEMO with a single strategy are the same as those used in ASDEMO. Other parameter settings of ASDEMO are discussed in section III. All the tests are run on a workstation with Intel 4-core i7 (4770K) 3.50 GHz CPU and 24GB RAM and the time consumptions reported are wall clock time.

A. EXAMPLE 1

The first example is a silicon electro-thermo-elastic micro-actuator (Fig. 2) with a Young’s modulus of 210 GPa and a Poisson’s ratio of 0.22. The coupled equations in (13) to (15) are solved subject to the boundary conditions in (16) to (21).

$$J = \gamma E = -\gamma \nabla V; \quad \gamma = 5 \times 10^4 Sm^{-1} \quad (13)$$

$$Q = J \cdot E = -\nabla \cdot (k \nabla T); \quad k = 34 Wm^{-1}K^{-1} \quad (14)$$

$$\alpha_T(T - T_{ref}) = \epsilon^{Th_s}; \quad \alpha_T = 2.6 \times 10^{-6} \quad (15)$$

where J is the current density, γ is the electrical conductivity, E is the electric field, Q is the heat due to Joule-effect, T is the temperature, k is the thermal conductivity, T_{ref} is the

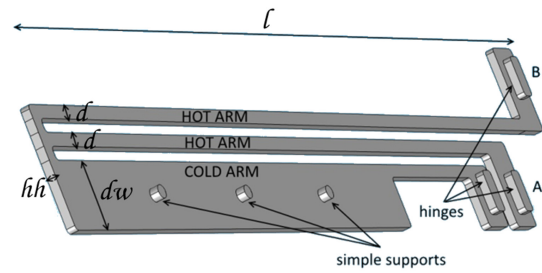


FIGURE 2. Layout of the electro-thermo-elastic micro-actuator.

reference temperature, h_s is the convection coefficient which is different in the upper and in the lower surface of the micro-actuator (Figure 2), α_T is the thermal expansion coefficient and ϵ^{Th_s} is the thermal part of the total strain.

The conduction current problem is:

$$V_A = const; \quad V_B = 0 \quad (16)$$

$$n_v \cdot J = 0 \quad (17)$$

elsewhere; where V_A and V_B are the terminal voltages at A and B respectively in (Figure 2) and n_v is the normal vector.

The thermal problem is:

$$T = T_{ext} = T_{ref} = 293.15K \quad (18)$$

at the simple supports and at the hinges.

$$-n_v \cdot (k \nabla T) = h_s(T_{ext} - T) \quad (19)$$

elsewhere; where T_{ext} is the external temperature.

The mechanical problem is:

$$U \cdot n_v = 0 \quad (20)$$

at the simple supports.

$$U = 0 \quad (21)$$

at the hinges; where U is the displacement.

The three problems are coupled via the thermal heat Q (electric and thermal problems) and via the temperature T (thermal and elastic problems). Because non-linearities of material parameters are not taken into account, it is possible to solve the three problems subsequently. Therefore, a weakly-coupled analysis problem is dealt with; to solve it, a cascade algorithm such as the successive substitution algorithm can be applied. The micro-actuator is modeled in COMSOL Multiphysics [55]. Its parametric finite-element model has a typical mesh composed of about 8,000 3-D elements and each simulation costs 25 minutes on average.

The design optimization problem is as follows. The design objective is to minimize the controllable area consumption of the micro-actuator (represented by $l \times (dw + 2d)$) subject to specifications on maximum temperature, stress and total displacement as shown in (22). It can be seen that the design specifications are very stringent, as the first and third specifications are narrow bound constraints despite that the

TABLE 1. Ranges of the design variables (all sizes in μm) (example 1).

Variables	l	hh	dw	d
Lower bound	56	2	7	1
Upper bound	300	5	30	7

constraint on stress is moderate. The ranges of the design variables are shown in Table 1. A geometric constraint defined by $dw \leq 2 \times d$ is used to ensure consistency in the model. This constraint is handled by repairing geometric infeasible candidate designs to the nearest feasible ones before simulation (i.e., setting $dw = 2 \times d$ when $dw > 2 \times d$).

Considering the number of design variables, the population/swarm/plant size is set to 30 for all the methods. The external penalty function method is used to handle constraints and the penalty factor for each constraint is 50. To make all the methods converge, the computing budget is as follows: 200 simulations for ASDEMO, SMAS, ASDEMO with a single mutation strategy (i.e., the strategies in (10), (11) and (12), respectively), 1500 simulations for DE, PSO, IWO, and L-SHADE. A total of 10 independent runs are carried out for surrogate model-assisted methods, and two independent runs are carried out for DE, PSO, IWO, and L-SHADE, respectively, because more runs are not affordable.

$$\begin{aligned}
 &\text{minimize } l \times (dw + 2d) \\
 &s.t. \ 697.5 K \leq ([T_{max}]_{V=V_{min}} + [T_{max}]_{V=V_{max}}) \leq 702.5 K \\
 &\quad S \leq 1.44 GPa \\
 &\quad 2.05 \mu m \leq ([U]_{V=V_{min}} + [U]_{V=V_{max}}) \leq 2.10 \mu m \quad (22)
 \end{aligned}$$

where $V_{min} = 1 V$, $V_{max} = 2 V$, T_{max} is the maximum temperature, S is the stress and U is the displacement.

The performances and convergence trends are shown in Table 2 and Figure 3, respectively. ASDEMO and DE satisfy all the design constraints in each run, while other methods fail to satisfy the design constraints in each run. In one PSO run, $T_{max} = 702.90 K$ and $U = 2.05 \mu m$ and the constraint on S is satisfied. In the other run, the constraints on T_{max} and S are satisfied, but $U = 2.03 \mu m$. Because the constraint violation for PSO is reasonably small, it is still included in the efficiency comparison.

TABLE 2. Statistics of the objective function values (in nm^2) for different methods (example 1).

Method	Best	Worst	Average	Std
ASDEMO	9.63	9.73	9.69	0.032
SMAS	N/A	N/A	N/A	N/A
ASDEMO (Only DE/rand-to-best/2)	N/A	N/A	N/A	N/A
ASDEMO (Only DE/rand/2/dir)	N/A	N/A	N/A	N/A
ASDEMO (Only DE/hybrid trig)	N/A	N/A	N/A	N/A
PSO	N/A	N/A	N/A	N/A
DE	9.69	9.74	9.71	0.037
IWO	N/A	N/A	N/A	N/A
L-SHADE	N/A	N/A	N/A	N/A

Note:
For methods that do not obtain a feasible design, not applicable (N/A) is used.

In both runs of IWO, the constraints on T_{max} and U are not satisfied with violations of around 165 K and 0.1 μm , respectively. In both runs of L-SHADE, the constraint on U is not satisfied with violations of around 0.04 μm . Note that in contrast to PSO where only one of the two runs violates the constraint on U with a violation of 0.04 μm , L-SHADE failed in both runs with worse results. In terms of our SMAS-based MEMS shape optimization method [23] (i.e., SMAS), and ASDEMO with a single mutation strategy, in the 10 independent runs, none of them satisfy the constraint on U with a violation of around 0.06 μm and they seldom satisfy the constraint on T_{max} with a typical violation of around 1 K. Based on the required performances in (22), it can be seen that the optimization abilities of SMAS, ASDEMO with a single mutation strategy, IWO and L-SHADE are insufficient for this high-performance MEMS shape optimization (i.e., stringent constraints). Hence, they are all excluded from the efficiency comparison.

As revealed in Figure 3, ASDEMO obtains the average objective value of 9.69 nm^2 using 199 simulations (about 4 days). A typical result of ASDEMO is 9.69 nm^2 for the objective function, $T_{max}=702.48 K$, $S=0.62 GPa$ and $U=2.05 \mu m$. The convergence criterion that we defined is that the design specifications are satisfied and the improvement of the objective function is less than 1 nm^2 after 30 simulations. Using

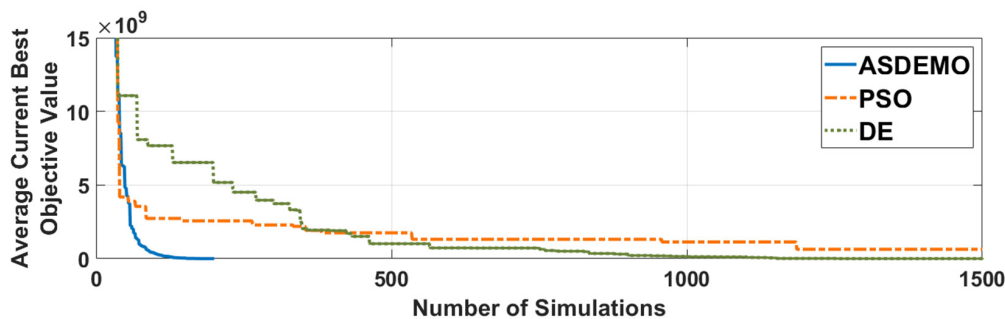


FIGURE 3. Convergence trends of ASDEMO (10 runs), PSO (2 runs) and DE (2 runs). Other methods are not included because all fail in all runs.

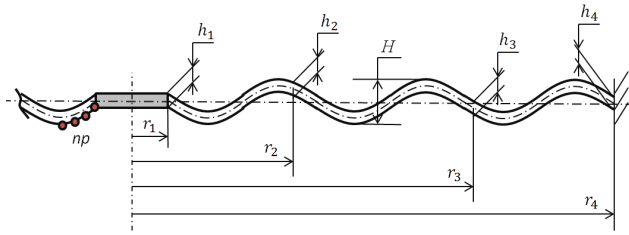


FIGURE 4. Layout of the corrugated membrane actuator.

this criterion, ASDEMO converges using an average of 163 simulations, with a standard deviation of 20.3 simulations. PSO and DE obtain average objective function values of 9.72 nm^2 and 9.71 nm^2 , respectively, using 1186 simulations (about 3 weeks) and 1468 simulations (about 4 weeks) each. It can be seen that ASDEMO obtains better results in terms of optimality compared to DE and PSO. According to Figure 3, to obtain the average objective function value of PSO and DE, ASDEMO needs 85 and 193 simulations, respectively. Hence, ASDEMO offers 14 times speed improvement over PSO and 7.6 times speed improvement over DE for this example.

B. EXAMPLE 2

The second example is a silicon corrugated membrane micro-actuator (Figure 4) with a Young’s modulus of 135 GPa and a Poisson’s ratio of 0.33. The micro-actuator is analyzed using Hooke’s law and the subsequent equilibrium and geometric equations in (23) to (25).

$$\sigma_{ij,j} + \rho f_i = 0 \tag{23}$$

where $\sigma_{ij,j}$ is the stress tensor components, ρ is the density and f is the loading vector component.

$$\epsilon_{ij,j} = \frac{1}{2}(U_{i,j} + U_{j,i} + \sum_l U_{l,i} U_{l,j}) \tag{24}$$

where $\epsilon_{ij,j}$ is the deformation tensor components, U is the displacement vector configuration.

$$\sigma_{ij,j} = \sum_{kl} C_{ijkl} \epsilon_{ij,j} \tag{25}$$

where C is the tensor components of the elastic constants.

The micro-actuator is modeled in ANSYS [56]. To make 15 runs for each method affordable for comparison, the micro-actuator is configured to have a typical mesh composed of about 500 2-D elements and each simulation costs 1 minute on average.

The ranges of the design variables are shown in Table 3. The design exploration goal is to satisfy specifications on critical pressure (P_c) and deflection at critical pressure ($D_f(@P_c)$) as shown in (26). Again, the design specifications are strict. Since this is a constraint satisfaction problem, the weighted sum of the violations of the two constraints of (26) serves as the objective function. To make the level of constraint violations at the same level, the penalty coefficients are 0.1 for P_c and 10 for $D_f @ P_c$. According to the number of

TABLE 3. Ranges of the design variables (all sizes in μm) (example 2).

Variables	H	r_1	r_2	r_3	r_4
Lower bound	75	30	150	400	550
Upper bound	80	70	390	500	7500
Variables	h_1	h_2	h_3	h_4	np
Lower bound	6	6	6	6	3
Upper bound	7.5	7.5	7.5	7.5	5

TABLE 4. Number of simulations required to meet specifications using different methods (example 2) (over 15 runs).

Method	Best	Worst	Average	Std
ASDEMO	28	207	123	58.3
SMAS	14	204	130	46.4
ASDEMO (Only DE/rand-to-best/2)	24	260	129	84.78
ASDEMO (Only DE/rand/2/dir)	53	296	154	83.41
ASDEMO (Only DE/hybrid trig)	38	244	133	63.58
PSO	52	1398	400	362.7
DE	20	723	273	197.5
IWO	N/A	N/A	N/A	N/A
L-SHADE	5	431	226	146.8

design variables, the population/swarm/plant size is set to 50 for all methods. To make all the methods converge, the computing budget is as follows: 250 simulations for ASDEMO, ASDEMO with a single mutation strategy, SMAS, 1500 simulations for PSO, DE, IWO, and L-SHADE. 15 independent runs are carried out for all the methods.

$$847.5 \text{ kPa} \leq P_c \leq 852.5 \text{ kPa} \tag{26}$$

$$170 \mu\text{m} \leq D_f(@P_c) \leq 175 \mu\text{m}$$

As shown in Table 4, in all the 15 runs, ASDEMO, SMAS, ASDEMO with a single mutation strategy (i.e., the strategies in (10), (11) and (12), respectively), PSO, DE, and L-SHADE obtain designs satisfying the design specifications in (26). IWO obtain designs satisfying the design specifications in (26) in only 2 runs within the computing budget of 1500 simulations. These two successful runs use 46 and 28 simulations, respectively, which are comparable to the best cases for other methods. Although IWO satisfies the specifications efficiently in two successful runs, its stability is insufficient. Hence, IWO is not included in the efficiency comparison. A typical result for ASDEMO is $P_c = 847.89 \text{ kPa}$ and $D_f(@P_c) = 171.53 \mu\text{m}$.

According to Table 4, SAEA methods (ASDEMO, SMAS, ASDEMO with a single mutation strategy) are more efficient than EA methods (DE, PSO, L-SHADE) given both kinds of methods satisfy the design specifications. Among SAEAs, considering the average number of necessary simulations, ASDEMO’s efficiency is slightly better than SMAS and ASDEMO with a single mutation strategy. It can also

be observed that the worst case of ASDEMO is 207 simulations. This is similar to SMAS and about 1.2-1.5 times better than ASDEMO with a single mutation strategy (i.e., the strategies in (10), (11) and (12), respectively). This verifies the stability of ASDEMO; in particular, the effect of the three selected DE mutation strategies and the self-adaptive adoption mechanism.

C. BENCHMARK PROBLEM Tests

To show the performance of ASDEMO for problems having similar landscape characteristics with high-performance MEMS optimization, but may have more local optima, a mathematical benchmark problem, the 15 – d Ackley function (27) [49] is used. Often, MEMS shape design optimization problems have fewer than 10 design variables. The use of 15 dimensions shows possible co-design with other devices. The Ackley function has a global minimum at $x = 0^d$ with a function value of 0. It has a narrow peak and numerous local optima. Considering the number of design variables in (27), the population/swarm/plant size is set to 50 for all methods. To make all methods converge, the computing budget is as follows: 650 evaluations for surrogate model-assisted optimization methods, and 7500 evaluations for PSO, DE, IWO, and L-SHADE, respectively. 30 independent runs are carried out for each method.

$$\begin{aligned} \min f(x) &= -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)} \dots \\ &+ 20 + e \\ x &\in [-30, 30], \quad i = 1, \dots, 15 \\ \text{minimum : } f(x^*) &= 0 \end{aligned} \quad (27)$$

The performances and convergence trends are shown in Table 5 and Figure 5, respectively. As revealed in Figure 5, ASDEMO obtains an average function value of 0.0895 using the assigned budget of 650 function evaluations. The convergence criterion that we defined is that the improvement of the objective function is less than $1e-4$ after 30 function evaluations. Using this criterion, ASDEMO converges using an average of 558 function evaluations. All other reference surrogate model-assisted optimization methods fail to obtain this average value after 650 function evaluations. PSO, DE, IWO, and in particular the famous L-SHADE also fail to obtain this average value after 7500 function evaluations,

TABLE 5. Statistics of the best objective function values using different methods (example 3) (over 30 runs).

Method	Best	Worst	Average	Std.
ASDEMO	2.88e-05	0.8565	0.0895	0.1811
SMAS	0.0836	1.0306	0.2789	0.1945
ASDEMO (Only DE/rand-to-best/2)	0.1071	5.4957	1.4221	1.3033
ASDEMO (Only DE/rand/2/dir)	0.0399	1.1017	0.2908	0.2517
ASDEMO (Only DE/hybrid trig)	0.0224	0.9444	0.1201	0.1909
PSO	6.21e-4	4.6122	0.4500	0.9951
DE	0.2706	1.6989	0.7380	0.2925
IWO	8.88e-16	1.8483	0.8092	0.3838
L-SHADE	2.67e-7	0.9313	0.1552	0.3530

respectively. To obtain the average function values of SMAS, ASDEMO with strategy in (10)), (11)) and (12)), PSO, DE, IWO and L-SHADE, ASDEMO needs 563, 320, 561, 626, 518, 440, 422 and 617 function evaluations, respectively as shown in Figure 5. Hence, ASDEMO offers 1.16, 2.03, 1.16, 1.04, 14.5, 17.1, 17.8 and 11.8 times speed improvement respectively, while obtaining better average result for this complex problem. In terms of the best result in the 30 runs, although the winner is IWO, the best result obtained by ASDEMO, PSO and L-SHADE are of comparable high quality. In terms of the worst result in the 30 runs, the winner is ASDEMO, showing its stable performance.

D. DISCUSSIONS

The three selected test cases involve various complexity of design landscapes, various number of design variables and various kinds of optimization problems (goal optimization, constraint satisfaction and constrained optimization). Based on the results, the following conclusions can be drawn: (1) ASDEMO can address high-performance MEMS design problems with stringent design specifications efficiently, while other reference methods, including dominant methods in MEMS shape optimization (DE and PSO), a novel heuristic method (IWO) and a state-of-the-art EA (L-SHADE), find these problems difficult to address. (2) ASDEMO is scalable to 15 design variables, which is often enough for MEMS design optimization problems. (3) Besides significantly improved efficiency, ASDEMO even shows improved

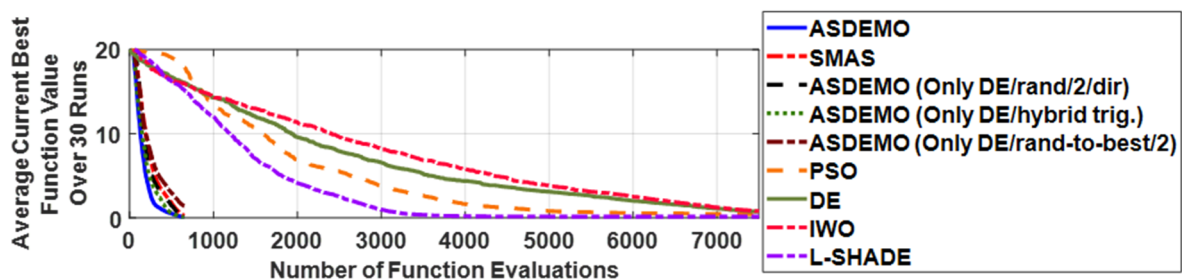


FIGURE 5. Convergence trends of ASDEMO, SMAS, ASDEMO (Only DE/rand/2/dir), ASDEMO (Only DE/hybrid trig.), ASDEMO (Only DE/rand-to-best/2), PSO, DE, IWO and L-SHADE (Average over 30 runs).

solution quality compared to DE, PSO, IWO and L-SHADE for the targeted problem. Often, surrogate model-assisted optimization methods are worse than pure EAs in terms of optimization quality due to the no-free-lunch theorem. (4) Compared to a popular surrogate model-assisted EA, SMAS, ASDEMO also shows both improved optimization quality and efficiency.

V. CONCLUSION

In this paper, the ASDEMO algorithm has been proposed for efficient high-performance MEMS design optimization. As demonstrated by two real-world MEMS design problems and a complex mathematical benchmark problem, ASDEMO can provide optimal designs that are better than DE, PSO, IWO and L-SHADE at a far lower computational cost. Therefore, ASDEMO achieves the goals in Section 1. These advantages are achieved by the core ideas of the new on-line SAEA framework, the three DE mutation strategies specially selected and combined for the targeted problem and the self-adaptive adoption of them. The ASDEMO algorithm can also serve as a reference for design optimization with stringent specifications in other engineering domains. Note that the robustness of the generated designs has not been considered in ASDEMO. Future works will focus on addressing robust high-performance MEMS design optimization and developing ASDEMO-based software tools.

ACKNOWLEDGMENT

The authors would like to thank Mrs. Anna Nikolaeva, Skolkovo Institute of Science and Technology, Russia, for valuable discussions.

REFERENCES

- [1] M. G.-E. Hak, *The MEMS Handbook*. Boca Raton, FL, USA: CRC Press, 2001.
- [2] J. L. Pons, *Wearable Robots: Biomechatronic Exoskeletons*. Hoboken, NJ, USA: Wiley, 2008.
- [3] I. Biswas, P. Roy (Kundu), M. Majumder, S. Sau, and A. K. Chakraborty, "Low-temperature synthesis of strain sensor based on flexible ZnO nanowire-cellulose paper composite," *Micro Nano Lett.*, vol. 12, no. 7, pp. 474–477, Jul. 2017.
- [4] D. Shin, A. Tanaka, N. Kim, and O. Khatib, "A centrifugal force-based configuration-independent High-Torque-Density passive brake for human-friendly robots," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 6, pp. 2827–2835, Dec. 2016.
- [5] D. Xiao, X. Zhou, Q. Li, Z. Hou, X. Xi, Y. Wu, and X. Wu, "Design of a disk resonator gyroscope with high mechanical sensitivity by optimizing the ring thickness distribution," *J. Microelectromech. Syst.*, vol. 25, no. 4, pp. 606–616, Aug. 2016.
- [6] T. Perrier, R. Levy, B. Bourgeteau-Verlhac, P. Kayser, J. Moulin, and S. Paquay, "Optimization of a MEMS magnetic thin film vibrating magnetometer," *IEEE Trans. Magn.*, vol. 53, no. 4, pp. 1–5, Apr. 2017.
- [7] M. M. Shalaby, M. A. Abdelmoneum, and K. Saitou, "Design of spring coupling for high-Q high-frequency MEMS filters for wireless applications," *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 1022–1030, Apr. 2009.
- [8] P. D. Barba and S. Wiak, "Evolutionary computing and optimal design of MEMS," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1660–1667, Aug. 2015.
- [9] J. K. Coulter, C. H. J. Fox, S. McWilliam, and A. R. Malvern, "Application of optimal and robust design methods to a MEMS accelerometer," *Sens. Actuators A, Phys.*, vol. 142, no. 1, pp. 88–96, Mar. 2008.
- [10] Z. Fan, J. Liu, T. Sorensen, and P. Wang, "Improved differential evolution based on stochastic ranking for robust layout synthesis of MEMS components," *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 937–948, Apr. 2009.
- [11] S. Nabavi and L. Zhang, "Design and optimization of piezoelectric MEMS vibration energy harvesters based on genetic algorithm," *IEEE Sensors J.*, vol. 17, no. 22, pp. 7372–7382, Nov. 2017.
- [12] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. New York, NY, USA: Springer-Verlag, 2005.
- [13] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. New York, NY, USA: Springer, 2011, pp. 760–766.
- [14] H. Ghodous and Z. Kordrostami, "Harvesting the ultimate electrical power from MEMS piezoelectric vibration energy harvesters: An optimization approach," *IEEE Sensors J.*, vol. 18, no. 21, pp. 8667–8675, Nov. 2018.
- [15] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
- [16] Q. Hao, Z. Zhou, Z. Wei, and G. Chen, "Parameters identification of photovoltaic models using a multi-strategy Success-History-Based adaptive differential evolution," *IEEE Access*, vol. 8, pp. 35979–35994, 2020.
- [17] X. Li, G. Dai, M. Wang, Z. Liao, and K. Ma, "A two-stage ensemble of differential evolution variants for numerical optimization," *IEEE Access*, vol. 7, pp. 56504–56519, 2019.
- [18] T. Tan, S. Roy, N. Thuy, and H. Huynh, "Streamlining the design of MEMS devices: An acceleration sensor," *IEEE Circuits Syst. Mag.*, vol. 8, no. 1, pp. 18–27, 1st Quart., 2008.
- [19] L. Cui, J. Deng, F. Liu, Y. Zhang, and M. Xu, "Investigation of RFID investment in a single retailer two-supplier supply chain with random demand to decrease inventory inaccuracy," *J. Cleaner Prod.*, vol. 142, pp. 2028–2044, Jan. 2017.
- [20] L. Cui, J. Deng, L. Wang, M. Xu, and Y. Zhang, "A novel locust swarm algorithm for the joint replenishment problem considering multiple discounts simultaneously," *Knowl.-Based Syst.*, vol. 111, pp. 51–62, 2016.
- [21] Y. Lee, Y. Park, F. Niu, and D. Filipovic, "Design and optimisation of one-port RF MEMS resonators and related integrated circuits using ANN-based macromodelling approach," *IEE Proc. Circuits, Devices Syst.*, vol. 153, no. 5, p. 480, Oct. 2006.
- [22] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, May 1987.
- [23] B. Liua and A. Nikolaeva, "Efficient global optimization of MEMS based on surrogate model assisted evolutionary algorithm," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2016, pp. 555–558.
- [24] Q. Zhou, Y. Wang, P. Jiang, X. Shao, S.-K. Choi, J. Hu, L. Cao, and X. Meng, "An active learning radial basis function modeling method based on self-organization maps for simulation-based design problems," *Knowl.-Based Syst.*, vol. 131, pp. 10–27, Sep. 2017.
- [25] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.
- [26] M. Sadoughi, C. Hu, B. Moghadassian, A. Sharma, J. Beck, and D. Mathiesen, "Sequential online dispatch in design of experiments for Single- and multiple-response surrogate modeling," *IEEE Trans. Autom. Sci. Eng.*, early access, Feb. 20, 2020, doi: 10.1109/TASE.2020.2969884.
- [27] X. Cai, L. Gao, and X. Li, "Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 365–379, Apr. 2020.
- [28] E. R. Ackermann, J. P. de Villiers, and P. J. Cilliers, "Nonlinear dynamic systems modeling using Gaussian processes: Predicting ionospheric total electron content over south africa," *J. Geophys. Res., Space Phys.*, vol. 116, no. A10, Oct. 2011, Art. no. A10303.
- [29] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 183–194, May 2005.
- [30] J. Dennis and V. Torczon, "Managing approximation models in optimization," in *Multidisciplinary Design Optimization: State-of-the-Art*. Philadelphia, PA, USA: SIAM, 1997, pp. 330–347.
- [31] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.
- [32] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

- [33] G. I. Hawe and J. K. Sykulski, "A scalarizing one-stage algorithm for efficient multi-objective optimization," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1094–1097, Jun. 2008.
- [34] B. Liu, D. Zhao, P. Reynaert, and G. G. E. Gielen, "GASPAD: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 2, pp. 169–182, Feb. 2014.
- [35] C. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Heidelberg, Germany, 2004, pp. 63–71.
- [36] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, 1998.
- [37] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numer.*, vol. 4, pp. 1–51, Jan. 1995.
- [38] V. Feoktistov and S. Janaqi, "Generalization of the strategies in differential evolution," in *Proc. 18th Int. Parallel Distrib. Process. Symp.*, Apr. 2004, pp. 165–170.
- [39] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
- [40] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *J. Global Optim.*, vol. 27, no. 1, pp. 105–129, 2003.
- [41] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.
- [42] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 66–76, Jan. 2007.
- [43] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. GECCO*, 2006, pp. 485–492.
- [44] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," *Found. Genetic Algorithms*, vol. 1, pp. 69–93, Jan. 1991.
- [45] P. Moscato, C. Cotta, and A. Mendes, "Memetic algorithms," in *New Optimization Techniques in Engineering*. Heidelberg, Germany: Springer, 2004, pp. 53–85.
- [46] B. Liu, V. Grout, and A. Nikolaeva, "Efficient global optimization of actuator based on a surrogate model assisted hybrid algorithm," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5712–5721, Jul. 2018.
- [47] B. Liu, H. Yang, and M. J. Lancaster, "Global optimization of microwave filters based on a surrogate model-assisted evolutionary algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1976–1985, Jun. 2017.
- [48] B. Liu, H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen, and P. Excell, "An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques," *IEEE Trans. Antennas Propag.*, vol. 62, no. 1, pp. 7–18, Jan. 2014.
- [49] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, p. 150, 2013.
- [50] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.
- [51] S. Karimkashi and A. A. Kishk, "Invasive weed optimization and its features in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 58, no. 4, pp. 1269–1278, Apr. 2010.
- [52] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecol. Informat.*, vol. 1, no. 4, pp. 355–366, Dec. 2006.
- [53] M. M. Roshanaei, C. Lucas, and A. R. Mehrabian, "Adaptive beamforming using a novel numerical optimisation algorithm," *IET Microw., Antennas Propag.*, vol. 3, no. 5, p. 765, Aug. 2009.
- [54] M. Geitle and R. Olsson, "A new baseline for automated hyper-parameter optimization," in *Machine Learning, Optimization, and Data Science (Lecture Notes in Computer Science)*, vol. 11943. Cham, Switzerland: Springer, 2019, pp. 521–530.
- [55] P. Di Barba, B. Liu, M. E. Mognaschi, P. Venini, and S. Wiak, "Multi-physics field analysis and evolutionary optimization: Design of an electrothermo-elastic microactuator," *Int. J. Appl. Electromagn. Mech.*, vol. 54, no. 3, pp. 433–448, Jul. 2017.
- [56] J. Han and D. P. Neikirk, "Deflection behavior of Fabry-Pérot pressure sensors having planar and corrugated membrane," *Proc. SPIE*, vol. 2882, pp. 79–90, Sep. 1996.



MOBAYODE O. AKINSOLU (Member, IEEE) received the M.Sc. degree (Hons.) in electrical and electronic engineering from the University of Bradford, U.K., in 2014, after undertaking his undergraduate studies and compulsory national service in Nigeria. He has been a Research Fellow (industrial attaché) and a Visiting Researcher with the National Space Research and Development Agency, Nigeria, and also been with the RFID Research Centre, African University of Science and Technology, Nigeria, since 2016. From 2016 to 2019, he was a Ph.D. Scholarship Awardee in recognition of a Joint Project between Wrexham Glyndŵr University, U.K., and the University of Birmingham, U.K. He is currently a Lecturer (Assistant Professor) in electronic and communication engineering with Wrexham Glyndŵr University. His research interest includes electromagnetic design automation using surrogate model-assisted evolutionary algorithms. He is a member of IET and a Registered Electrical Engineer with COREN.



BO LIU (Senior Member, IEEE) received the B.Sc. degree from Tsinghua University, China, in 2008, and the Ph.D. degree from the University of Leuven (KU Leuven), Belgium, in 2012. From 2012 to 2013, he was a Humboldt Research Fellow with the Technical University of Dortmund, Germany. In 2013, he was a Lecturer (Assistant Professor) with Wrexham Glyndŵr University, U.K., where he was promoted to a Reader (Associate Professor), in 2016. Since 2020, he has been a Senior Lecturer (Associate Professor) with the University of Glasgow, U.K. He has authored or coauthored one book and more than 60 articles in renowned international journals, edited books, and conference proceedings. His research interests include AI-driven design methodologies of analog/RF integrated circuits, microwave devices, MEMS, evolutionary computation, and machine learning.



PAVLOS I. LAZARIDIS (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the Aristotle University of Thessaloniki, Greece, in 1990, the M.Sc. degree in electronics from Université Pierre and Marie Curie, Paris, France, in 1992, and the Ph.D. degree from ENST Paris, Paris, in 1996. From 1991 to 1996, he was involved in research on semiconductor lasers and wave propagation with France Télécom and Teaching with ENST Paris. In 1997, he was the Head of the Antennas and Propagation Laboratory, TDFC2R Metz France (Télécom Research Center/Télédiffusion de France). From 1998 to 2002, he was a Senior Examiner with the European Patent Office (EPO), Den Haag, The Netherlands. From 2002 to 2014, he was involved in teaching and research with ATEL, Thessaloniki, Greece, and Brunel University London. He is currently a Professor in electronic and electrical engineering with the University of Huddersfield, U.K. He is a member of IET and the Higher Education Academy.



KEYUR K. MISTRY (Graduate Student Member, IEEE) received the B.Tech. degree in electronics and telecommunication engineering from NMIMS University, Mumbai, India, in 2015, and the master's degree in electronics and communications engineering from the University of Huddersfield, U.K., in 2016. He is currently pursuing the Ph.D. degree in electronics and communications engineering. His research interests include antenna design and measurements, wireless partial discharge sensors, UHD TV broadcasting, and developing optimization algorithms. He received the Vice-Chancellor Scholarship with the University of Huddersfield for his Ph.D. degree. He was a recipient of the First Prize with the Best Design for the Proof of Concept Design Competition organized by ICE/HoneyPot, in 2016.



MARIA EVELINA MOGNASCHI (Member, IEEE) received the B.Sc. and M.Sc. (*Laurea*) degrees in computer science and the Ph.D. degree in electronics, computer science, and electrical engineering from The University of Pavia, Pavia, Italy, in 2003 and 2007, respectively. She is currently an Assistant Professor with the Department of Electrical, Computer, and Biomedical Engineering, The University of Pavia. She has authored or coauthored more than 80 ISI- or Scopus-indexed articles, either presented to international conferences or published in international journals. Her research interests include inverse problems, in particular multiobjective optimization, identification problems in electromagnetism, and biological systems.



PAOLO DI BARBA (Member, IEEE) received the M.Eng. degree in electrical engineering from The University of Pavia, Pavia, Italy, in 1988, and the Ph.D. and D.Sc. degrees in electrical engineering from the Lodz University of Technology, Lodz, Poland, in 2002 and 2013, respectively. He is currently a Full Professor of electrical engineering with the Department of Electrical, Computer, and Biomedical Engineering, The University of Pavia. He has authored or coauthored more than 150 articles, either presented to international conferences or published with international journals, such as the book *Field Models in Electricity and Magnetism* (Springer, 2008), the monograph *Multiobjective Shape Design in Electricity and Magnetism* (Springer, 2010), and the monograph *MEMS: Field Models and Optimal Design* (Springer, 2019). His research interests include the computer-aided design of electric and magnetic devices, with special emphasis on the methods for field synthesis and automated optimal design.



ZAHARIAS D. ZAHARIS (Senior Member, IEEE) received the B.Sc. degree in physics, the M.Sc. degree in electronics, the Ph.D. degree in antennas and propagation modeling for mobile communications, and the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1987, 1994, 2000, and 2011, respectively. From 2002 to 2013, he was with the Administration of the Telecommunications Network, Aristotle University of Thessaloniki, where he has been with the Department of Electrical and Computer Engineering, since 2013. His current research interests include design and optimization of antennas and microwave circuits, signal processing on smart antennas, development of evolutionary optimization algorithms, and neural networks. He is a member of the Technical Chamber of Greece. He serves as an Associate Editor for IEEE ACCESS.

...