Improved Differential Evolution Based on Stochastic Ranking for Robust Layout Synthesis of MEMS Components

Fan, Zhun; Liu, Jinchao; Sørensen, Torben; Wang, P.

Published in:
IEEE Transactions on Industrial Electronics

Link to article, DOI:
10.1109/TIE.2008.2006935

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Improved Differential Evolution Based on Stochastic Ranking for Robust Layout Synthesis of MEMS Components

Zhun Fan, *Member, IEEE*, Jinchao Liu, Torben Sørensen, and Pan Wang, *Member, IEEE*

***Abstract*—This paper introduces an improved differential evolution (DE) algorithm for robust layout synthesis of microelectromechanical system components subject to inherent geometric uncertainties. A case study of the layout synthesis of a comb-driven microresonator shows that the approach proposed in this paper can lead to design results that meet the target performance and are less sensitive to geometric uncertainties than the typical designs. It is also demonstrated that the algorithm proposed in this paper cannot only obtain better results than the standard DE algorithm but also outperform some other state-of-the-art algorithms in constrained optimization.**

***Index Terms*—Differential evolution (DE), microelectromechanical systems (MEMS), robust design, stochastic ranking (SR).**

I. Introduction

MICROELECTROMECHANICAL systems (MEMS) are tiny mechanical devices that are built upon semiconductor chips and are measured in micrometers. They usually integrate a number of functions, including fluidics, optics, mechanics, and electronics, across different physical domains and are used to make numerous devices such as pressure sensors, gyroscopes, engines, accelerometers, etc. Many designs of MEMS are made through engineering experience and back-of-the-envelope calculations and are highly dependent on the knowledge and experience of the designers.

One reason for this is the complexity involved in the modeling, design, and fabrication of MEMS—there are many constraints in designing and fabricating MEMS devices due to the limitations of current fabrication techniques. However, as process technologies become more stable, research emphasis can be shifted from developing specific process technologies toward the design of systems with a large number of reusable components, such as resonators, accelerometers, gyroscopes, and micromirrors. It greatly benefits the MEMS designers if the routine design of frequently used components can be optimized automatically by computer programs, while the designers can take more time in contemplating the more creative conceptual designs.

It has been shown that the performance of individual components influences the quality of the whole system. For example, the frequency stability of a MEMS resonator can directly affect the quality of the MEMS RF system in which it serves as a component of a filter or an oscillator [1]. Because microresonators are basically 2.5 dimensional devices, the design automation of microresonators boils down to a layout synthesis problem, which has been carried out by many researchers. Some notable research uses deterministic numeric approaches [2]–[4] and metaheuristic approaches such as evolutionary computation [5], [6] and simulated annealing [7], [8].

It is notable that, with current micromachining techniques, the fabrication process variation in MEMS is inevitable when devices are miniaturized to the point of process limitations [1]. For example, it is reported in [9] that the width of a typical suspension beam has a fabrication tolerance of about 10%. How to design MEMS that is most insensitive to fabrication process variation is therefore an important issue in MEMS design. This paper also addresses this important issue, i.e., incorporating uncertainty in MEMS design through robust optimization.

Many approaches exist in the literature to incorporate uncertainty in a design formulation. Taguchi [10] introduced the concept of robust design to improve the quality of products with significant variations in their manufacturing process by reducing the sensitivity of the design performance to possible sources of variations without an attempt to eliminate the sources. Robust design has been developed and applied in many areas. Some examples include robust optical coating design [11], [12], robust design of a vibratory microgyroscope [13], an active micromixer [14], and brushless dc motors [15].

In this paper, we present a robust optimization approach for designing MEMS subject to process-induced geometrical uncertainties. In this approach, we first formulate the robust design problem as a multiobjective constrained optimization problem [16], [17] and then solve it using an improved differential evolution (DE). DE is a strong and efficient optimization algorithm capable of handling nonlinear, nondifferentiable, and multimodal objective functions [18]. A case study based on the layout synthesis of comb-driven microresonator shows that the robust designs nominally meet the target performance and are less sensitive to geometric uncertainties. It is also demonstrated

Manuscript received April 24, 2008; revised September 2, 2008. First published October 31, 2008; current version published April 1, 2009. This work was supported in part by the DCAMM research school of Denmark.

Z. Fan is with the Department of Management Engineering, Technical University of Denmark, 2800 Lyngby, Denmark (e-mail: zf@mek.dtu.dk).

J. Liu is with the Department of Management Engineering, Technical University of Denmark, 2800 Lyngby, Denmark, and also with the FORCE Technology, 2605 Brøndby, Denmark (e-mail: jql@force.dk).

T. Sørensen, deceased, was with the Department of Management Engineering, Technical University of Denmark, 2800 Lyngby, Denmark.

P. Wang is with the School of Automation, Wuhan University of Technology, Wuhan 430072, China (e-mail: jfpwang@yeah.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2008.2006935

that the algorithm proposed in this paper cannot only obtain better results than the standard DE algorithm but also outperform some other state-of-the-art algorithms in constrained optimization.

The remainder of this paper is organized as follows. Section II introduces the formulation of the general robust optimization problem that is used to formulate the MEMS layout synthesis problem in this paper. Section III describes the method of modeling uncertainty in MEMS fabrication process. Section IV explains in details the improved DE algorithm (IDE) used to do the robust layout synthesis. Section V gives the description of the case study of comb-driven microresonator design, and Section VI presents experimental results. Section VII concludes this paper with a brief summary.

II. Formulation of the Robust Optimization Problem

This paper considers the application of a general robust optimization problem that can be formulated as the following [17].

Let $\vec{x} = \{x_1, x_2 \ldots, x_n\}$ be an array of design variables of a given design problem. We assume that the uncertainty $\vec{\delta} = \{\delta_1, \delta_2, \ldots, \delta_n\}$ can be characterized as a random vector with the following statistics:

$$E(\vec{\delta}) = 0_{nx1} \tag{1}$$

$$E(\vec{\delta}\vec{\delta}^{\mathrm{T}}) = \Omega \in \Re^{nxn} \tag{2}$$

where Ω is the covariance matrix and is positive semidefinite. If the uncertainties are uncorrelated, then Ω is diagonal; otherwise, the off-diagonal entries are nonzero when correlation exists.

Given a function $f(\vec{x}, \vec{\delta})$ describing the performance of a design merit, the robust design problem that we aim to solve is to minimize the expected value of the squared error between the actual and target performances. We can write this as

$$\min_x E\left(f(\vec{x}, \vec{\delta}) - \bar{f}\right)^2 \quad \text{subject to} \quad g_i(\vec{x}) \leq 0 \tag{3}$$

where $\bar{f}$ is the target performance, and the expectation is taken over the random vector $\vec{\delta}$. In addition, $g_i(\vec{x}) \leq 0$ represents a list of constraints to be satisfied.

The problem posed in (3) is a difficult robust optimization problem to solve in general. To simplify the problem, we choose to approximate $f(\vec{x}, \vec{\delta})$ with a first order Taylor series expansion in $\vec{\delta}$ as

$$f(\vec{x}, \vec{\delta}) \cong f(\vec{x}, 0) + \nabla_\delta f(\vec{x}, 0)\vec{\delta} \tag{4}$$

where $\nabla_\delta f(\vec{x}, 0)$ is the gradient of $f(\vec{x}, \vec{\delta})$ with respect to $\vec{\delta}$. Using this approximation, we can expand the expression of $(f(\vec{x}, \vec{\delta}) - \bar{f})^2$ into

$$\left(f(\vec{x}, \vec{\delta}) - \bar{f}\right)^2 \cong \left(f(\vec{x}, 0) - \bar{f}\right)^2 + 2\left(f(\vec{x}, 0) - \bar{f}\right) \times \nabla_\delta f(\vec{x}, 0)\vec{\delta} + \nabla_\delta f(\vec{x}, 0)\vec{\delta}\vec{\delta}^{\mathrm{T}}\nabla_\delta^{\mathrm{T}} f(\vec{x}, 0). \tag{5}$$

Taking the expectation of the aforementioned equation, we can get

$$E\left(f(\vec{x}, \vec{\delta}) - \bar{f}\right)^2 \cong \left(f(\vec{x}, 0) - \bar{f}\right)^2 + 2\left(f(\vec{x}, 0) - \bar{f}\right) \times \nabla_\delta f(\vec{x}, 0)E(\vec{\delta}) + \nabla_\delta f(\vec{x}, 0)E(\vec{\delta}\vec{\delta}^{\mathrm{T}})\nabla_\delta^{\mathrm{T}} f(\vec{x}, 0). \tag{6}$$

By reducing (6), based on our assumptions about the mean and covariance of $\vec{\delta}$ according to (1) and (2), we obtain

$$E\left(f(\vec{x}, \vec{\delta}) - \bar{f}\right)^2 \cong \left(f(\vec{x}, 0) - \bar{f}\right)^2 + \nabla_\delta f(\vec{x}, 0)\Omega\nabla_\delta^{\mathrm{T}} f(\vec{x}, 0). \tag{7}$$

Substituting the approximation in (7) back into the original design problem posed in (3) yields

$$\min_x \left\{\left(f(\vec{x}, 0) - \bar{f}\right)^2 + \nabla_\delta f(\vec{x}, 0)\Omega\nabla_\delta^{\mathrm{T}} f(\vec{x}, 0)\right\} \quad \text{subject to } g_i(\vec{x}) \leq 0. \tag{8}$$

To normalize the cost function, we decide to divide through by $\bar{f}^2$. We then refer to the following expression as our robust design problem:

$$\min_x \left(\left(\frac{f(\vec{x}, 0) - \bar{f}}{\bar{f}}\right)^2 + \frac{1}{\bar{f}^2}\left(\nabla_\delta f(\vec{x}, 0)\Omega\nabla_\delta^{\mathrm{T}} f(\vec{x}, 0)\right)\right) \quad \text{subject to } g_i(\vec{x}) \leq 0. \tag{9}$$

It is now easy to see that the expression we want to minimize has two distinct terms. For notational convenience, we will label the two terms as

$$N(\vec{x}) \equiv \left(\frac{f(\vec{x}, 0) - \bar{f}}{\bar{f}}\right)^2 \tag{10}$$

$$D(\vec{x}, \Omega) \equiv \frac{1}{\bar{f}^2}\nabla_\delta f(\vec{x}, 0)\Omega\nabla_\delta^{\mathrm{T}} f(\vec{x}, 0). \tag{11}$$

With the aforementioned definitions, the robust design problem posed in (9) becomes

$$\min_x \{N(\vec{x}), D(\vec{x}, \Omega)\} \quad \text{subject to } g_i(\vec{x}) \leq 0. \tag{12}$$

The first term $N(\vec{x})$ penalizes the deviation of the nominal solution $f(\vec{x}, 0)$ from the target $\bar{f}$, while the second term $D(\vec{x}, \Omega)$ penalizes the sensitivity of the design with respect to $\vec{\delta}$. The first term is a performance index, while the second term is a robustness index. Since there are two objectives in the formation of the cost function to be minimized, a tradeoff is usually needed to be made by the designer. In practice, if we want to obtain robust designs, a simple way is to sum the two objectives into a single one and take it as the objective to be minimized. One disadvantage of applying this method in our work is that it cannot guarantee that $D(\vec{x}, \Omega)$ is reduced after optimization. Theoretically, it is possible that $D(\vec{x}, \Omega)$ increases if the decrease of $N(\vec{x})$ is more than the increase of $D(\vec{x}, \Omega)$ within the reduced sum. In our approach, we set a small threshold constraint $N(\vec{x}) \leq 1.0e-6$ and then focus on minimizing the value of $D(\vec{x}, \Omega)$.

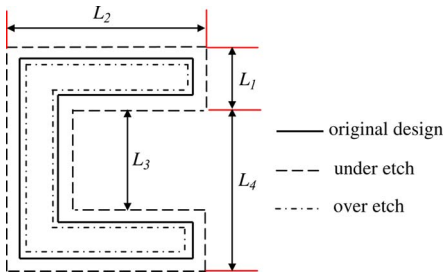


Fig. 1. Under- and overetch of a MEMS structure.

III. Modeling Uncertainty in MEMS Fabrication Process

We assume that the uncertainty in the fabrication process is introduced by etch- or lithograph-induced variations in linewidth, and the structure is etched uniformly.

Fig. 1 shows the two uniform etch scenarios on a structure—overetch and underetch. Take the underetch situation, for example, after process variation is introduced, some design variables may increase (such as L1 and L2) and other design variables (such as L3) may decrease, while some others may stay unchanged (such as L4).

We can model the geometric process variations using a simple additive uncertain model

$$\tilde{x} = \vec{x} + \vec{\delta} \tag{13}$$

where $\tilde{x}$ is the uncertain (actual) design vector, and for the aforementioned simple example, $\vec{\delta} = \{\delta_{L_1}, \delta_{L_2}, \delta_{L_3}, \delta_{L_4}\}$.

Since the structure is etched uniformly, if we define ρ to be a normal random variable with zero mean and standard deviation of σ, then we can write

$$\vec{\delta} = \rho\zeta \tag{14}$$

where $\zeta = [1, 1, -1, 0]^{\mathrm{T}}$ and is called a variation vector. Note that, in the condition of underetch, L1 and L2 increase, L3 decreases, and L4 is not changed. Also note that, in this case, ρ is positive; the aforementioned facts can easily be verified by (14).

Because ρ is a normal random variable, it can also be used to model the overetch situation, in which ρ will take a negative value.

According to (2), we can obtain

$$\Omega = E(\vec{\delta}\vec{\delta}^{\mathrm{T}}) = \sigma^2\zeta\zeta^{\mathrm{T}}. \tag{15}$$

IV. Robust Optimization Using IDE

Many types of evolutionary approaches have been developed and implemented as design optimization tools [19], [20]. Genetic algorithms with a robust solution searching scheme were first presented by Tsutsui and Ghosh [21] and later discussed by Deb and Gupta [22], [23]. Jin and Branke [24] made a thorough survey of applying evolutionary computation in uncertain environments. Most recently, Lee *et al.* [25] use robust evolutionary algorithms for unmanned (combat) aerial vehicle aerodynamics and radio cross section design optimization. One advantage of using genetic algorithms is its convenience to solve the optimization problem with both discrete and continuous design variables. While it is very difficult for many numerical optimization approaches (for example, gradient-based approaches) to include considerations of feature size constraints in MEMS design [4], it is quite convenient for genetic algorithms to do so. We need to modify the objective function only slightly, mapping real values of design variables to integer multiples of the feature size before using them in the formulations of constraints and objectives. No modifications to the genetic algorithm are needed. In this paper, we always set the feature size as 0.09 μm. It is also very convenient for evolutionary computation to integrate integer design variables such as the number of comb fingers used in a microresonator.

An IDE based on Stochastic Ranking (IDE-SR) was developed and used to solve the robust layout synthesis problem in this paper. The succeeding sections will first introduce the standard DE algorithm and then explain the novel mechanisms developed in IDE-SR in details.

A. Standard DE

DE is one of the most recent evolutionary algorithms for solving real-parameter optimization problems. In each iteration, DE creates one new offspring individual by combining one parental individual and the differences of several other individuals in the same population. The generated offspring individual replaces the parental individual only if it is better. In general, DE has three parameters that can impact its performance significantly: scaling factor F, crossover control parameter p_{CR}, and population size N_P.

The population of DE contains N_P n-dimensional individuals

$$\vec{x}_{i,G} = \{x_{i,1,G}, x_{i,2,G}, \ldots, x_{i,n,G}\}, \qquad i = 1, 2, \ldots N_P \tag{16}$$

where G denotes the generation number. Because it is considered beneficial to the search process if the initial population can be statistically evenly distributed over the entire search space, each variable of all individuals in the initial population is randomly decided by a uniform distribution between lower and upper bounds predefined for each variable.

At each generation, a target vector $\vec{x}_{i,G}$ is first selected randomly, and then, a mutant vector $\vec{\nu}_{i,G}$ is created by disturbing the target vector using a mutation operation; after that, a trial vector $\vec{u}_{i,G}$ is formed by applying crossover operation between the target and mutant vectors. Finally, a selection operation is executed between the trial and target vectors to decide which vector goes to the next generation. The procedure is repeated N_P times to create all individuals for an offspring generation. The main procedure for DE is shown in Fig. 2 and explained in detail as follows.

1) Mutation Operation: For each target vector $\vec{x}_{i,G}$ at generation G, an associated mutant vector $\vec{\nu}_{i,G} = \{\nu_{i,1,G}, \nu_{i,2,G}, \ldots, \nu_{i,n,G}\}$ can be created by using one of the mutation strategies. The most commonly used strategies are as follows,

1. **Initialize**

   Parameters: N_P, P_{CR}, F, t, m, where N_P denotes the size of population; P_{CR} denotes the probability of crossover, F is a scaling factor, t denotes the generation, and m represents the number of individuals participating in mutation operation(e.g. 3 or 5).

   Generate the initial generation P_0, $t = 1$;
2. **While** termination criteria not satisfied **do:**
3. Evaluate population $P_t = \{x_1^t \quad x_2^t \quad \ldots \quad x_{N_P}^t\}$.
4. **For** $k = 1$ to N_P **do**
5. Select $x_{r_1}^t, x_{r_2}^t, x_{r_m}^t \in P_t$ at random;
6. $u_k^t \leftarrow mutate(x_{r_1}^t, x_{r_2}^t, x_{r_m}^t F)$;
7. $v_k^t \leftarrow crossover(u_k^t, x_k^t, P_{CR})$;
8. **If** v_k^t is better than x_k^t **then**
9. $x_k^t \leftarrow v_k^t$;
10. **End if**
11. **End for**
12. $t \leftarrow t + 1$;
13. **End while**

Fig. 2. Pseudocode of the iterative search procedure of DE.

where the indexes r_1, r_2, r_3, r_4, and r_5 represent the random and mutually different integers generated between 1 and N_P and $\vec{x}_{\text{best},G}$ is the best individual at generation G:

"rand/1" $\vec{\nu}_{i,G} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G})$;

"best/1" $\vec{\nu}_{i,G} = \vec{x}_{\text{best},G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G})$;

"current to best/1" $\vec{\nu}_{i,G} = \vec{x}_{i,G} + F(\vec{x}_{\text{best},G} - \vec{x}_{i,G}) + F(\vec{x}_{r1,G} - \vec{x}_{r2,G})$;

"best/2" $\vec{\nu}_{i,G} = \vec{x}_{\text{best},G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G}) + F(\vec{x}_{r3,G} - \vec{x}_{r4,G})$;

"rand/2" $\vec{\nu}_{i,G} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) + F(\vec{x}_{r4,G} - \vec{x}_{r5,G})$.

Different strategies have different features for different applications. It is also possible to use a combination of two or more strategies to better cope with certain application.

2) Crossover Operation: After mutation, a "binary" crossover operation is applied to form the final trial vector $\vec{u}_{i,G}$, according to its corresponding target vector $\vec{x}_{i,G}$ and mutant vector $\vec{\nu}_{i,G}$

$$\vec{u}_{i,j,G} = \begin{cases} \vec{\nu}_{i,j,G}, & \text{if rand} \le p_{\text{CR}} \text{ or } j = j_{\text{rand}} \\ \vec{x}_{i,j,G}, & \text{otherwise} \end{cases} \tag{17}$$

where $i = 1, 2, \ldots, N_P$; $j = 1, 2, \ldots, n$; and index j_{rand} is a randomly chosen integer within the range $[1, n]$. By making use of j_{rand}, it can be guaranteed that the trial vector $\vec{u}_{i,G}$ will differ from its target vector $\vec{x}_{i,G}$ by at least one parameter.

3) Selection Operation: After evaluating the target vector $\vec{x}_{i,G}$ and the corresponding trial vector $\vec{u}_{i,G}$, a "knockout" competition is played between them, and the vector with smaller objective function value is selected and added to the next population

$$\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G} & \text{if } f(\vec{u}_{i,G}) \le f(\vec{x}_{i,G}) \\ \vec{x}_{i,G}, & \text{otherwise.} \end{cases} \tag{18}$$

Because each individual has both the values of objective function and constraint violation for comparison, it is important to use some rules for the purpose of comparison. According to our empirical experience, different rules of handling constraints used can actually lead to very different results in constrained optimization algorithms.

1. $I_j = j, \forall j \in \{1,\ldots,\lambda\}$
2. **for** $i = 1$ to N **do**
3. **for** $j = 1$ to $\lambda - 1$ **do**
4. sample $\mu \in U(0,1)$
5. **if** $\phi(I_j) = \phi(I_{j+1}) = 0$ or $(\mu < P_f)$ **then**
6. **if** $f(I_j) > f(I_{j+1})$ **then**
7. swap(I_j, I_{j+1});
8. **end if**
9. **else**
10. **if** $\phi(I_j) > \phi(I_{j+1})$ then
11. swap(I_j, I_{j+1});
12. **end if**
13. **end if**
14. **end if**
15. **end for**
16. **if** no *swap* done **break**
17. **end for**

Fig. 3. Pseudocode of SR [29].

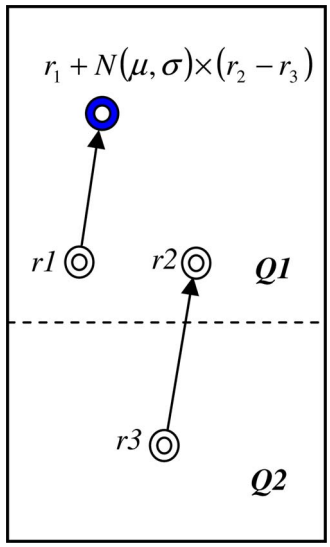


Fig. 4. Illustration of the modified mutation strategy in IDE-SR. Note that the population ranked by SR is divided into upper part Q1 and lower part Q2. The difference of one randomly selected individual r2 from Q1 and one randomly selected individual r3 from Q2 forms a differential vector pointing toward r2.

B. Different Rules of Handling Constraints

Very few constraint handling techniques have been reported in DE. Two very important and similar techniques are

Initialize:

Parameters: N_P, p_f, μ, σ, p_{CR}, γ;where N_P denotes the size of population; p_f is a parameter used in stochastic ranking., μ, σ denote the mean and variance of normal distribution, respectively; p_{CR} denotes the probability of crossover, γ represents the number of individuals in the upper part of the population Q_1.

Generate the initial generation P_0;

While termination criteria not satisfied **do:**

Evaluate population P_t: $(f,\varphi) = eval(P_t)$; where f,φ denote objective and violation of constraints, respectively.

Rank population using stochastic ranking: $I = stochastic\_rank(f,\varphi,p_f)$;

Divide population into two sets:

$$Q_1^t = \left\{x_{I(1)}^t \quad x_{I(2)}^t \quad \ldots \quad x_{I(\gamma)}^t\right\};$$

$$Q_2^t = \left\{x_{I(\gamma+1)}^t \quad x_{I(\gamma+2)}^t \quad \ldots \quad x_{I(\lambda)}^t\right\};$$

For $k = 1$ to N_P **do**

Select $x_{r_1}^t, x_{r_2}^t \in Q_1^t$ at random;

Select $x_{r_3}^t \in Q_2^t$ at random;

$$u_k^t \leftarrow x_{r_1}^t + N(\mu,\sigma)\times\left(x_{r_2}^t - x_{r_3}^t\right);$$

$$v_k^t \leftarrow crossover\left(u_k^t, x_k^t, P_{CR}\right);$$

If v_k^t is better than x_k **then**

$$x_k^t \leftarrow v_k^t;$$

End if

End for

$t \leftarrow t+1$;

End while

Fig. 5. Pseudocode of the iterative search procedure of the IDE based on SR: IDE-SR.

proposed by Lampinen [26] and Becerra and Coello [27]. Both techniques use three rules for the replacement during the selection procedure, and the first two are the same. They are as follows.

1) A feasible individual is always better than an infeasible individual.
2) If both individuals are feasible, the one with better value of the objective function is selected for the next generation.

The third rule, regarding the situation when both individuals are infeasible, is different. In Lampinen's approach, the comparison is made in the Pareto sense in the constraint violation space. It can be expressed as follows.

1) If both individuals are infeasible, the parent is replaced if the new individual has lower or equal violation for all the constraints.

In Becerra and Coello's approach, a sum of normalized constraint violations is used for comparison and can be written as follows.

1) If both individuals are infeasible, the individual with less level of constraint violations is better. The level of constraint violation is measured with the normalized constraints with the expression of $\mathrm{viol}(x_j) = \sum_{c=1}^{\mathrm{constr}} (g_c(x)/g_{\max c})$, where $g_c(x)$ denotes the violated constraints of the problem and $g_{\max c}$ is the largest violation of the constraint $g_c(x)$ found so far.

It is worthwhile to point out that both approaches bear some resemblance with an approach proposed by Deb [28] previously, even though Deb's approach is not based in DE. The key difference also lies in the comparison for the case of two infeasible individuals: Lampinen's method makes the comparison in the Pareto sense, Deb's method sums all the constraint violations and compares a single value, and Becerra and Coello's method makes normalization for the constraint violations before summing them together.

Like the selection of mutation strategies, the selection of proper constraint handling techniques is highly dependent on applications. In this paper, Becerra and Coello's approach was selected because it outperformed the others.

C. IDE-SR: An Improved DE Based on SR

The "rand/1" mutation strategy used in the standard DE provides no information of direction toward the global optimum. If the information of direction can be obtained and utilized in the search process, the performance of the algorithm has a potential to be improved. To avoid the search to be stuck in local minimum, however, the direction information should not

be local but global. To define a "global direction" information for the whole population is not an easy task, particularly when each individual has actually two features to compare with the others in a constraint optimization problem—one feature is the objective value, and the other is the level of constraint violation. How to optimally balance them in the comparison procedure presents a challenge.

SR [29] provides a convenient and powerful mechanism to balance the dominance in ranking the whole population with both objective value and constraint violation as comparison criteria. The pseudocode of SR is provided in Fig. 3.

The IDE-SR is designed with a focus on a modified mutation strategy, which can be described in more details as the following: For the generation of trial vectors, the whole population is first made to undergo an SR procedure. Then, the ranked population is divided into two parts—upper and lower parts. The upper part comprises of the "better" individuals who have been ranked high after the SR procedure. For each individual trial vector, the upper part contributes two "good" randomly selected individuals, and the lower part contributes one randomly selected individual that is "less good." The three individuals then make a mutation operation according to "rand/1" strategy, with the difference vector obtained through extracting one "good" individual with the "less-good" individual. It is notable that, in this way, the difference vector will always be directed toward the upper part of the population, thus leading the population to search upwards (Fig. 4). This procedure is repeated until the whole population of trail vectors is obtained. The rest of the algorithm is almost the same as the standard DE, with the exception that the scaling factor F can become a random variable as a variation of the algorithm. The overall procedure of the IDE-SR algorithm can be illustrated using the pseudocode listed in Fig. 5.

V. Case Study

A case study in the area of MEMS design was carried out to verify the effectiveness of the aforementioned robust optimization method using the IDE-SR. The design problem is a comb-drive microresonator with 15 mixed-type design variables and 24 design constraints, both linear and nonlinear. The following section gives a more detailed description of the case study.

A. Comb-Driven Microresonator Design

The comb-driven microresonator problem was originally taken from [4]. The goal of the design is to robustly match the resonant frequency to the predefined target frequency in the presence of geometric process variations. The comb-drive microresonator is fabricated in a polysilicon surface microstructural process. The layout of the comb-driven microresonator is shown in Fig. 6(a) and can be specified by 15 design variables, as shown in Fig. 6(b) [4]. The vector of design variables can then be defined as follows:

$$\vec{x}=[L_b,w_b,L_t,w_t,L_{\rm sy},w_{\rm sy},w_{\rm sa},w_{\rm cy},L_{\rm cy},L_c,w_c,L_{\rm sa},x_o,V,N_c] \tag{19}$$

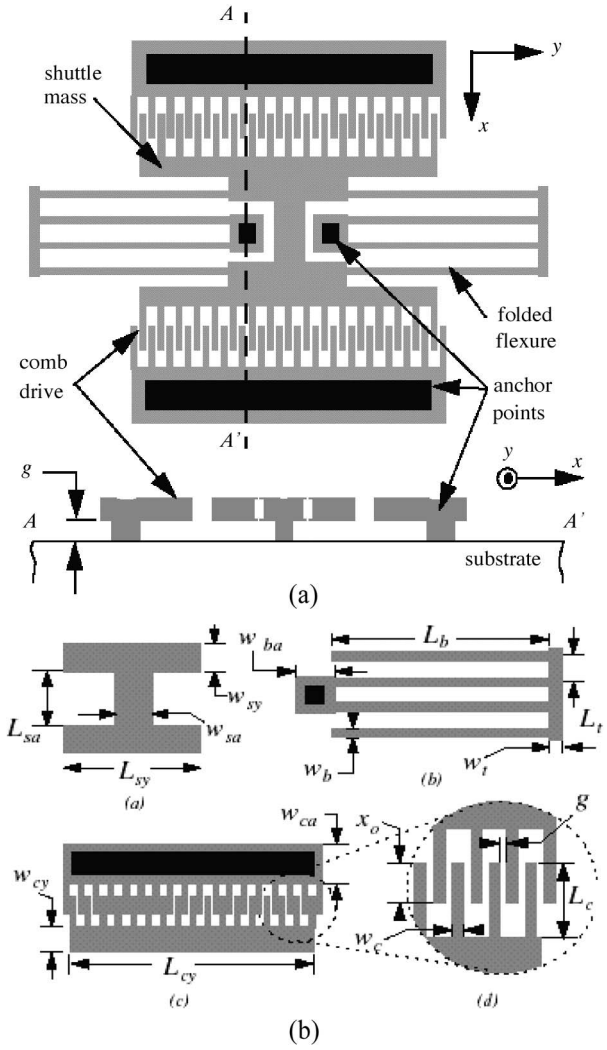


Fig. 6. (a) Comb-drive microresonator fabricated in a polysilicon surface microstructural process [4]. (b) Major design variables for comb-driven microresonators [4].

where

L_b and w_b	the length and width of flexture beam, respectively;
L_t and w_t	the length and width of truss beam, respectively;
$L_{\rm sy}$ and $w_{\rm sy}$	the length and width of shuttle yoke, respectively;
$L_{\rm cy}$ and $w_{\rm cy}$	the length and width of comb yoke, respectively;
L_c and w_c	the length and width of comb fingers, respectively;
$w_{\rm sa}$	the width of shuttle axle;
g	the gap between comb fingers;
x_o	the comb finger overlap;
V	the voltage amplitude;
N_c	the number of rotor comb fingers.

It is noted that the first 13 design variables have units of micrometers. They are discrete variables because they can only be integer multiples of the feature size, which is set to be 0.09 μm in this paper. The fourteenth design variable has units

of volts and is a continuous variable. The fifteenth variable has no unit and is an integer variable.

The constraints for the design variables are also listed as follows: $2 \le L_b \le 400$, $2 \le w_b \le 20$, $2 \le L_t \le 400$, $2 \le w_t \le 20$, $2 \le L_{\rm sy} \le 400$, $10 \le w_{\rm sy} \le 400$, $10 \le w_{\rm sa} \le 400$, $10 \le w_{\rm cy} \le 400$, $2 \le L_{\rm cy} \le 700$, $8 \le L_c \le 400$, $2 \le w_c \le 20$, $2 \le L_{\rm sa} \le 400$, $4 \le x_o \le 400$, $0 \le V \le 50$, and $3 \le N_c \le 50$.

In addition, we assume that $w_c = g = d$ in our design, for the special case of equal comb finger width, gap, and spacing above the substrate. Some design variables are predefined: They are $w_{\rm ba} = 11$, $w_{\rm ca} = 14$, $\gamma = 4$, and $t = 2$, in which $w_{\rm ba}$ is the width of beam anchors, $w_{\rm ca}$ is the width of stator comb, and t is the thickness of the microresonator.

There are a number of design constraints to be considered for the comb-driven microresonator cell component, including both geometric and functional constraints. In this paper, without loss of generality, we consider the following 24 constraints:

$$g_1(x) : -(L_{\rm cy} + 2g + 2w_c) \le 0$$
$$g_2(x) : L_{\rm cy} + 2g + 2w_c - 700 \le 0$$
$$g_3(x) : -(L_{\rm sy} + 2L_b + 2w_t) \le 0$$
$$g_4(x) : L_{\rm sy} + 2L_b + 2w_t - 700 \le 0$$
$$g_5(x) : -(3L_t + w_{\rm sy} + 4L_c - 2x_0 + 2w_{\rm cy} + 2w_{\rm ca}) \le 0$$
$$g_6(x) : 3L_t + w_{\rm sy} + 4L_c - 2x_0 + 2w_{\rm cy} + 2w_{\rm ca} - 700 \le 0$$
$$g_7(x) : L_c - (x_0 + x_{\rm disp}) - 200 \le 0$$
$$g_8(x) : 4 - L_c + (x_0 + x_{\rm disp}) \le 0$$
$$g_9(x) : -((2N_c + 1)W_c + 2N_c g - L_{\rm cy}) \le 0$$
$$g_{10}(x) : (2N_c + 1)W_c + 2N_c g - L_{\rm cy} - 700 \le 0$$
$$g_{11}(x) : -(x_0 - x_{\rm disp} - 4) \le 0$$
$$g_{12}(x) : x_0 - x_{\rm disp} - 200 \le 0$$
$$g_{13}(x) : 4 - (L_t - x_{\rm disp} - (W_{\rm sy} + W_b)/2) \le 0$$
$$g_{14}(x) : L_t - x_{\rm disp} - (W_{\rm sy} + W_b)/2 - 200 \le 0$$
$$g_{15}(x) : 2 - (L_{\rm sy} - 2W_{\rm ba} - W_{\rm sa})/2 \le 0$$
$$g_{16}(x) : (L_{\rm sy} - 2W_{\rm ba} - W_{\rm sa})/2 - 200 \le 0$$
$$g_{17}(x) : 2 - x_{\rm disp} \le 0$$
$$g_{18}(x) : x_{\rm disp} - 100 \le 0$$
$$g_{19}(x) : 5 - Q \le 0$$
$$g_{20}(x) : Q - 1e5 \le 0$$
$$g_{21}(x) : -x_{\rm disp}/L_b \le 0$$
$$g_{22}(x) : x_{\rm disp}/L_b - 0.1 \le 0$$
$$g_{23}(x) : -K_{e,y}/K_y \le 0$$
$$g_{24}(x) : K_{e,y}/K_y - 1/3 \le 0.$$

Among them, the first sixteen are linear constraints, and the last eight are nonlinear constraints

$$x_{\rm disp} = QF_{e,x}/K_x \tag{20}$$

TABLE I
LIST OF PARAMETERS USED BY IDE-SR

symbol	Meaning of parameter	value
NP	Population size	100
p_f	A parameter used in stochastic ranking	0.45
μ	Mean value of the randomized scaling factor	1
σ	Standard deviation of the randomized scaling factor	0.25
p_{CR}	Crossover probability	0.8
γ	Size of the upper part of the population	0.7

where Q is quality factor and can be represented as

$$Q = \sqrt{m_x K_x / B_x^2}. \tag{21}$$

$F_{e,x}$ is the force generated by the comb drive. The force is proportional to the square of the voltage V applied across the comb fingers

$$F_{e,x} = 1.12\varepsilon_0 N_c \frac{t}{g} V^2 \tag{22}$$

where ε_0 is the permittivity of air.

We also have

$$K_x = \frac{2EtW_b^3}{L_b^3} \frac{L_t^2 + 14\alpha L_t L_b + 36\alpha^2 L_b^2}{4L_t^2 + 41\alpha L_t L_b + 36\alpha^2 L_b^2} \tag{23}$$

where

$$\alpha = (W_t/W_b)^3 \tag{24}$$
$$B_x = \mu \left[(A_s + 0.5A_t + 0.5A_b) \left(\frac{1}{d} + \frac{1}{\gamma} \right) + \frac{A_c}{g} \right] \tag{25}$$

where μ is the viscosity of air, and A_s, A_t, A_b, and A_c are the bloated layout areas of the shuttle, truss beams, flexure beams, and comb finger sidewalls, respectively.

Moreover, we know that

$$m_x = m_s + \frac{1}{4} m_t + \frac{12}{35} m_b \tag{26}$$

where $m_s = \rho A_s t$, $m_t = \rho A_t t$, and $m_b = \rho A_b t$

$$A_s = w_{\rm sa} L_{\rm sa} + 2w_{\rm sy} L_{\rm sy} \tag{27}$$
$$A_t = 2w_{\rm ca} L_{\rm cy} \tag{28}$$
$$A_b = 8L_b w_b + 2w_t(2L_t + w_a + 2w_b) \tag{29}$$
$$A_c = 2N_c w_c L_c. \tag{30}$$

The natural frequency ω_n is defined as

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{K_x}{m_x}}. \tag{31}$$

The design objective of comb-driven microresonator is to robustly match the natural frequency of the comb-driven microresonator with a predefined natural frequency.

TABLE II
RESULTS OF NONROBUST LAYOUT SYNTHESIS OF COMB-DRIVEN MICRORESONATOR

RUN NO.	1	2	3	4	5	6	7	8	9	10
$L_b(\mu m)$	91.71	153.09	164.34	126.63	123.39	170.73	168.48	129.78	104.76	163.35
$w_b(\mu m)$	4.05	5.22	8.46	5.67	4.68	8.82	11.88	5.4	4.59	5.31
$L_t(\mu m)$	42.84	44.37	29.7	62.37	90.36	47.7	40.32	29.61	59.4	46.89
$w_t(\mu m)$	4.32	8.91	15.39	14.58	16.38	5.31	15.12	12.33	9	11.7
$L_{sy}(\mu m)$	253.5	121.5	127.98	146.52	164.97	58.41	199.53	141.21	375.84	77.4
$w_{sy}(\mu m)$	18.81	17.1	18.63	12.15	16.38	24.48	44.37	18.09	10.62	19.08
$w_{sa}(\mu m)$	25.83	24.21	38.88	54.99	32.67	20.61	37.62	16.92	19.35	12.87
$w_{cy}(\mu m)$	62.64	78.21	17.37	96.57	13.95	48.6	21.87	28.98	38.25	56.88
$L_{cy}(\mu m)$	336.2	231.57	395.1	330.48	377.64	303.03	554.94	319.5	452.16	560.25
$L_c(\mu m)$	70.02	103.14	76.5	65.07	48.42	43.65	44.19	64.26	105.03	74.07
$w_c(\mu m)$	5.4	7.2	4.77	4.86	5.67	3.69	2.79	4.23	4.5	8.19
$L_{sa}(\mu m)$	71.1	103.14	351.9	147.33	16.83	123.57	224.82	135.09	66.42	82.08
$x_o(\mu m)$	19.26	65.79	28.44	35.01	34.65	33.57	34.2	27.81	79.2	24.03
$V(volt)$	47.96	46.18	43.25	44.95	42.92	44.56	44.34	47.66	47.48	46.57
N_C	31	22	56	51	38	51	93	43	58	21
$\left\|\frac{f_x - 200k}{200k}\right\|$	8.55e-4	5.99e-4	2.49e-4	3.30e-4	3.24e-4	3.85e-4	3.34e-4	1.57e-4	2.20e-4	1.29e-4
$D(\vec{x},\Omega)$	13.06e-3	7.992e-3	6.956e-3	9.370e-3	7.310e-3	6.368e-3	8.354e-3	8.028e-3	8.263e-3	3.215e-3

In other words, in this particular case study, the definition of $f(\vec{x},0)$ in (10) can be expressed as

$$f(\vec{x},0) = \omega_n(\vec{x},0) \tag{32}$$

where $\bar{f}$ can be predefined by users. In this paper, without loss of generality, $\bar{f} = 200$ kHz.

VI. EXPERIMENTS

As shown in (10) and (11), there are two design objectives to minimize in the robust design problem. The first objective relates to the design performance, while the second objective reflects the robustness of the design. To verify that involving the robustness consideration in the optimization process can help to reduce the sensitivity of the resulting designs to the variations of the design variables, we carry out a comparative study. In the first set of runs of IDE-SR, we only consider the first design objective, i.e., the performance objective. In the second set of runs of IDE-SR, we consider both performance and robustness objectives.

To verify that the performance of IDE-SR is competitive, a comparison study was also made among IDE-SR, standard DE, and two other state-of-the-art approaches in constrained evolutionary approaches.

A. Results of Nonrobust Layout Synthesis

In the first set of runs, we only considered the performance objective $f_{\rm obj} = N(\vec{x})$, as described in (10). Ten runs of experiments using IDE-SR algorithm were repeated with $\bar{f} = 200$ kHz. The parameters of the constrained genetic algorithm are listed in Table I.

The experimental data were obtained in Table II. It is noted that ten results represent ten different designs that all satisfy the design constraints and have natural frequencies very closely matching to the target $\bar{f} = 200$ kHz.

B. Results of Robust Layout Synthesis

For robust layout synthesis of the comb-driven microresonator, we need to consider both objectives in (10) and (11). To calculate the robustness index in (11), we need to know the variation vector according to (15). By examining the layout schematic of the comb-driven microresonator, we found that the variation vector can be set as follows:

$$\xi = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad -1 \quad 1 \quad 0 \quad 0].$$

According to (15), to obtain the robustness index in (11), we also need to make an assumption about σ. In this paper, we assume $\sigma = 0.1$ μm.

In the robust layout synthesis, we took the robustness index as the optimization objective $f_{\rm obj} = D(\vec{x},\Omega)$. In addition, another constraint $N(\vec{x}) \leq 1.0e^{-6}$ is added to the constraint list. Ten runs of experiments using IDE-SR algorithm were repeated, with the same parameters defined in Table I. The experimental data were listed in Table III. It can be seen from Tables II and III that the values of the objective $D(\vec{x},\Omega)$ are smaller in the case of robust designs than those in the case of nonrobust designs. The next section demonstrates that the reduced objective values of $D(\vec{x},\Omega)$ lead to more robust designs.

TABLE III
RESULTS OF ROBUST LAYOUT SYNTHESIS OF COMB-DRIVEN MICRORESONATOR

RUN NO.	1	2	3	4	5	6	7	8	9	10
$L_b(\mu m)$	305.73	323.37	316.44	329.76	294.66	274.77	320.4	306.63	299.97	290.88
$w_b(\mu m)$	15.75	20.07	16.56	19.80	19.62	17.10	20.07	17.82	15.84	15.57
$L_t(\mu m)$	198.36	201.78	134.91	190.17	199.71	181.71	201.33	188.37	133.38	121.59
$w_t(\mu m)$	14.49	2.07	15.48	2.07	2.07	9	2.07	2.07	15.21	14.85
$L_{sy}(\mu m)$	59.49	47.79	36.09	36.09	104.85	119.34	53.64	67.86	69.57	88.29
$w_{sy}(\mu m)$	18.27	10.08	202.95	42.30	16.47	70.47	10.53	50.4	60.57	112.32
$w_{sa}(\mu m)$	33.48	15.93	10.08	10.08	67.32	93.33	27.36	34.47	43.47	31.50
$w_{cy}(\mu m)$	10.44	10.08	12.51	11.43	10.08	10.08	10.17	10.08	47.16	15.12
$L_{cy}(\mu m)$	652.68	644.94	657.27	253.35	314.28	617.67	607.41	636.93	649.89	649.44
$L_c(\mu m)$	12.06	12.06	12.06	12.06	12.06	12.06	12.15	12.06	29.34	12.15
$w_c(\mu m)$	10.17	12.33	9.36	10.62	9.63	9.9	11.97	13.77	8.82	9
$L_{sa}(\mu m)$	398.07	399.51	382.59	29.07	104.85	20.34	239.94	27.99	244.08	96.03
$x_o(\mu m)$	6.03	6.03	6.03	6.03	6.03	6.03	6.12	6.03	21.51	6.03
$V(volt)$	50.00	50.00	50.00	50.00	50.00	50.00	49.99	50.00	49.99	49.99
N_C	33	27	36	22	26	33	27	24	38	37
$\left\|\frac{f_x - 200k}{200k}\right\|$	9.981e-4	9.997e-4	9.998e-4	9.888e-4	9.990e-4	9.943e-4	9.946e-4	1.0e-3	9.999e-4	9.999e-4
$D(\vec{x},\Omega)$	1.691e-3	9.277e-4	1.786e-3	2.011e-3	2.197e-3	2.201e-3	1.001e-3	9.682e-4	1.935e-3	2.096e-3

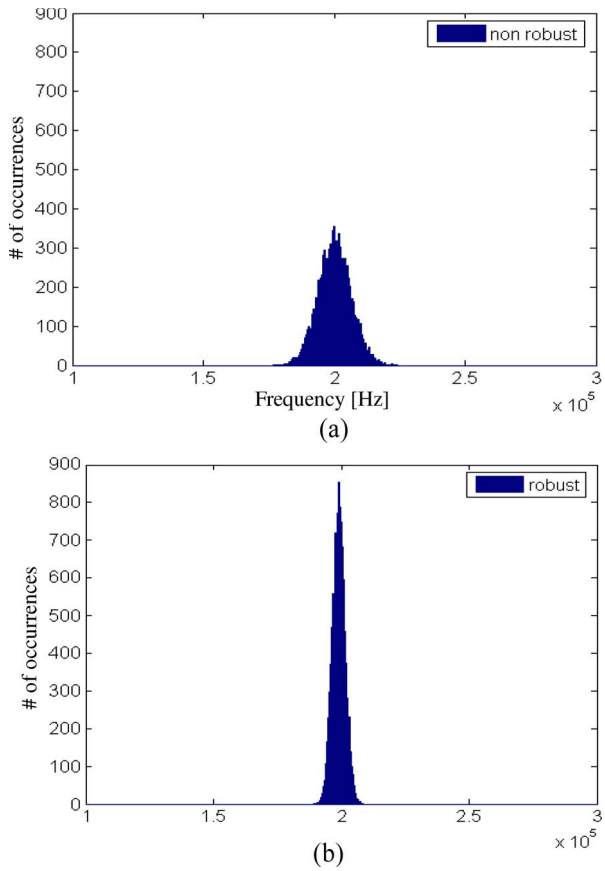


Fig. 7. (a) Histogram of natural frequencies of the nonrobust design of comb-driven microresonator subject to uncertainties. (b) Histogram of natural frequencies of the robust design of comb-driven microresonator subject to uncertainties.

C. Comparison of Robust and Nonrobust Results

It is noted that, in the robust design process, we minimized the robustness objective. To verify that doing this helps the resulting designs to increase their insensitivity to geometric uncertainties, we designed a comparative study as the following: We put two designs in one group for comparison by selecting one design from the robust design group and the other from the nonrobust design group. We then ran Monte Carlo simulations to model uncertain MEMS fabrication processes. We introduced the same variations to the design variables of both designs to emulate uniform overetch and/or underetch situations. To represent the variations in the process, we generated 10 000 Gaussian random vectors with a standard deviation σ of 0.1 μm. The natural frequencies of both the robust and nonrobust designs were calculated, and the histograms of them were plotted, as shown in Fig. 7.

According to Fig. 7, we can see that robust design has a much tighter distribution of natural frequencies and therefore is much less sensitive to geometric variations. The tests of other design candidates from both robust and nonrobust design groups revealed similar results. Figs. 8 and 9 drawn with SUGAR [30] show the layout of two exemplar nonrobust and robust designs, respectively.

D. Comparison of Different Optimization Algorithms

A comparison study was also made to compare the performance of IDE-SR with the standard DE and two other state-of-the-art evolutionary constraint optimization algorithms, which are improved stochastic ranking based evolutionary strategy (ISRES) [29] and nondominated sorting genetic algorithm

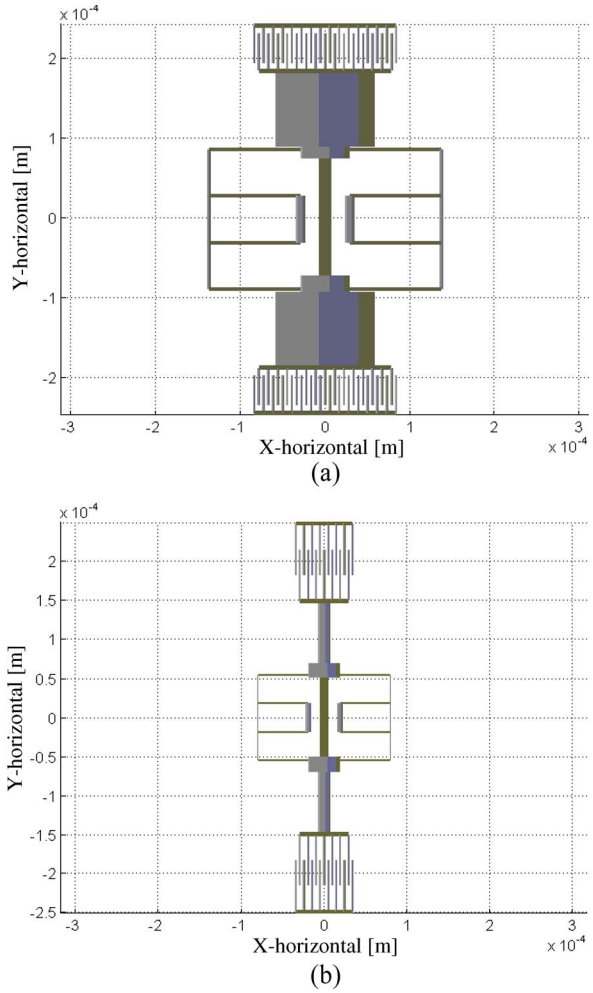


Fig. 8. (a) Layout of nonrobust solution I with a natural frequency of 200 kHz. (b) Layout of nonrobust solution II with a natural frequency of 200 kHz.

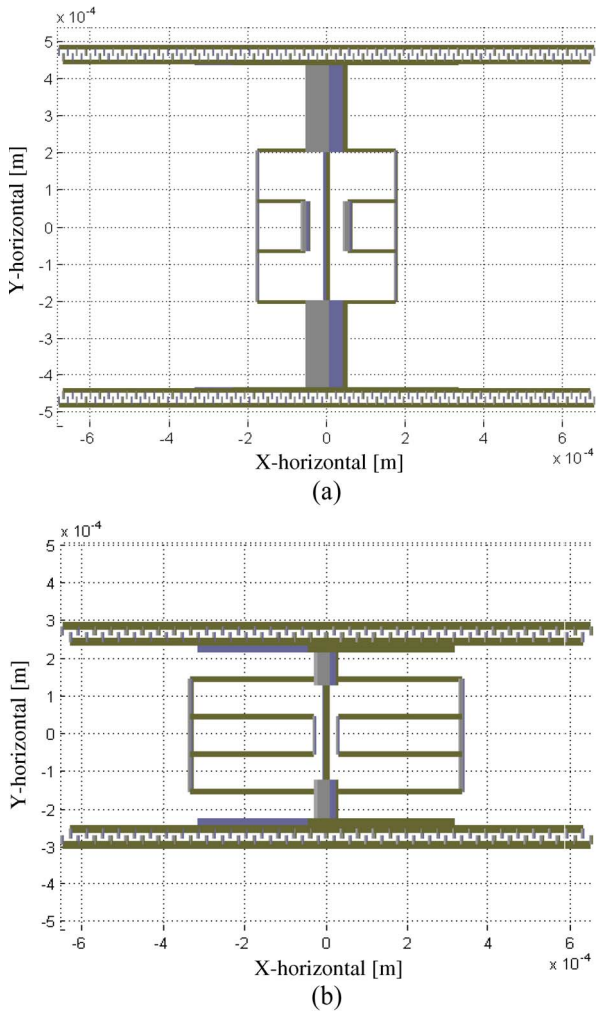


Fig. 9. (a) Layout of robust solution I with a natural frequency of 200 kHz. (b) Layout of robust solution II with a natural frequency of 200 kHz.

(NSGA)-II [31]. For each algorithm, 50 independent runs were carried out, with the best, mean, worst, and standard deviation of the obtained results that are all recorded in Table IV for comparison purposes. Bold values in Table IV indicate the best results among different algorithms.

It is clear from Table IV that IDE-SR outperforms DE, ISRES, and NSGA-II in terms of best, mean, and worst results. The most important criterion to be compared is the best result, because usually, we choose the design vector related to the best result to make the design. It is also important to note that DE, NSGA-II, and IDE-SR all performed very stably and could successfully find feasible solutions that satisfy all the constraints every time out of 50 independent runs. Due to their stochastic nature, evolutionary algorithms cannot guarantee convergence every time. However, the aforementioned three algorithms show very good consistency in this particular problem. ISRES, however, failed to do so in five times out of 50 independent runs. It is also notable that, if we use a population size of 100 in ISRES, it could not find a feasible solution. The reported results for ISRES were obtained with a population size of 200, which is double the population size used in other algorithms.

Fig. 10 shows the curves of objective values versus generation number recorded in one exemplar evolutionary process of both algorithms—IDE-SR and standard DE. It can be seen that IDE-SR has a stronger capability to find better objective values.

TABLE IV
COMPARING IDE-SR WITH OTHER ALGORITHMS

	DE	ISRES	NSGA-II	IDE-SR
best	1.741e-3	9.682e-4	1.001e-3	**9.277e-4**
mean	2.016e-3	1.691e-3	1.612e-3	**1.569e-3**
worst	2.455e-3	2.249e-3	2.940e-3	**1.823e-3**
SR	**50(50)**	45(50)	**50(50)**	**50(50)**
PS	**100**	200	**100**	**100**

SR: success rate; PS: population size

VII. CONCLUSION

Layout synthesis is an important stage for a structured design of MEMS [32], [33], after the stage of the system-level design [34]. This paper has developed a novel constrained optimization algorithm IDE-SR, which is an IDE based on SR, and reports a method of robust layout synthesis of MEMS based on it. The method transforms the robust design problem into a multiobjective constrained optimization problem and then solves it by using IDE-SR. Simulation results based on a case study of the layout synthesis of a comb-driven microresonator show that the

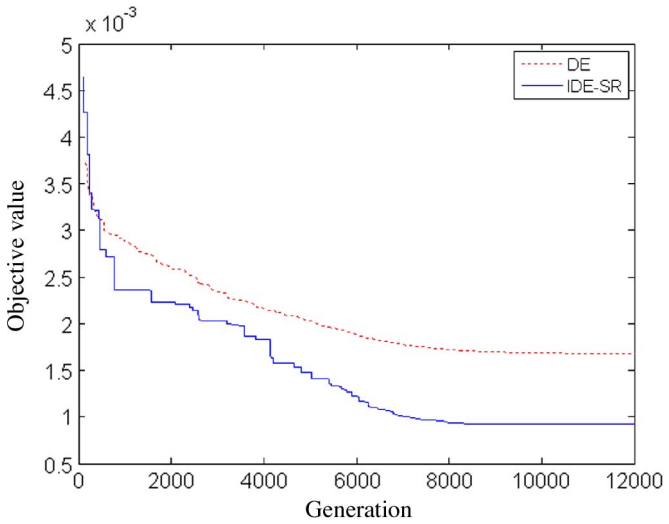


Fig. 10. Curves of objective value versus generation of both DE and IDE-SR.

design solutions obtained using the method proposed in this paper are much less sensitive to process-induced uncertainties. This paper has also shown that the IDE-SR algorithm cannot only obtain better results than the standard DE algorithm but also outperform some other state-of-the-art evolutionary constrained optimization algorithms.

The next step is to combine the detailed design described in this paper with system-level design so that a hierarchical design procedure can be automated. In addition, another work of testing the performance of IDE-SR on a comprehensive set of benchmark problems [35] is being conducted.

ACKNOWLEDGMENT

The authors would like to thank E. Goodman for his valuable comments and suggestions in revising and finalizing the manuscript. The authors would also like to thank Z. Cai and Y. Wang for their general discussions on the DE algorithm when they visited the Technical University of Denmark as Research Scholars during April–August 2007. The visit was gratefully sponsored by the Otto Mønsted Fund of Denmark.

REFERENCES

[1] R. Liu, B. Paden, and K. Turner, "MEMS resonators that are robust to process-induced feature width variations," *J. Microelectromech. Syst.*, vol. 11, no. 5, pp. 505–511, Oct. 2002.

[2] W. Yeh, S. Mukherjee, and N. C. MacDonald, "Optimal shape design of an electrostatic comb drive in microelectromechanical systems," *J. Microelectromech. Syst.*, vol. 7, no. 1, pp. 16–26, Mar. 1998.

[3] T. Mukherjee, S. Iyer, and G. K. Feeder, "Optimization-based synthesis of microresonators," *Sens. Actuators A, Phys.*, vol. 70, no. 1/2, pp. 118–127, 1998.

[4] G. Fedder, S. V. Iyer, and T. Mukherjee, "Automated optimal synthesis of microresonators," in *Proc. 9th Int. Conf. Solid State Sens. Actuators (TRANSDUCERS)*, Chicago, IL, Jun. 16–19, 1997, pp. 1109–1112.

[5] R. Kamalian, H. Takagi, and A. M. Agogino, "Optimized design of MEMS by evolutionary multi-objective optimization with interactive evolutionary computation," in *Proc. GECCO*, K. Deb *et al.*, Eds. Berlin, Germany: Springer-Verlag, 2004, vol. 3103, LNCS, pp. 1030–1041.

[6] L. Ma and E. K. Antonsson, "Automated mask-layout and process synthesis for MEMS," in *Proc. Int. Conf. Model. Simul. Microsyst.*, 2000, pp. 20–23.

[7] A. Ongkodjojo and F. E. H. Tay, "Global optimization and design for microelectromechanical systems devices based on simulated annealing," *J. Micromech. Microeng.*, vol. 12, no. 6, pp. 878–897, Nov. 2002.

[8] G. Fedder and T. Mukherjee, "Physical design for surface-micromachined MEMS," in *Proc. 5th ACM/SIGDA Phys. Des. Workshop*, Apr. 1996, pp. 53–60.

[9] Y. S. Hong, J. H. Lee, and S. H. Kim, "A laterally driven symmetric microresonator for gyroscopic applications," *J. Micromech. Microeng.*, vol. 10, no. 3, pp. 452–458, 2000.

[10] G. Taguchi, *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*. New York: ASME Press, 1993.

[11] H. Greiner, "Robust optical coating design with evolutionary strategies," *Appl. Opt.*, vol. 35, no. 28, pp. 5477–5483, Oct. 1996.

[12] D. Wiesmann, U. Hammel, and T. Back, "Robust design of multilayer optical coatings by means of evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 16–162, Nov. 1998.

[13] K. Hwang, K. Lee, G. Park, B. Lee, Y. Cho, and S. Lee, "Robust design of a vibratory gyroscope with an unbalanced inner torsion gimbal using axiomatic design," *J. Micromech. Microeng.*, vol. 13, no. 1, pp. 8–17, Jan. 2003.

[14] J.-Y. Park, Y.-D. Kim *et al.*, "Robust design of an active micro-mixer based on the Taguchi method," *Sens. Actuators B, Chem.*, vol. 129, no. 2, pp. 790–798, Feb. 2008.

[15] T.-S. Low, S. Chen, and X. Gao, "Robust torque optimization for BLDC spindle motors," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 656–663, Jun. 2001.

[16] Y. Jin and B. Sendhoff, "Trade-off between optimality and robustness: An evolutionary multiobjective approach," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optimization*, C. M. Fonseca *et al.*, Eds. New York: Springer-Verlag, 2003, vol. 2632, LNCS, pp. 237–251.

[17] P. J. Sedivec, "Robust optimization: Design in MEMS," M.S. thesis, Univ. California, Berkeley, CA, 2002.

[18] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[19] K.-S. Low and T.-S. Wong, "A multiobjective genetic algorithm for optimizing the performance of hard disk drive motion control system," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1716–1725, Jun. 2007.

[20] Y. Yu and Y. Xinjie, "Cooperative coevolutionary genetic algorithm for digital IIR filter design," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1311–1318, Jun. 2007.

[21] S. Tsutsui and A. Ghosh, "Genetic algorithms with a robust solution searching scheme," *IEEE Trans. Evol. Comput.*, vol. 1, no. 3, pp. 201–208, Sep. 1997.

[22] K. Deb and H. Gupta, "A constraint handling strategy for robust multicriterion optimization," Kanpur Genetic Algorithms Lab., Kanpur, India, KanGAL Rep. 2005001, 2005.

[23] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization," Kanpur Genetic Algorithms Lab., Kanpur, India, KanGAL Rep. 2004016, 2004.

[24] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[25] D. S. Lee, L. F. Gonzalez, K. Srinivas, and J. Periaux, "Robust evolutionary algorithms for UAV/UCAV aerodynamics and RCS design optimisation," *Comput. Fluids*, vol. 37, pp. 547–564, 2008.

[26] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in *Proc. CEC*. Piscataway, NJ: IEEE Service Center, May 2002, vol. 2, pp. 872–877.

[27] R. L. Becerra and C. A. C. Coello, "Cultured differential evolution for constrained optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 33–36, pp. 4303–4322, Jul. 2006.

[28] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2–4, pp. 311–338, Jun. 2000.

[29] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000.

[30] N. Zhou, J. V. Clark, and K. S. J. Pister, "Nodal simulation for MEMS design using SUGAR v0.5," in *Proc. Int. Conf. Model. Simul. Microsyst. Semicond., Sens. Actuators*, Santa Clara, CA, Apr. 6–8, 1998, pp. 308–313.

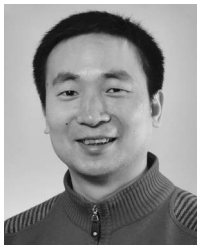
[31] K. Deb, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[32] G. Fedder, "A vision of structured CAD for MEMS," in *Proc. 5th ACM/SIGDA Phys. Des. Workshop*, Apr. 1996, pp. 76–80.

[33] T. Mukherjee and G. Fedder, "Structured design of microelectromechanical systems," in *Proc. 34th DAC*, Anaheim, CA, Jun. 9–13, 1997, pp. 680–685.

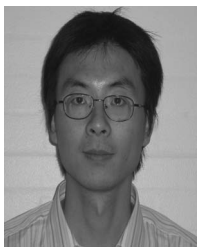
[34] Z. Fan, K. Seo, R. Rosenberg, J. Hu, and E. Goodman, "System-level synthesis of MEMS via genetic programming and bond graphs," in *Proc. Genetic Evol. Comput. Conf.* New York: Springer-Verlag, Jul. 2003, Lecture Notes in Computer Science, pp. 2058–2071. Chicago, IL.

[35] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006," in "Special Session on Constrained Real-Parameter Optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., Mar. 2006.



Zhun Fan (S'01–M'04) received the B.S. and M.S. degrees in control engineering from Huazhong University of Science and Technology, Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, in 2004.

From 2004 to 2007, he was an Assistant Professor with the Department of Mechanical Engineering, Technical University of Denmark, Lyngby, Denmark. He is currently an Associate Professor with the Department of Management Engineering, Technical University of Denmark. His research interests include MEMS, mechatronics, robotics, design automation and optimization, evolutionary computation, computational intelligence, computer vision, and intelligent transportation and power system.



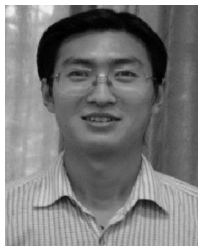
Jinchao Liu received the B.S. and M.S. degrees in control science and engineering from Wuhan University of Technology, Wuhan, China, in 2004 and 2007, respectively. He is currently working toward the Ph.D. degree in the Department of Management Engineering, Technical University of Denmark, Kongens Lyngby, Denmark.

He is also with FORCE Technology, Brøndby, Denmark. His main research interests include computer vision, image processing, welding automation, microelectromechanical systems, robotics control, and evolutionary computation.

Torben Sørensen received the M.S. and Ph.D. degrees in robotics and control engineering from the Technical University of Denmark (DTU), Kongens Lyngby, Denmark, in 1986 and 1996, respectively.

He was an Assistant Professor of control and engineering design with DTU from 1996 to 1998, where he was an Associate Professor in the Department of Mechanical Engineering from 2001 to 2007 and an Associate Professor in the Department of Management Engineering in 2008. He has participated as a Work Group Leader in three international scientific projects and authored or coauthored several scientific papers, as well as teaching materials including lecture notes, etc. His main research interests include robotics control, welding automation, image processing, and mechanical systems.

Dr. Sørensen was nominated for the title "Lecturer of the Year at DTU" in 2003. He was a committee member of the Danish Industrial Robot Association (in 2004–2008). He tragically passed away in 2008.



Pan Wang (S'01–M'08) received the B.S. degree in industrial automation from Wuhan University of Technology, Wuhan, China, in 1994, and the M.S. and Ph.D. degrees in systems engineering from Huazhong University of Science and Technology, Wuhan, in 1998 and 2003, respectively.

He is an Associate Professor and the Director of the Laboratory of Control and Decision, School of Automation, Wuhan University of Technology. He has published over 35 journal papers, three monographs, and 25 conference papers. His research interests include intelligent control and optimization, decision analysis, and biomedical intelligent information systems.

Dr. Wang has been the recipient of about ten academic or teaching awards in China.