# Case-Based Reasoning for Evolutionary MEMS Design

**Corie L. Cobb**
Palo Alto Research Center (PARC),
3333 Coyote Hill Road,
Palo Alto, CA 94304
e-mail: ccobb@parc.com

**Alice M. Agogino**
Department of Mechanical Engineering,
University of California at Berkeley,
6102 Etcheverry Hall,
Berkeley, CA 94720-1740
e-mail: agogino@berkeley.edu

*A knowledge-based computer-aided design tool for microelectromechanical systems (MEMS) design synthesis called case-based synthesis of MEMS (CaSyn-MEMS) has been developed. MEMS-based technologies have the potential to revolutionize many consumer products and to create new market opportunities in areas such as biotechnology, aerospace, and data communications. However, the commercialization of MEMS still faces many challenges due to a lack of efficient computer-aided design tools that can assist designers during the early conceptual phases of the design process. CaSyn-MEMS combines a case-based reasoning (CBR) algorithm and a MEMS case library with parametric optimization and a multi-objective genetic algorithm (MOGA) to synthesize new MEMS design topologies that meet or improve upon a designer's specifications. CBR is an artificial intelligence methodology that uses past design solutions and adapts them to solve current problems. Having the ability to draw upon past design knowledge is advantageous to MEMS designers, allowing reuse and modification of previously successful designs to accelerate the design process. To enable knowledge reuse, a hierarchical MEMS case library has been created. A reasoning algorithm retrieves cases with solved problems similar to the current design problem. Focusing on resonators as a case study, case retrieval demonstrated an 82% success rate. Using the retrieved cases, approximate design solutions were proposed by first adapting cases with parametric optimization, resulting in a 25% reduction in design area on average while bringing designs within 2% of the frequency goal. In situations where parametric optimization was not sufficient, a more radical design adaptation was performed through the use of MOGA. CBR provided MOGA with good starting points for optimization, allowing efficient convergence to higher quantities of Pareto optimal design concepts while reducing design area by up to 43% and meeting frequency goals within 5%.* [DOI: 10.1115/1.3462920]

## 1 Introduction

This paper presents a design synthesis tool, called case-based synthesis of microelectromechanical systems (CaSyn-MEMS), for early stage MEMS design. CaSyn-MEMS is a computer-aided design (CAD) tool that assists MEMS designers with concept development by utilizing past design structures to synthesize new structures. CaSyn-MEMS integrates case-based reasoning (CBR), a knowledge reasoning algorithm, with parametric optimization and a multi-objective genetic algorithm (MOGA). Using previously successful MEMS designs indexed in a hierarchical case library, CaSyn-MEMS synthesizes and optimizes new design structures that meet a designer's current set of design requirements. This paper will demonstrate how CBR can support design ideation during the initial conceptual phases of the design process while enabling large stochastic search methods, such as genetic algorithms (GAs), to converge to new promising design solutions.

MOGA algorithms developed previously [1–3] have proven successful in the design of resonant MEMS structures. Zhang et al. [1] noted that seeding MOGA with a good initial design is essential to helping MOGA converge to better design solutions in a practical number of evolutions. One shortfall is that they worked with the same initial design for all of their synthesis processes, limiting the range and quality of solutions their MOGA algorithm could generate. In addition, the burden of selecting a seed design was placed on the user of MOGA. What previous studies lacked was an efficient automated knowledge base, which removes from the human designer the burden of seeding the synthesis algorithm with good starting designs. The CBR tool developed in this paper benefits MEMS design by giving a wider range of good starting design cases for optimization and adaptation processes. The CBR starting cases increase the quantity and quality of optimal design solutions synthesized by evolutionary algorithms and enable convergence to a wider range of optimal design solutions.

**1.1 Introduction to MEMS.** MEMS are microscale electronic and mechanical components made with fabrication technology adapted from the field of integrated circuits. MEMS range from simple beams and electrostatic gaps to more complex sensors and actuators that include fluidic, magnetic, and thermal systems. MEMS can provide cost and size advantages as well as new functionality in products and different application areas.

MEMS design efforts date back to the late 1960s when Nathanson et al. at Westinghouse Research developed microscale resonant structures for filtering applications [4]. Today, micromechanical elements along with circuitry can be combined together on a common silicon substrate to create devices such as micro-accelerometers for deploying car air bags [5], microactuators for disk drives [6], micromirror arrays for projection display technology [7], and microfluidic devices for ink-jet print heads [8].

**1.2 Research Motivation.** Maseeh [9] surveyed MEMS companies and discovered that the length of the product development cycle was the most critical challenge to MEMS product commercialization. The median time to develop a prototype device was 3.2 years, with some taking as long as 8 years. In particular, it is not uncommon for MEMS designers to rely on a "trial-and-error" method during the initial stages of development. MEMS product development is an extremely costly and time-consuming process when compared with the average 14–27 month product development cycles of consumer products [10].

Efficient CAD tools can help MEMS designers foresee potential design problems and generate optimal design structures

Contributed by the AI/Knowledge Based Systems Committee of ASME for publication in the Journal of Computing Information Science and Engineering. Manuscript received July 9, 2009; final manuscript received February 12, 2010; published online September 1, 2010. Guest Editors: Ashok K. Goel and Andrés Gómez de Silva Garza.

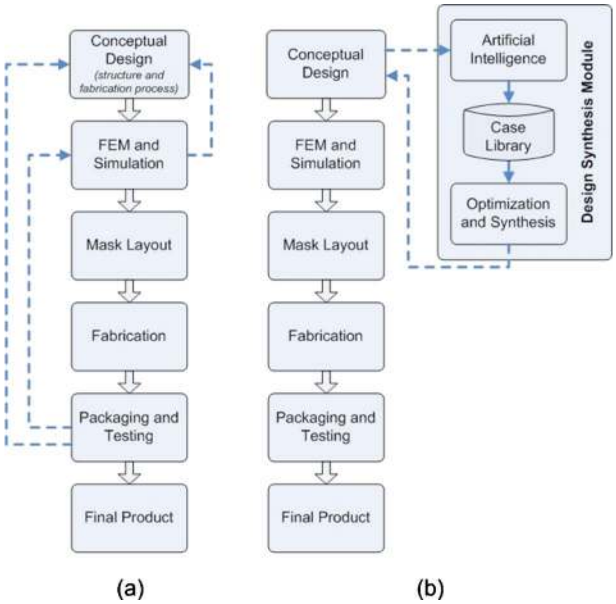


**Fig. 1** (*a*) **MEMS design process;** (*b*) **MEMS design process with case-based design synthesis**

quickly. This can help decrease the product development time for many MEMS devices by reducing time spent fabricating and testing suboptimal design concepts. Better design synthesis tools encourage more complex MEMS designs by enabling the reuse of various components, thereby reducing the need to redesign low-level design components, enabling designers to focus on larger structures and layouts. Knowledge-based CAD tools enable designers to weed out bad design ideas early on and iterate more with simulation than with costly manufacturing processes.

There are many variations in the MEMS product development process [11,12], but the general process can be explained by the diagram in Fig. 1(*a*). Once an initial design concept is conceived, time is spent iterating through mask layout and fabrication in order to produce the best performing design. Figure 1(*b*) shows how MEMS case-based design synthesis can change the current MEMS product development cycle by allowing designers to iterate more effectively through the conceptual design phases with a library of MEMS knowledge, artificial intelligence, and optimization tools in addition to quick, high-level simulation. Providing more structured design methods for MEMS can help improve the MEMS design process by significantly reducing the costly and time-consuming trial-and-error process for devices.

## 2 Related Work

**2.1 MEMS Simulation Tools.** Efficient CAD tools for MEMS have the potential to revolutionize MEMS design just as they revolutionized very large-scale integration (VLSI) circuit design with electronic design automation tools. VLSI circuit design grew tremendously through the use of CAD tools, which enabled designers to build complex circuit designs by drawing from basic building blocks and hierarchies of design elements [13,14]. VLSI simulation capabilities remain far more advanced than MEMS simulation capabilities when it comes to mixed-domain simulation.

MEMS CAD has matured to the point that there are now commercial CAD programs, such as CADENCE®, INTELLISUITE®, and ANSYS®, which offer MEMS designers preconfigured cell libraries with reusable components; however, there is little automatic reasoning in place for the user on how and when these components should be used. These simulation packages are not very helpful during exploratory phases of design as they require detailed modeling knowledge and can take hours or even days to analyze one design. There is a need for simulation and design tools that can enable faster concept generation during the initial stages of the design process. As more MEMS fabrication processes become standardized, MEMS designers can focus more on device design and layout with simulation and synthesis tools. CaSyn-MEMS aims to advance the MEMS field by utilizing successful MEMS design knowledge and building blocks as the analog to VLSI circuit design. CaSyn-MEMS can potentially reduce the cost and time of the MEMS product development cycle by helping MEMS designers iterate more during the initial conceptual phases of product development and reduce unnecessary design iterations during the testing and fabrication stages of development.

**2.2 MEMS Design Synthesis.** MEMS design synthesis involves the creation of new MEMS structures that meet a set of given design requirements. The majority of the research conducted to date on MEMS design synthesis has invoked some form of optimization, such as genetic algorithms, parametric optimization, or shape grammars. Design synthesis has been applied to all aspects of MEMS design, ranging from mechanical design and circuit design to mask layout and process development.

Zhou et al. [2] were the first to demonstrate that MOGA combined with SUGAR [15], an open-source MEMS simulation tool, can synthesize MEMS meandering resonators, produce new conceptual structures, and outperform human designers. Kamalian et al. [3,16] extended Zhou's work to more advanced MEMS problems and explored interactive evolutionary computation (IEC), integrating human expertise into the synthesis loop to leverage the strength of human expertise with computational efficiencies. Zhang et al. [1] implemented a hierarchical MEMS synthesis and optimization architecture, integrating an object-oriented data structure with SUGAR and two types of optimization: GAs and local gradient-based refinement. They noted that the MOGA approach needed a means for automating the starting populations for MOGA that would enable a larger sampling of the solution space of MEMS designs.

Mukherjee et al. [17] conducted work on MEMS synthesis for accelerometers using structured optimization methods. Their work focused on parametric optimization of a predefined MEMS topology. Because the configuration of their MEMS design had a fixed topology, it did not allow for a more radical design space exploration. Mukherjee and Fedder [18] also worked on a mixed-domain simulation, synthesis, and extraction methodology for MEMS.

Agarwal et al. [19] developed a 2D shape grammar for the synthesis of MEMS resonators. Shape grammars create designs by applying shape transformation rules continuously to an initial starting design. Agarwal et al. noted that MEMS designs have a strong form-function coupling, meaning that the slightest change in the topology of a design can drastically alter the design's performance. Wang et al. [20] approached MEMS synthesis by utilizing bond graphs (BGs) and genetic programming (GP) with a treelike structure of modeling building blocks to incorporate knowledge into the evolutionary process. Knowledge in Wang's synthesis system is formed as generalized system models rather than explicit cases of previously successful designs.

Li and Antonsson [21] applied GAs to the mask-layout aspect of MEMS synthesis. Ma and Antonsson [22] also used GAs for automated mask-layout synthesis but extended their work to include process synthesis for MEMS. Given a desired MEMS device topology and fabrication process, their tool could produce mask layouts and associated fabrication steps for a particular MEMS device. GAs were used to evolve an optimal mask layout given a user-defined shape. Li et al. [23] also concentrated on developing an automatic fabrication process planning for MEMS devices for the later stages of MEMS product development once a designer always has a defined device concept.

There is a lack of an efficient hierarchical library of MEMS designs and a means of retrieving the necessary designs for users embedded in these aforementioned MEMS design synthesis tools.

The burden of finding an appropriate initial design topology for a synthesis process is mostly left up to the user. The general lack of design tools for the initial stages of the MEMS design process and the need to select better starting designs for synthesis processes motivated our development of case-based reasoning capabilities for MEMS synthesis.

## 3 Case-Based Reasoning Background

CBR is an artificial intelligence method that utilizes knowledge from a past situation to help deal with new complex problems [24]. The beginnings of CBR can be traced back to the work of Shank and Abelson [25] in their introduction of "scripts" for representing knowledge about abstract problem situations. They proposed that these scripts or problem cases could be used to solve new problems by finding and appropriately modifying the closest matching item in memory. Shank's work produced a cognitive model upon which many CBR applications are based today.

The creation of a CBR system involves the indexing of information and the use of domain knowledge to improve search. CBR often works with a smaller hand-tailored collection of data where cases are structured according to domain-specific rules. Indexing and knowledge representation are the two initial stages of CBR, determining the ultimate performance of a CBR engine. The CBR cycle has many variations depending on the application, but at a high level, CBR can be summarized by the steps [26] below:

1. *Retrieve* the most similar cases from memory.
2. *Reuse* the retrieved cases and attempt to solve the problem at hand.
3. *Revise* the proposed solution (if necessary).
4. *Retain* the new solution in the case memory for future use.

**3.1 CBR Design Systems.** Human designers continuously draw upon their past experiences and expertise to solve design problems. It is with this realization that many CBR applications have been developed. CBR has been applied to various domains ranging from cooking recipes to the design of mechanical devices. Applying CBR to new design applications is a challenging task involving knowledge representation and management issues [27]. CYRUS, developed by Kolodner, was one of the first case-based reasoners and included processes for choosing indexes, reorganizing memory, making generalizations, and searching memory [24].

Other notable systems include CADET [28] and KRITIK [29]. CADET focused on the behavioral synthesis of devices with a dynamic or non-monotonic behavior. KRITIK generated both conceptual and qualitative designs for physical systems such as electrical circuits utilizing a structure-behavior-function (SBF) model that explicitly specified the structure and the functions of a device along with its internal casual behaviors to explain how the structure delivers the functions and how the device functions are composed from the functions of structural components. KRITIK2 expanded the functionality to a wider range of designs such as electromagnetic devices and operational amplifiers.

CADSYN [30] is a CBR system based on case decomposition and transformation knowledge to organize structural engineering cases for reuse. CLAVIER [31] is a CBR system used for determining loads of composite material parts for curing in an autoclave. CLAVIER was one of the first successful applications of CBR in industry. CLAVIER demonstrated that even with a small initial library of only 20 cases, of which some were incomplete, it was still found to be a useful tool by Lockheed employees. Tsai et al. [32] used CBR to develop a defect prediction system for new printed circuit board products. A vantage-based indexing scheme was developed to quickly retrieve good candidate cases by using clustering algorithms to partition similar cases into groups based on design specifications.

Boyle et al. created CAFIXD [33], a CBR system for fixture design. Boyle et al. focused on the indexing and case representation aspects of CBR with CAFIXD. CAFIXD developed two distinct case libraries: one case library for conceptual design and another for individual parametrized design solutions, with indices into the libraries based on axiomatic design decomposition. Tor et al. created a CBR system for a stamping die design using a relation graph representation scheme and a dual-step similarity retrieval strategy [34].

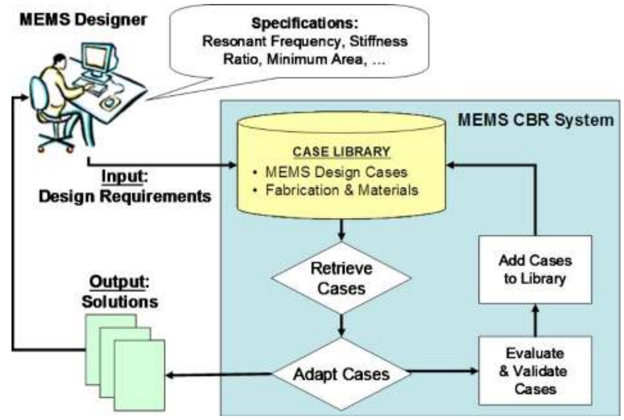


**Fig. 2 CaSyn-MEMS**

**3.2 CaSyn-MEMS Design Methodology.** Based upon past successes and lessons learned from the CBR design systems and MEMS synthesis applications, a methodology has been created for applying CBR to the design of MEMS. The overall goals of the CBR methodology for MEMS design are as follows:

1. Develop a hierarchical case library containing MEMS building blocks and devices at varying levels of complexity.
2. Create an organizational scheme and representation for cases that will enable an efficient and accurate retrieval of cases.
3. Implement a computationally inexpensive retrieval method that can focus the search space while returning the best designs possible.
4. Employ a method to effectively adapt selected cases to the current design situation and verify adaptation accuracy.

Previous work [35,36] documents items 1 and 2 as implemented in CaSyn-MEMS. Items 3 and 4 of the CaSyn-MEMS design methodology are the focus of this paper. Figure 2 illustrates the information flow for CaSyn-MEMS. A designer approaches the system and inputs their initial design specifications. The most relevant cases are retrieved from the case library using efficient retrieval algorithms. The case library contains MEMS components, building blocks, and preconfigured devices. Once cases are retrieved, they are adapted (if needed) to fit the current design problem using parametric optimization and GAs. Cases are initially evaluated with the MEMS simulation tool SUGAR. If new unique designs are synthesized by the system, the designs can be validated with fabrication and more advanced modeling techniques, such as finite element analysis (FEA), before being added to the case library to further expand the knowledge base. A small case library can expand and grow with use as newly synthesized design cases are stored in the library for future retrieval.

## 4 Case Retrieval

Case retrieval is the process of finding cases in the case library that best match a current design problem. Various approaches to case retrieval have been explored in research. The most commonly used methods are nearest neighbor, induction, and template retrieval [24,26]. These retrieval methods are often used alone or in combination with other strategies to form hybrid retrieval algorithms. CaSyn-MEMS employs a combination of retrieval methods, namely, a variation in nearest neighbor and inductive/structured query language (SQL)-template retrieval.

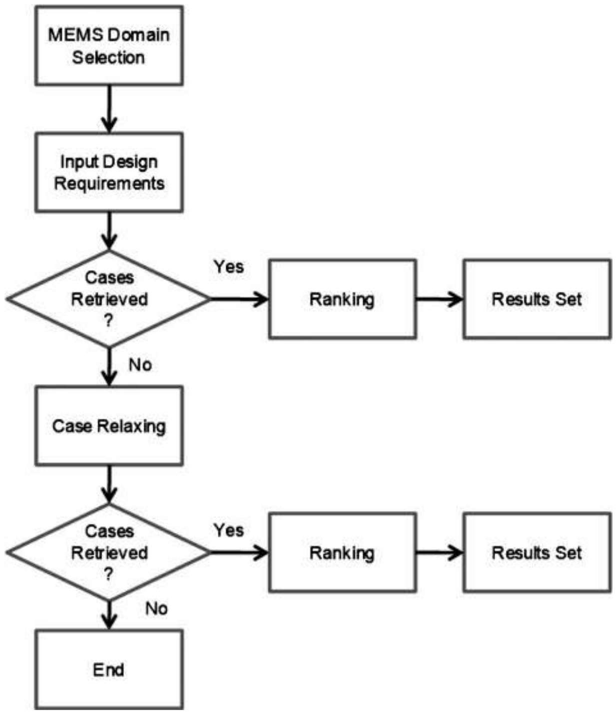


**Fig. 3 Case retrieval flowchart**

The nearest neighbor algorithm is used in CBR when computing the degree of similarity between items is desired and is one of the most popular approaches for case retrieval. This approach involves the assessment of similarity between stored cases and a new input case based on a weighted sum of features [24]. Each feature in a new problem is matched to the corresponding feature in stored cases. The degree of match is then computed for each feature of a case, and a total weighted score is calculated.

**4.1 CaSyn-MEMS Case Retrieval Methodology.** The foundation of CaSyn-MEMS is a case library of previously successful MEMS designs extracted from an extensive design documentation search [35]. The extracted designs were simulated and represented with the developed hierarchical function-structure case representation scheme [36] and were stored in a relational database to enable case retrieval. MEMS designs were also represented in a SUGAR, the back-end MEMS simulator used in CaSyn-MEMS, and MOGA-compatible data structure to enable more efficient simulation.

Because the MEMS case library is structured in a hierarchical manner, a template search is used first to focus on the most relevant section of the case library. The designer selects an application based on MEMS pre-indexed design domains (Fig. 3). Design requirements are entered, and a nearest neighbor similarity function is applied to weight and rank the matching cases retrieved from the initial search. Weights for the similarity search can be chosen by the user or set to a default where all design requirements carry equal weight. Cases that fall within $\pm 20\%$ of all the newly specified design requirements are considered good candidates for parametric optimization or future design adaptation with MOGA. The $\pm 20\%$ bounds are chosen in order to not place rigid bounds on design requirements, which can be improved upon with case adaptation algorithms.

In the event that design requirements are overconstrained and cannot pull a candidate case from the library, the retrieval algorithm relaxes secondary design requirements (such as area and stiffness for resonators) in increments of 10% and searches for cases with the frequency operation band as the most important criterion until cases are retrieved. Frequency requirements are not relaxed because a resonant MEMS device is not viable if it does not operate at its given frequency. If this search still yields no results, the user can create a new resonator instance based on components stored in the SUGAR component library.

**Table 1 Bounds on randomly generated resonator design requirements**

| Input design requirement | Bounds |
|---|---|
| Resonant frequency ($f_0$) | $2\ \text{kHz} \leq f_0 \leq 200\ \text{kHz}$ |
| Stiffness ratio ($K_x/K_y$) | $1 \leq K_x/K_y \leq 2500$ |
| Area | $6\times 10^{-8}\ \text{m}^2 \leq \text{Area} \leq 4\times 10^{-7}\ \text{m}^2$ |

After a group of candidate cases has been identified, a similarity measure is applied to compare the input and output cases. A ranking score between 0 and 1 is given to each case. Cases with a rank score closer to 1 are considered the best match with the input case. We will operate under the assumption that the closer two cases are in similarity, the better suited they are for the current design problem. The nearest neighbor variant used for ranking cases is presented in Eqs. (1)–(3) below. Assume that an input case ($C^I$) and a retrieved case ($C^R$) are each represented as vectors where $i$ is the $i$th feature of the case:

$$\text{Rank}_i = \frac{\sum_{i=1}^{n} w_i \times \text{sim}(C_i^I, C_i^R)}{\sum_{i=1}^{n} w_i} \tag{1}$$

where a conditional is applied for the similarity score,

$$\text{sim}(C_i^I, C_i^R) = \begin{cases} 1 - \dfrac{|C_i^I - C_i^R|}{\max|C_i^I - C_i^R|} & [2.1] \\ \dfrac{|C_i^I - C_i^R|}{\max|C_i^I - C_i^R|} & [2.2] \end{cases} \tag{2}$$

and

$$\sum_{i=1}^{n} w_i = 1 \tag{3}$$

[2.1] in Eq. (2) is the similarity measure used when CaSyn-MEMS is calculating rank for a case feature that is a double-sided constraint such as frequency. [2.2] in Eq. (2) is the similarity measure used when CaSyn-MEMS is concerned with a one-sided constraint such as minimum area. This dual similarity measure was implemented because there are many one-sided and two-sided constraints that exist for adapting MEMS designs.

**4.2 Case Retrieval Experiment.** Resonators were chosen to test the CaSyn-MEMS retrieval algorithm since they are currently the largest facet of the case library with 38 instances. To arrive at the resonator portion of the case library, a user must first select a "sensor" as the functional domain of their device and then indicate an electrical input-output device. From there, a fabrication method can be selected. Without loss of generality, the following retrieval examples will focus on designs fabricated with the MCNC poly-multi-user MEMS processes (MUMPS). This means the structural material layer of all designs will be polysilicon and 2 $\mu$m thick. Resonant frequency, stiffness ratio, and surface area are the global performance constraints, and the resonator area envelope is to be minimized.

Design requirement inputs to the system were randomly generated with MATLAB® in order to sample a wide range of possible design scenarios within the realm of the current cases represented. The values for the randomly generated requirements were bounded by the range of design requirements the case library currently covers (Table 1).

Cobb and Agogino [35] performed a retrieval experiment over the first generation case library design. A second retrieval experiment utilizing the same bounds (Table 1) was performed utilizing

Table 2 Resonator retrieval experiment results (* indicates designs only retrieved with the relaxing of input constraints)

| Test case | Input design requirements | | | Score of top ranked design | Number of cases retrieved (no constraint relaxing) | Number of cases retrieved (with constraint relaxing) |
|---|---|---|---|---|---|---|
| 1 | $f_0$=69.3 kHz | $K_x/K_y \geq 2320$ | Area $\leq 3.4\times10^{-7}$ m$^2$ | – | 0 | 0 |
| 2 | $f_0$=55.1 kHz | $K_x/K_y \geq 7$ | Area $\leq 2.3\times10^{-7}$ m$^2$ | 0.73 | 4 | – |
| 3 | $f_0$=123.5 kHz | $K_x/K_y \geq 207$ | Area $\leq 4.1\times10^{-7}$ m$^2$ | – | 1 | – |
| 4 | $f_0$=8.3 kHz | $K_x/K_y \geq 29$ | Area $\leq 3.7\times10^{-7}$ m$^2$ | – | 1 | – |
| 5 | $f_0$=27.3 kHz | $K_x/K_y \geq 32$ | Area $\leq 6.2\times10^{-8}$ m$^2$ | 0.41 | 0 | 2* |
| 6 | $f_0$=40.8 kHz | $K_x/K_y \geq 993$ | Area $\leq 3.6\times10^{-7}$ m$^2$ | 0.50 | 0 | 3* |
| 7 | $f_0$=24.9 kHz | $K_x/K_y \geq 8$ | Area $\leq 2.1\times10^{-7}$ m$^2$ | 0.87 | 8 | – |
| 8 | $f_0$=193.0 kHz | $K_x/K_y \geq 65$ | Area $\leq 3.6\times10^{-7}$ m$^2$ | – | 0 | 0 |
| 9 | $f_0$=23.4 kHz | $K_x/K_y \geq 76$ | Area $\leq 1.1\times10^{-7}$ m$^2$ | 0.72 | 4 | – |
| 10 | $f_0$=32.7 kHz | $K_x/K_y \geq 163$ | Area $\leq 2.7\times10^{-7}$ m$^2$ | 0.83 | 4 | – |
| 11 | $f_0$=53.4 kHz | $K_x/K_y \geq 121$ | Area $\leq 1.1\times10^{-7}$ m$^2$ | 0.55 | 3 | – |
| | | | | Overall success rate | 64% | 82% |

the second generation case library design, which contains more MEMS devices and building blocks. For the resonator retrieval experiment over the second generation case library, the same 11 sets of design requirements [35] were used (Table 2). Of the 11 sets, 4 initially returned no results (test cases 1, 5, 6, and 8). A 64% success rate was achieved for the first search. It is important to note that success rate in this instance refers to the reasoner retrieving at least one case, which meets the randomly generated design requirements. For the searches that returned zero results, CaSyn-MEMS incrementally relaxed the stiffness and area constraints by decreasing the stiffness ratio and increasing the device area while maintaining the same frequency constraint as before. After the stiffness and area constraints were relaxed, two of the four aforementioned unsuccessful retrieval test cases returned results (test cases 5 and 6), increasing the success rate to 82%.

For the purposes of case ranking in this experiment, all case features are assigned equal importance, but the system gives a designer the flexibility to assign different preference weights ($w_i$) to each case feature based on the needs of a current design application. As a result, this can change the ranking results for a given design application.

The results contained a diverse mix of resonator topologies, but often the folded flexure resonator (seen in test cases 1, 5, 6, 7, 9, 10, and 11) dominated the result set. This design dominated the case library (see design in Fig. 8(*c*)) because of its ability to achieve a high stiffness ratio in a small area. The folded flexure is a prime example of how engineering expertise from MEMS designers can be reused for new design applications.

## 5 Case Adaptation With Optimization

Once cases have been retrieved and ranked, the next crucial step is to adapt the retrieved cases to better meet the current design objectives. Typically, cases retrieved from the case library will not be an exact match with a current design problem; thus, retrieved cases will usually require design modification for a new problem. Case adaptation is the stage of the CBR process where designs are modified or synthesized to match a given set of design requirements. The retrieved cases are starting points for design adaption and exploration. Various approaches to case adaptation have been utilized by CBR researchers, including fuzzy logic, neural networks, Bayesian networks, rule-based approaches, case or subcase substitution, and optimization methods.

**5.1 Parametric Optimization.** MEMS structures, like many dynamical systems, experience strong function-structure coupling. Often, a slight adjustment to the length or width of a structure can drastically impact the performance of a design. For this reason, we found it beneficial to first explore parametric optimization as a case adaptation method. In instances where parametric optimization does not lead to the convergence of a solution, emergent methods, namely, GAs, are employed.

Parametric optimization works best in situations where a design starting point is close to a local minimum. The goal with this method is to minimize a given objective function (subjected to defined constraints) in order to drive the initial input problem closer to one's desired objectives. Parametric optimization requires a problem to be represented in parametric form for optimization. The decomposed MEMS designs from literature were instantiated in a parametrized format [35,36], enabling parametric optimization to more readily be applied to retrieved CBR designs.

In previous work, Cobb and Agogino [35] presented the results of an initial parametric optimization over retrieved resonator cases. This scheme involved minimizing area deviation while enforcing any deviation from the resonant frequency goal as a penalty factor. This initial test did not explicitly enforce a stiffness ratio constraint or contain weighted objectives. To extend this work, a new objective function for resonators was implemented and tested in CaSyn-MEMS, as presented in Eq. (4). The goal is to minimize the device area and the deviation from the frequency goal. If the stiffness ratio of a design falls below the stiffness ratio requirement, a penalty factor is added to the objective function. If the stiffness ratio requirement is met, then a penalty value of zero is used in the objective function. In addition, objective weights can be varied by the user, where $\alpha_1+\alpha_2=1$. For the purposes of this paper, $\alpha_1=\alpha_2$,

$$\begin{aligned}\text{obj}(\vec{x}) &= \alpha_1 \cdot \text{Area}(\vec{x}) + \alpha_2\left(\frac{\text{Freq}(\vec{x}) - \text{Freq}_{\text{goal}}}{\text{Freq}_{\text{goal}}}\right)^2 \\ &\quad + r_p \cdot \max\left(0, \frac{K_{x\ \text{goal}}}{K_{y\ \text{goal}}} - \frac{K_x(\vec{x})}{K_y(\vec{x})}\right)\end{aligned} \tag{4}$$

where $\text{obj}(\vec{x})$ is the objective function, $(\vec{x})$ is the vector of variables representing a case, $\alpha_1$ is the weight assigned to the area objective, $\alpha_2$ is the weight assigned to the frequency objective, $\text{Area}(\vec{x})$ is the design area as a function of geometrical parameters, $\text{Freq}(\vec{x})$ is the resonant frequency as a function of geometrical parameters, $K_x(\vec{x})/K_y(\vec{x})$ is the stiffness ratio as a function of geometrical parameters, and $r_p$ is the penalty coefficient on stiffness ratio violation.

The retrieved resonators from Table 2 were used as starting points for a parametric optimization process. Table 3 illustrates the results of parametric optimization for retrieved resonators from the resonator experiment. Each design that converged to a feasible solution was able to meet the stiffness ratio requirement. As a result, only the frequency and area results are compared. The

Table 3 Parametric optimization results. Frequency error and area improvement calculations are based on a comparison with the initial requirements from Table 2 (* indicates cases retrieved with design requirement relaxing).

| Test case | Total cases | No. of cases that converged to a solution | Frequency error before optimization (%) | Frequency error after optimization (%) | Area reduction below minimum required before optimization (%) | Area reduction below minimum required after optimization (%) |
|---|---|---|---|---|---|---|
| 2 | 4 | 4 | 3.1–17.1 | **0.04–0.4** | 65.1 | **66.8** |
| 3 | 1 | 1 | 8.0 | **2.0** | 77.5 | **78.8** |
| 4 | 1 | 1 | 16.4 | **0.7** | 29.3 | **21.7** |
| 5* | 2 | 2 | 8.3–11.0 | **0.04–0.07** | −62.4 | −70.4 |
| 6* | 3 | 0 | (N/A) | (N/A) | (N/A) | (N/A) |
| 7 | 8 | 8 | 0.1–19.1 | **0.0–0.08** | 51.2 | **45.9** |
| 9 | 4 | 4 | 2.9–14.2 | **0.09–0.09** | −0.9 | −1.8 |
| 10 | 4 | 4 | 7.3–17.4 | **0.09–0.09** | 57.8 | **55.9** |
| 11 | 3 | 3 | 5.0–14.4 | **1.7**–26.1 | 26.7 | **0.0** |

minimum area achieved for each test case before and after optimization is displayed along with the frequency error range. The optimization results that met or outperformed the initial design objectives are in bold font.

It is interesting to note that parametric optimization was able to converge to better solutions in 7 out of 11 of the original resonator test cases from Sec. 4. Test cases 1 and 8 retrieved zero cases as discussed previously, while test case 6 could not converge to a solution in a practical number of iterations. Upon closer examination, we hypothesize that this is due to the conflicting design requirements that are out of the range achievable by the retrieved design structure and the selected polysilicon material. Test case 6 requirements consist of an extremely high stiffness ratio, and the only designs in the case library capable of such a stiffness value are in a higher frequency band. Test case 11 converged to solutions that were worse in terms of minimum area and frequency error. Parametric optimization attempted to lengthen and widen the suspensions of the retrieved designs for test case 11 in order to meet the stiffness ratio requirement but lowered frequency and increased design area in the process.

Employing parametric optimization before a more computationally expensive optimization algorithm such as MOGA is a quick and efficient way to adapt a case for a new design problem. As more constraints are added to an optimization problem, we must deal with more tradeoffs in design performance. Previous work [35] showed that not enforcing the stiffness ratio enabled all retrieved resonator designs to converge to a feasible design solution. However, this section has shown that enforcing the stiffness ratio requirement as a penalty factor resulted in one case (test case 6) not converging to a feasible design solution. If more objectives such as cross-axis sensitivity and stress were added to the objective function, we expect fewer of the retrieved cases to converge to a feasible solution with parametric optimization.

**5.2 Case Adaptation With MOGA.** Evolutionary computation expands the capabilities of CBR by further adapting cases and synthesizing more conceptual MEMS designs when the CBR retrieved designs are not sufficient for the current problem. Parameter optimization may not always lead the designer to an acceptable solution after CBR; thus, it is advantageous to apply GAs for a more exploratory design search. GAs have been shown to be useful for CBR case adaptation [37,38]. MOGAs are a global stochastic search method based on the principles of evolution, allowing multiple objectives and constraints to be considered simultaneously. GAs were first introduced by Holland [39] to explain the adaptive processes of evolving natural systems and for creating new artificial systems in a similar way, but GAs became more popular when they were applied to search, optimization, and machine learning by Goldberg [40].

The MOGA algorithm used in this paper for case adaptation was initially developed by Zhang et al. [1]. The MOGA library in its initial form contained components ranging from lower level atomic building blocks, such as anchors and single beams, to some higher level clusters such as comb drives. We further expanded the building block data structure to accommodate additional design components used in the CaSyn-MEMS case library (such as enclosed frame masses, crab-leg suspensions, and folded flexures).

To begin the MOGA process (Fig. 4), the design objectives, constraints, stopping criteria, and an initial valid design are loaded into the MOGA adaptation module. Whereas selecting an initial valid design for MOGA was left up to the user of the tool in previous research, in this paper, the initial designs are selected by CBR. During the MOGA process, the module drastically mutates an initial design to create a population for the first generation. The addition of case-based designs to the MOGA library contains specialized mutation operations, which mutate component geometrical parameters or overall topology. After each generation of the MOGA process, the best designs, based on a Pareto ranking of all designs seen so far, are stored in a design archive to prevent losing good designs during the evolutionary process. At the end of the MOGA adaptation process, the designer is presented with multiple good design topologies from which to choose for further design exploration and analysis.

**5.3 MOGA Adaptation Results.** This section will demonstrate that combining valuable MEMS engineering knowledge from previously fabricated devices with MOGA can help generate new MEMS designs that may go beyond those conceptualized by

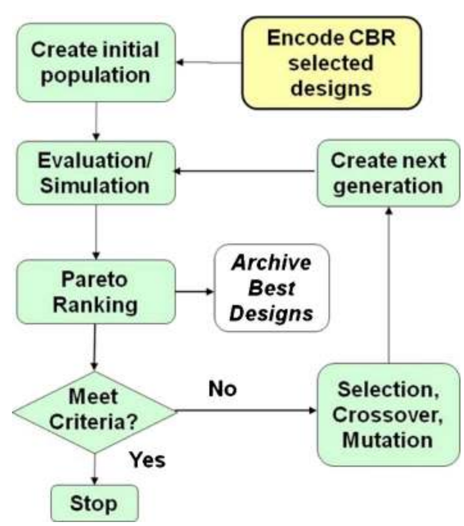


Fig. 4 MOGA process with initial population seeded by CBR

Table 4 **MOGA case adaptation results summary (* indicates test cases where design requirements were relaxed during case retrieval)**

| Test case | Number of CBR cases retrieved | Average number of solutions per symmetry constraint case | Total quantity of solutions generated | Performance of best design | | |
|---|---|---|---|---|---|---|
| 2 | 4 | 68 | 340 | Area $=7.78\times10^{-8}$ m$^2$ | $f_0=55.1$ kHz | $K_x/K_y=122.3$ |
| 4 | 1 | 58 | 175 | Area $=1.04\times10^{-7}$ m$^2$ | $f_0=8.3$ kHz | $K_x/K_y=46.6$ |
| 5* | 2 | 32 | 64 | Area $=9.83\times10^{-8}$ m$^2$ | $f_0=27.3$ kHz | $K_x/K_y=77.5$ |
| 6* | 3 | 23 | 45 | Area $=8.14\times10^{-8}$ m$^2$ | $f_0=40.7$ kHz | $K_x/K_y=10.4$ |
| 7 | 8 | 84 | 929 | Area $=9.38\times10^{-8}$ m$^2$ | $f_0=24.9$ kHz | $K_x/K_y=8.35$ |
| 9 | 4 | 69 | 275 | Area $=1.09\times10^{-7}$ m$^2$ | $f_0=23.5$ kHz | $K_x/K_y=75.7$ |
| 10 | 4 | 56 | 223 | Area $=1.04\times10^{-7}$ m$^2$ | $f_0=32.8$ kHz | $K_x/K_y=289.8$ |
| 11 | 3 | 36 | 71 | Area $=9.50\times10^{-8}$ m$^2$ | $f_0=53.2$ kHz | $K_x/K_y=225.1$ |

the designer. We will further explore what happens when past MEMS structures are used to generate new ones. The cases retrieved from the CBR experiment (Table 2) were evaluated for use as candidates in a MOGA process based on feedback from MEMS designers. Some of the CBR-generated design cases were determined to be suboptimal candidates for a MOGA synthesis process due to their relatively fixed topologies. The designs in Figs. 8(*a*) and 8(*b*) contain complicated connected graphed suspension layouts, which benefit more from local optimization than from MOGA optimization. All designs through parameter optimization achieved a resonant frequency within 0.04–2.00% of the frequency target while simultaneously minimizing area and meeting the stiffness ratio requirement.

For each MOGA synthesis run, a population of 400 for 50 generations was used. The synthesis metrics and geometry and symmetry constraints were based on Zhang et al.'s efficiency tests [1] and Kamalian et al.'s [3] research on performance, geometry constraints, and symmetry. Below is the different symmetry constraint cases used for the MOGA process for CBR-generated initial cases. The MOGA adaptation process focuses on the mutation of MEMS suspensions and not mass topologies as the suspensions have the largest impact on design performance. Each constraint case had five runs of the MOGA process in order to view a good range of generated design topologies:

- *no symmetry*—suspension blocks are free to mutate with no restrictions on angle orientation
- *y-axis symmetry*—suspension blocks are free to mutate, but the design is required to be symmetric about the *y*-axis
- *xy-axis symmetry*—suspension blocks are free to mutate, but the design is required to be symmetric about the *x*- and *y*-axis

Unlike typical four "leg" suspension MOGA cases (sample MEMS structures are provided in Fig. 8), the retrieved folded flexure suspensions (Fig. 8(*c*)) will only undergo an asymmetry constraint case for each MOGA synthesis run. If *y*-axis or *xy*-axis symmetry constraints were to be applied, the resonator would experience very little topology mutation and the results would closely match those that were presented during parametric optimization. The folded flexure suspension is already optimized in design due to human ingenuity. Thus, the most effective MOGA synthesis process is one that constrains the folded flexure designs the least in terms of symmetry, allowing us to view a wider range of unique topologies.

The results of the MOGA case adaptation process are summarized in Table 4. Due to the vast quantities of optimal design solutions generated, we will only highlight some of the best designs. All designs synthesized are within 5% of the specified frequency goal. Designs that fell on the Pareto frontier for frequency and area were deemed as some of the best MEMS designs.

Test cases 2 and 7 both retrieved resonator designs with an enclosed frame mass with crab-leg suspensions contained inside the mass. Although the crab-leg suspensions were free to rotate outside of the mass frame during the MOGA process, because the enclosed mass topology predetermines the minimum possible design area, many of the designs that had crab-leg suspensions located outside of the frame mass were not the best Pareto ranked designs since all designs with suspensions inside the frame had the minimum achievable area (see Fig. 8(*d*)).

Test cases 2 and 7 also retrieved folded flexure designs, and they differ by the size of their comb drives and folded flexure parameters (see Fig. 8(*c*)). In each folded flexure case instance, the stiffness ratio and area objectives were improved upon, while the resonant frequency was brought within 5% of the target goal. Figure 8(*h*) shows the best representative design for test case 7, while Fig. 8(*i*) shows the best design for test case 11, which also had a similar starting folded flexure design.

Test case 4 had only one design case retrieved from the CBR system, and that was a lateral resonator with a rectangular ring mass and serpentine springs. The asymmetric results for test case 4 performed well based on the frequency, stiffness, and minimum area objectives; however, these designs may no longer be viable if other design objectives are introduced into the process such as cross-axis sensitivity, stress, and rotational stiffness (the best design is provided in Fig. 8(*e*)). MEMS designers may be inclined toward picking more symmetric designs, while the asymmetric designs are somewhat unconventional and would require further analysis using a small batch fabrication run or FEA.

MOGA optimization is especially beneficial in test case 6 because parametric optimization methods were not able to converge to a feasible design solution, which simultaneously met all of the given design constraints. MOGA was able to generate multiple designs for this scenario due to the wider design space exploration MOGA offers over parametric optimization methods. In test case 6, there was an extremely high stiffness ratio requirement that was hard to achieve. The retrieved folded flexure design (see Fig. 8(*c*)) involves a tradeoff with frequency accuracy. These MOGA designs generated for test case 6 are still of value for the designer to observe, as they meet two of three design objectives (see Fig. 8(*g*)). In this instance, a designer would need to revaluate their design requirements to see if such a high stiffness ratio is necessary for a given design application.

With the exception of test case 5, the best minimum area achieved by each test case was better than that of the initial design. The first retrieved design in test case 5 had a best minimum area value that only matched that of the initial design, meaning that area was not improved, but it was not worsened during the MOGA process (see Fig. 8(*f*)).

Table 4 also shows a comparison of the quantity of Pareto optimal solutions generated for CBR selected designs, highlighting

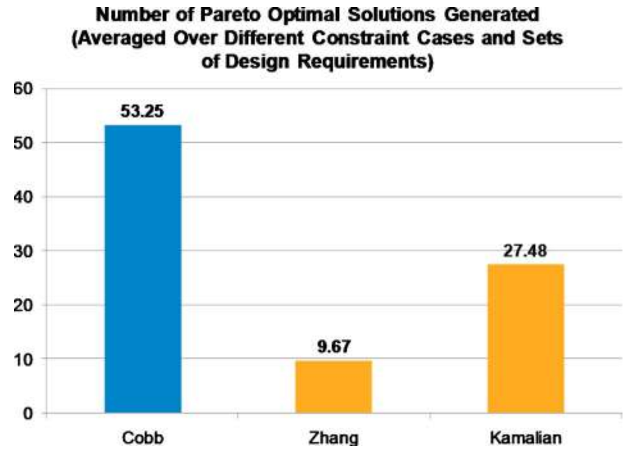


Fig. 5 Comparison with MOGA results generated by Kamalian [3] and Zhang [41]

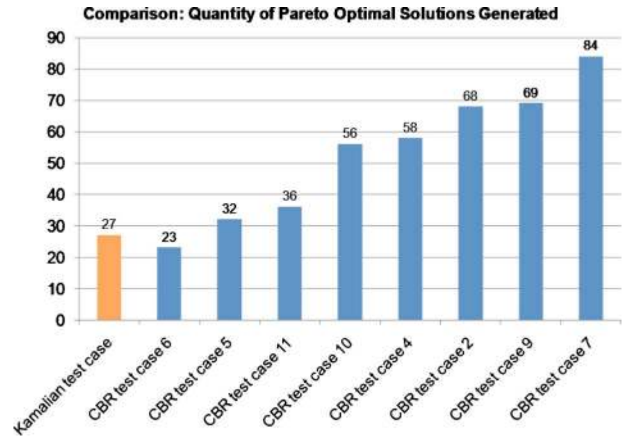


Fig. 6 Comparison of quantity of Pareto optimal solutions generated

the best design from each test case. One interesting point is the amazingly large quantity of Pareto optimal solutions generated. Whereas parametric optimization will only converge to a single optimal solution based on one starting point, MOGA can converge to a multitude of new optimized design topologies with just one initial design. The quantity and quality of solutions presented can give MEMS designers a strong range of design options to pursue during the conceptual phases of the design process. MOGA generated many radical new design topologies that go beyond what MEMS designers may initially conceptualize on their own.

**5.4 MOGA Results Comparison.** To better understand the impact of the results presented thus far, a comparison of the quantity of Pareto optimal solutions generated and the resultant minimum area is made with previous test cases generated by Kamalian et al. [3] and Zhang [41]. For all of their resonator MOGA experiments, Zhang and Kamalian both utilized the same resonator structure. The starting design employed a special mutation functionality that allowed MEMS suspensions to freely mutate not only in length, width, and angle, but also in the number of beams used to create each suspension, forming interesting spider-like suspension layouts in many instances. Though Zhang and Kamalian's design was deemed a reasonable starting point, the CBR reasoning system yielded a variety of stronger starting designs for the MOGA process by incorporating and reusing MEMS design knowledge in the form of successful past MEMS design cases.

Focusing on the quantity of solutions, Zhang generated an average of ten designs for each symmetry constraint case used. A similar average was taken for a test case Kamalian used with the same initial design; however, only one set of design requirements was utilized ($f_0=10$ kHz, $K_x \geq K_y$, and minimize device area). Kamalian garnered an average of 27 solutions. If we take the average for the results across all of the different CBR design requirement test cases (Table 4), CBR starting designs generate 53 solutions on average; the CBR number is almost double of what Kamalian was able to generate with his starting MEMS design and about six times more than what Zhang was able to generate (see Fig. 5).

It is important to note, however, that Zhang's results are lower due to the fact that her process was run for shorter periods of time and only for *xy*-axis symmetry constraints. Due to these limitations on her data set, it is not surprising that Zhang's quantity of Pareto optimal design solutions has the lowest value. Kamalian, however, ran his MOGA process for the longest period of time (ten runs with 500 generations over a population of 400). The MOGA designs presented in our work resulted from five runs of the MOGA process for only 50 generations over a population of 400 and employed asymmetry, *y*-axis symmetry, and *xy*-axis symmetry constraints, similar to Kamalian. Figure 6 illustrates a relative comparison of the quantity of Pareto optimal solutions gen-

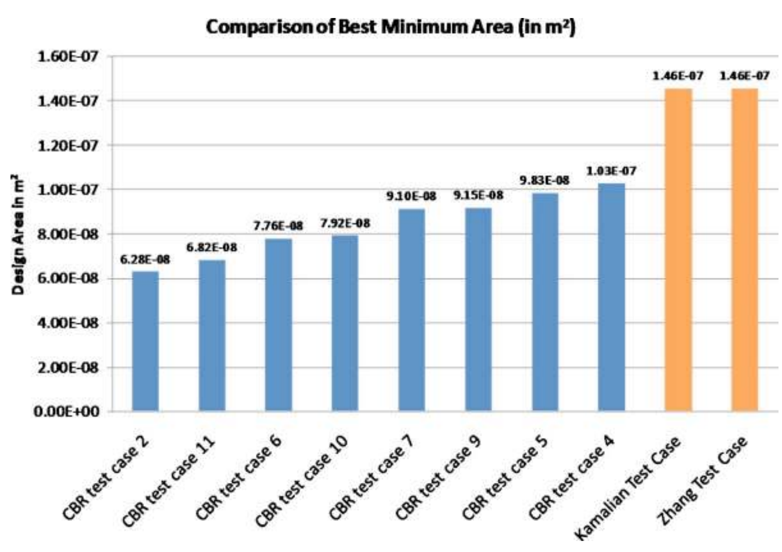


Fig. 7 Comparison of best minimum area

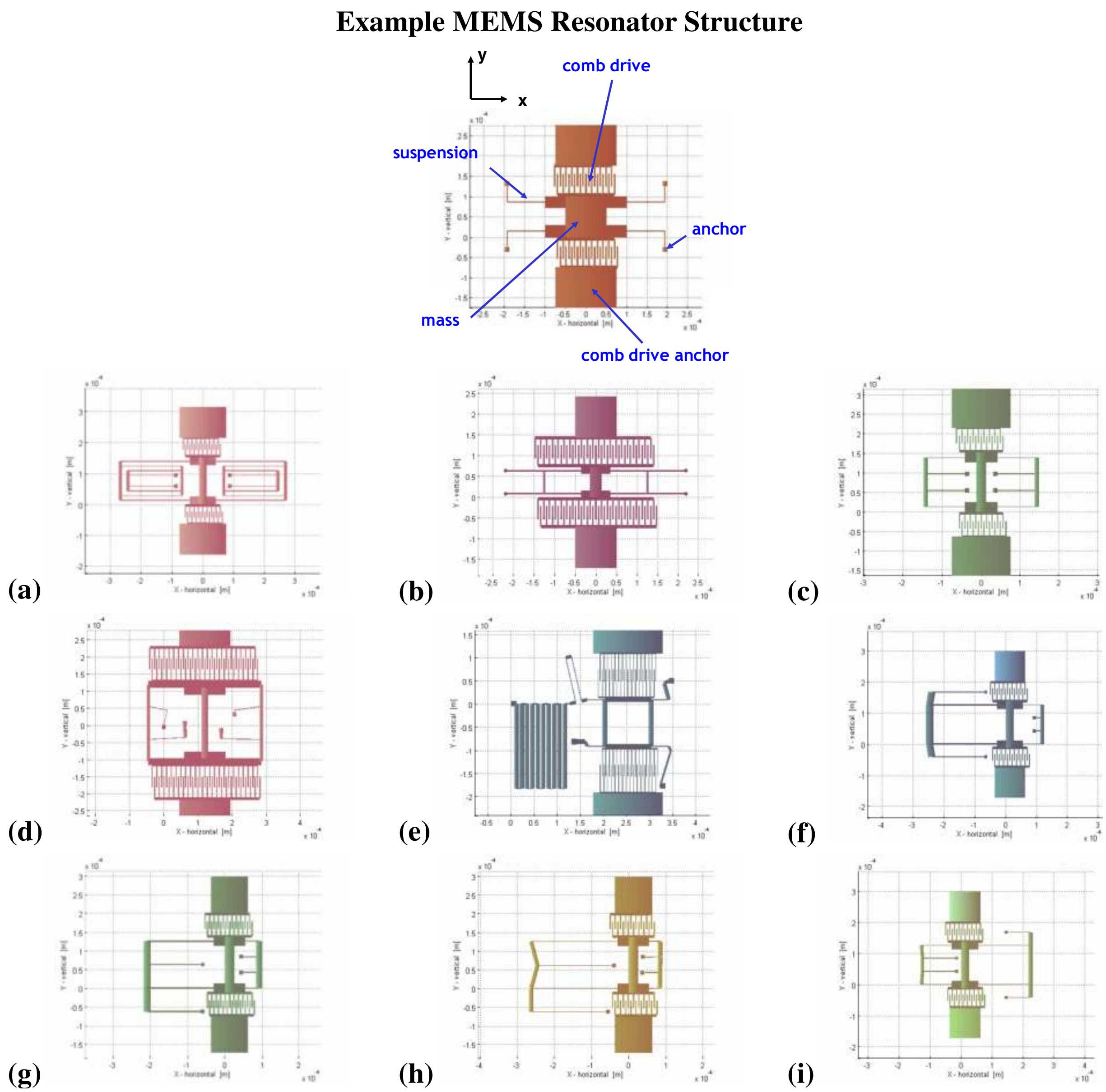


**Fig. 8 CaSyn-MEMS resonator designs**

erated by different sets of design requirement test cases.

It is interesting to note that only one CBR test case underperforms Kamalian's test case. CBR test case 6, which was the poorest performing CBR test case due to the fact that design requirement relaxing was necessary during case retrieval, makes the CBR initial designs for this test case weaker than those shown in the other test cases. Test case 5 also required design constraint relaxing but was still able to generate multiple Pareto optimal solutions with MOGA.

Because we are concerned with device area minimization in our design problem, a comparison of the best minimum area achieved (Fig. 7) is also examined. Again, CBR starting designs help outperform Kamalian's and Zhang's MOGA results. All of the CBR test cases had a best minimum design area that beat the best minimum design area generated by Kamalian and Zhang's starting design. This result is due to the fact that the CBR initial designs varied in their mass, comb drive, and suspension topologies, allowing MOGA to explore a new space for area minimization.

## 6 Conclusions

A new MEMS design synthesis process, integrating CBR with parametric optimization and a multi-objective genetic algorithm (MOGA) for case adaptation, was introduced and evaluated. MEMS design cases retrieved during a CBR retrieval experiment were evaluated for their potential as initial designs for both a parametric optimization and MOGA adaptation in order to further generate design concepts for MEMS designers (see Fig. 8).

Focusing on case retrieval, the functional similarity retrieval method over the MEMS resonator test cases benefitted from the relaxing of design constraints, demonstrating an 82% success rate for resonators. As a comparison benchmark, CLAVIER [31], a CBR system for autoclave loading, began with a case library of only 20 designs and demonstrated an initial retrieval success rate of only 30%. However, with only the initial 30% retrieval, users still found CLAVIER to be helpful in their engineering work. The resonator retrieval rates presented here exceed CLAVIER's values and will only continue to grow as new cases are added to the knowledge base of the system.

Parametric optimization for case adaptation generated optimal solutions for 7 out of 11 (64%) of the resonator retrieval test cases. The lesson learned from the parametric optimization analysis is that designs retrieved from the case library with the relaxing of design constraints are less likely to converge to better performing design solutions when conflicting constraints and objectives are added to the optimization problem. Designs successfully adapted with parametric optimization are immediately good candidates for a design application. They have a predetermined topology and are exemplars of MEMS design taken from literature, having been previously fabricated by their respective authors and optimized within feasible fabrication bounds.

MOGA case adaptation synthesized new radical layouts that met or beat those generated by parametric optimization. In addition, for every initial design case, parametric optimization only generates one optimal design concept. However, MOGA generates a Pareto frontier of optimal results. One CBR case can generate 23–84 Pareto optimal solutions on average while improving upon the minimum design area, exceeding previous results [3,41]. The new designs generated by combining valuable engineering knowledge in the form of CBR and MOGA show promise for MEMS design applications. The quantity and quality of Pareto optimal designs presented by MOGA adaptation can help MEMS designers ideate more during the early phases of the design process and increase the likelihood of success in the later stages.

## Acknowledgment

This work was supported, in part, by NSF Grant No. CCR-DES/CC-0306557 and a Bell Labs Graduate Research Fellowship.

## References

[1] Zhang, Y., Kamalian, R., Agogino, A. M., and Séquin, C. H., 2006, "Design Synthesis of Microelectromechanical Systems Using Genetic Algorithms With Component-Based Genotype Representation," *Proceedings of the Eighth Annual Conference on Genetic and Evolutionary Computation*, ACM, New York, pp. 731–738.
[2] Zhou, N., Zhu, B., Agogino, A. M., and Pister, K. S. J., 2001, "Evolutionary Synthesis of Microelectromechanical Systems (MEMS) Design," *Proceedings of ANNIE 2001: Intelligent Engineering System through Artificial Networks*, ASME, New York, Vol. 11, pp. 197–202.
[3] Kamalian, R., Agogino, A. M., and Takagi, H., 2004, "The Role of Constraints and Human Interaction in Evolving MEMS Designs: Microresonator Case Study," *Proceedings of the 2004 ASME International Design Engineering Technical Conferences*, Paper No. DETC2004-57462.
[4] Nathanson, H. C., Newell, W. E., Wickstrom, R. A., and Davis, J. R., 1967, "The Resonant Gate Transistor," IEEE Trans. Electron Devices, **14**(3), pp. 117–133.
[5] Dixon, R. H., and Bouchaud, J., 2006, "Markets and Applications for MEMS Inertial Sensors," *Proceedings of SPIE: MEMS/MOEMS Components and Their Applications III*, Vol. 6113, Paper No. 611306.
[6] Imamura, T., Katayama, M., Ikegawa, Y., Ohwe, T., Koishi, R., and Koshikawa, T., 1998, "MEMS-Based Integrated Head/Actuator/Slider for Hard Disk Drives," IEEE/ASME Trans. Mechatron., **3**(3), pp. 166–174.
[7] Van Kessel, P. F., Hornbeck, L. J., Meier, R. E., and Douglass, M. R., 1998, "A MEMS-Based Projection Display," Proc. IEEE, **86**(8), pp. 1687–1704.
[8] Kamisuki, S., Hagata, T., Tezuka, C., Nose, Y., Fujii, M., and Atobe, M., 1998, "A Low Power, Small, Electrostatically-Driven Commercial Inkjet Head," *Proceedings of the 11th Annual International Workshop on Micro Electro Mechanical Systems*, pp. 63–68.
[9] Maseeh, F., 2000, *Reducing MEMS Product Development and Commercialization Time*, Future Fab International, Mill Valley, CA, Vol. 8.
[10] Minahan, T. A., 2004, "Product Development in Consumer Industries Benchmark Report: Leading the Next Wave of Growth," Aberdeen Group.
[11] Antonsson, E. K., ed., 1995, "Structured Design Methods for MEMS–Final Report," NSF Sponsored Workshop on Structured Design Methods for MEMS, pp. 63–68.
[12] Fedder, G. K., 1999, "Structured Design of Integrated MEMS," *Proceedings of the 12th IEEE International Conference on MicroElectroMechanical Systems*, pp. 1–8.
[13] Koh, H. Y., Séquin, C. H., and Gray, P. R., 1990, "OPASYN: A Compiler for CMOS Operational Amplifiers," IEEE Trans. Comput.-Aided Des., **9**(2), pp. 113–125.
[14] Stone, D. C., Schroeder, J. E., and Smith, A. R., 1984, "Analog CMOS Building Blocks for Custom and Semicustom Applications," IEEE J. Solid-State Circuits, **19**(1), pp. 55–61.
[15] Clark, J. V., Zhou, N., Bindel, D., Schenato, L., Wu, W., Demmel, J., and Pister, K. S. J., 2000, "3D MEMS Simulation Modeling Using Modified Nodal Analysis," *Proceedings of the 2000 Microscale Systems: Mechanics and Measurements Symposium*, pp. 68–75.
[16] Kamalian, R., Zhang, Y., Agogino, A. M., and Takagi, H., 2006, "Evolutionary Synthesis of Micromachines Using Supervisory Multiobjective Interactive Evolutionary Computation," *Advances in Machine Learning and Cybernetics*, Springer, Berlin, Vol. 3930, pp. 428–437.
[17] Mukherjee, T., Zhou, Y., and Fedder, G., 1999, "Automated Optimal Synthesis of Microaccelerometers," *Proceedings of the 12th IEEE International Conference on MicroElectroMechanical Systems*, pp. 326–331.
[18] Mukherjee, T., and Fedder, G. K., 1999, "Hierarchical Mixed-Domain Circuit Simulation, Synthesis and Extraction Methodology for MEMS," J. VLSI Signal Proc. Syst. Signal, Image, Video Technol., **21**(3), pp. 233–249.
[19] Agarwal, M., Cagan, J., and Stiny, G., 2000, "A Micro Language: Generating MEMS Resonators by Using a Coupled Form-Function Shape Grammar," Environ. Plan. B: Plan. Des., **27**(4), pp. 615–626.
[20] Wang, J., Fan, Z., Terpenny, J. P., and Goodman, E. D., 2005, "Knowledge Interaction With Genetic Programming in Mechatronics Systems Design Using Bond Graphs," IEEE Trans. Syst. Man Cybern., Part C Appl. Rev., **35**(2), pp. 172–182.
[21] Li, H., and Antonsson, E. K., 1998, "Evolutionary Techniques in MEMS Synthesis," *Proceedings of the 1998 ASME Design Engineering Technical Conferences*, Paper No. DETC98/MECH-5840.
[22] Ma, L., and Antonsson, E. K., 2000, "Automated Mask-Layout and Process Synthesis for MEMS," *Proceedings of the International Conference on Modeling and Simulation of Microsystems*, pp. 20–23.
[23] Li, J., Gao, S., and Liu, Y., 2007, "Solid-Based CAPP for Surface Micromachined MEMS Devices," Comput.-Aided Des., **39**(3), pp. 190–201.
[24] Kolodner, J., 1993, *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, CA.
[25] Shank, R., and Abelson, R., 1997, *Scripts, Plans, Goals and Understanding: An Inquiry Into Human Knowledge Structures*, Lawrence Erlbaum, Hillsdale, NJ.
[26] Watson, I., and Marir, F., 1994, "Case-Based Reasoning: A Review," Knowl. Eng. Rev., **9**(4), pp. 327–354.
[27] Maher, M. L., and Gómez de Silva Garza, A., 1997, "Case-Based Reasoning in Design," IEEE Expert, **12**(2), pp. 34–41.
[28] Narasimhan, S., Sycara, K. P., and Navin-Chandra, D., 1997, "Representation and Synthesis of Non-Monotonic Mechanical Devices," *Issues and Applications of Case-Based Reasoning in Design*, M. Maher and P. Pu, eds., Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 187–222.
[29] Goel, A. K., Bhatta, S., and Stroulia, E., 1997, "Kritik: An Early Case-Based Design System," *Issues and Applications of Case-Based Reasoning in Design*, M. Maher and P. Pu, eds., Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 87–132.
[30] Maher, M. L., and Gomez de Silva Garza, A., 1996, "Developing Case-Based Reasoning for Structural Design," IEEE Expert, **11**(3), pp. 42–52.
[31] Hennessy, D., and Hinkle, D., 1992, "Applying Case-Based Reasoning to Autoclave Loading," IEEE Expert, **7**(5), pp. 21–26.
[32] Tsai, C.-Y., Chiu, C.-C., and Chen, J.-S., 2005, "A Case-Based Reasoning System for PCB Defect Prediction," Expert Sys. Applic., **28**(4), pp. 813–822.
[33] Boyle, I. M., Rong, K., and Brown, D. C., 2006, "CAFixD: A Case-Based Reasoning Fixture Design Method. Framework and Indexing Mechanisms," ASME J. Comput. Inf. Sci. Eng., **6**(1), pp. 40–48.
[34] Tor, S. B., Britton, G. A., and Zhang, W. Y., 2003, "Indexing and Retrieval in Metal Stamping Die Design Using Case-Based Reasoning," ASME J. Comput. Inf. Sci. Eng., **3**(4), pp. 353–362.
[35] Cobb, C. L., and Agogino, A. M., 2006, "Case-Based Reasoning for the Design of Micro-Electro-Mechanical Systems," *Proceedings of the 2006 ASME International Design Engineering Technical Conferences*, Paper No. DETC2006-99120.
[36] Cobb, C. L., Zhang, Y., Agogino, A. M., and Mangold, J., 2008, "Knowledge-Based Evolutionary Linkage in MEMS Design Synthesis," *Linkage in Evolutionary Computation*, Y.-P. Chen and M.-H. Lim, eds., Springer, Berlin/Heidelberg, Vol. 157, pp. 461–483.
[37] Gómez de Silva Garza, A., and Maher, M. L., 1999, "An Evolutionary Approach to Case Adaptation," *Lecture Notes in Artificial Intelligence*, K.-D. Althoff, R. Bergmann, and L. K. Branting, eds., Vol. 1650, pp. 162–173.
[38] Domer, B., and Smith, I. F. C., 2005, "An Active Structure That Learns," J. Comput. Civ. Eng., **19**(1), pp. 16–24.
[39] Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI.
[40] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley/Longman, Boston, MA.
[41] Zhang, Y., and Agogino, A. M., 2010, "Interactive Hybrid Evolutionary Computation for MEMS Design Synthesis," *Advances in Neural Network Research and Applications*, Z. Zeng and J. Wang, eds., Springer, Berlin/Heidelberg, Vol. 67, pp. 211–218.