A COMPARISON OF MEMS SYNTHESIS TECHNIQUES

R. Kamalian[1], N. Zhou[1], and A. M. Agogino[1]

[1]*Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA*
email: raffi@me.berkeley.edu

Abstract

Recent results in the development of synthesis tools for Micro-Electromechanical Systems (MEMS) are presented. Single-objective and Multi-objective Genetic Algorithms (MOGA) have been successfully applied to several MEMS synthesis examples. Genetic Algorithms (GA) are compared to the optimization technique of Simulated Annealing (SA) in terms of robustness, effectiveness and speed. Results show that SA can synthesize valid designs faster then Genetic Algorithms but does not handle multiple objectives as well and tends to have more difficulty in finding the global optimum.

Keywords: MEMS Synthesis; Genetic Algorithms; Simulated Annealing; SUGAR

---

1. Introduction

The goal of design synthesis is to help engineers develop rapid, optimal configurations for a given set of performance and constraint guidelines. Synthesis tools for MEMS utilizing different optimization algorithms have been developed as part of a larger research program aimed at developing a practical synthesis tool that can create both the topology and sizing of MEMS devices, incorporating an indexed library of device and sub-component examples from which to draw initial designs.

Multi-objective Genetic Algorithms (MOGA) have been successfully implemented for MEMS design in previous research [1,2]. This method was chosen because of its generality, robustness and ability to optimize for multiple design objectives. As part of our evaluation of MEMS synthesis techniques, we will contrast the Single Objective Genetic Algorithm (SOGA) and MOGA against Simulated Annealing (SA) optimization.

There has been much research on the comparison of different algorithms, including GA and SA in the area of numerical optimization [3,4], but none in the area of MEMS synthesis. There has been limited research in automating MEMS design [5,6] and even less in the synthesis of device topology using stochastic techniques.

MEMS synthesis using SA will be compared to the GA approaches in terms of robustness, effectiveness and speed for two examples.

2. Theory

MOGA uses an evolutionary approach to developing a population of optimal solutions (*figure 1)*: Given a higher-level description of the device's desired behavior, an initial population of candidate designs is generated randomly from a number of available components such as anchors, beams, electrostatic gaps and combs. Each design is checked for geometrical validity and its performance is evaluated. MOGA is then applied to the initial population to iteratively search for functional designs by applying the genetic operations of selection, elitism, crossover and mutation to create the next generation of designs. This process continues until an optimal set of "pareto optimal" solutions is synthesized.

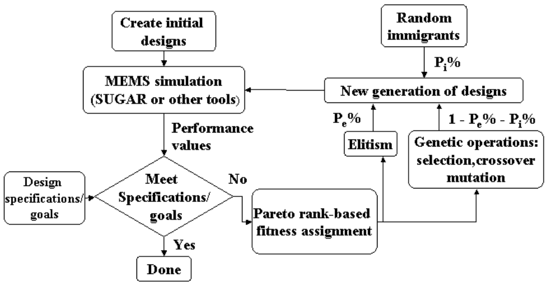

Figure 1: Evolutionary synthesis

As the name implies, Simulated Annealing exploits an analogy between the way in which a heated metal cools into a minimum energy state and a stochastic optimization algorithm that slowly "lowers the temperature" in stages to eventually "freeze" at the global optimum. SA randomly perturbs a given initial design, whose variations are accepted as the new design with a threshold probability, known as the Metropolis Condition, that decreases as the computation proceeds. The slower the rate of probability decrease, the more likely the algorithm is to find an optimal or near-optimal solution (*figure 2*).

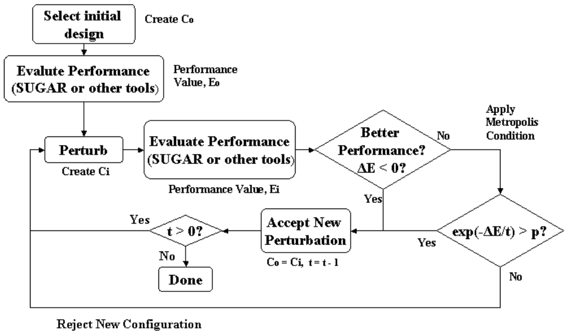

Figure 2: Simulated annealing synthesis

Whereas MOGA is designed to handle multiple separable objectives, in order to represent multiple objectives in SA they must be combined into a single function. For this paper we presented two examples: in the first we apply SOGA and SA to a meandering resonator design with one objective and in the second we apply MOGA and SA to an electrostatic actuator system with multiple objectives.

3. Implementation

The modified nodal analysis (MNA) tool SUGAR is used to evaluate the performance of the MEMS design. The use of SUGAR's much quicker MNA method, rather then the relatively slow finite element simulation, is critical to its computational tractability [7].

SUGAR defines elements as discrete elements such as beams, anchors, and electrostatic gaps connected at nodes. These primitive elements can be grouped in clusters or subassemblies such as meandering springs or comb drives. The location and frequency of these clusters (e.g., number of comb drives per device) can be changed in addition to their internal composition (e.g., number of fingers per comb).

A meandering resonator was chosen as the first example to synthesize. It is comprised of a center mass connected to four clusters comprised of a series of beams and an anchor (*see figure 3*).

The design goal for this example – to have a lowest natural frequency *f* = 93723 – was taken from an existing resonator that has been fabricated and characterized. Maximum and minimum values were set for the beam properties (length, width, angle) and a maximum number of beams per resonator leg were also specified. The center mass properties were fixed.

To better compare with SA, we chose to use a single objective, resonant frequency, for this paper. Therefore SOGA was utilized rather than MOGA. In [2] Zhou presents full results of MOGA synthesis for this example with three objectives, resonant frequency, and x and y stiffness.

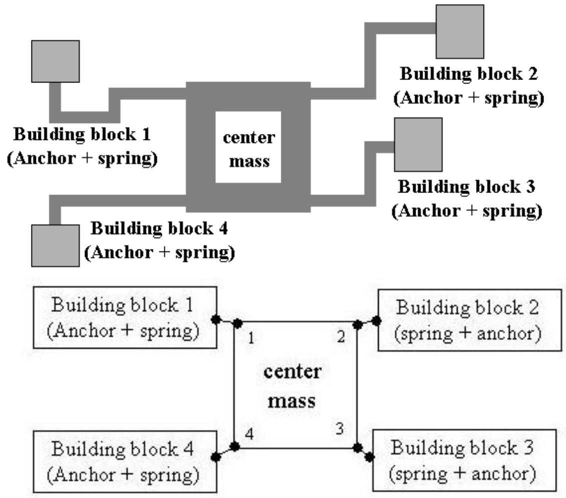

Figure 3: Example 1 – Schematic representations of meandering resonator

For the SOGA implementation, the crossover rate was set at 70%, mutation rate at 10%, elitism rate was 5% with new immigrants making up the balance.

For the SA implementation, the probability of changing any single beam parameter was set to 10%, as was the probability of adding or deleting a beam segment from each of the four clusters. Maximum number of iterations was set to 5000, but a conditional statement can stop the synthesis if the objective error is below an accepted value (frequency error < 100 Hz).

To further demonstrate the feasibility of these design tools, SA was also used with more strict constraints, including forcing Manhattan geometry and forcing symmetry between the clusters.

The second example presented is to synthesize the optimum design for an electrostatic actuator/spring device. This example is taken from a homework assignment in UC Berkeley's MEMS Introductory course: *create as small a device as possible that that can achieve a displacement of 20 ?m at an input voltage of 15V utilizing electrostatic actuation.* The representation chosen for synthesis is comprised of two clusters, a comb-drive cluster (where the number of combs, number of fingers per comb, length of fingers, gap between fingers can all be varied) and a serpentine spring cluster (where the beam width, length, and number of crenulations can be varied) (see *figure 4*).

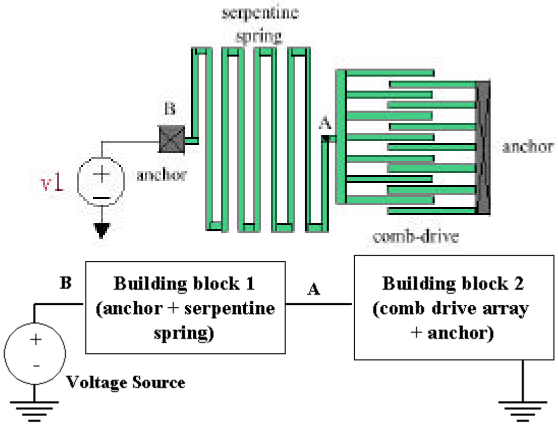

Figure 4: Example 2 – Schematic representations of electrostatic actuator

For example 2, the MOGA implementation has two objectives: 1) achieve a displacement greater then 20 ? m at 15V and 2) minimize the overall area of the entire structure. MOGA is well suited to tackle these two competing objectives. The same genetic operation settings were used for this example as the previous one - the crossover rate=70%, mutation rate=10%, elitism=5%. A population of 400 was used for 25 generations.

For SA to reach a solution, the minimum required displacement is added as a constraint through the use of a penalty function. If the constraint is violated, a penalty is added to the objective we wish to minimize (the device area). As the SA minimizes the objective function it also minimizes the constraint violation. The objective function is:

$$F(x) ? f(x) ? kg(x)^2$$

where *f(x)* is the area of the device, *g(x)* is:

if displacement > 20??m, g(x) = 0

else g(x) = 20??m - displacement

k is chosen to be large enough so that the penalty function dominates when *g(x)* is non-zero.

4. Results

For the meandering resonator example SOGA produced a population of good designs that converged to the objective quickly. Figure 5 shows an optimal design, along with a plot of the frequency of the best-ranked design in the population for each generation. The best-ranked design converges to the objective value within 15 generations. On a Pentium III-450 PC (used for all tests), the 30-generation run took 4-5 hours on average to complete.

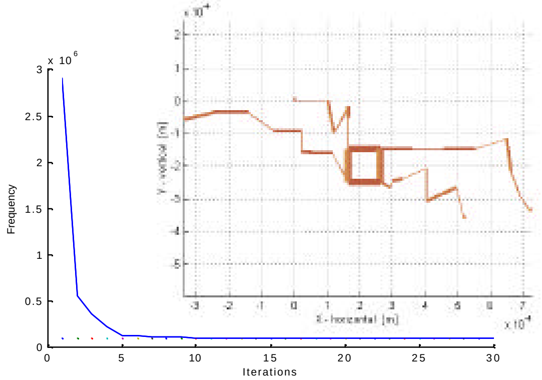

Figure 5: SOGA results for Example 1 – meandering resonator. Natural frequency of best solution per generation. Inset: layout of best-ranked design.

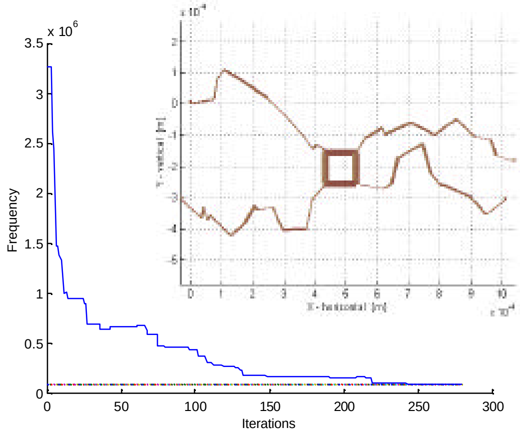

Figure 6: SA results for Example 1 – meandering resonator. Solution reached in fewer then 300 iterations. Inset: layout of final design.

The SA implementation was also able to achieve a good solution quickly. *Figure 6* shows a solution along with a plot of the resonant frequency per iteration. In this case the solution was reached quickly, only taking 279 iterations (approximately 1 hour). For different starting points, the average time was in the range of 4 to 5 hours. *Figure* 7 shows a solution with forced Manhattan geometry and symmetry between the clusters. This more constrained version reached valid solutions anywhere from 0.5 to 1 hours.

For the electrostatic actuator example, MOGA was able to develop an optimal solution within approximately 9 hours on average. *Figure 8* shows the objective values for the top 20 solutions found. Of these, the top 8 are valid (displacement greater then 20? m), with number 8 (highlighted) having the smallest area, 4.57e-8 m^2. *Figure 9* shows the layout of this design.

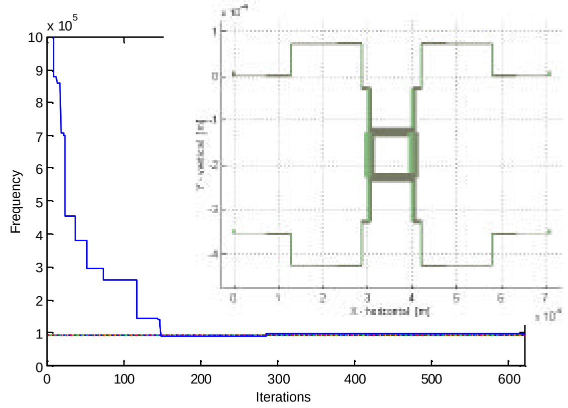

Figure 7: SA results for modified example 1 – meandering resonator with Manhattan geometry and symmetry constraints. Solution was reached in 620 iterations. Inset: layout of final design.

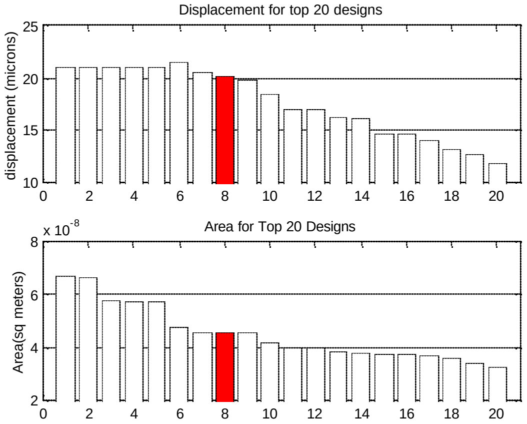

Figure 8: Example 2 – displacement versus area of top 20 MOGA solutions. The optimum is highlighted.

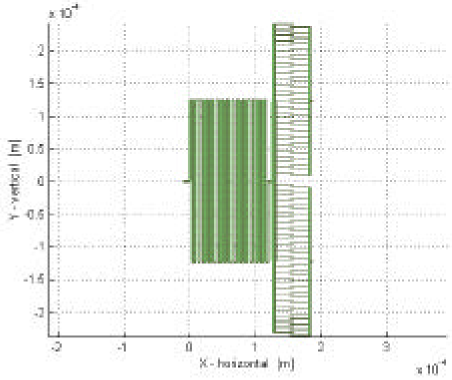

Figure 9: Example 2 – layout of one of the pareto optimal solution found by MOGA

For example 2, SA was not able to reach the globally optimum solution reached by MOGA. The best SA trial only reached an area of 8.58e-8 m^2 even after 10000 iterations (17+ hours). It is believed that this is due to an overly conservative cooling rate, causing the SA to get stuck in a local minimum. Further testing is necessary to see if better cooling rates can overcome these problems.

5. Conclusions

The results show that Simulated Annealing can in some cases synthesize valid designs faster then Genetic Algorithms. However its dependence on a single objective function and the difficulty in finding the global optimum indicate that it is a less robust method for many MEMS synthesis problems.

A major area for future work will be in incorporating more complete sets of fabrication and design rules into the existing synthesis tools, and then verifying their performance by fabricating and characterizing the synthesized designs.

6. References

1. N. Zhou, B. Zhu, A.M. Agogino, K.S.J. Pister, “Evolutionary Synthesis of MEMS (Microelectronic Mechanical Systems) Design”, *Proceedings of ANNIE 2001*, *Intelligent Engineering Systems through Artificial Neural Networks*, Volume 11, ASME Press, pp. 197-202.

2. N. Zhou and A. M. Agogino, "Automated Design Synthesis for Micro-Electro-Mechanical Systems (MEMS)", to appear in the *Proceedings of the ASME Design Automation Conference*, Sept. 29-Oct. 2 2002, Montreal, Canada.

3. J.C. Spall, S.D. Hill, D.R. Stark, “Some Theoretical Comparisons of Stochastic Optimization Approaches” *Proceedings of the 2000 American Control Conference*. vol.3, pp. 1904-8.

4. T. Lacksonen, “Empirical Comparison Of Search Algorithms For Discrete Event Simulation” *Computers & Industrial Engineering*, vol.40, (no.1-2), June 2001. pp.133-48.

5. H. Kotera, M. Senga, T. Hirasawa, S. Shima, “Optimization of Thickness Distribution of Micro-Membrane by Genetic Algorithm” *Proceedings of the 2nd International Conference on Intelligent Processing and Manufacturing of Materials*. IPMM'99, Honolulu, 10-15 July 1999, pp.245-50.

6. T. Mukherjee, G.K. Fedder, **“**Design Methodology For Mixed-Domain Systems-On-A-Chip [MEMS Design].” *Proceedings IEEE Computer Society Workshop on VLSI'98 System Level Design*, Orlando, FL, 16-17 April 1998, pp.96-101.

7. J.V. Clark, N. Zhou, et al, “Addressing the Needs of Complex MEMS Design”, *Proceedings of the 15th IEEE MEMS Conference*, Las Vegas, January 20-24, 2002, pp. 204-209.