Mask-Layout Synthesis Through an Evolutionary Algorithm

Hui Li, Erik K. Antonsson

California Institute of Technology, Mail Code: 104-44, 1200 E. California Blvd., Pasadena, 91125, USA

ABSTRACT

An automatic method for synthesizing MEMS mask-layouts is presented. This method uses evolutionary algorithm techniques to optimize the mask-layouts for a forward simulation of fabrication. Initially, a random population of mask-layouts is generated. The fabrication of each layout is simulated through a digital process simulator to produce a 3D fabricated shape, which is compared to a user-specified desired shape. Each evolutionary loop governs the stochastic searching behavior such that the mask-layouts whose simulated shapes are closer to the desired shape are more likely to survive. More importantly, the “better” masks are more likely to be evolved among those survived mask-layouts for the next loop. Through such evolutionary iterations, a near global “optimum” mask-layout is likely to be found. A test loop is constructed for the bulk wet etching mask synthesis by incorporating a 3D wet etching simulator. The current emerging results demonstrate the feasibility of this approach to mask-layout synthesis.

$Keywords$: MEMS, design, synthesis, genetic algorithm, bulk wet etching

INTRODUCTION

As the complexity of device-making through Microelectromechanical System (MEMS) increases, a clean separation between design process and fabrication process becomes more desirable. Such separation requires the synthesis techniques which automatically provide design and fabrication solutions to realize user-specified functions. This paper is devoted to realizing mask synthesis which will generate optimum mask-layouts for a micro-fabrication process to produce the user-desired functional shapes. Due to the complex and empirical nature of micromachining processes, most of the process simulations appear impossible to be reversed (so that a 2D mask-layout might be produced), which could be a direct approach to achieve the mask synthesis. Here we introduce an approach to realize mask synthesis through an evolutionary algorithm.

Evolutionary algorithm is a global stochastic optimization technique based on the adaptive mechanics of natural genetics [6]. It maintains a population of candidate solutions known as individuals. Typically, each individual is encoded into a string of characters or digits, through which the original solution space can be converted into an encoded space which becomes easier to search for the global optimum. By analogy with genetics, the individuals in the original search space are referred as phenotypes and those in the encoded space are named as genotypes. The initial population are generated randomly. During each iteration, within a population known as a generation, all individuals are evaluated to get their performance value called fitness values and then go through genetic operations such as selection, crossover *etc.* to form individuals of the next generation. Such iterative searching stops whenever the performance of an individual is satisfied or the limit of the search effort is reached. In general, evolutionary algorithms are a non-problem specific technique which can be applied to virtually any problem if its objective function measurements are available. In particular, what makes evolutionary algorithm distinct among other stochastic optimization techniques is the involvement of a generation of individuals. Such individual pool participation provides efficient solution information used by genetic operations to control the bias for the searching region as well as increase the robustness to overcome the deception traps during the searching. Evolutionary algorithms have been shown to successfully solve problems in various complex domains [5].

Consider the problem of mask synthesis, a general evolutionary algorithm loop can be constructed to evolve optimal mask-layouts for various process simulators as shown in Figure 1. During the evolution, an initial population of mask-layouts are randomly generated. The fabrication of each mask-layout is simulated by a specified process simulator to produce a 3D shape. The performance of each tried mask-layout is measured through the shape comparison between the produced shape and the user-desired shape. During each evolutionary loop, genetic operations are used to control the stochastic searching behavior such that the well-performed mask-layouts are more likely to survive and the survived individuals have potentials to evolve even better ones for the next loop. Through such iterative evolution, a near global optimally performed mask-layout is likely to be found. In this way, the existing forward simulations of fabrication processes are used to achieve mask-layout synthesis, where reversing those processes seems not to be possible.

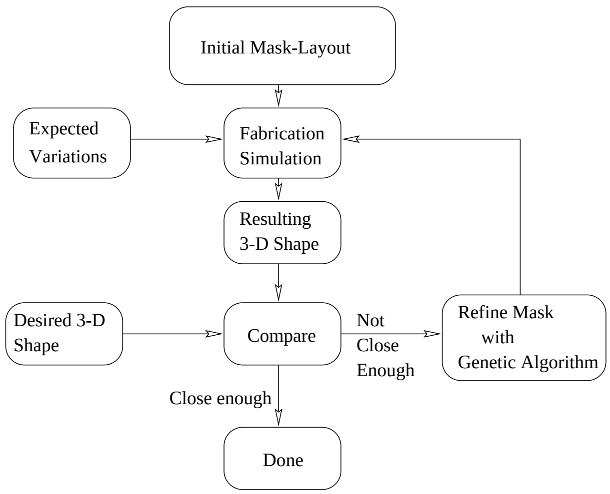

Figure 1: A Schematic Representation Of An Evolutionary MEMS Synthesis Technique.

ALGORITHM ARCHITECTURE

An object-oriented software architecture has been created to implement such an evolutionary algorithm based framework. The framework mainly consists of three independent modules, namely: mask genetics module, evolutionary strategy module and MEMS simulation module. The mask genetics module provides heuristic genetic operations on mask-layouts, which includes random mask generation, random crossovers, mutations, and local exploitations. The evolutionary strategy module contains strategy routines to control the convergency of searching process such as stochastic selection schemes, hybrid searching schemes and other strategic schemes to balance the searching effort between exploration versus exploitation. The MEMS simulation module is the user input module which contains user specified MEMS fabrication simulations and the desired fabricated shape.

With the use of object abstraction, the development of the three modules becomes separated and thus achieves a high level of software modularity. Figure 2 shows the design of three layers of object inheritance. The base layer consists of two abstract object types called GENOTYPE and PHENOTYPE which mnemonically represent genotype and phenotype individuals respectively. GENOTYPE provides the interfaces of coding and genetic operations. PHENOTYPE provides the interface of performance evaluation. The second layer has the derived object types MASK GENOTYPE and MASK PHENOTYPE devoted to mask synthesis application. The geometry of mask-layouts is stored in this layer. The third layer has the final derived types such as REAL MASK GENOTYPE and SHAPE MATCH MASK PHENOTYPE. REAL MASK GENOTYPE refers to the MASK GENOTYPE with a real coding scheme of mask-layouts. It provides the implementations of the interfaces defined by GENOTYPE. SHAPE MATCH MASK PHENOTYPE fulfills the major task of evaluating the performance of the mask-layout

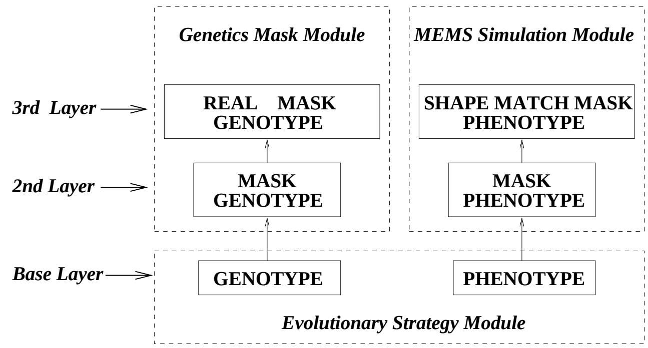

Figure 2: An Object-oriented Architecture On Evolutionary Algorithm.

through a specified process simulation and shape closeness measurements. Figure 2 also shows the relationship between the three modules and the three layers. The mask genetics module is built upon the base type MASK GENOTYPE and its derived types such as REAL MASK GENOTYPE; while the simulation module is developed based on MASK PHENOTYPE and its derived types such as SHAPE MATCH MASK PHENOTYPE. In this way, mask genetics module and simulation module are completely separated by the different inheritance chains. The evolutionary strategy module is entirely constructed from the base level objects GENOTYPE and PHENOTYPE and thus is insulated from the other two modules by the entire second inheritance layer.

THE EVOLUTIONARY TECHNIQUES

In our problem, we treat mask-layouts geometrically as 2D simple polygons, which form the underlying solution space. The entire searching for the optimum mask polygon is split into two stages: first find the optimum polygon shape (invariant to size and translation); then find the optimum polygon size. The second stage is simply carried through a greedy-based searching since it is reasonable to expect that the size of a mask polygon is proportional to the size of its fabricated shape. The evolution of an optimum polygon shape is rather challenging due to the complexity of shapeness and thus major searching effort has been devoted to the this stage.

Coding Scheme

Mask polygons are the searched solution points which serve as the phenotypes. To form the corresponding genotypes, we need an appropriate coding scheme. A real coding mechanism chosen because it provides adequate precision with a much shorter string length. The coding scheme only needs to encode the shapeness of mask polygon, which is captured through edge directional angles ranging from 0 to 2π and edge lengths scaled by a common factor (called edge lengths from now on). Such a coding scheme provides a convenient way to design the associated crossover and mutation schemes as illustrated in later sections. Two

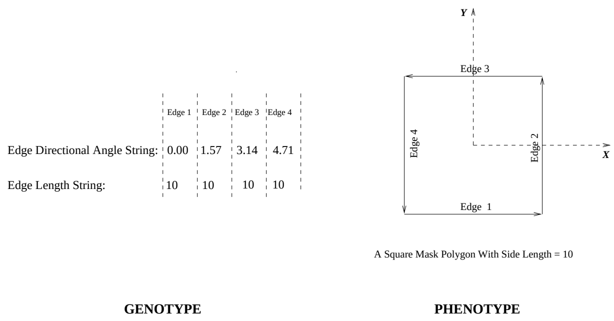

Figure 3: A Schematic Illustration Of The GA Coding Scheme.

real strings are used with one called "angle string" which contains edge directional angles and the other called "distance string" which contains edge lengths. The size of each string is equal to the number of mask polygon sides. Two elements from each string with the same element position describe a polygon edge. Such a coding scheme is illustrated in Figure 3. Through the encoding, the searching for the optimal mask polygon solution in the phenotype space is converted into the searching for the optimal individual with the two real strings in the genotype space.

Initialization

At the initialization stage, the ideal goal is to generate uniformly distributed sample points in the genotype space while meet the constraints that all the corresponding mask polygons are simple ones. To meet such a goal, it is equivalent to require that for each edge, the length is uniformly spread over a specified searching range and the directional angle is evenly distributed from 0 to 2π. A heuristic generation mechanism is used to incrementally obtain each edge by keeping randomly generating the edge length and angle until no intersection exists between the edge and any previously generated ones. It is easy to see that such an approach can only guarantee the uniform length and angle distributions of the first generated edge and the uniformities of such distributions becomes poorer as the later edges are generated. To dilute this effect, the first edge of the mask polygon is randomly selected among all the generated edges. The polygons generated in this way will have various sizes and thus need to be uniformly scaled to a common size to capture the shape.

Genetic Operations

The genetic operations mainly involve crossover and mutation. Crossover serves as the driving genetic operator to evolve offsprings from a given pair-wise genotype parents. Mutation is used only if the implementation of crossover on certain genotypes fails.

During each crossover, since each genotype consists of two real strings, for any two genotypes, a pair of edge directional angle strings and a pair of edge length strings are formed separately. Within each string pair, the crossover operator is applied onto paired elements. The number of the paired elements and which one of them participates in the crossover can be either specified or randomly determined. The crossover operator is the blend crossover BLX-α with α value set as 0.5 [4]. Such a crossover operator may produce the element value exceeding the valid bounds, in which case, for edge length, the simple cut-off to the nearest boundary value is used and for edge angle, due to the wrap-around, any angle value is converted into the range of 0 to 2π. Heuristics are incorporated to improve the performance and ensure the validity. First, before applying the crossover, the polygons of parent genotypes are geometrically aligned and then string elements are paired according to the aligned edges. Second, a self-intersection routine is used to make sure that each generated child edge does not intersect with all the previously generated ones and thus the simplicity of all the evolved mask polygons are guaranteed.

Mutation is applied on a single genotype. An edge is randomly selected and its length and directional angle are mutated through the random regeneration between the boundary values. And then, the underlying polygon is reconstructed accordingly. Note that a valid polygon may not always be reconstructed, in which case the above process is retried until a new valid one is produced.

Selection Scheme

The selection scheme is decomposed into the selection algorithm and the sampling algorithm. The selection algorithm assigns a sampling rate to each individual in the generation based on the performance value. The sampling algorithm samples individuals from the current generation based on the assigned sampling rates. Through such separation, various selection schemes can be constructed with different combinations of a selection algorithm and a sampling algorithm. In the application shown below, a rank-based selection algorithm is chosen to assign the sampling rates based on individual's rank [2] through the following formula:

$$\text{Sampling rate}(i) = 1.0 + \text{bias} - 2 * \text{bias} * i/\text{size}$$

i	=	individual i
size	=	population size
bias	=	extra fitness awarded to the top-ranked individual compared to the average-ranked individual whose fitness is 1.0

In this way, the single bias variable can be used to control the selection pressure. Currently, the bias value is gradually increased as the genetic iteration proceeds to control the rate of convergence.

Stochastic universal sampling is chosen as the sampling algorithm to eliminate the sampling bias and reduce the sampling spread [3].

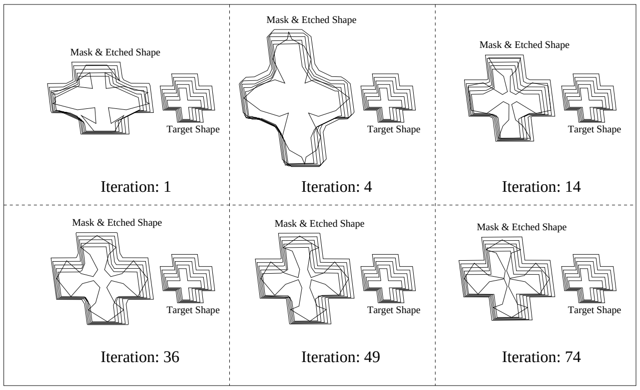

Figure 4: Evolutionary Algorithm Synthesis Result For Void Cross Target Shape.

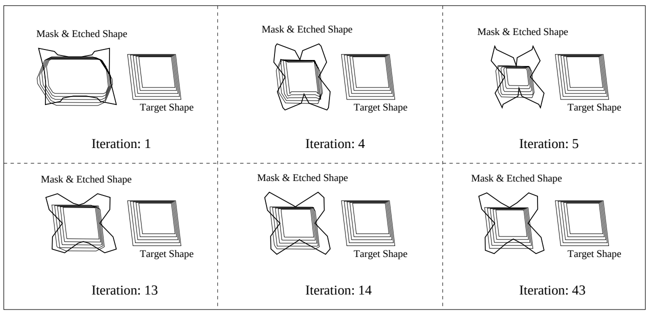

Figure 5: Evolutionary Algorithm Synthesis Result For Solid Square Target Shape.

APPLICATION

A simulation module is constructed including a wet etching simulator called SEGS [7] and a 3D shape matching algorithm. The module is embedded into the evolutionary algorithm framework to synthesize the mask-layouts for bulk wet etching. The goal is to find a mask which produces a bulk-etched 3D shape which matches a specified target shape as closely as possible.

The shape matching algorithm calculates the shape mismatch value between two 3D shapes, which is subtracted by a specified maximum mismatch value to obtain a fitness value. All the shapes are represented by a stack of horizontal polygonal layers. Each mismatch value can be constructed as the weighted sum of the mismatch between the two polygon layers in each vertical level. The mismatch value between two polygons are obtained by the method introduced in [1]. The method calculates the minimum difference called L1 distance between the accumulated turning angles of each edge weighted by the scaled length of that edge.

Figure 4 and Figure 5 demonstrates the test results. Each cell in each figure includes two shapes. The right shape is the user-specified shape which is the same throughout the cells. The left shape is the synthesized mask-layout (indicated by the darker outline) and its etched shape. The iteration numbers are presented to show the converging search process. In both tests, the generation size is set as 80. Crossover is applied to all the paired string elements. Crossover could fail if no valid children mask-layouts are generated, in which case, mutation is carried out. The target shape in Figure 4 is a cross-shaped hole. The challenge in this test is the complexity of the shape with both convex and concave corners involved. And the convex corners need to be compensated by the mask-layout. Figure 5 shows the synthesis for a solid peg shape which requires the compensations for the four square corners in the mask-layout.

CONCLUSIONS

Thus far we have created a general mask-layout synthesis framework driven by an evolutionary algorithm. With the use of object-oriented architecture, the framework is able to incorporate any forward process simulator. The bulk wet etching simulator SEGS has been plugged into the framework to demonstrate a synthesis outcome. The high complexity level of wet etching mask-layout synthesis comes from the fact that there is no simple geometrical correlation between masks and the evolved 3D structures. Successful results have been obtained for several challenging tests. Of course, the full verification of such synthesis methodology has to be based on further tests on various other applications. Nevertheless, the demonstrated success over such complexity level of synthesis gives enough justifications to continuously pursue the application of such methodology.

REFERENCES

[1] Esther M. Arkin, L. Paul Chew, Daniel P. Huttenlocher, Klara Kedem, and Joseph S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:209–215, 1991.

[2] James E. Baker. Adaptive selection methods for genetic algorithm. In John Grefenstette, editor, *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pages 101–111, 1985.

[3] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In *The Second International Conference on Genetic Algorithms*, 1987.

[4] Larry J. Eshelman and J.David Schaffer. Real-coded genetic algorithms and interval-schemata. *Foundation of Genetic Algorithms*, pages 187–202, July 1991.

[5] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[6] John H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[7] Ted J. Hubbard and Erik K. Antonsson. Design of MEMS via Efficient Simulation of Fabrication. In *Design for Manufacturing Conference*. ASME, August 1996.