Evolutionary Algorithms for Planar MEMS Design Optimisation: A Comparative Study

Elhadj Benkhelifa, Michael Farnsworth[*], Ashutosh Tiwari, and Meiling Zhu

Abstract. The evolutionary approach in the design optimisation of MEMS is a novel and promising research area. The problem is of a multi-objective nature; hence, multi-objective evolutionary algorithms (MOEA) are used. The literature shows that two main classes of MOEA have been used in MEMS evolutionary design Optimisation, NSGA-II and MOGA-II. However, no one has provided a justification for using either NSGA-II or MOGA-II. This paper presents a comparative investigation into the performance of these two MOEA on a number of MEMS design optimisation case studies. MOGA-II proved to be superior to NSGA-II. Experiments are, herein, described and results are discussed.

1 Introduction

Micro-electro-mechanical systems (MEMS) or micro-machines [1] are a field grown out of the integrated circuit (IC) industry, utilizing fabrication techniques from the technology of Very-Large-Scale-Integration (VLSI). The goal is to develop smart micro devices which can interact with their environment in some form.

Elhadj Benkhelifa · Michael Farnsworth · Ashutosh Tiwari
Decision Engineering Centre, Building 50, School of Applied Sciences, Cranfield Campus, Cranfield University College Road, Cranfield Bedfordshire, MK43 0AL
e-mail: {e.benkhelifa,m.j.farnsworth,a.tiwari}@cranfield.ac.uk

Meiling Zhu
Microsystems and Nanotechnology, Building 40, School of Applied Sciences, Cranfield Campus, Cranfield University College Road, Cranfield Bedfordshire, MK43 0AL
e-mail: m.zhu@cranfield.ac.uk

[*] Corresponding author.

J.R. González et al. (Eds.): NICSO 2010, SCI 284, pp. 199–210, 2010.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

The paradigm of MEMS is well established within both the commercial and academic fields. At present encompassing more than just the mechanical and electrical [2], MEMS devices now cover a broad range of domains, including the fluidic, thermal, chemical, biological and magnetic systems. This has resulted in a host of applications to arise, from micro-resonators and actuators, gyroscopes, micro-fluidic devices [3], and biological lab on chip devices [4], to name but a few.

Developing MEMS by silicon micromachining fabrication techniques [5] requires both many prototypes and a long line of experimentation (design process). The process of MEMS design itself is broken down into many levels into which a designer may provide input and ultimately model, analyse and optimise a device. The process itself has been outlined by both fedder [6] and also senturia [7]. Normally, designs are produced in a trial and error approach dependant on user experience and naturally an antithesis to the goal of allowing designers the ability to focus on device and system design. This approach, nominally coined a "Build and Break" iterative, is both time-consuming and expensive.

A number of Computer Aided Design (CAD) tools and simulators have been developed and used to facilitate an improvement in the design process; however, this does not solve a fundamental problem with the current approach to MEMS design optimisation. The development of a design optimisation environment, which can allow MEMS designers to automate the process of modelling, simulation and optimisation at all levels of the MEMS design process, is fundamental to the eventual progress in MEMS Industry. Such an environment reduces the burden put on the designer and providing mediums that will potentially produce optimal devices within design constraints [20]. Work in design automation and optimisation can be seen to fall into two distinct areas; firstly the more traditional approaches found within numerical methods such as gradient-based search [8] [9]; and secondly the use of more powerful stochastic methods such as simulated annealing [10] and/or Evolutionary Algorithms (EAs) [11][12][16]. The current work has employed the latter to evolve and optimise new MEMS devices. Different researchers have used different classes of EAs in this subject domain [10],[12],[13],[14],[15],[17], however it is not clear which particular EA approach is the most appropriate and efficient in MEMS design synthesis and optimisation. This paper presents a comparative investigation into the performance of, particularly, two well known and widely used EAs (Multi-Objective Genetic Algorithm: MOGA-II [18] and NSGA-II [19]) on a number of MEMS design optimisation case studies. MOGA-II proved to be superior to NSGA-II. Experiments are, herein, described and results are discussed.This study also allows the validation of a design optimisation framework, by coupling both areas of MEMS simulation and analysis with optimization routines.

The next section describes the evolutionary design optimisation environment for MEMS. The subsequent section presents the experimental setup for three case studies of increasing complexity, followed by results and discussions in sections 4 and 5, respectively. Finally, a conclusion of the findings is presented in section 6.

2 MEMS Design Optimisation Framework

It is important as a designer to be able to undertake automated design optimisation whenever possible in order to speed up the design process. In response to this we establish a design optimization framework which links a powerful optimization environment tool based on EAs, with MEMS simulator SUGAR Fig 1. The MOEAs follow an iterative process, selecting designs based upon their performance in respect to the objectives set out, evolving them using powerful operators. Analysis is then undertaken by the simulator which is passed a parameters structure which overrides a default model design. Finally analysis is retrieved and designs are evaluated and ranked and finally replacement operators tune out worse designs by replacing them with better offspring.

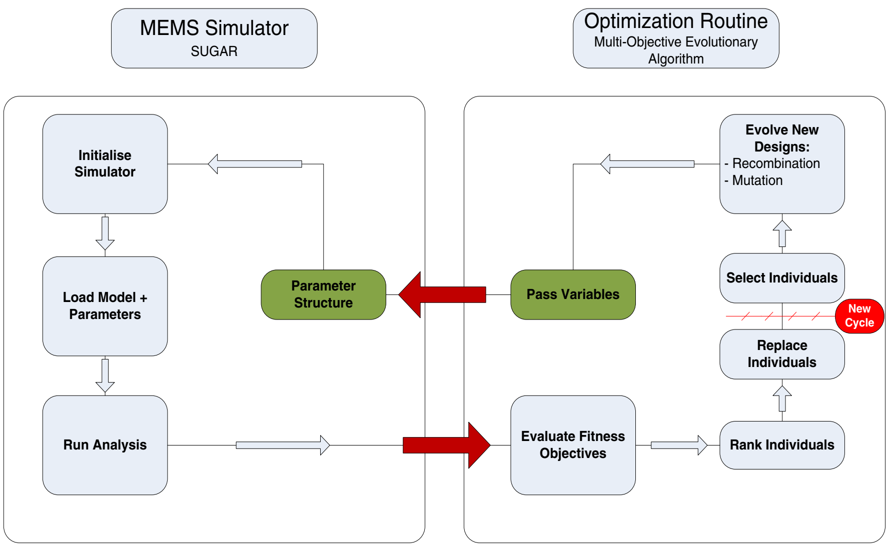

Fig. 1 An Evolutionary Design Optimisation Framework for MEMS

3 Experiments Set Up

Drawing on previous work undertaken in the field [15][10][21], planar MEMS devices form the basis for our evaluation of our design optimisation approach. A set of three case studies of increasing complexity have been implemented within our design optimisation environment, which forms a suitable strategy to evaluate the performance of the algorithms in question. The experiments investigate the performance of MOGA-II and NSGA-II for the design and optimisation of MEMS through these case studies. For each case study five experimental runs of each algorithm are conducted. MOGA-II is an improved version of MOGA by Poloni [22],

Table 1 Experimental parameter settings for MOGAII and NSGAII

MOGA-II		NSGA-II	
Probability of directional crossover	80%	Probability of SBX crossover	80%
Probability of classical crossover	14%	Probability of Mutation	1%
Probability of Mutation	1%	Distribution Index for crossover	20
DNA string Mutation ratio	5%	Distribution Index for mutation	20
Population	100	Population	100
Generations	100	Generations	100

utilizing a smart multi search elitism, and a triad of operators (classical one-point crossover, directional crossover and bit flip mutation) each with their own probability of invocation. As with classical MOGA, the representation is a binary string and in order to simulate continuous variables a sufficiently high base value must be used to divide between upper and lower bounds the possible variable values. NSGA-II is an elite preserving multi objective genetic algorithm, which also includes a diversity heuristic to maintain a uniform spread on the Pareto front. Unlike the standard MOGA, NSGA-II uses a real-valued representation, and therefore both recombination and mutation operators revolve around these real values. Both algorithms use some form of elitism based generational evolution and a breakdown of each is found below. The algorithms' parameters are fixed as shown in Table 1.

Algorithm 1: MOGA-II Pseudo Code

1. Initialize population
   a. Generate random population of size N and elite set E = θ
2. Evaluate objective values
3. Assign rank based on Pareto dominance - 'Sort'
4. Generate offspring population
   a. Combine both population and elite sets P' = P $\cup$E
   b. If the cardinality of P' is greater than the cardinality of P reduce P' removing randomly the exceeding points.
   c. Compute the evolution from P' to P" applying MOGA operators
      i. Randomly assign one operator (Local tournament selection, directional crossover, one point crossover or bit flip mutation) based upon probability of invocation.
5. Evaluate objective values of population P"
6. Assign rank to P" individuals based on Pareto dominance - 'Sort'
7. Copy all non-dominated designs of P" to E - 'Sort'
8. Update E by removing duplicated or dominated designs
9. Resize the elite set E if it is bigger than the generation size N removing randomly the exceeding individuals
10. Return to step 2 considering P" as the new P until termination

Algorithm 2: NSGA-II Pseudo Code

1. Initialize population.
   a. Generate random population P of size N.
2. Evaluate objective values.
3. Assign rank based on Pareto dominance - 'Fast-Sort'.
4. Generate offspring population.
   a. Create population P' using tournament selection and apply variation operators (Simulated binary crossover and mutation).
5. Evaluate objective values of population P'.
6. Combine both population sets P and P' to give set of size 2N P".
7. Assign rank to P" individuals based on Pareto dominance - 'Fast-Sort'.
   a. Fill new P set with non-dominated fronts until cardinality is reached from set P"
   b. If the cardinality of new set P is greater than the size N reduce P by computing the crowding distance of the last front set to be added and fill remaining slots using crowded-comparison operator.
8. Return to step 4 until termination.

The case studies experimented with are; a simple meandering spring, a meandering resonator and finally a real world example of an ADXL150 accelerometer. For each case study, the performance of MOGA-II and NSGA-II is evaluated.

3.1 Case Study: Meandering Spring

The core topology of a large class of MEMS, such as micro-resonators and accelerometers consists generally of a spring + mass system, where a mass is suspended by a spring like structure anchored to a substrate, and the shape and topology of which effects the behaviour of the device. Therefore the ability to evolve spring like structures which match certain behaviour is important for the eventual design optimisation of more complex spring + mass systems such as a micro-resonator. Following previous work [15] we look to synthesize a simple meandering spring, composed of several beams, each of which has three variables, length, width and angle. The variable design parameters can be seen in Table 2.

Table 2 Variables design parameters used in spring case study, taken from [15]

Min Width	Max Width	Min Length	Max Length	Angle Min	Angle Max	Min Beam No	Max Beam No
2E-06	2E-05	2E-05	4E-04	-90	90	1	6

The objectives chosen for the experiment were to evolve designs that matched a certain behaviour in this instance each spring was to have a stiffness in the x direction Kx = 2N/m, and a stiffness in the y direction Ky = 2N/m following a force applied deflection. In this instance the objectives simply become the minimization of error from the design goal of 2 N/m.

3.2 Case Study: Meandering Resonator

It is important to be able to design micro resonators to match a certain frequency which can be integrated into a band-pass filter device. Following previous work [12] we look to evolve a MEMS resonator in order to match certain behaviour and design objectives. For this case study a set of four meandering springs are evolved each of which consists of several beams. The same variable parameters as described in table 2 are used and the central mass shape is fixed as in [10]. In order to reduce the search space complexity, a symmetry constraint to the design is applied, where one spring is evolved and then mirrored in both the x and y directions. The objectives for each design remain the same as for the spring, but also a third objective of having a first mode resonance frequency of 93723 Rad/s as taken from [12] are shown in Table 3.

Table 3 Design Objectives for Meandering Resonator

Objective	Target
Stiffness Kx N/m	2.0 (Minimize Error)
Stiffness Kx N/m	2.0 (Minimize Error)
Frequency Rad/s	93723 Rad/s (Minimize Error)

3.3 Case Study: ADXL150 Accelerometer

The goal to produce devices that mimic already viable real world macro designs but at a much smaller and more energy efficient way is a possibility with MEMS technology. The ADXL accelerometer series is a device which has been fabricated and tested in real world applications and seen it replacing its macro counterpart. This device can detect acceleration, as a result of force and gravity. This is crucial in one of the applications of this device that of car airbag deployment. Upon impact with another vehicle, acceleration as a result of the force occurs, this is then detected via the accelerometer device and if over a given threshold the signal can trigger the deployment of the airbag and thus save lives. The design variables follow that of previous work undertaken in [21] and are summarised in table 4. They consist of a central mass and a special case spring known as a “serpentine” spring, along with the sensing comb that runs alongside the mass. In this particular case study a symmetry constraint is applied to the serpentine springs, as a result only one spring is evolved and then mirrored in the x and y directions. The design objectives for these experiments are shown in Table 5.

Table 4 Design Objectives for ADXL150 Accelerometer

Variable	Lower Bounds	Upper Bounds
Mass Length	300μm	600 μm
Mass Width	50 μm	150 μm
Finger Length	30 μm	130 μm
Finger Width	2 μm	4 μm
Short Beam Length	10 μm	10 μm
Short Beam Width	2 μm	2 μm
Long Beam Length	10 μm	100 μm
Long Beam Width	2 μm	2 μm
Crenulations	1	6

Table 5 ADXL150 Case Study design Objectives

Objective	Target
Frequency Rad/s	150,796 Rad/s (Minimize Error)
Total Area μm^2	Minimize
Sense Capacitance fF	Maximize

4 Results

For each case study results are represented in four sets of values; firstly the number of pareto solutions that were present at the end for each experiment (exp) run; secondly the number of pareto solutions from a particular experiment that remained when all five sets were combined, thirdly the number that remained when constraints on objective values were added and finally near the bottom we compare the number of pareto solutions from these sets that remain for each algorithm when MOGA-II and NSGA-II pareto individuals are combined.

Table 6 Table 7 and Table 9 shows the number of Pareto optimal solutions found within each experimental run for case studies described in sections 3.1, 3.2, and 3.3,

Table 6 MOGAII v NSGAII Experimental Results for the Meandering Spring

MOGA-II				NSGA-II			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol < 1% Error per Obj	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol < 1% Error per Obj
1	4316	2299	0	1	2944	1	0
2	26	0	0	2	2322	0	0
3	910	910	910	3	2866	0	0
4	2919	2873	1	4	2886	0	0
5	1920	0	0	5	209	209	209
Total	10091	6082	911	Total	11227	210	209
Total MOGAII v NSGAII	-	-	1	Total MOGAII v NSGAII	-	-	209

Table 7 MOGAII v NSGAII Experimental Results for the Meandering Resonator

MOGA-II				NSGA-II			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol < 1% Error per Obj	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol < 1% Error per Obj
1	66	22	7	1	220	10	0
2	31	18	12	2	16	0	0
3	49	1	0	3	87	82	8
4	215	1	0	4	182	31	0
5	42	20	10	5	164	1	0
Total	403	62	29	Total	669	124	8
Total MOGAII v NSGAII	-	-	29	Total MOGAII v NSGAII	-	-	2

Table 8 MOGAII v NSGAII Top 10 Frequency Results for culled < 1% set for the Meandering Resonator

MOGA-II					NSGA-II				
Exp	ID	Freq Error Rad/s	Kx Error N/m	Ky Error N/m	Exp	ID	Freq Error Rad/s	Kx Error N/m	Ky Error N/m
1	9630	1.07	1.253E-02	6.911E-03	3	29624	20.15	1.851E-02	8.094E-03
5	47112	2.75	1.644E-02	4.814E-03	3	29693	30.08	1.445E-02	7.766E-03
5	47555	3.93	5.284E-03	3.878E-04	3	29734	33.24	6.372E-03	6.175E-03
1	8608	4.04	3.722E-04	9.670E-03	3	29806	72.17	4.353E-03	2.785E-03
1	9053	4.04	3.722E-04	9.670E-03	3	29660	122.65	4.875E-04	3.309E-03
1	9172	4.04	3.722E-04	9.670E-03	3	28978	245.77	5.755E-03	2.047E-03
1	9880	4.04	3.722E-04	9.670E-03	3	29563	279.95	8.898E-03	1.072E-03
2	19326	10.45	3.079E-03	8.302E-03	3	28733	668.50	1.678E-03	4.429E-05
2	18276	13.29	1.664E-05	6.495E-03	-	-	-	-	-
2	19258	13.91	3.165E-03	3.322E-03	-	-	-	-	-

Table 9 MOGAII v NSGAII Experimental Results for the ADXL150 Accelerometer

MOGA-II				NSGA-II			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol < 1% Error per Obj	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol < 1% Error per Obj
1	1525	551	47	1	1741	684	36
2	1389	646	69	2	1781	289	34
3	1613	547	88	3	1298	382	7
4	1494	940	146	4	1325	857	19
5	1464	695	134	5	1229	449	22
Total	7485	3379	484	Total	7374	2661	118
Total MOGAII v NSGAII	-	-	484	Total MOGAII v NSGAII	-	-	18

Table 10 MOGAII v NSGAII Top 10 Total Area Results for culled $< 1\%$ set for the ADXL150 Accelerometer

MOGA-II					NSGA-II				
Exp	ID	Freq Error Rad/s	Kx Error N/m	Ky Error N/m	Exp	ID	Freq Error Rad/s	Kx Error N/m	Ky Error N/m
1	9630	1.07	1.253E-02	6.911E-03	3	29624	20.15	1.851E-02	8.094E-03
5	47112	2.75	1.644E-02	4.814E-03	3	29693	30.08	1.445E-02	7.766E-03
5	47555	3.93	5.284E-03	3.878E-04	3	29734	33.24	6.372E-03	6.175E-03
1	8608	4.04	3.722E-04	9.670E-03	3	29806	72.17	4.353E-03	2.785E-03
1	9053	4.04	3.722E-04	9.670E-03	3	29660	122.65	4.875E-04	3.309E-03
1	9172	4.04	3.722E-04	9.670E-03	3	28978	245.77	5.755E-03	2.047E-03
1	9880	4.04	3.722E-04	9.670E-03	3	29563	279.95	8.898E-03	1.072E-03
2	19326	10.45	3.079E-03	8.302E-03	3	28733	668.50	1.678E-03	4.429E-05
2	18276	13.29	1.664E-05	6.495E-03	-	-	-	-	-
2	19258	13.91	3.165E-03	3.322E-03	-	-	-	-	-

respectively. For the constrained set all individuals which did not have an error value within 1% of the target for each objective were removed and in the case of the ADXL150 accelerometer an additional constraint of designs with a minimum sensitivity of 133fF was applied.

Table 8 highlights the top ten results from the culled 1% set, ranked by frequency error objective for the Meandering Resonator case study. Table 10 highlights the top ten results from our culled 1% set, ranked by total area objective for the ADXL150 Accelerometer case study.

5 Comparison and Discussion

From the above presented results one can begin to paint a picture into the performance of the two selected algorithms on this particular subset of case studies for MEMS design optimisation. To begin with, it seems that both algorithms are robust enough to provide similar sets of Pareto fronts from each experimental run when they are collated. However of the two algorithms, MOGA-II provides results which fair better, with NSGA-II falling down somewhat with the meandering spring case study. Of the number of Pareto solutions found within the target constraints for each case study, MOGA-II outperforms NSGA-II, generally producing two thirds more solutions for all three case studies. A direct comparison between the final Pareto sets for each algorithm provides a similar result, with MOGA-II providing more individuals within the Pareto front for the ADXL150 Accelerometer and the Meandering Resonator case-studies, with only NSGA-II providing better results on the Meandering Spring example.

Reasons behind such discrepancies could fall into the differences found within each algorithm; these can lay either in, naturally, the choice of representation, the role of each algorithms variation operators, or some of the diversity heuristics used. If one picks up on the third case study, the ADXL150 Accelerometer, from Table 10 one can see a deviation in terms of behaviour between MOGA-II and NSGA-II. For this case study, the constraints focused upon designs which had as small a total area as possible while maintaining a sensitivity value above 133 fF and a minimum frequency error below 1% of the target goal. Though both MOGA-II and NSGA-II were able to provide individuals which lied within these constraints, NSGA-II designs seem to lie heavily towards an increased sensitivity, rather than focusing upon reduced total area.

From the outset this seems to cast a shadow on the performance of NSGA-II in this example; however it may be an unfair assessment and hence requires further analysis. NSGA-II employs a diversity heuristic in the form of a crowded distance operator to enforce a uniform spread of Pareto solutions while MOGA-II does not in any specific way emulate this behavior. The ADXL150 example contains two particular objectives which somewhat work in tandem, that being total area and sensitivity. In this instance changing the mass length of the device can either result in a decrease in total area and subsequent decrease in sensitivity or provide the opposite effect. In parallel, increasing finger length can increase sensitivity and the total area and vice versa. These two variable changes seem to have the most accessible influence in objective function performance, and as such most likely drive our algorithms for fitter individuals. As NSGA-II looks to find a suitable spread it will produce designs which lie upon the whole gradient between these two competing objectives, while MOGA-II does not feel this selective pressure and can perhaps begin to concentrate on designs which target improved frequency objectives. As a result MOGA-II can possibly produce designs which target all three objectives more easily than NSGA-II. Given the final constraint where we want to focus on one particular area of a front, something NSGA-II looks to avoid, the MOGA-II is not encumbered by this and as a result it seems able to produce superior designs. This is only a speculative explanation and requires further investigation into what effect NSGA-II's crowded diversity heuristic has in terms of performance and the reasons why.

Finally, it is a possibility that each algorithm local search approach, be it MOGA-II's single bit flip operator or NSGA-II's real valued polynomial mutation may provide a profound difference in performance when it comes to local search performance at near optimal design spaces. In the field of MEMS design it is important for operators to cope with such small scales and is therefore something for further investigation.

6 Conclusions

The paper presents an important study that compares the performance of two widely known and used evolutionary algorithms for the design optimisation of MEMS devices, namely, NSGA-II and MOGA-II. Experiments are conducted on three MEMS case studies with increasing complexity. Initial results clearly show the superiority of MOGA-II over NSGA-II. Speculative explanations are discussed in section 5, however, further work is needed to evaluate the reasons why the performance of the two algorithms differed, and essentially the role of various heuristics and operators in the evolutionary design optimisation of this application domain.

References

[1] Fujita, H.: Two Decades of MEMS– from Surprise to Enterprise. In: Proceedings of MEMS, Kobe, Japan, January 21-25 (2007)

[2] Hsu, T.R.: MEMS and Microsystems, 2nd edn. Wiley, Chichester (2008)

[3] Isoda, T., Ishida, Y.: Seperation of Cells using Fluidic MEMS Device and a Quantitative Analysis of Cell Movement. Transactions of the Institute of Electrical Engineering of Japan 126(11), 583–589 (2006)

[4] Hostis, F.l., Green, N.G., Morgan, H., Akaisi, M.: Solid state AC electroosmosis micro pump on a Chip. In: International Conference on Nanoscience and Nanotechnology, ICONN, Brisbane, Qld, July 2006, pp. 282–285 (2006)

[5] Hao, Y., Zhang, D.: Silicon-based MEMS process and standardization. In: Proceedings of the 7th International Conference on Solid-State and Integrated Circuits Technology 2004, vol. 3, pp. 1835–1838 (2004)

[6] Fedder, G.: Structured Design of Integrated MEM. In: Twelfth IEEE International Conference on Micro Electro Mechanical Systems, MEMS 1999, Orlando, FL, USA, pp. 1–8 (1999)

[7] Senturia, S.D.: Microsystem Design. Kluwer Academic Publishers, Dordrecht (2001)

[8] Haronain, D.: Maximizing microelectromechanical sensor and actuator sensitivity by optimizing geometry. Sensors and Actuators A 50, 223–236 (1995)

[9] Iyer, S., Mukherjee, T., Fedder, G.: Automated Optimal Synthesis of Microresonators. In: Solid-State Sensors and Actuators, Chicago, IL, pp. 12–19 (1997)

[10] Kamalian, R., Zhou, N., Agogino, A.M.: A Comparison of MEMS Synthesis Techniques. In: Proceedings of the 1st Pacific Rim Workshop on Transducers and Micro/Nano Technologies, Xiamen, China, July 22-24, pp. 239–242 (2002)

[11] Li, H., Antonsson, E.K.: Evolutionary Techniques in MEMS Synthesis. In: Proc. DETC 1998, 1998 ASME Design Engineering Technical Conferences, Atlanta, GA (1998)

[12] Zhou, N., Agogino, A.M., Pister, K.S.: Automated Design Synthesis for Micro-Electro-Mechanical Systems (MEMS). In: Proceedings of the ASME Design Automation Conference, ASME CD ROM, Montreal, Canada, September 29-October 2 (2002)

[13] Kamalian, R.H., Takagi, H., Agogino, A.M.: Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 1030–1041. Springer, Heidelberg (2004)

[14] Zhang, Y., Kamalian, R., Agogino, A.M., Sequin, C.: Hierarchical MEMS Synthesis and Optimization. In: Varadan, V.K. (ed.) Proceedings of SPIE, Smart Structures and Materials 2005: Smart Electronics, MEMS, BioMEMS, and Nanotechnology. International Society for Optical Engineering, CD ROM. Paper 5763-12, vol. 5763, pp. 96–106 (2005)

[15] Zhou, N., Zhu, B., Agogino, A.M., Pister, K.: Evolutionary Synthesis of MEMS (Microelectronic Mechanical Systems) Design. In: Proceedings of ANNIE 2001, IEEE Neural Networks Council and Smart Engineering Systems Laboratory, Marriott Pavilion Hotel, St. Louis, Missouri, November 4-7, vol. 11, pp. 197–202. ASME Press (2001)

[16] Benkhelifa, E., Farnsworth, M., Tiwari, A., Zhu, M.: An Integrated Framework for MEMS Design Optimisation using modeFrontier. In: EnginSoft International Conference 2009, CAE Technologies For Industry and ANSYS Italian Conference 2009 (2009)

[17] Lohn, J.D., Kraus, W.F., Hornby, G.S.: Automated Design of a MEMS Resonator. In: Proceedings of the Congress on Evolutionary Computation, pp. 3486–3491 (2007)

[18] Poles, S.: MOGA-II An Improved Multi-Objective Genetic Algorithm. Technical report 2003-006, Esteco, Trieste (2003)

[19] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schonauer, M., et al. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)

[20] Benkhelifa, E., Farnsworth, M., Bandi, G., Tiwari, A., Zhu, M., Ramsden, J.: Design and Optimisation of Microelectromechanical Systems: A Review of the State-of-the-Art. International Journal of Design Engineering, Special Issue Evolutionary Computing for Engineering Design (2009) (accepted to be published)

[21] Zhang, Y., Kamalian, R., Agogino, A.M., Séquin, C.H.: Design Synthesis of Microelectromechanical Systems Using Genetic Algorithms with Component-Based Genotype Representation. In: Proc. of GECCO 2006 (Genetic and Evolutionary Computation Conference), Seattle, July 8-12, vol. 1, pp. 731–738 (2006) ISBN 1-59593 187-2

[22] Poloni, C., Pediroda, V.: GA coupled with computationally expensive simulations: tools to improve efficiency. In: Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, pp. 267–288. John Wiley and Sons, England (1997)