Microsystem Technologies (2021) 27:2889–2901
https://doi.org/10.1007/s00542-020-05194-w

TECHNICAL PAPER

Two-objective metaheuristic optimization for floating gate transistor-based CMOS-MEMS inertial sensors

B. Granados-Rojas[1] · M. A. Reyes-Barranca[1] · Y. E. González-Navarro[2] · G. S. Abarca-Jiménez[3] · M. A. Alemán-Arce[4] · S. Mendoza-Acevedo[4] · L. M. Flores-Nava[1]

Received: 3 September 2020 / Accepted: 16 December 2020 / Published online: 4 January 2021

Abstract
In this work, a study case is presented in which the design of the layout for a CMOS sensor cell is partially automated by implementing a metaheuristic algorithm to find the best tradeoff between two conflicting objectives (two quantitative opposite and not totally independent yet desired performance or design qualities) among the set of feasible layout and electronic device configurations within a constricted search space. The feasibility of a solution (a particular configuration) and its capability to fulfill every requested objective, is determined by its compliance to the CMOS-MEMS design rules and fabrication process. Any given solution besides showing optimal or very near-to-the-optimal characteristics, must be suitable to be fabricated in the CMOS conventional process which for this case is a 0.5 μm, 3-metal 2-poly N-well fabrication, beside this, since monolithic inertial sensors generally contains embedded movable electromechanical parts a surface micromachining must be considered. Simulation data and behavior of the bio-inspired metaheuristic algorithm used during the design process are presented, as well as electromechanical simulation results based the automatic-generated solutions.

Keywords MEMS · CMOS-MEMS · Genetic algorithm · Floating-gate · FGMOS · Optimization

1 Introduction

Inertial sensors such as accelerometers are nowadays present in many and varied industries and research areas ranging from IoT to vehicle safety and aerospace developments, but also in consumer electronics as they can be found in most communication and entertainment devices. This kind of instruments can be catalogued as MEMS due to their dimensions and close relation with the integrated circuit fabrication process. At a microfabrication level, MEMS are generally conformed by two subsystems: one or more microelectromechanical structures with mechanical, thermal, optical, and related properties and the associated electronic circuit for signal conditioning and processing.

✉ B. Granados-Rojas
bgranadosr@cinvestav.mx

M. A. Reyes-Barranca
mreyes@cinvestav.mx

Y. E. González-Navarro
ygonzalezn@ipn.mx

G. S. Abarca-Jiménez
gabarcaj@ipn.mx

M. A. Alemán-Arce
maleman@ipn.mx

S. Mendoza-Acevedo
smendozaa@ipn.mx

L. M. Flores-Nava
lmflores@cinvestav.mx

[1] Electrical Engineering Department, CINVESTAV IPN, Av. Instituto Politécnico Nacional 2508, 07360 Mexico City, Mexico

[2] Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas - Instituto Politécnico Nacional, Av. Instituto Politécnico Nacional 2580, Gustavo A. Madero, 07340 Mexico City, Mexico

[3] Unidad Profesional Interdisciplinaria de Ingeniería Campus Hidalgo - Instituto Politécnico Nacional, Boulevard Circuito La Concepción 3, San Agustín Tlaxiaca, 42162 Hidalgo, Mexico

[4] Centro de Nanociencias y Micro y Nanotecnología - Instituto Politécnico Nacional, Av. Luis Enrique Erro s/n, Gustavo A. Madero, 07738 Mexico City, Mexico

These two stages in the measurement system are not always intended to be integrated in a single chip since the IC fabrication process might not be fully compatible with the electromechanical counterpart. In accelerometers for example, the mobile proof mass (which serves as a varactor proportional to its displacement) is usually thicker than those metal layers interconnecting the IC in-chip electronic devices. As seen in Fig. 1, a typical MEMS sensor consist of two separated chips which leads to higher costs and additional prototyping issues such as wire bonding. An alternative to the solution described above is a single-chip (monolithic) CMOS-MEMS system which integrates both electronic and electromechanical capabilities in a single silicon die.

Nevertheless, a mechanism to effectively transduce one physical quantity into another of electrical nature within the integrated chip must be considered. As reported in Abarca-Jimnez (2018) and Abarca-Jimenez (2013) the floating-gate MOS transistor (FGMOS) is suitable to be the transducer element in accelerometer designs.

The capacitive coupling attribute of the FGMOS transistor will be further discussed as its high dependency to the particular topology and layout geometry established by the design engineer is what would make it interesting to be computationally optimized by means of metaheuristic methods and algorithms.

The present study case represents first approach from our group to the Evolutionary Multi-Objective Optimization (EMOO) in the field of CMOS layout design. State-of-the-art bio-inspired algorithms usually deal with way more sophisticated design and computational problems and are continuously improved and compared one to another. The selected parameters and characteristics in the applied algorithm may not go beyond a proof-of-concept context and are not intended to optimize the performance of the algorithm itself but to find a preliminary design solution.

The so called genetic algorithms are a versatile group of bio-inspired metaheuristic techniques based on the natural evolutionary process that along with other Evolutionary Multi Objective Optimization (EMOO) methods are commonly used in engineering for optimization (Coello-Coello 2015) and also popular in the field of logistic operations and scheduling (Murata et al. 1996). This kind of iterative search methods are well known to effectively handle very large modeling parameter quantities in problems with up to more than a few thousand of variables (Yang et al. 2008) and a couple dozens of objective functions (Lopez-Jaimes and Coello-Coello 2015). This makes the multi-objective evolutionary optimization a fertile research area nowadays. Another research topic in vogue within this framework is the appropriate constraint handling for a given modeled problem, this due to the multiple natural and technical restrictions that the real-world design problems usually involve, aside from many other mathematical models with not necessarily physical counterparts, carefully designed to benchmark different algorithms and methods.

When it comes to mechanical, microelectronic or other geometry-based engineering design problems, especially for those where the shape and form factor of the designed element can be deconstructed into smaller parts, each one with its particular parameters set, the main use of genetic algorithms is the codification and random-like selection of the parameters that otherwise can be manually adjusted by the designer. The whole algorithm is a sophisticated computational attempt to emulate the also complex biological process by which a population of individuals combine their genetic information in order to achieve a better overall performance before a given situation, with every generation as a new group of individuals replacing the one that preceded it.

In terms of a genetic algorithm, when applied to an engineering design problem, each variable is codified with a certain number of bits, usually a binary string long enough according to the physical nature of the variable and its useful range in the real numbers within the boundaries of the search space. We call chromosome (Fig. 2) to an array of n variables conveniently defined in length and position that are concatenated in a single binary string, this array contains all the variables involved in the models for one or more objective functions. A chromosome also represents an individual among the population, since any of its particular characteristics can be extracted from the bits sequence and used to evaluate the objective function associated to a design requirement.

For a single-objective problem, when evaluated with a particular combination of either continuous or discretized values for each variable, the n-variable objective function (1) describes the fitness of the selected individual, that is, how well it fulfills the goal of the modeled problem quantitatively. During the evolutionary process, individuals are randomly selected and tried out to determine their fitness, also, mechanisms to determine the best element

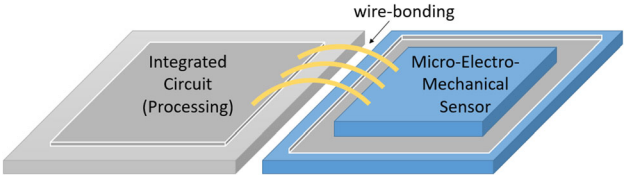

Fig. 1 Typical MEMS sensor consisting in two separated chips

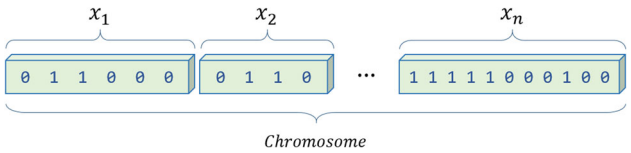

Fig. 2 A chromosome including all codified variables in a single string

within a population for a given generation (elitism), to arbitrarily deviate the tendency avoiding local minima and maxima (mutation) and to combine features of two or more individuals interpolating their chromosomes (crossover) are implemented to iteratively achieve a global optima.

$$fitness = f(x_1, x_2, ..., x_n) \tag{1}$$

In the case of a multi-objective design task, it is only useful to implement this kind of method when every pursued objective is conflicting with one or more other objectives, so increasing the fitness for a particular goal goes in detriment of the complementary objectives. This is why a Pareto-like analysis must be added to the process in order to find the best or most profitable trade-off between objectives. Usually, the nearest point to the ideal solution in the Pareto Frontier (Fig. 3) is selected to be a final solution, since for a problem with objectives in conflict the optimal for every goal cannot be reached at the same time (Coello-Coello 2001). The Pareto Frontier, is constructed from the set of non-dominated solutions for a given search space. A non-dominated solution is that for which there is no other solution better in all the attributes at once. This solution dominate some others by being better in at least one objective and equal or better in all the others.

2 The FGMOS transistor in terms of its geometry

Floating-gate transistors, as well as every other device within the complementary MOS technology, are designed by configuring layers of different materials in a given topological structure. The floating-gate transistor, also known as FGMOS, has been primarily used in the fabrication of digital storage devices (Baker 2005, pp. 113, 466). The information is allocated in the transistor in the form of electric charge that is transferred to and from (programming and erasing) the so called floating gate terminal, the FGMOS main feature is that of including a double polysilicon gate stacked structure (Fig. 4) in which one of them, the closest to the transistor channel, is electrically isolated from other terminals and circuit paths, nevertheless this floating terminal might be electrostatically induced to a potential that will ultimately drive the current across the transistor channel.

When it comes to a conventional n-channel MOS transistor in saturation $(V_{DS} \geq V_{GS} - V_{TH}, V_{GS} \geq V_{TH}, V_S = V_B = 0)$ (Baker 2005, p. 144), the drain current I_D driven through the channel (between drain and source terminals), is given by Eq. (2). Where the device-specific parameter β is given by $\beta = KP_n \cdot W/L$, KP_n is the n-channel transconductance parameter given by the product of the semiconductor electron mobility μ_n and the gate oxide capacitance C_{ox}, W and L are the width and length of the polysilicon gate terminal and therefore of the transistor's channel from a top view. At this point, the channel modulation effects are neglected in this approach in order to simplify the algorithm implementation. Nevertheless, further discussion will show that the channel modulation parameter allows to fully match the obtained results from the algorithm with SPICE device simulations.

$$\begin{aligned} I_D &= \frac{\beta}{2}(V_{GS} - V_{TH})^2 = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - VTH)^2 \\ I_D &= \frac{\mu_n C_{ox}}{2} \cdot \frac{W}{L}(V_{GS} - VTH)^2 \end{aligned} \tag{2}$$

In CMOS-based transistor designs, parameters W and L will take particular relevance since layout geometry is the only field left to the engineer to freely intervene. This applies for most fabrication processes where there are well defined design rules and restrictions including quantity and type of available materials, sheet resistivity and thicknesses for each layer.

In the case of floating-gate transistors, we must consider that the effective applied voltage in the (floating) gate, now on called floating potential V_{FG} is the one driving the drain electrical current I_D and is at the time induced by the potential supplied to the control gate V_{CG} as in Eqs. (3) and

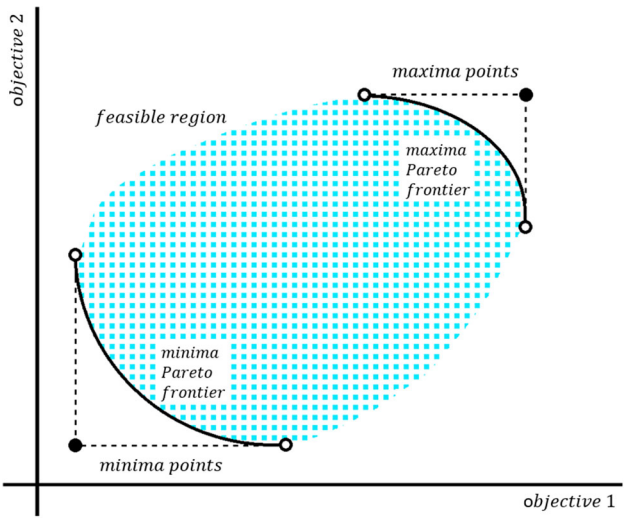

Fig. 3 Location of Pareto-optimal Frontiers and ideal (utopian) optimal point, for a single all-connected feasible region of the search space

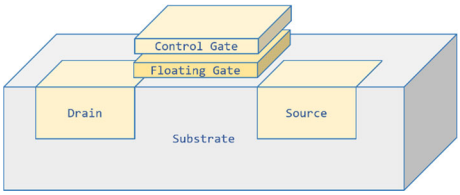

Fig. 4 Typical floating-gate transistor terminals and structure

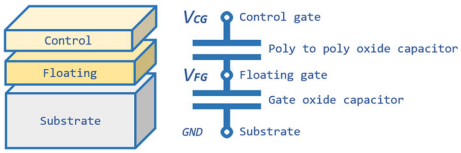

Fig. 5 Floating gate's equivalent capacitive divider

(4) taking into account the capacitive coupling factor k_c deduced from the capacitive voltage divider shown in Fig. 5 and widely discussed in Abarca-Jimnez (2018) and Granados-Rojas (2017).

$$V_{FG} = k_c \cdot V_{CG} \tag{3}$$

$$k_c = \frac{C_{pp}}{C_{pp} + C_{ox}} \tag{4}$$

C_{ox} and C_{pp} are respectively the capacitances between substrate and floating gate and between floating and control gates. In this study case, parasitic capacitors such as gate-to-drain and gate-to-source capacitances are neglected. Regarding further FGMOS analysis, Eq. (2) can be modified as follows:

$$I_D = \frac{\beta}{2}(V_{GS} - V_{TH})^2 = \frac{\mu_n C_{ox}}{2} \cdot \frac{W}{L}(k_c V_{CG} - VTH)^2 \tag{5}$$

At this point, the geometry-dependent parameters are all capacitive and related to the *W*/*L* ratio. A layout top view (Fig. 6) reveals some other details to be considered. As can be seen, due to the design rules and photolithography compliance requirements would not be appropriate to locate the poly-to-poly capacitor right above the gate oxide and gate terminal, instead, the top polysilicon (darker red) terminal lies over in an area right aside (above in the figure) to the transistor active regions, this rectangular pattern

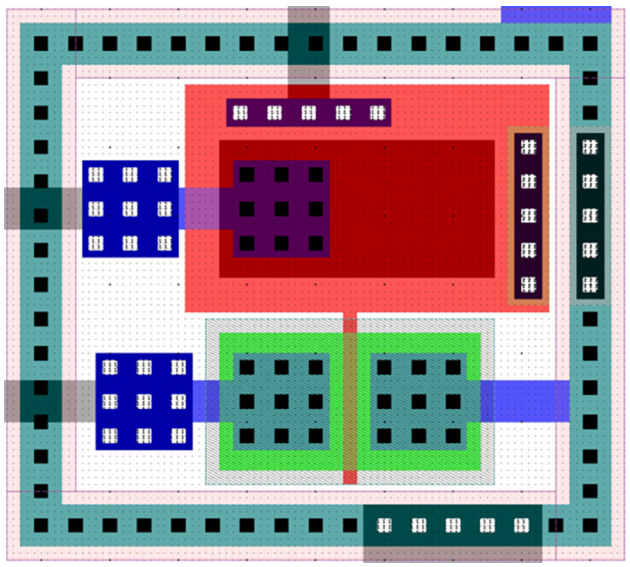

Fig. 6 CAD layout for a floating-gate transistor

in the Poly 2 layer constitutes the control gate and is directly connected to the exterior with via contacts. Floating gate terminal (lighter red) has a geometry pattern consisting in two rectangles, a narrow one $(W \gg L)$ actually forming the n-channel between the drain and source active areas, and other way wider polysilicon plate featuring a convenient area to form a poly-to-poly capacitor. Many connection and terminals seems to connect this plate with non-floating terminals but as is described later the floating gate is completely isolated (surrounded only by dielectric silicon dioxide and air gaps).

Recalling Fig. 6, another thing to point out is that, as for the C5 fabrication process (ON Semiconductor C5X, 0.5 Micron Technology Design Rules 4500099 Rev. X, 2020, p. 99), the gate silicon dioxide t_{ox} with about to 13.5nm is quite thin in comparison to the field oxide t_{fox} $(400nm)$ underneath the floating gate main plate. So, for the capacitive floating factor k_c a two-parallel-capacitor system must be considered in further detailed analysis. As shown in Fig. 7, during the multi-stage photolithography process, the lower polysilicon layer (Poly 1) is grown over two oxide regions with different thicknesses, not involving any significant mechanical issue. Control gate (Poly 2) due to design rules must be drawn with specific margins within Poly 1 surface in order to generate a poly-to-poly capacitor. The dielectric thickness between Poly 1 and Poly 2 layer is $t_{pp} = 39nm$, as for the mentioned fabrication process.

As described in Granados-Rojas (2017), Abarca-Jimnez (2018), and Abarca-Jimenez (2013) floating gate transistor may be used as transductive active elements when it comes to inertial micro-sensors such as capacitive accelerometers, moreover, in Granados-Rojas (2016) alternative interdigital structures are proposed in order to achieve higher sensitivities based on the geometry of the movable parts within the CMOS–MEMS chip or device.

Figure 8 depicts how floating gate gets connected to a CMOS-MEMS variable capacitor while remaining electrically isolated. The movable capacitive structure, also known as varactor consist in two metallic terminals facing each other along the substrate plane, one of them is fixed while the other is coupled mechanically by a spring system and released by means of MEMS surface micromachining procedures to freely move while in presence of accelerated

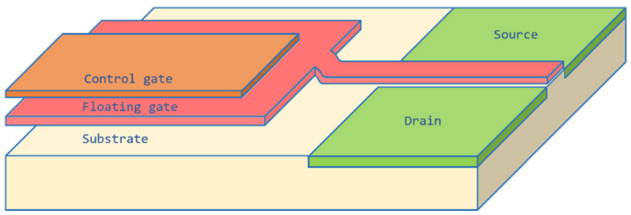

Fig. 7 Poly-to-poly and poly-to-substrate capacitor (interleaved insulating oxide layers not represented)

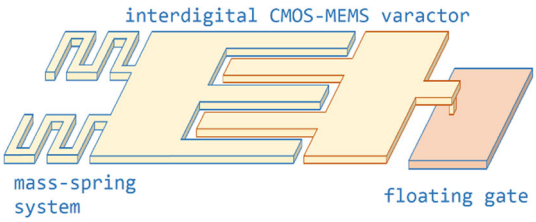

Fig. 8 Schematic for a simplified FGMOS-based CMOS-MEMS inertial sensor

movements. Figure 8 must be considered as a simplification of the actual varactor designed, showing just a few fingers at each end of the varactor and considering only Metal 1.

Figure 9 summarizes the components of a conventional MEMS FGMOS-based sensor subsystem within a CMOS chip. Shall we highlight the inclusion of a usually value–fixed voltage V_Y supplied to the movable terminal of the varactor. At this point the control gate in the poly–to–poly capacitor seen before and the movable capacitive structure, having each their respective applied potentials, works together in the same floating gate reformulating Eq.s (3) and (4) with a weighted additive effect seen in Eqs. (6) and (7), where C_Y as in Eq. (8) is the capacitance in the MEMS varactor from a comprehensive, yet idealistic equation model presented in Granados-Rojas (2016) and Granados-Rojas (2017) that neglects gravitational side effects. Figure 10 shows a micrograph of an actual MEMS capacitive interdigital structure which capacitance in case of a displacement Δy along the y axis can be modeled by Eq. (8).

$$V_{FG} = k_{cY}(V_Y + V_{CG}) \tag{6}$$

$$V_{FG} = \frac{C_Y V_Y + C_{pp} V_{CG}}{C_Y + V_{pp} + C_{ox} + V_{fox}} \tag{7}$$

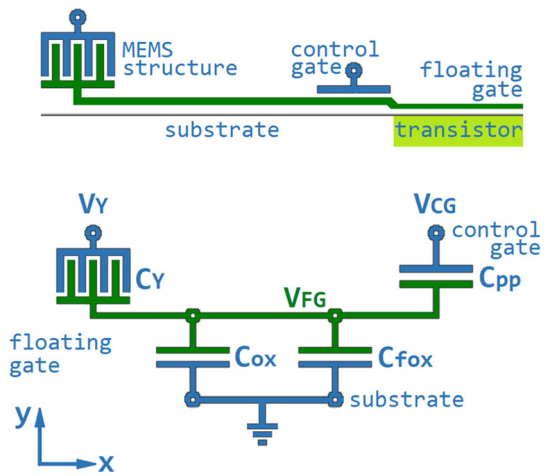

Fig. 9 Components of an FGMOS-based inertial sensor

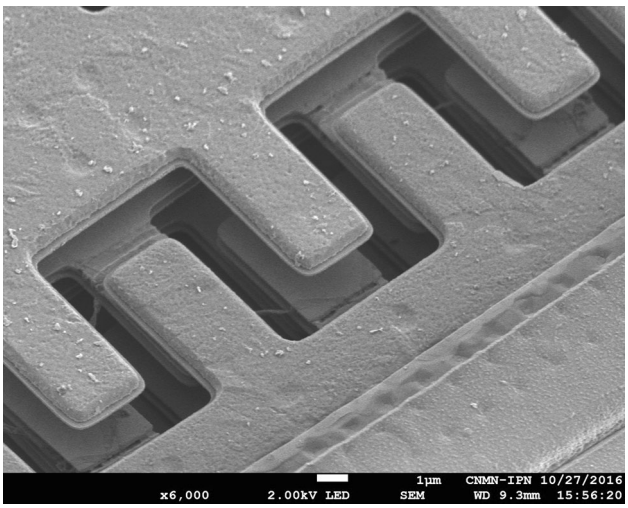

Fig. 10 SEM micrograph of a CMOS-MEMS capacitive structure

$$V_{FG} = \frac{2 \cdot n \cdot \epsilon_0 \cdot W_f \cdot (L_f - \Delta y)}{d_{ox}} + \frac{(n-1) \cdot \epsilon_0 \cdot (t_{M1} + t_{M2} + t_{M3}) \cdot (L_f - \Delta y)}{d_{fin}} + \frac{n \cdot \epsilon_0 \cdot W_f \cdot (t_{M1} + t_{M2} + t_{M3})}{d_{tip} + \Delta y} \tag{8}$$

Figure 10 was obtained by SEM microscopy on a previously micro-machined conventional CMOS chip, the three-layered capacitive structure is the same described in Granados-Rojas (2016), this variable capacitor is intended to achieve about $5 \pm 1fF$ while in presence of $\pm 1G$ of acceleration.

According to the kind of capacitive structures presented before and Eq. (8), the geometry-related variables are n, the number of interleaved fingers, W_f and L_f, the width and length of every finger, respectively, Δy is the displacement due to acceleration in the direction of the y axis as stated in Fig. 9, d_{ox} is the separation gap between metal layers, about $1.1\mu m$ of silicon dioxide separating Metal 1 from Metal 2 as well as Metal 2 from Metal 3 , this encapsulating dielectric is chemically removed by wet etching, d_{fin} is the gap between contiguous fingers along the x axis and d_{tip} is the gap between a fingertip and the plate facing it in the y direction. Beside the variables, the model also consider ϵ_0 for the electrical permittivity constant in air (similar to vacuum) and t_{M1}, t_{M2} and t_{M3}, that are constant parameters as well, for the thickness of all the three metal layers available in the C5 fabrication process from On Semi, 0.64, 0.57 and $0.77\mu m$, respectively.

From knowing the parallel-plate capacitor Eq. (9) we transform it into a geometry-based Eq. (10). Where the i-th area A_i of a capacitor C_i to be modeled is given by their dimensions W_i and L_i in the substrate plane and the

thickness of dielectric oxide between conductor or semiconductor layers. Equation (10) represents the floating potential actuating on the transistor channel conduction in terms of the topological design parameters, being L_{pp} and W_{pp} the dimensions of the poly-to-poly capacitor with dielectric thickness t_{pp}.

$$C_i = \frac{\epsilon_0 A_i}{d_i} = \frac{\epsilon_0 W_i L_i}{t_i} \tag{9}$$

$$C_i = \frac{C_Y V_Y + \frac{\epsilon_0 W_{pp} L_{pp}}{t_{pp}} V_{CG}}{C_Y + \epsilon_0 \left[\frac{W_{pp} L_{pp}}{t_{pp}} + \frac{W_{fox} L_{fox}}{t_{fox}} + \frac{W_{ox} L_{ox}}{t_{ox}}\right]} \tag{10}$$

This data set instead of proposed by the layout designer might be optimized by means of implementing a heuristic algorithm.

3 The two conflicting optimization objectives

As seen in Granados-Rojas (2018) high sensitivity is a desired characteristic of an inertial sensor. For an accelerometer, this parameter can be defined in terms of how much the output signal changes according to the ongoing acceleration magnitude. A given acceleration produces a displacement in the sensor proof-mass which is usually suspended by two or more metallic springs, let's consider the structure shown in Fig. 11 as the one providing mechanical movement. Since the optimization of the dimensions and properties of each spring is out of the scope of this two-objective analysis our design area comprises only the interdigital capacitive structure (varactor) and the dimensions of the elements conforming the floating-gate transistor.

Considering that any capacitive magnitude is proportional to the area of the conductive (or semiconductive)

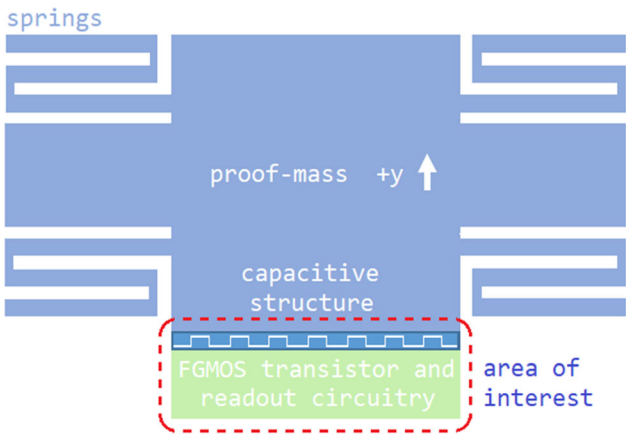

Fig. 11 Mass-spring system with four springs and one capacitive MEMS structure at the bottom

surfaces involved, and also being the drain current I_D proportional to the floating potential V_{FG} and therefore proportional to the capacitances in both the MEMS structure and the active device itself, we are in conditions to state that the increment of the sensitivity S_{acc} is in conflict with the reduction of the design area A_D.

In other words, at a given proof-mass displacement (due to an acceleration), the larger the capacitive structure is, the bigger the changes in the capacitance, the floating voltage, and the drain current what corresponds to an increment in the sensitivity (desired to be greater) defined as seen in (11) and (12). But, in the other hand, if a capacitive structure is big so it is the design area (desired to be smaller).

To have both a very sensitive device fabricated in a very small area is quite idealistic. This work is an approach to find topological and geometrical parameters systematically selected to be in agreement with a fair good tradeoff between both objectives.

$$S_{acc} = \frac{\partial I_D}{\partial C_Y} \tag{11}$$

$$S_{acc} = KP_n \cdot \frac{W}{L} \cdot \left(\frac{V_{MS} C_{MS} + V_{CG} C_{CG}}{C_{MS} + C_{CG} + C_{ox} + C_{fox}} - V_{TH}\right) \cdot \left(\frac{V_{MS} C_{CG} + V_{MS} C_{ox} - V_{CG} C_{CG}}{\left(C_{MS} + C_{CG} + C_{ox} + C_{fox}\right)^2}\right) \tag{12}$$

Once the floating voltage V_{FG} in (10) is reformulated for the structure in Fig. 9 and included in expression (5), the sensitivity to an acceleration S_{acc} in (12) must be interpreted as the change in drain current due to a change in the capacitance of the MEMS structure, being C_Y the only capacitive variable term. This sensitivity is measured in $\mu A/fF$. On the other hand, expression (13) is the design area (needed to be reduced) that is the accumulative of the different regions of interest including the transistor channel, the polysilicon plate of the floating gate, and the multiple surfaces of the varactor structure the air gaps between them. The area occupied by interconnecting lines, vias, pads and other active regions is discarded at this stage in order to simplify the models and focus in the role of the FGMOS as element of transduction.

$$A_D = WL + W_{FG} L_{FG} + n\left(W_F L_F + 2d_{tip} W_F\right) + (n-1)\left(d_{fin}\left(L_F + 2d_{tip}\right)\right) \tag{13}$$

In agreement with Fig. 12, the design area A_D depends on the width W_{FG} and length L_{FG} of the floating-gate plate, but also in the dimensions of the capacitive structure mentioned before, as can be seen, the dimensions W_{CG} and L_{CG} of the control gate do not affect the resulting area since they are inscribed into the floating-gate area. For the

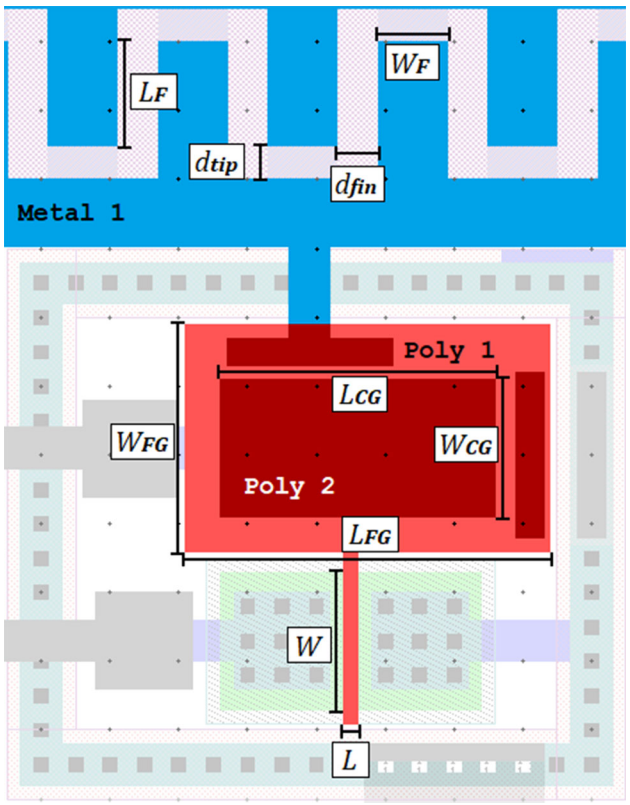

Fig. 12 Topological variables in the FGMOS and the MEMS multi-layer capacitive structure

transistor itself we are right now taking into account only the channel and not the whole active area (drain and source) since we are considering this area is highly dependent in the rules for the specific fabrication process (layer overlaps, number of active contacts, etc.) and they do not have a significant effect in the electrical sensitivity model as the channel area certainly does.

4 Codification of variables

The study case presented is characterized as a 2-objective, 11-variable optimization problem, each variable represents a geometrical parameter either of the FGMOS or the MEMS variable capacitor and each one of these codified in a convenient way so the genetic operator of the meta-heuristic algorithm can be applied. The selected codification is to represent each variable with the binary Gray code string, it is important to consider Gray codification due to its inherent minimum Hamming distance (Bykov and Aleksei L'vovich Perezhogin 2017) between consecutive elements of the discrete variable. Moreover, the discrete nature of the layout design and fabrication process for CMOS technology allows to describe any magnitude as an integer multiple of the layout unit λ ($0.3\mu m$ in the C5 fabrication process). For example, for Metal 1 layer the minimum feature is a wire $3\lambda = 0.9\mu m$ long, this can be considered the offset and assigned to the Gray zero for the variable, now, let's say we are not interested in wires or plates larger than $9\mu m$, this is $8.1\mu m = 27\lambda$ away from the offset and a 5-bit string is needed to codify the variable where 00000_{Gray} is equal to $0.9\mu m(3\lambda)$, 00001_{Gray} is $1.2\mu m$ (4λ) and so on up to $10110_{Gray} = 11011_b$ that represents $9\mu m = 30\lambda$. Table 1 summarizes the variable collection with their binary length and respective boundaries within the search space of the algorithm.

These binary strings are concatenated in a single string known as chromosome (49-bit long) and any given chromosome represents a unique individual of the simulated population, by decoding it (retrieving the value of the variables back) and evaluating the objective functions with its numerical values we obtain the fitness for the particular individuals. By the ways, there might be binary string which numerical attribute is out of the boundaries of a given variable, these individuals must be treated as non-feasible solutions to the problem and filtered out of the population by means of restriction compliance mechanisms, preventing the population to evolve into that direction.

5 Bio-inspired metaheuristic algorithm

We define the search space of our algorithm as all the possible points within the boundaries of our restrictions. Restrictions are set regarding compliance with design rules, limitations in the fabrication process or any other facts known to be impractical. After several evaluations of the

Table 1 Range and codification of variables

Var	Range	Min	Max	Bits
W	$10-30\lambda$	3 µm	9 µm	5
L	$2-10\lambda$	0.6 µm	3 µm	4
W_{FG}	$10-30\lambda$	3 µm	9 µm	5
L_{FG}	$10-30\lambda$	3 µm	9 µm	5
W_{CG}	$5-25\lambda$	1.5 µm	7.5 µm	5
L_{CG}	$5-25\lambda$	1.5 µm	7.5 µm	5
n	5 – 25	5 fingers	25 fingers	5
W_F	$5-20\lambda$	1.5 µm	6 µm	4
L_F	$5-30\lambda$	1.5 µm	9 µm	5
d_{tip}	$3-10\lambda$	0.9 µm	3 µm	3
d_{fin}	$3-10\lambda$	0.9 µm	3 µm	3
		Total chromosome length		49

objective functions (12) and (13) with randomly generated values for the 11-variable chromosome string yet observing the correspondent restrictions such as keep the control gate smaller than the floating gate, we reveal an approach to the shape of the search space in Fig. 13. This data set is treated with a preliminary scaling factor in order to get rid of many decimals as both Design Area and Sensitivity are quite small values.

As described before, we cannot achieve optimal values for two or more objective function in conflict at the same time, the main goal of this implementation is to achieve the Pareto Optimality (Censor 1977) which is related to the best tradeoff between conflicting premises. When using this kind of strategy it is convenient to minimize both objective functions, therefore, for those where we actually want to get greater values, i.e. sensitivity, we invert the objective function and look for the minimum. So, as for figures 3 and 13 we will find our viable solutions in the left-bottom end.

Figure 14 summarize the data flow and key features of the implemented algorithm. Most of the blocks are common to many bio-inspired meta-heuristics especially those based on genetics and evolution of populations. This algorithms uses the so called secondary population which is a repository file used to store and update the non-dominated solutions of each generation, after further non-dominated selections on the secondary population we obtain the effective Pareto Frontier.

There are three main processes all along the algorithm, some of them occur in more than one situation or context:

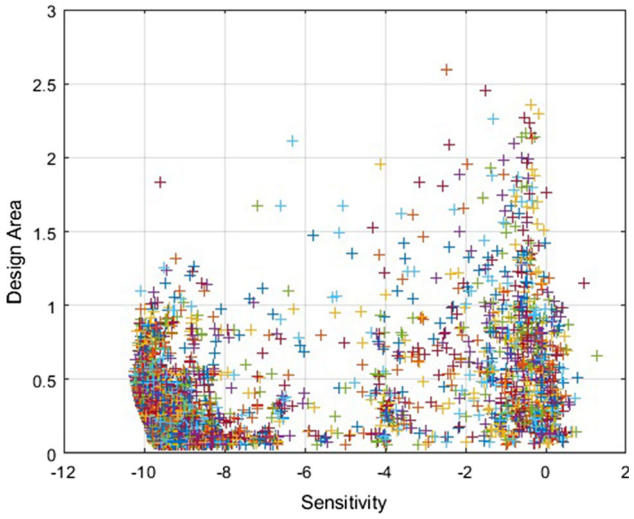

Fig. 13 Approach to the search space

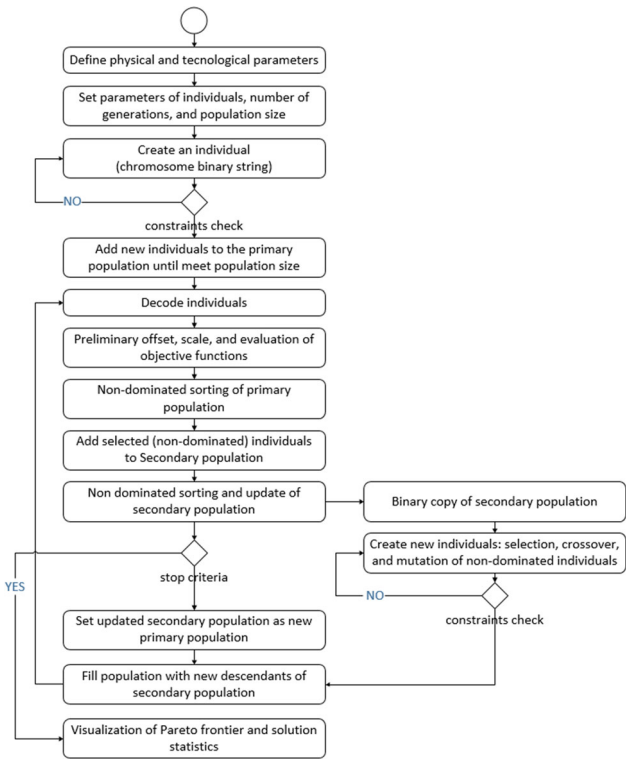

Fig. 14 Flow chart of the genetic algorithm used

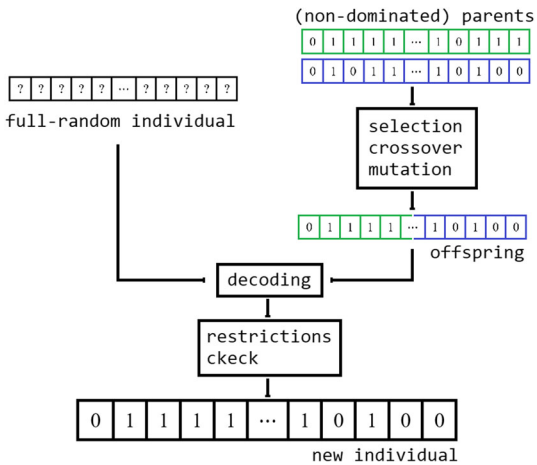

Fig. 15 Creation of new individuals

5.1 Creation of individuals

As depicted in Fig. 15, all new individuals in the population come from one of two sources and to be included is because it has already cleared a restriction check (so it belongs to the feasible region). The first source of individuals is used only for the initial population (in the first generation) and consist in randomly generate binary bits with a uniform probability. For the second generation and on, new individuals are created out of a set successful (non-dominated) individuals from the previous generation by

means of the selection and crossover of parents and the mutation (binary toggle) of one of bits of the generated offspring. Both crossover and mutation processes control the diversity among individuals and are tuned by probability of occurrence parameters with quite high (75%) a very low values (1%), respectively, in this study case.

5.2 Decoding and evaluation

Decoding and evaluation of an individual is needed to know its particular values of fitness for each objective function and therefore its position in the search. Figure 16 illustrates how by decoding predefined segments of the binary string we obtain an array of real-number parameters related to every physical parameter included in the modeling of the problem. Next, we evaluate the objective functions with the numerical values of each variable and the resulting fitness values can be represented as the coordinates of the individual in the search space.

5.3 Non-dominated sorting

The non-dominated sorting is a one-on-one comparison between all the elements of a uniform data set, in this case, all the individuals of a given population.

As in Fig. 17a, when looking for minimum values in each attribute our optimal solutions will be located in the left-bottom end, a given solution sol 1 dominates its whole right-upper quadrant by being better or equal than all those other solutions in both attributes, so does sol 2 with its respective right-top vicinity. The Pareto Frontier is the set of solution in the nearest region of the idealistic optimal characterized for consist in points that do not belong to the dominance region of any other solution, this is why they are called non-dominated, however, one or more of these non-dominated solutions are the closest to the idealistic optimal and determining its Euclidean distance is often a

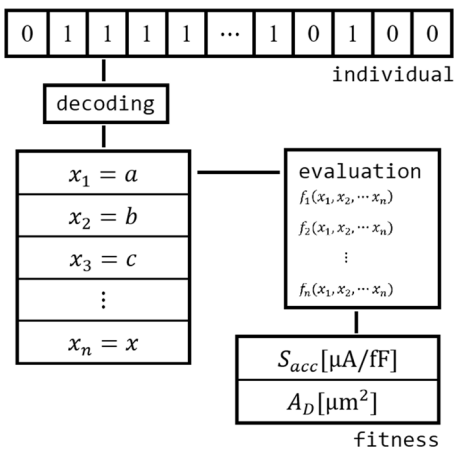

Fig. 16 Decoding and evaluation of individuals

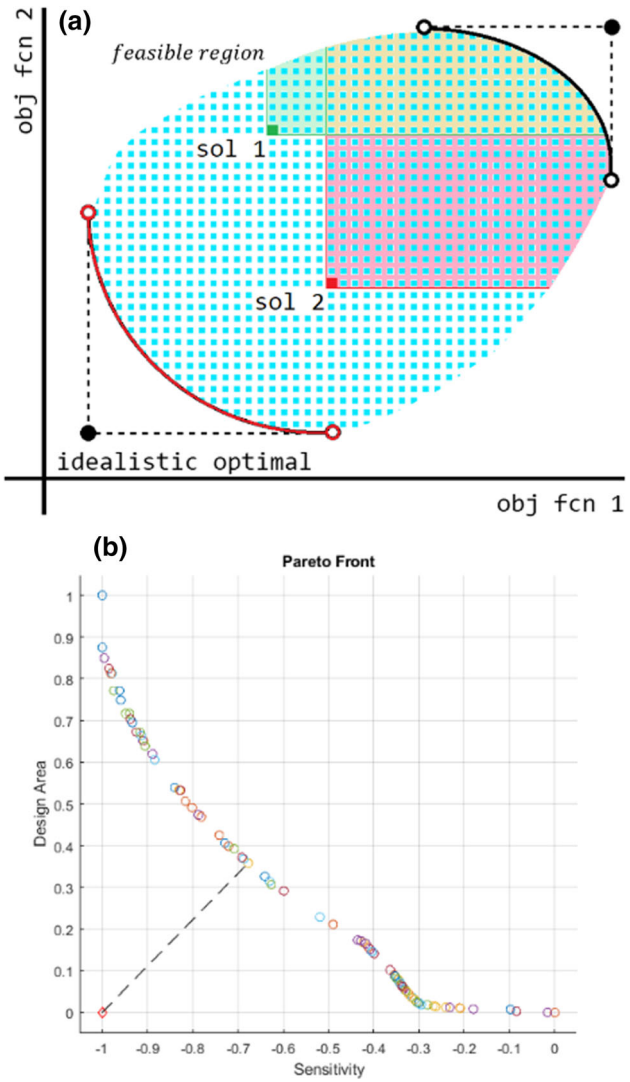

Fig. 17 **a** Set of non-dominated solutions, **b** the closest solution in the Pareto Frontier to the idealistic optimal for a given execution

criteria to choose a final solution. The idealistic optimal is a coordinated point calculated with the overall minimum values reached in the data set even when the associated individuals are far from it.

The graph presented in Fig. 17b is one of many obtained while looking for a better solution to this design optimization problem, it represents the Pareto Frontier calculated after ten executions (program runs) of the main algorithm with a population of 1000 individuals evolving during 100 generations. Due to the stochastic nature of the general process it is convenient to merge the information of many executions and perform a new sorting to get a better approach to the real Pareto Frontier. Figure 17b also shows the selected individual which is the one with the shortest distance to the idealistic optimal. For this measurement a second scaling is performed, a normalization within the minimum and maximum for both objectives so all fitness

values are mapped into a 0 to 1 range. In order to retrieve the selected solution back and decode its chromosome, all individuals must be labeled with their original position in the population array and execution number and keep this data intact during the non-dominated sorting processes.

6 Results and further simulation

This work consists of a series of tests of the algorithm, in order to support the one result ultimately analyzed six sets of tests were considered with results summarized as follows in Table 2.

The highlighted data corresponds to a set of only five executions that excels all other by reaching a way more dense and close Pareto Frontier, even when the sensitivity calculation is slightly lower, the design area saving is significant in relation to other results. In Fig. 18 all the five execution are plotted together in terms of the amount of non-dominated solutions found in every generation, the average of solution stored for this set is about 119 per run, for a total in the set F of 594 solutions, after the last non-dominated sorting process, only 160 solutions remained in the Pareto Frontier, Fig. 19 reports the contribution of each execution, as can be seen, approximately 70 of the frontier point were calculated in run #2 while the entire curve generated from run #3 was outperformed by the rest.

The selected solution in the previous step is processed to decode and reinterpret its associated binary string, for this particular case, the solution #41 of the fourth run was the one with the best performance, this individual correspond with the parameter list shown below in Table 3.

Finally in the semi-automated design process we obtain a plot (Fig. 20) for the definitive Pareto frontier, in this

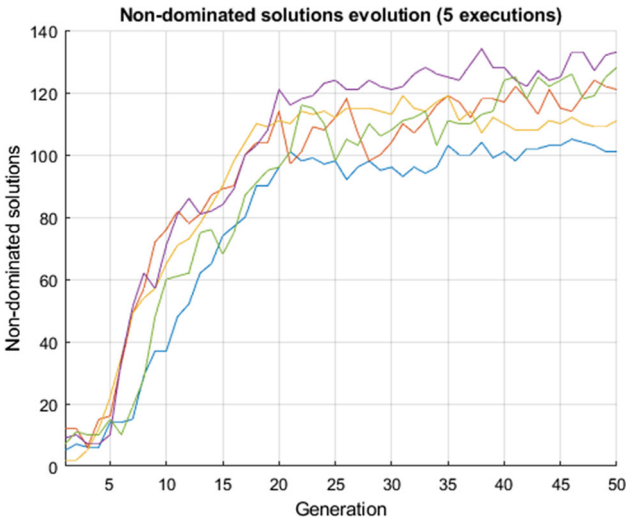

Fig. 18 Evolution of the number of non-dominated solutions

normalized scale the sensitivity is about the 75% of the maximum feasible and the design area drops to near the 10% of the largest calculated, it is up to the design engineer to decide if the tradeoff is good enough for any particular purposes.

The selected solution in the previous step is processed to decode and reinterpret its associated binary string, for this particular case, the solution #41 of the fourth run was the one with the best performance, this individual correspond with the parameter list shown below in Table 3.

Further electromechanical and electrical simulation can be done in order to validate the expected performance. By means of FEA software we get an estimate of the capacitive properties behavior of the design proposal. Figure 21 shows the 3D model of the MEMS capacitive structure along with its stationary analysis for capacitance.

Table 2 Summary of test results

Set	Total execs	Pop size		ND sols found	
A	10	1000	100	83	$S_{acc} = -101.55$ $D_A = 328.32$
B	10	2000	100	69	$S_{acc} = -102.144$ $D_A = 380.16$
C	10	1000	200	71	$S_{acc} = -101.414$ $D_A = 313.740$
D	20	1000	100	79	$S_{acc} = -101.018$ $D_A = 328.640$
E	10	3000	100	87	$S_{acc} = -101.872$ $D_A = 355.230$
F	5	1000	50	160	$S_{acc} = -98.164$ $D_A = 121.416$

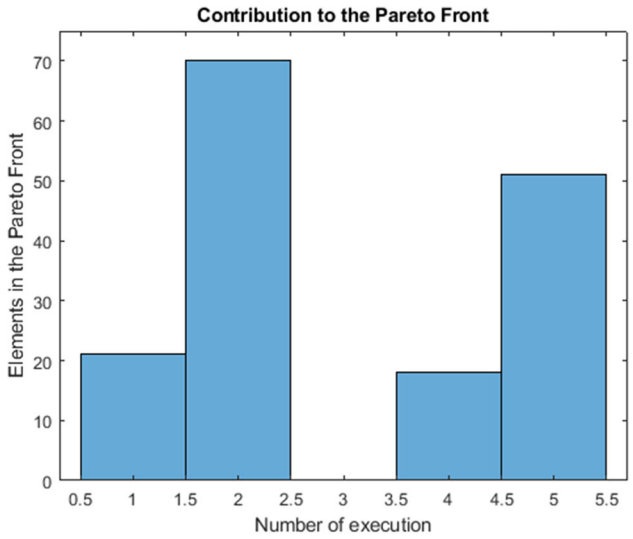

Fig. 19 Contribution of each execution to the Pareto Frontier according to the number of non-dominated solutions

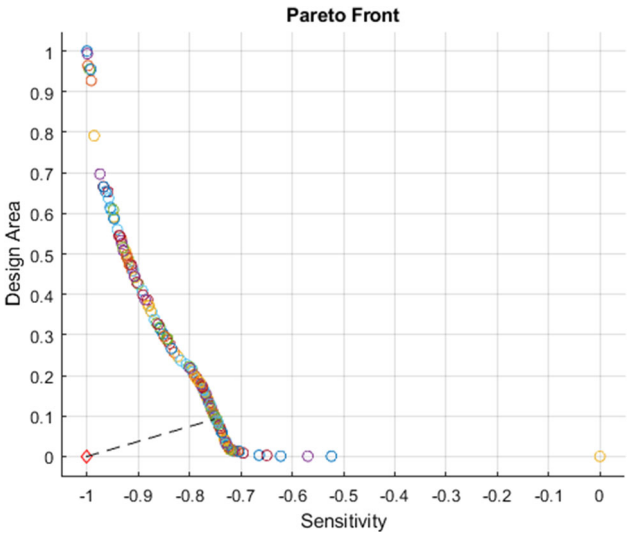

Fig. 20 Pareto Frontier

Table 3 Topology of the final solution

Parameter	Value	Units
Channel width	9.0	μm
Channel length	0.6	μm
Floating gate width	3.6	μm
Floating gate length	5.4	μm
Control gate width	2.7	μm
Control gate length	4.5	μm
Number of fingers	5	μm
Finger width	1.5	μm
Finger length	6.9	μm
Tip gap	0.9	μm
Finger gap	0.9	μm

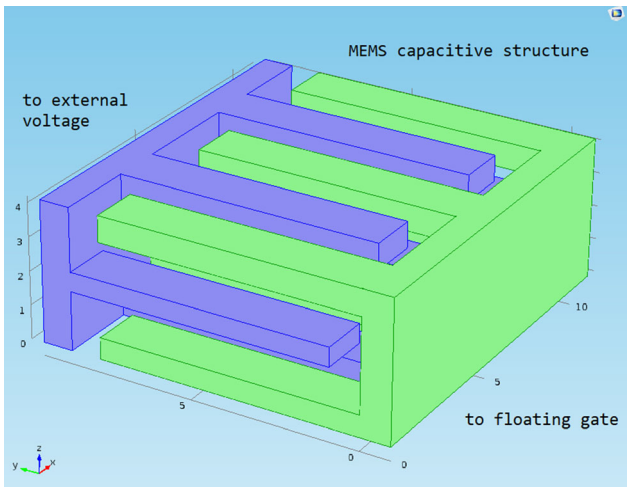

Fig. 21 CAD model of the 3-layered interdigital capacitor

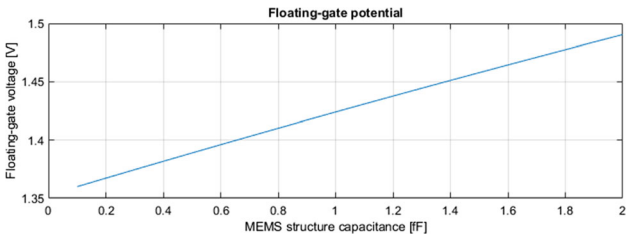

Fig. 22 Floating potential according to the MEMS structure capacitance

As for Eqs. (8) and (10) with this set of parameters the capacitance in the MEMS structure at zero displacement (and therefore 0G acceleration) is $C_Y = 1.099fF$, what produces a floating-gate potential of $V_{FG} = 1.43V$ and a static value for drain current of $I_D = 584.21\mu A$. Figure 22 shows the correlation between the variable capacitance and the floating gate potential, as well as Fig. 23 depicts the result of SPICE simulations based on the BSIM Level 1 that is in agreement with the Shichman-Hodges model (Razavi 2002, p. 592), in simulation a potential is applied to the gate terminal in an NMOS transistor for both a full-range DC sweep and a transient analysis in the time domain, in the later, the voltage at the transistor gate is applied according to what the floating potential should be when an full-displacement-range oscillatory acceleration changes the varactor capacitance.

7 Conclusions

This particular algorithm is proven to be an effective tool intended to assist the design engineer at the first stages of an ASIC project, it allows to add partial yet comprehensive automation and computational decision-making elements

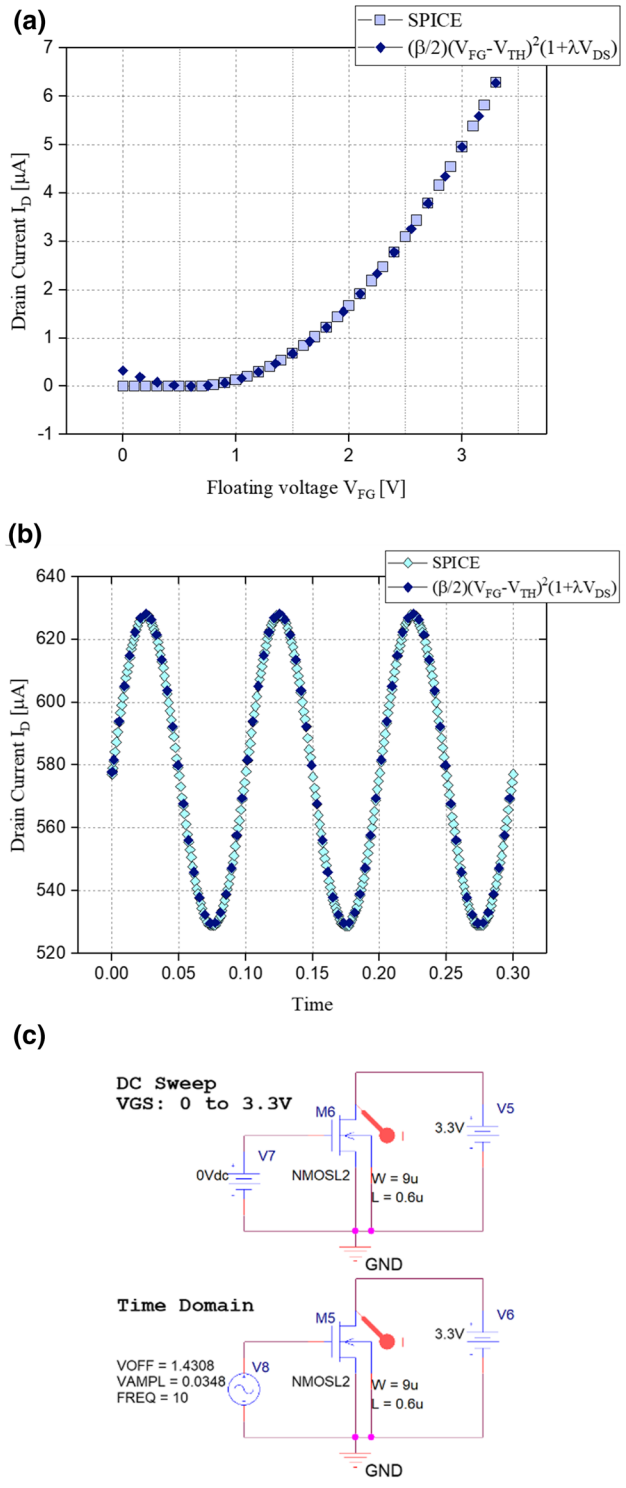

Fig. 23 **a** Match between the SPICE simulation and the drain current model used during the evaluation of the algorithm, **b** approach to the response to an oscillatory input within the range of displacement of the capacitive structure, **c** setup for the SPICE simulations

to the CMOS standard layout design process. The results retrieved by the computational system can for sure be taken as a start point for the parameter selection at a given requirement having in mind the opportune addition of minor adjustments and user preferences. As treated in this work, the automation characteristics can be extended to micro-electro-mechanical components where not only electrical but also mechanical parameters are included.

By the other hand, the cost of increase the degree of confidence in the algorithm and the relevance of its results is to consider a bigger collection of variables and parameters for the mathematical modeling of the objective functions. Nevertheless, even when having access to more detailed, realistic and exact data, handle too many parameters is often impractical and may require of especial attention at the moment of confect the appropriate models and configure the bio–inspired mechanisms to be used. As an example, even when we used the term search space for the two-dimensional projection of possible solutions in the objective-function space, in this study case the actual search space is a 11-dimensional one where reside all the feasible configuration of a 49–bit long representative vector. As simple as this engineering problem is in comparison with state-of-the-art computational benchmarks we strongly recommend to take into account the rapid rise in the complexity of the implementation as the detail in the modeling of the problem grows.

As true for most bio-inspired meta-heuristic algorithms, the convenience of using this kind of techniques must be pondered according to the number of variables and objective functions to handle. Meta-heuristics are often said to be the last option when having access to a wide catalogue of classical optimization tools many of them way simpler to implement while saving computational resources.

This work presented valuable information and results to extend the outreach of ongoing VLSI and CMOS-MEMS projects, future work will be oriented to explore the optimization of a number of mechanical component and capabilities, go deeper into the electrical models and look for suitable applications.

Acknowledgements Authors would like to thank to the VLSI group members at Cinvestav for the support given to this project and to Conacyt for the scholarship #295930 granted to B. Granados-Rojas.

References

Abarca-Jimenez GS, Reyes-Barranca MA, Mendoza-Acevedo S (2013) MEMS capacitive sensor using FGMOS. In: 2013 10th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2013, pp 421–426

Abarca-Jimnez GS et al (2018) Inertial sensing MEMS device using a floating-gate MOS transistor as transducer by means of modifying the capacitance associated to the floating gate. Microsyst Technol 24(6):2753–2764

Jacob BR (2005) CMOS: circuit design, layout, and simulation

Bykov IS, Aleksei LP (2017) On distance Gray codes. J Appl Ind Math 11(2):185–192

Censor Y (1977) Pareto optimality in multiobjective problems. Appl Math Optim 4(1):41–59

Coello-Coello CA (2001) A short tutorial on evolutionary multiobjective optimization. In: International Conference on evolutionary multi-criterion optimization. Springer, pp 21–40

Coello-Coello CA (2015) Multi-objective evolutionary algorithms in real-world applications: some recent results and current challenges. In: Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences. Springer, pp 3–18

Granados-Rojas B et al. (2016) 3-layered capacitive structure design for MEMS inertial sensing. In: 2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp 1–5

Granados-Rojas B et al. (2017) Basic readout circuit applied on FGMOS-based CMOS-MEMS inertial sensing prototypes. In: 2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp 1–6

Granados-Rojas B (2018) Application and Resulting Suitability of a Genetic Algorithm in the Design of FGMOS-based CMOS-MEMS Transducers. In: et al (2018) 15th International Conference on Electrical Engineering. Computing Science and Automatic Control, CCE, p 2018

Lopez-Jaimes CC (2015) Many-objective problems: challenges and methods. Springer handbook of computational intelligence. Springer, New York, pp 1033–1046

Murata Tadahiko, Ishibuchi Hisao, Tanaka Hideo (1996) Genetic algorithms for flowshop scheduling problems. Comput Ind Eng 30(4):1061–1071

ON Semiconductor C5X (2020), 0.5 Micron Technology Design Rules 4500099 Rev. X, p 99

Razavi B (2002) Design of analog CMOS integrated circuits. Tata McGraw-Hill Education, London

Yang Z, Tang K, Yao X (2008) Large scale evolutionary optimization using cooperative coevolution. Inform Sci 178(15):2985–2999