New Genetic Algorithm Approach For Dynamic Biochemical Sensor Measurements Characterization

Deepak Gantla, Hoda S. Abdel-Aty-Zohdy, Robert L. Ewing*

Microelectronics System Design Laboratory
Department of Electrical & Systems Engineering
Oakland University, Rochester, MI, USA.

*Wright Patterson Air Force Base
Information Directorate
WPAFB, OH, USA.

Abstract

In an olfaction system (E-Nose) hardware implementation, outputs from the GA approach are used as inputs to an intelligent neural network system for biochemical detection and decision-making. In this paper we present a Genetic Algorithm for measurement characterization with dynamic inputs. Input measurements are from a given range and are assumed in parallel from chemical-sensor array. An input multiplexer/controller/Analog-Digital converter preprocessing stage is used to control these input measurements. The new dynamical approach presents measurement characterization and also optimum fused measurements without loosing the integrity of incoming signals. Through a novel mutation and crossover approach (Half Sibling and A Clone) optimum characteristic weight chromosomes are achieved. HSAC represents both crossover and mutation. A key feature of the new approach is that no pre- assigned minimum error is specified, rather error is dynamically evaluated based on measurements. Simulation results of the new GA with dynamic measurements are compared with one of the approaches from GAlib (A library of genetic Algorithm approaches from MIT) and proved the new approach has minimum error and a early convergence. MATLAB has been used as the simulation tool.

1. Introduction

Genetic algorithm (GA) is a search and optimization technique, based on evolutionary principle of natural chromosomes. Specifically, the evolution of chromosomes due to the action of crossover, mutation and natural selection of chromosomes based on Darwin's survival-of-the-fittest principles are mostly used to constitute a robust search and procedure.

GAs encode each point in a parameter space into a binary bit string called a ***chromosome***, and each point is associated with a "fitness" value that, for maximization is usually equal to the objective function evaluated at the point instead of a single point, GA usually keeps a set of points as a ***population***, which is then evolved repeatedly towards a better overall fitness value. In each ***generation***, the GA constructs a new population using ***genetic operators*** such as crossover and mutation; members of higher fitness values are more likely to survive and to participate in mating (crossover) operations. After a number of generations, the population contains members with better fitness values.

In [1], a GA system for optimal sensor measurements was proposed. In this system an effective mutation and crossover scheme Half Sibling and A Clone (HSAC) has been developed, simulated and implemented. The HSAC approach is applied at various iteration levels and kept a dynamic error to a minimum. By this scheme, a single converging chromosome value is achieved within 10 iterations. The new approach considers input signals as controlled parallel 128 measurements. Measurement characterization and optimum fused measurements without loosing the integrity of incoming signals are achieved by assigning initial random weights to each feature of each incoming sensor measurement. Vectors of these weights are considered the chromosomes in our evolutionary approach. Dynamic inputs represent real life sensor measurements (readings from biochemical sensor).

2. Basic Operation

A GA uses the fitness criteria to determine the best choice of weights that should be applied to input data in order to achieve measurements characterization. GAs randomly select solutions from a predefined solution space, and start applying the fitting criteria, which is followed by mutation and crossover to obtain the best solution according to the fitting criteria.

A random population of data is initially created by a random chromosome generator, which is called chromosome population (i.e., the solution of space in this case). The population has the size of 64 chromosomes by 3 genes and each gene constitutes of 2 bits. An error fitness criteria is then applied on the population and on the input data measurements. In order to calculate the error, calculate the fused measurement. The inputs F1, F2 and F3 are given in a range of values shown in Fig. 1. Fig. 2 plots the output fused measurement value, which is identical to the given range of inputs as shown in Fig. 1. The following two steps resemble fitness criteria. Fused measurement is calculated using the equation:
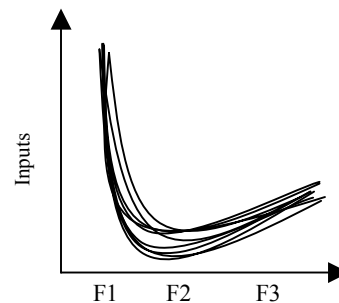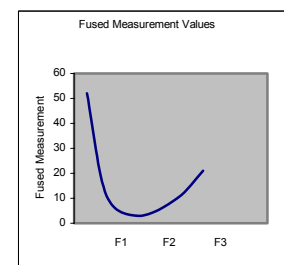

Fig. 1. Input Measurements Range

Fig. 2. Fused Measurement Values

$$YF = \frac{Y_1 \, x \, W_1 + Y_2 \, x \, W_2 + Y_3 \, x \, W_3}{W_1 + W_2 + W_3} \qquad (1)$$

The fused measurement calculated, will then be used to find the error of the input value with respect to this calculated value. Error is calculated using the equation:

$$Error = |F_1\text{-}YF| + |F_2\text{-}YF| + |F_3\text{-}YF| \qquad (2)$$

The calculated error signal is then used to sort the initially created random matrix in ascending order.

The sorted matrix is now available for crossover and mutation. The top most chromosome is called as the elite chromosome. Two parents are selected sequentially and they are crossovered and mutated to get a new child. The GA approach used here is represented in Fig. 3.
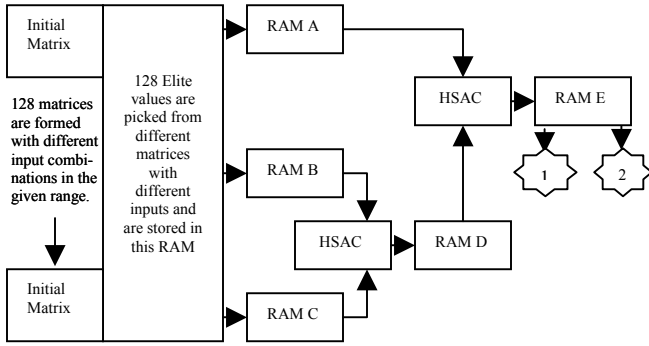
1→ Fused Measurements; 2→ Optimum Chromosome.
Fig. 3. Block Diagram Representation of GA Approach.

The crossover and mutation scheme employed here is *'Half Sibling and A Clone'*. In HSAC two parents and a random chromosome of same dimensions are considered. One of the parent and random chromosome are considered as parents and the other parent acts as a decider. If the parent decider bit is 1 it will take the new child bit as that of parent else the random chromosome bit. This is done for all the bits of the chromosome. Block diagram representing HSAC is represented in Fig 4.
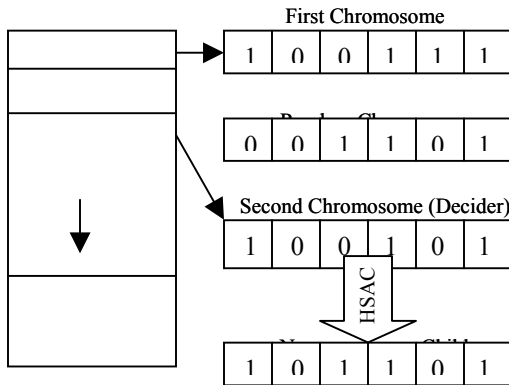

Fig. 4. Block Diagram Representation of HSAC.

This is done for all the chromosomes in the matrix. The matrix is now replaced by children except the elite chromosome. The algorithm goes into a number of iterations to get a single converging value and is the elite chromosome.

The above procedure is repeated with 128 different input values and thereby getting 128 elite chromosomes. Error is calculated for the above 128 chromosomes and the new matrix is arranged in ascending order. The matrix is divided into three parts A, B & C each of 42 chromosomes ignoring the last two chromosomes. HSAC is implemented for matrix B & C. Here each chromsome of B is crossovered and mutated with C. When the first chromosome of B is crossovered and mutated with all the chromosomes of C error is calculated for the new child's using the input measurements of B. The new population is sorted in ascending order based on error. This is done for all the chromosomes of B. Elite values of all those matrices form a new matrix called D.

Now HSAC is implemented for A and D. Here each chromosome of A is crossovered and mutated with D. Error is calculated for the new child's using the input measurements of A. The new population is sorted in ascending order based on error. Elite value of all those matrices form a final matrix called E.

3. Symbolic Formulation

The equations of the approach are represented in Bäck's Notation [7].

$EA = (I, \Phi, \Omega, s, \iota, \mu, \lambda)$ is called the Dynamic State Genetic Algorithm.

(1) $I = $ Space of Individuals drawn from arbitrary sets (vector or matrix), $A_X \, x \, A_S$.

(2) $\forall \, \vec{a} \in I : \Phi(\vec{a}) = \delta(f(\vec{a}_k), \Theta_\delta)$, Where $\delta : I\Re \times \Theta_\delta \to I\Re^+$

denotes a scaling function as in

$\delta(f((\vec{a}_k), \{ c_0, P(t) \}) = c_0 \bullet f(\vec{a}_k) - \min\{ f(\vec{a}_k) \, | \, \vec{a}_k \in P(t) \}, ,$ where P(t)

denotes the current population and $c_0 \in IR - \{0\}$.

(3) The crossover operator $r_{\{1,1\}}$ denotes a crossover rate of 1 and is a one-point crossover.

$\overset{\to l}{s} = (s_1, ..., s_{\chi-1}, s_\chi, v_{\chi+1}, ..., v_1)$

$\overset{\to l}{v} = (v_1, ..., v_{\chi-1}, v\chi, s_{\chi+1}, ..., s_1)$

(4) The selection operator is represented by:
$\Psi(P) = s(r_{\{1,1\}}(P))$

(5) S: $I^\mu x I^\mu \to I^\mu$, the proportional selection operator, samples individuals according to the probability density function given by:

$p_s(\vec{a}_k''(t)) = \dfrac{\Phi(\vec{a}_k''(t))}{\displaystyle\sum_{j=1}^{\mu} \Phi(\vec{a}_j''(t))}$

(6) The termination criterion ι is given by:

$\iota(P1(t)) = \begin{cases} true, \text{ if } t > t_{max} \\ false, \text{ otherwise.} \end{cases}$

(7) $|\lambda| = |\mu|$.

The following table explains different steps of symbolic formulation:

Step	Description
Step 1:	Initializing the matrix.
Step 2:	Validates the fitness function with linear dynamic scaling.
Step 3:	Crossover is performed for chromosomes S and V.
Step 4:	Describes the complete process of

	Population into a subsequent one by applying Genetic operators and selection.
Step 5:	Represents the rate of fitness function to that of mean fitness function.
Step 6:	Represents the termination criteria.
Step 7:	Verifies whether the population size is same or different after application of genetic operators.

TABLE 1: Symbolic Formulation Description

Algorithm Representation:

$t := 0$

$initialize\ P(0) = \left\{\vec{a}_1, \ldots, \vec{a}_\mu\right\} \in I^\mu$

$where\ I = \{0,1\}$

$evaluate\ \left\{P(0) = \left\{\Phi(\vec{a}_1(0)), \ldots, \Phi(\vec{a}_\mu(0))\right\}\right\}$

$where\ \Phi(\vec{a}_k(0)) = \delta(M_0(\vec{a}_k(0)), Pl(0))$

While termination criterion $\neq$ true loop

$for (t = 1:10)$

$\vec{s}^I = (s_1, \ldots, s_{\chi-1}, s_\chi, v_{\chi+1}, \ldots y_1)$

$\vec{v}^I = (v_1, \ldots, y_{\chi-1}, v\chi, s_{\chi+1}, \ldots s_1)$

$evaluate\ P'(t) := \left\{\vec{a}_1''(t), \ldots, \vec{a}_\mu''(t)\right\}$

$\left\{\Phi(\vec{a}_1''(t)), \ldots, \Phi(\vec{a}_\mu''(t))\right\} where$

$\Phi(\vec{a}_k''(t)) = \delta(f(\vec{a}_k''(t)), P(t-\omega));$

$select\ P(t+1) := s(P'(t))$

$where\ p\ s(\vec{a}_k''(t)) = \Phi(\vec{a}_k''(t)) / \sum_{j=1}^\mu \Phi(\vec{a}j''(t));$

$t = t+1;$

od

The terms in the symbolic formulation are described in the following table:

| Step	Symbols and Description
Step 1:	I → Space of individuals; A_X x A_S→ Arbitrary sets.
Step 2:	Φ → Fitness function; $\vec{a}$ → Element of population;
Step 3:	S & V→ Chromosomes.
Step 4:	Ψ→ Generation transition function; s→ Selection operator; p→ Population; $r_{\{1,1\}}$→ Crossover (percentage crossover, crossover rate)
Step 5:	$(I^\mu x I^\mu \to I^\mu)$→ New population
Step 6:	ι→ Termination; t→ Iteration number.
Step 7:	λ→ Number of children; μ→ Number of Parents.

TABLE 2: Symbolic Description.

3. Simulation Results & Comparison

The results of this approach are compared with the results of the approach from GAlib (A library of GA approaches from MIT). The algorithm was implemented with one of the approaches from the GAlib. A table of comparison of the both the approaches are shown in table 3. Fig. 5 plots the average error of the present approach and Fig. 6 plots the average error of the GAlib approach. From the plots it is clear that the error is min and steady in HSAC. The error in GAlib approach is not steady.
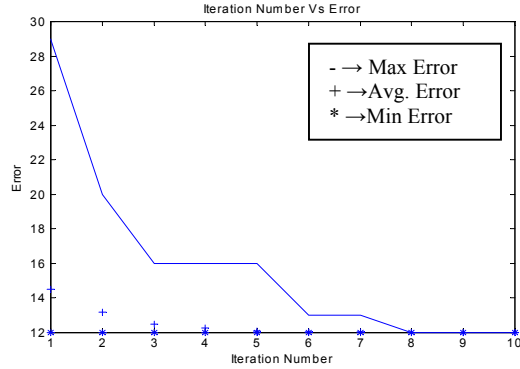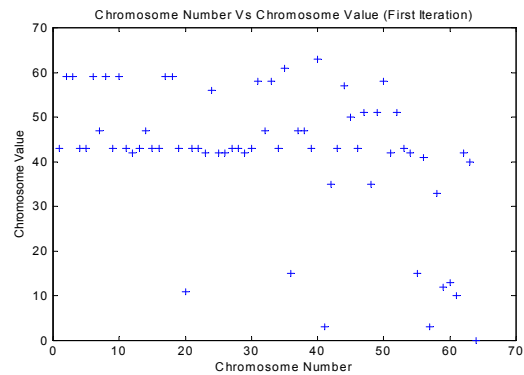

Fig. 5. Avg. Error Vs Iteration Number (HSAC).
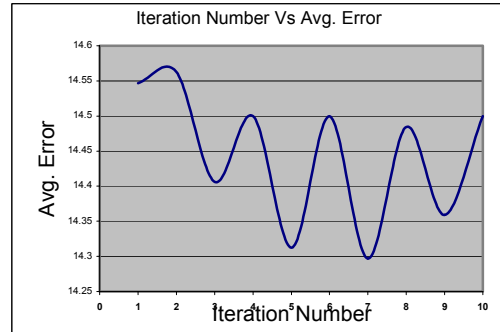
Fig. 6. Avg. Error Vs Iteration Number (GAlib).

Fig. 7 represents the 64 chromosome values after first iteration with HSAC approach. Fig. 8 shows the chromosome values after first iteration with GAlib approach. From the figures it is shown that in both the cases the values are random.

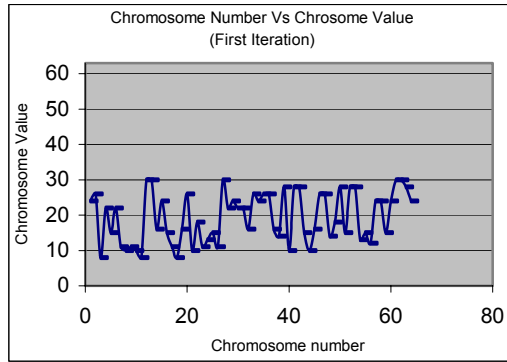**Fig. 7.** First Iteration Chromosome Values (HSAC).

Fig. 8. First Iteration Chromosome Values (GAlib).

Fig. 9 and Fig. 10 represent the chromosome values of all 10 iterations with HSAC and GAlib approaches respectively. From the figures it is shown that the chromosome values in HSAC approach converge to one single value whereas in GAlib approach the value does not converge and continues its randomness.
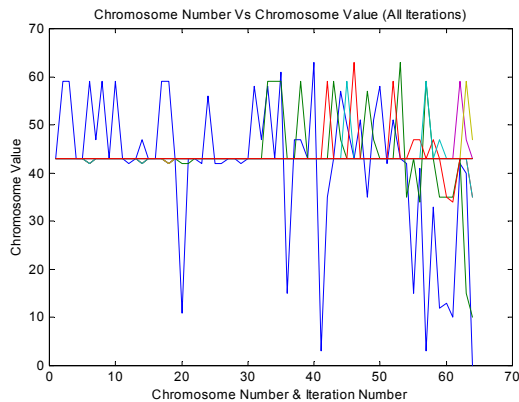

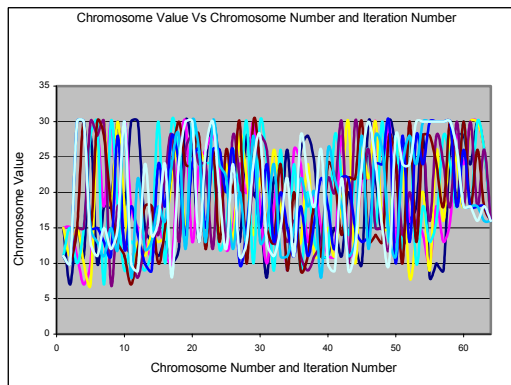**Fig. 9.** Chromosome Values For All Iterations (HSAC).

Fig. 10. Chromosome Values For All Iterations (GAlib).

	HSAC APPROACH	GALIB APPROACH
1.	The Error rapidly decreases and remains constant from one point.	The Error is not decreasing and is not constant.

2.	The Avg. Error is minimum.	The Avg. Error is more than in HSAC.
3.	By HSAC we can find the best inputs from the given range.	In this approach we should select the inputs.
4.	In HSAC a converging chromosome value is obtained within ten iterations.	In GAlib approach even after ten iterations the chromosome value is not converged.

Table 3: Comparison of Approaches.

5.   Conclusions

The GA using HSAC approach proved to be successful in both exploration of new chromosomes and exploitation of the available chromosomes. The system proved that it is not restricted to local minima. The fused measurement plot is identical to the given input range. The gain of the present GA approach is minimum error, which is not predefined rather calculated. The GA approach using HSAC aided in getting minimum error. The presented GA system provides sensor fused measurement and representative chromosomes to intelligent Bio-inspired signal processing.

REFERENCES

[1] Mohammad S. Sharawi, Hoda S. Abdel-Aty-Zohdy, and Robert L. Ewing, "*Optimal- Weights Sensors-Measurement Fusion Using Genetic Algorithms,*" in Proc. of IEEE Midwest symposium, Aug. 2001, pp. 537-542.

[2] Hoda S. Abdel-Aty-Zohdy, and Mahmoud Al-Nsour, *Reinforcement Learning Neural Network Circuits For Electronic Nose,*" IEEE Int. symposium on Circuits and Systems (ISCAS), May 1999, pp. 379-382.

[3] James W. Hauser, and Carla N. Purdy, "*Sensor Data Processing Genetic Algorithms.*" InProc. Of IEEE Midwest symposium, Aug. 2001, pp. 1112-1115.

[4] Hoda S. Abdel-Aty-Zohdy, "*Artificial Neural Network Electronic Nose For Volatile Organic Compounds.*" IEEE Computer Society press for the eighth great lakes Symposium on VLSI, pp. 122-125, Feb 1998.

[5] Hoda S. Abdel-Aty-Zohdy, Robert L. Ewing, Misson Mah, and Lihyeh Liou "*Integrated circuits and systems for solving bio-chemical sensor development detection problems,*" fifth joint conf. on standoff detection for chemical and biological defense, 2001.

[6] Hoda S. Abdel-Aty-Zohdy, Robert L. Ewing, Misson Mah, Lihyeh Liou "*Integrated circuits and systems for solving bio- chemical sensor development and detection problems.*" ASEE North central section, April 2002.

[7] Bäck Dr. Thomas. Evolutionary Algorithms in Theory and Practice. Oxford University Press. New York: 1996.

[8] Matthew B. Wall. "*A Genetic Algorithm for Resource constrained scheduling*", Ph.D Thesis, Massachusetts Institute of Technology, 1996.

[9] H.S.Abdel-Aty-Zohdy, R.L Ewing and G.B. Lamount, "*Multidisciplinary Biotechnology Domains for Physical Sensors and detectors*", in Proc. of IEEE 4th European workshop on Microelectronics Education, pp. 137-140, May 2002.