Universe Scientific Publishing

ORIGINAL ARTICLE

A new approach to an adaptive filter high performance based on abstract data types for processing ECG signals

Maciel A.[1*], Vieira R. V.[2]

[1] Federal Rural University of Pernambuco - IKAS / CODAI, São Lourenço da Mata, Pernambuco, Brazil; Email: andrilenef@yahoo.com.br

[2] Computer Institute, Federal University of Alagoas -UFAL, Maceió, Alagoas – Brazil; Email:rvvl@ic.ufal.br

Abstract: The implementation of genetic algorithms (GAs) inspired by the Holland model in hardware to filter signals aims to speed up the convergence time of these algorithms by implementing the modules considered a bottleneck for a software implementation. However, these modules have the same problems with the representation of the chromosome, dependence on genetic operators, representation adopted for the chromosome and population, and the loss of chromosomes with relevant features for the solution of the problem to which the GA has been applied. In this paper, we present a new approach to an adaptive filter high performance based on abstract data types for processing ECG signals, tested in conventional central processing unit (CPU) and Graphics Processing Unit (GPU), evaluated for its efficiency from the samples obtained from the MIT-BIH, arrhythmia database. The algorithm used is based on a compact genetic algorithm that was based on abstract data types, implemented in MATLAB in architecture CUDA. The results have shown that the compact genetic algorithm can be implemented in high-performance systems, aiming to improve the health care systems of treatment is provided to patients with cardiovascular problems.

Keywords: Electrocardiogram; Genetic Algorithm; Adaptive Filter; Graphics Processing Unit (GPU); Compute Unified Device Architecture (CUDA).

1. Introduction

Estimates from the World Health Organization (WHO) indicate that diseases chronic non-communicable (NCDs) are responsible for 38 million deaths worldwide annually. Government measures are needed to achieve the global targets aimed at reducing the burden of chronic diseases and prevent 16 million premature deaths (before age 70), caused by cardiovascular disease, cancer and diabetes[1,2]. According to the new WHO report[2], most early deaths caused by chronic noncommunicable diseases are preventable. Of the 38 million lives lost in 2012 by NCDs, 16 million, ie 42% were premature and preventable (an increase of 14.6 million deaths concerning 2000)[2].

Cardiovascular diseases (CVDs), or diseases of the circulatory system they make up the bulk of all endemics[1] in developed countries and the same behavior has occurred in recent decades in emerging countries, for which health statistics show that cardiovascular diseases occupy the first or second as a cause of death[3].

Heart disease known as heart disease or disease cardiovascular can be described as heart failure, arrhythmias, hypertension, coronary artery disease, endocardial diseases, myocardial and pericardium, aortic diseases, among others[1,3].

The diseases associated with patients with heart disease may worsen their cardiovascular conditions, as well as interfere with the diagnosis and therapy of these diseases[4].

[1] This differs from the epidemic because it is of a more continuous and restricted character[5].

doi: 10.18686/csit.v1i1.1254

The identification of such disease is carried out in general by tests electrocardiogram, which record the heart's function over time[5].

The wave-detection process of an electrocardiogram presents difficulties related to its analysis due to oscillations of the signal, lack of uniformity in the morphology of the waves that compose it and the emergence of noise during its extraction[5].

These noises can, in turn, be displayed for collecting these signals and hide important features. Typical cardiovascular diseases described in the morphology of the cardiac signal, among which there are arrhythmias[1]. Cardiac arrhythmias represent any changes in the regularity, frequency, place of impulse origin, or abnormality in conducting this pulse, to modify the normal sequence of depolarization of the atria and ventricle[3].

Often the diagnosis of cardiac arrhythmias could be difficult to identify even with the aid of electrocardiogram and other methods of clinical research[1]. The main points to the problem of irregular heartbeat is related to the identification of the waves that form the electrocardiogram signal (ECG): P wave, tracking PR, QRS complex, ST segment and U wave[1]. The wave-detection process of an electrocardiogram has some difficulties, because of the oscillations in the signal, lack of uniformity in wave morphology and the appearance of noises during the extraction of the ECG signal[5].

The effectiveness of the detection system, extraction and viewing the morphological characteristics of the automatic ECG signals can be considered a very important process to save lives, and directly influences the early diagnosis of cardiac diseases, account for the majority of existing sudden deaths worldwide [1,6]. If the performance of the arrhythmia detection system is considered inefficient, that may render the arrhythmias system impracticable due to the processing time for ECG feature extraction.

The analysis of an electrocardiogram is directly related to the ECG signal quality of the collection, able to extract the relevant characteristics of the waves that composes the signal (amplitude, duration, angles, and form of waves)[1]. With the data obtained from the signal is performed a study which aims to find arrhythmias, then allow a related diagnosis on structural and functional changes of the heart.

Treatment of dimensionality of the data processing of the ECG signal, almost always requires the combination of several algorithms for processing, are: Cluster analysis[7], Adaptive Lifting Scheme (ALS)[8], Discrete Fourier Transform (DFT)[9], Decimation In Time (DIT)[9], Decimation in Frequency (DIF)[9], Discrete Wavelet Transform (DWT)[9], Fast Fourier Transform (FFT),[9] Finite Impulse Response (FIR)[10-12], Genetic Algorithm (GA)[13], K-Nearest Neighbors (KNN)[14], Independent Component Analysis[15] Wavelet Transform (WT)[16], Parallel Genetic Algorithm (PGA)[17,18], have computational approaches able to perform the filtering task identification and diagnosis without much precision, thus bringing about loss of information and compromising the diagnosis of cardiac arrhythmias. In general, we use more than one algorithm for representing and processing these tasks in a sequential isolation, making the process slow and not feasible for rapid diagnosis of abnormalities.

Most of the algorithms dealing with the functions of filtering, identification, and diagnosis for processing ECG signals, each of these features separately, because there is no one in the literature algorithm that performs all the functions at once.

In these aspects, it has been even to deal with the proper conduct of the cleaning signal that has a high dimensionality without loss of information with the processing of point data fixed or floating with modeling algorithm so that it can be applied to any similar problem processing ECG signals.

The implementation of genetic algorithms (GAs) inspired by the Holland model[19] in hardware to filter signals aims to speed up the convergence time of these algorithms by implementing the modules considered a bottleneck for a software implementation.

However, these modules have the same problems with the representation of the chromosome, dependence on genetic operators, representation adopted for the chromosome and population, and the loss of chromosomes with relevant features for the solution of the problem to which the AG has been applied.

Genetic algorithms (GA) inspired by the traditional Holland model[19], are only used for optimization

functions and identification of the filter coefficients, rarely used for signal cleaning, due to the computational effort during processing time, especially when the chromosome and the population of the GA represents high dimensionality.

The genetic algorithm based on abstract data types (GAADT) developed by Vieira[20], differs from traditional genetic algorithms, for obtaining a stratified view of the chromosome and the evolution revolves around the environment. In traditional algorithms, the evolutionary process takes place from the population, which is typically binary and does not apply to most of the problems. Although all implementations the algorithm GAADT [22,23,24], aiming seek the best results, in the course of its processing, the main disadvantage is the exponential explosion of population to find the most appropriate result for the convergence process; this may take, at least, one (1) week and, at most months for execution.

Despite GAADT have already been implemented in software in the works[22-25], it is considered a convergent algorithm and the stratification of chromosome can be applied in most of the problems addressed by genetic algorithms. Its main disadvantage is related to the computational effort brought in processing algorithm in conventional CPUs and treat people with high dimensional data.

In this paper, a specification of an adaptive filter is discussed, inspired by the genetic algorithm based on abstract data types, able to perform all genetic operations performed by abstract GAs and solve the high dimensionality population problem, particularly about cleaning ECG signal. In the implementation of the adaptive filter, a compact version of the genetic algorithm based on abstract data types will be displayed (CGAADT), to improve the performance of the algorithm developed in software. Also presenting the results obtained using this system with the sampled ECG signals from the base MIT-BIH arrhythmia data[21].

2. The Monitoring System of Electrocardiogram Signals

The medical monitoring system of electrocardiogram signals correspond to systems used for continuous monitoring of information obtained from patients. The main records of the normal electrocardiogram can be seen in **Figure 1.**

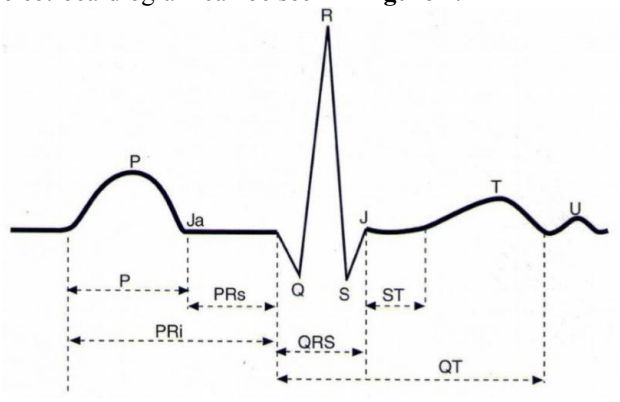

Figure 1. Composition of the waves of ECG signals[5].

The extraction of ECG is a noninvasive, low cost, quickly and efficiently method to provide important features for analysis and diagnosis of heart diseases. They present a succession of waves that correspond to those cardiac cycles. The first electrical activity of each cardiac cycle recorded by ECG is the P-wave (**Figure 1**), which represents the activation of atria. Then it is recorded an isoelectric line called the PR segment, that coupled with the P wave sets the interval PR. Right after, subscribes to the QRS complex can consist of one phase (single-phase), two phases (biphasic) or three phases (three-phase) wave, which is the ventricular activation. Again, if registers called isoelectric line segment ST, to constitute the interval between the ventricular activation and the start of ventricular repolarization. Finally, it subscribes to the T wave, corresponding to the electrocardiographic representation of ventricular depolarization, followed by U wave. Remains, in the end, set the QT interval, comprising the time between the onset of ventricular depolarization and repolarization[5]. The systems of monitoring ECG signals are responsible for the identification of the waves, which are analyzed by a doctor for a possible arrhythmia diagnosis, suggested treatments and medications according the clinical condition of the patient. For proper operation of the system, it must be able to demonstrate quality ECG wave morphologies, from the processing and signal cleaning.

This work approach features a high-performance application for genetic algorithm acceleration based on abstract data types in GPU / CUDA platform aims to present a specification of a compact version of a genetic algorithm based on abstract data types (CGAADT), used in processing and cleaning of cardiac signals. Also, the

results obtained will be presented using this system with samples of ECG signals from the base MIT-BIH Arrhythmia data[21].

3. Specification of ECG Waves on Genetic Algorithm in Compact High-Performance Architecture

The modeling of the GAADT in Vieira [20], presents the basic definitions of genetic algorithm based on abstract data types, with the representation of the chromosome types, gene, and base. The GAADT architecture, the process of formation of alphabets (gene, base, and chromosome), representing 70% of the algorithm processing time when run on conventional CPUs. In the new compact version of CGAADT the chromosome representation is divided into two levels:

• The abstract type basis (B) is represented by B_{Host}, run directly in the central processing unit (CPU);

• The chromosome types and gene run in Compute Unified Device Architecture (CUDA).

The CGAADT algorithm starts the process of forming the elementary units to generate the alphabets. This process allows the algorithm to perform the genetic material formation in different platforms, from the creation of the bases in the CPU and the genes and chromosomes in the GPU, so that the genetic material is fragmented and processed in parallel. In **Figure 2**, it presents the block diagram referring to the data flow of the CGAADT algorithm. The input data are represented by ECG signals containing the 12 (twelve) leads.

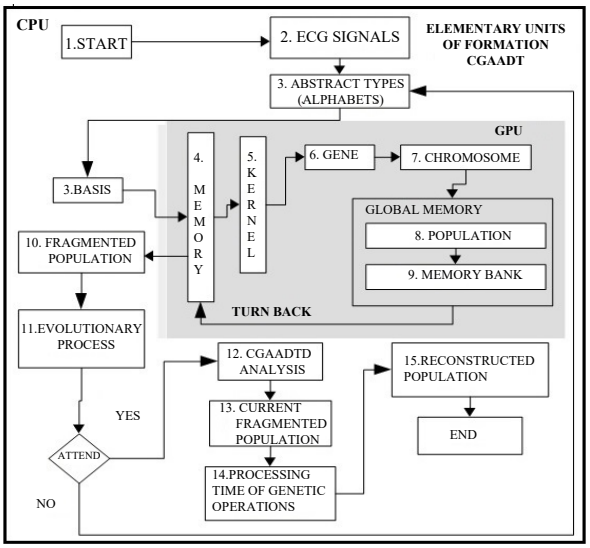

Figure 2. CGAADT Data Flow[26].

The CGAADT algorithm initializes the process according to its steps:

1. Input data is represented by the ECG signal containing the 12 (twelve) leads.
2. These signals are read.
3. Begin the process of forming elementary units to generate the alphabet. This process allows the algorithm to perform genetic material formation in different architectures. It starts the process of creating the base abstract term, which runs on the CPU.
4. The abstract base term is routed to GPU memory.
5. Subsequently being automatically forwarded to the GPU kernel.
6. After the formation of the abstract base term, the genetic material is fragmented in the GPU so that it can form the complementary genetic material (gene and chromosome), run in parallel.
7. Fragments of the abstract gene and chromosome terms are stored in GPU global memory.
8. The process of creating a new population begins.
9. After the creation of this new fragmented population, it returns the memory of GPU (4), to be further fragmented to the CPU.
10. With this new fragmented population, the evolutionary process begins.
11. The evolutionary process performs the functions of genetic operators (selection, crossing, reproduction, mutation, and insertion of descendants in the population).
12. If the evolutionary process is satisfied, this population is directed to the new module called CGAADT ANALYSIS.
13. In this module, the analysis of population quality at the noise level is verified.
14. The processing time of genetic operations is evaluated.
15. The new population is rebuilt without loss of information.

The application of CGAADT requires a definition of specific elements in an environment featuring the problem in focus.

3.1 Basic types

3.1.1. Definition (Basis)

The basis type for the construction of the adaptive filter instantiated by CGAADT for processing cardiac signals is formed by $B_{PointsWavesHost}$ the set times of the ECG for each lead executed on the CPU. The ensemble $B_{NamesWavesHost}$ e the set $B_{Host\lambda}$ containing innocuous leads.

$$B_{Host} = B_{PointsWavesHost} \cup B_{NamesWavesHost} \cup B_{host\lambda} \quad (1)$$

The term $B_{host\lambda}$ is formed by the electrical phenomenon recorded by the electrocardiogram (ECG), the deflections that form a particular junction and the periods representing the end of ventricular depolarization and the start of repolarization.

The term $B_{NamesWavesHost}$ is the set $\{wave_P, wave_Q, wave_R, wave_S, wave_T, wave_U\}$ which contains all the waves segments and complexes, which may be detected in the ECG examination, contained in the database MIT-Databases[21]. The identification and classification of waves that make up the ECG signals, obey the standard model of Willen Einthoven[5].

The elements are the set $B_{PointsWavesHost}$ are ordered pairs X = (x, y), where $X \in \mathbb{N} \times \Re$ that contains values that we can extract the morphological properties of waves in the ECG host (amplitude and time intervals).

The ensemble $B_{Host\lambda}$ formed by the λ element, which represents the wave whose morphology is within the normal pattern.

The characteristics (genes) in the relevant GPU to the problem treated in this paper are $g_D = \{wave_P, wave_Q, wave_R, wave_S, wave_T, wave_U\}$, where D is the device, that are part of the same derivation recorded by the ECG. The set representing these waves is $G_{DElements}$, which is formed by the junction of the basis of sets B_{Host}. The elements $G_{DElements}$ contain values from which we can extract the morphological properties of the ECG elements (amplitude, duration, and interval) used in the detection process. The structure adopted for the whole $G_{DElements}$ elements is $element_i = (name,(x-,y-),(x_p,y_p),(x+,y+)$ where i $\in$ # (don't care symbol can be replaced by an alphabet symbol adopted for chromosome), where:

$$name \in B_{NameswavesHost} \quad (2)$$

$$(x-,y),(x_p,y_p),(x+,y+) \in B_{PointsWavesHost} \quad (3)$$

For example, if a gene GPU represents the QRS complex shown in **Figure 3** (where point S is the starting point of the wave, M is the maximum and the endpoint F). The resulting gene would be:

$$g_D = (Complex_{QRS},(x_s,y_s),(x_m,y_m),(x_f,y_f)) \quad (4)$$

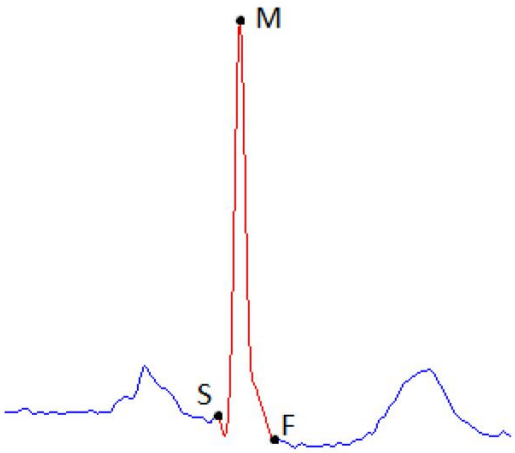

Figure 3. Location in a wave sample from a QRS complex with points S, M and F, which are responsible for determining the information wave[24].

The interpretation adopted for the elements $element_i$, where i the element name; x^- is the lower value of x coordinate of the element to the wave; x_p is the value of x coordinate to the peak of the element; x^+ is the highest value of the x coordinate for the element; It is the y^- coordinate value y for the element during x^- ; y_p is the value of y^- coordinate to the peak of the element; and y^+ is the y coordinate value for the *"Element"* element during x^+. When in an ECG is not entered a given wave during a period that the parameter name wave receiving λ value.

Definition 3.1.2. (Gene) - The abstract type gene G_D is a set of all elements $G_D = <b_{h1},b_{h2},b_{h3},b_{h4}> \in G_D$ in GPU, represented by h (host), formed by the elements of abstract base type B_{Host}, where D is the device (GPU).

The characteristics law be represented by the set of Axioms of Genes Formation in Device (AGFD), which shall be defined in each case, according to the semantics attributed to the gene in the CUDA device, as described in item 3.1.1.

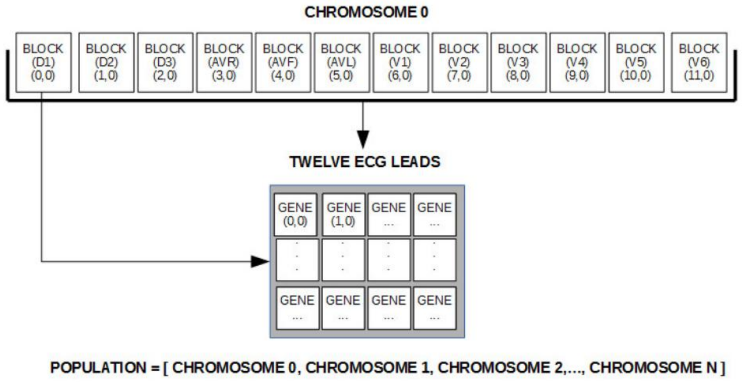

Figure 4. The formation of genes g_D on the block grid on the GPU [26].

The axioms of set AGFD establish that:

- The basis $b_{h1} \in B_{\text{NamesWavesHost}}$;
- Basis b_{h2},b_{h3},$b_{h4} \in B_{\text{PointsWavesHost}}$;
- For every gene $g_D = <b_{h1},b_{h2},b_{h3},b_{h4}> \in G_d = [\text{bk}_{ij}]_{mxn}$;
- For every gene $g_D = <b_{h1},b_{h2},b_{h3},b_{h4}> \in G_d = [\text{bk}_{ij}]_{mxn}$, of size (16x12), which is the best formation of genes and chromosomes in the grid, where i and j represent respectively the row and column that the element occupies in $B_{\text{NamesWavesHost}}$;
- The ordered pair b_{h2}, should be a point whose occurrence is a period less than or equal to the ordered pair b_{h3} on the ECG, at where $\text{agfd1} = \forall g_D = <b_{h1},b_{h2},b_{h3},b_{h4}> \in G_{D,x-} \leq x_p$;
- For every gene $g_D = <b_{h1},b_{h2},b_{h3},b_{h4}>$, the ordered $h_3$pair must be a point whose occurrence is a period less than or equal to the ordered pair b_{h4} of the ECG, at where $\text{agfd2} = \forall g_D = <b_{h1},b_{h2},b_{h3},b_{h4}> \in G_D$, $x_p \leq x^+$;
- The $G_{D\lambda}$ is set innocuous gene formed by the basis $g_{D\lambda} = <b_{h1},b_{h2},b_{h3},b_{h4}>$ such that: The b_{h1}=λ. The elements of this set are represented by $g_{D\lambda} = [\text{gene}_{D0},...,\text{gene}_{Dn}]$ where $n \geq 0 \in \{0, ..., 191\}$ number of threads per block;

Definition 3.1.3. (Chromosome) - Abstract chromosome type C_D is the set of all genes constructed by the definitions established by the Axiom of Chromosome Formation in Device (ACFD);

Thus, the ACFD set is specified as:

- The elements must occur in sequences of waves *P, QRS, T*, and *U*, represented by $C_{D_{Period}}$;

- The non-occurrence of an element in the block is characterized by the replacement of the gene for the missing element in the block, by innocuous gene $g_{D\lambda}$;

- Occurrence ranges of the waves do not intercept, that is, x^+ element $p \leq x^-$ $element^{QRS}$ and x^+ $element^{QRS} \leq x^-$ $element^T \leq x^+$ $element^U$;

• For every gene $g_D = <b_{h1},b_{h2},b_{h3},b_{h4}> \in G_D = [bk_{ij}]_{mxn}$, of size (16x12), which is the best formation of genes and chromosomes in the grid, where i and j represent respectively the row and column that the element occupies in $B_{NamesWavesHost}$;

- For a given period may not be elements in the block (bk) of the same type, where:

$$\text{acfd}_1 = \forall C_{D1} \in C_D(\forall(g_{D1i},g_{D2i},\text{name}^{gD1i} \neq \text{name}^{gD2i}))) \quad (5)$$

Where the name is a function which returns the gene base which stores the value of the element name b_{h1};

- Each bk is composed of a lead ECG, represented by D1, D2, D3, aVR, aVL, aVF, V1, V2, v3, v4, v5, v6 (twelve lead), the grid of blocks in GPU;

- The set C_D forms a one-dimensional plane (**Figure 4**) $\forall C_D \in G_d$;

- The set streaming multiprocessor (SM), represented by SM={sm_0,...,sm_n}where $n \geq 0 \in \{0,.., 3\}$, form C_D a size $2^{11} \times 16$ so that there is the exponential explosion of population in finding the most appropriate outcome during the process convergence.

- Each sm will have a maximum of 32 threads that form 6 wraps (W), representing scaling units needed to improve the processing of genetic operations. The calculation for the quantity wraps results from the following expression:

$$W = \frac{\text{Th}_{\text{Max}_{bk}}}{\text{Th}_{\text{Max}_{SM}}} \quad (6)$$

Where W represents the number of wraps $Th_{Max_{bk}}$(maximum number of threads in the block), and

$Th_{Max_{SM}}$(maximum number of threads per Sms).

- Chromosome set innocuous denoted by $C_{D\lambda}$ is formed by all sets of innocuous genes that satisfy the constraints established by Axiom Chromosome Formation in the Device (ACFD), according to equation 7.

$$C_D = \{g_{D1}, g_{D2}, \ldots, g_{Dn}\}, \text{ at where } n \geq 0 \in \mathbb{N} \quad (7)$$

The population is defined by the compact genetic based on abstract data types, the model in CUDA performs partitioning of the population in the stream processing (SM).

The system was developed representing the following configuration: an accelerator card from NVIDIA, GeForce GT 740M, Resolution 1366 x 768, 60 Hz, and a host machine, adopted an Intel (R) Core TM i7 4500 U CPU 1.80 GHz 8GB RAM, MATLAB R2014a[27-29]. The characterization of the problem to clean the ECG signal and the CGAADT algorithm acceleration in high-performance platform is described below.

Definition 3.1.4. (Population) - The abstract type P_{DFRAG} is the set of all the chromosomes built according to Definition 3.1.3, which is $P_{DFRAG} \geq 2^{15}$ the size of the most adapted population.

3.2 Genetic Operators

The specification of abstract data types: basis, gene, chromosome, and population, preserved the requirements contained in the definition of CGAADT the specification of the functions and relationships necessary to calculate the function CGAADT must me*et all* preconditions to its original setting. The following are presented the definitions of functions and relations whose specifications for problem is more concrete than the origin setting, getting subtended the roles and relationships that are not redefined in this section preserve their original definition by GAADT[20].

Given an element "element$_o$" of the patient's ECG, and limits standards of height and width for this element in the device, the function CompareElement returns true if the element is within the range for the height and width provided for atrial flutter arrhythmias, atrial fibrillation and other irregularities found in ECG (arrhythmias), otherwise it returns FALSE.

Definition 3.2.1. (Compare Element) - The function compares element is formally defined as:

$$\text{CompareElement}: B_{DElement}\ \Re x \Re \rightarrow B$$

$$CompareElement(Element_0, X, Y) =$$

$$\begin{cases} True\ if\ ((x_0^+ - x_0^-) \leq x) \wedge \\ (((y_0^+ \geq y_0^-) \wedge (y_0^p - y_0^-) \leq \\ (y_0^+ - y_0^-) \wedge (y_0^p - y_0^+) \leq y\ False \\ otherwise \end{cases} \quad (8)$$

Where $o \in \{P, QRS, T, U\}$, B is the set of Boolean values in the device.

The degree of adaptation of the chromosome for the detection of elements must consider the height and width of the waves (P, T, Q, R, S, U), are represented in the given gene. For each element of the chromosome that meet this verification must be added another to their degree of adaptation. So to calculate the degree of chromosome adaptation we must first define a function to return the wave patterns.

Given a chromosome $C_D = \{g_{D1}, g_{D2}, g_{Dn}\}$ and a standard element format, the default function returns the value 1 if one of the properties to g_D meet the metric properties registered for the wave, and zero otherwise.

Definition 3.2.2. (Standard) - The occurrence of an element in a given period of ECG is provided by standard$_D$ function of the following type:

$$standard_D(C_D, O_D) =$$

$$\begin{cases} 1\ if\ (P \in wave_P(O_D, II) \Rightarrow compareWave(wave_P, Y, X)) \wedge \\ (QRS \in wave_{QRS}(O_D, II) \Rightarrow comparewave(wave_{QRS}, Y, X)) \wedge \\ (T \in wave_T(O_D, II) \Rightarrow comparewave) wave R, Y, X)) \wedge (U \\ \in wave_T(O_D, II) \Rightarrow comparewave(wave_P, Y, X) \\ 0 \quad otherwise \end{cases} \quad (9)$$

Where: $C_D = \{g_{D1}, g_{D2}, g_{Dn}\}$ and II contains the names of all the waves.

Definition 3.2.2. (degree) The degree of adaptation of a gene is a function degree of the following type:

G → K such that for each gene g_D, $g_D \in G_D$ is associated with a unique number k, $k \in K$ (K is an ordered body[2]), called degree$_D$ (g_D) and reflects,

[2] It is an algebraic structure, with two operations, without its own zero dividers and provided with an order[20].

according to the interpretation adopted for the problem, a comparative stratification between gene adaptation[20].

The dominant gene is identified by the *domi* function that receives a pair of genes, one from each of the parent chromosomes, and returns the most adaptive gene if the given genes express the same trait. If the given genes do not express the same trait, then the *domi* function will return g_λ.

$$\text{degree}_D{:}G_D \rightarrow \Re\ \text{adaptation}_D(g_D) = \sum_{a\in I} \text{standard}_D\ (g_D{,}\text{o}_D) \quad (10)$$

I contain the names of all the waves registered in the system.

The weight given to a gene g_{Di} of a chromosome is equal to j + 1, where j is the number of waves whose characteristics meet the standards. For example, if a chromosome has been formed by genes that meet the specifications of the waves *P*, *T*, *QRS* complex and *U*, then the function degree_D will return the value 1 for each gene of this chromosome, and the total adaptation of chromosome is equal to 10.

The specification for the detection of CGAADT waves working with genetic crossover and mutation operators between adjacent chromosomes, or chromosomes that are neighbors in the temporal space. The crossover occurs when a chromosome is found that has at least one harmless gene. In this case, it generates a new chromosome with a harmless gene being replaced by a gene on chromosome neighbor.

This operation allows errors in the detection of waves and/or grouping of genes for the formation of chromosomes is corrected, as shown in **Figure 5.**

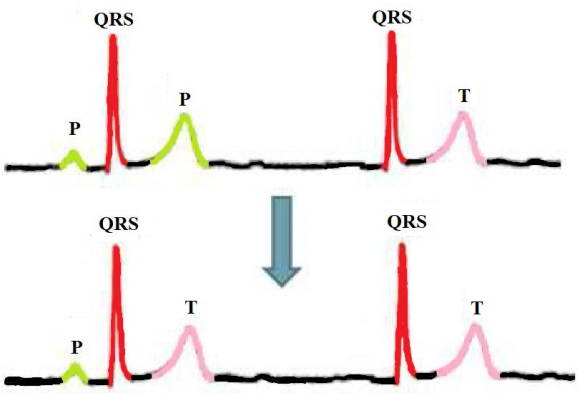

Figure 5. Crossover operation carried out in two chromosomes[24].

During the reproduction process shown in **Figure 5**, the algorithm identifies the pattern of *P* wave in the normal ECG from $g_{D\lambda}$, the first identifiable record is the *P* wave, representing the depolarization of the atria. The duration of the normal *R* wave generally obtained in D2 the ranges from 0.08 to 0,11s in adults[5].

According to Sanches[5], the *P* wave is rounded and single-phase most of the leads. Occasionally, they have small indentations and, in such cases, the distance between a peak and the other should not exceed 0,03s. In normal individuals, the maximum amplitude of the *P* wave approaches the 0,30mV 0.25 (2.5 to 3mm), observed in $D2$. However, the amounts commonly found ranging from 0.5 to 2mm[5].

The *T* wave represents ventricular repolarization. The repolarization different portions of the right ventricle and the left ventricle occurs more heterogeneous so that the depolarization is recorded wider T-waves and smaller amplitude, i.e., usually below 6 mm[5].

He rounded morphology and asymmetric, the first being longer than the second portion. Typically, the deflection of the *T*-wave has the same direction as the QRS complex, ie multiple shunts the two phenomena positive deflections register[5].

The innocuous gene stores the patterns of waves, during the crossing process, the algorithm performs the comparison of the wave pattern established in performing the comparison and performs the crossover operation at that point in the ECG wave is the T wave according with the standards set by $g_{D\lambda}$.

Improvements can still be obtained by crossing operation through the sums of the adjustments to the chromosomes. However, when parent's chromosomes are less adapted to the children-chromosomes, they will be replaced by his descendants.

The mutation process will occasionally chromosomes having at least one gene of adaptation equal to "0". This operation will consist of an exchange of names of genes to try to increase the resulting adaptation of chromosomes, see **Figure 6.** This operation is also designed to try to correct possible errors in detecting the waves, following the 50% probability of chromosome size is due to the fact that the mutations occurred in a chromosome of a given species are too large, then this chromosome would be repelled by the chromosomes of its kind, not to be considered more an equal to these[20].

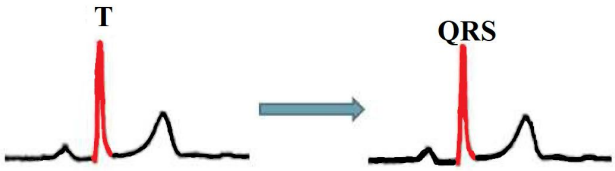

Figure 6. Example of name change held in mutation operation[24].

The stopping criterion adopted by CGAADT function is the maximum numbers of desired iterations and the value of the average adjustment of the current population defined by $P_{point\_off\_cut}$ (set of chromosomes below average) and are considered satisfactory for the result of the problem analysis.

These criteria are also part of the problem set requirements R_{qD} . To represent the most suitable chromosome, several experiments were carried out with values of 25 to 125 iterations, until the value of the fittest chromosome was not changed during the twenty iterations followed. Thus, we conclude that to reach 100 iterations, the population had more chromosome adapted to the problem.

3.2.3. Adaptation

The adaptation of a chromosome is an adaptation function of the following type:

$$adaptation: C_D \rightarrow K\ adaptation\ (c_D) \sum_{g \in c} \theta_{c,g}\ x\ degree(g_D) \quad (11)$$

Where $\theta_{c,g}$ is the weight with which the g gene contributes to the adaptation of chromosome c [20].

To perform the crossover operation, we will need to be aware of two other functions, which are selection and reproduction functions. The mating operation receives two parent chromosomes, capable of mating, and returns a population whose chromosomes are formed only by the dominant genes of the given chromosomes.

The selection function filters the chromosomes that can cross and thus undergo a reproduction process, which occurs in the reproduction function, where they will be crossed. The function will return the dominant set of genes for all existing parent chromosome characteristics. The selection function receives a population *P* and returns the P_{DFRAG} subpopulation formed by the chromosomes that satisfy the requirements of problem *r* (selection function), described by first-order logic formula, which indicates when a given chromosome is considered to cross, Further details on the selection, crossing and reproduction functions can be seen in Vieira[20] and Maciel[26].

3.3. Features GAADT / CGAADT

The algorithm consists of the abstract data type, has bases in their basic training units. According to Vieira[20], the only requirement for the abstract type base GAADT is that it has at least one base b_λ, called base innocuous to be used in the construction of the gene-chromosome innocuous c_λ and harmless. The constants of the GAADT base innocuous gene g_λ - chromosome harmless and innocuous-should have its value set at the time of the instantiation of GAADT to a given problem.

Figure 7 shows the block diagram of elementary units forming the GAADT, called gene, chromosome base and are directly executed in CPU.

The GAADT algorithm in the current modeling (**Figure 7**), made it impossible to improve the algorithm processing time, as can be seen in results.

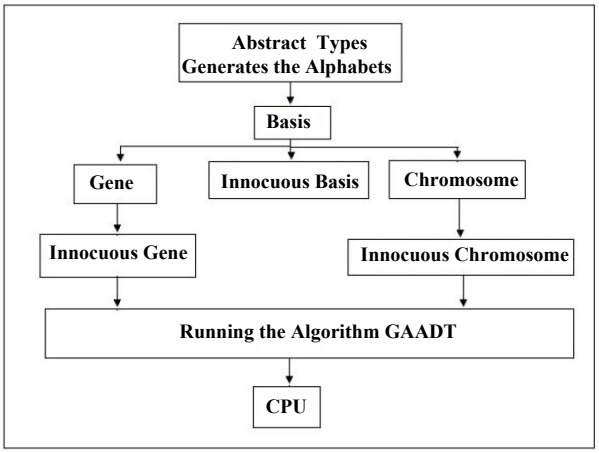

Figure 7. Block diagram of the elementary units forming the GAADT[26].

The new version of GAADT in CUDA, called CGAADT (**Figure 8**), only the base abstract term runs in the CPU, while the other abstract terms called gene and chromosome run on the GPU.

This processing enables at least ten (10) chromosomes are executed in parallel, in the case of fixed-size population according to size executed on the GPU grid of blocks, whereas the number of chromosomes is limited according to the size of the database performed in constant grid of blocks on the GPU.

Therefore, the larger the size of the database and the

grid of blocks, the greater the number of chromosomes in CGAADT run in parallel, which can be modified according to the adjustment algorithm architecture.

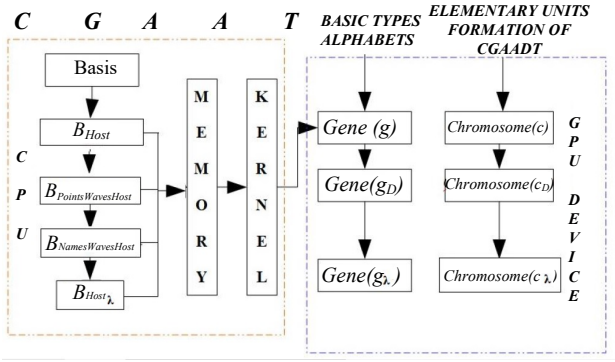

Figure 8. Block diagram of the elementary units forming the CGAADT[26].

3.4. CGAADT Architecture

The **Figure 9**, shows the architecture of CGAADT algorithm instantiated by GAADT, both algorithms work on an environment that can be modified according to the problem being addressed and how populations of chromosomes that will evolve.

In the GAADT, the environment A is made up of 8-tuple $< P, IP, Rq, AFG, AFC, Tx, \sum, P_0 >$, in this environment, there is no concern with memory and all environment runs in CPU.

In the CGAADT environment exposed in the architecture (**Figure 9**), a genetic algorithm operates on populations of chromosomes that occur in the grid of blocks GPU (device) according to the characteristics of the environment A. This environment is presented by 13-tuples, $[P_{DFRAG}, IP_{DFRAG}, Rq_D, M_m, G_D, AGFD, AGFD, Tx_D, SMs, \sum, P_{DFRAG0}, P_{DFRAGt}, P_{Rec}]$ where:

1. P_{DFRAG} is the population in GPU (device);
2. $IP(P_{DFRAG})$ is the set power;
3. Rq_D is the set of requirements (features expressed by formulas in a first-order language) the problem that influences the genealogy of the population P_{DFRAG};
4. M_m is the set of memories $\{M_k, G_M, M_{PDFRAG}\}$, where M_k is the data transfer memory to the kernel, GM is the global memory on the GPU (device) and M_{PDFRAG} is the population transfer memory $P_{DFRAG}$to the initial population P_{DFRAG0} in the host, for algorithm execution;
5. G_D is the thread block grid on the GPU (device);
6. AGFD is the axioms of set of formation of the population of genes in chromosomes P_{DFRAG} in the GPU (device);
7. ACFD is the axioms of set of formation of the population of chromosomes P_{DFRAG} in the GPU (device);
8. Tx_D is the set of pairs of chromosomes (x, y), which x is a chromosome C_D constructed from the chromosome y, by the action of crossing operation or mutation, thus registering the genealogy of chromosomes belonging to the populations generated by CGAADT during its execution;
9. SM is parallelization of chromosomes in streaming multiprocessor for scheduling the operations of the AG;

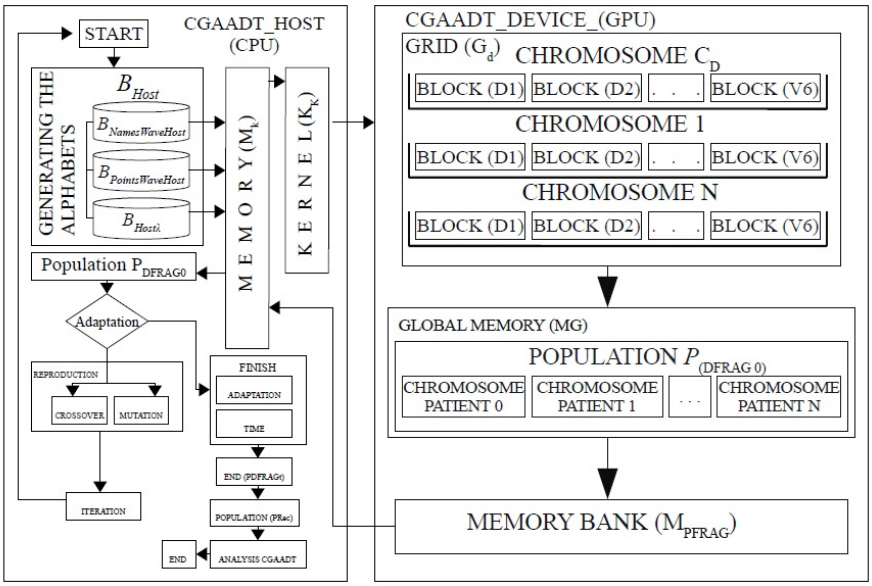

Figure 9. Architecture CGAADT in CUDA[26].

10. Σ is a set of genealogical operators acting on the population P_{DFRAG};
11. P_{DFRAG0} is a subpopulation belonging to IP(P_{DFRAG0}) called the initial population, with at least one chromosome.
12. P_{DFRAGt} is the most suitable people to be transferred to reconstruction by $P_{\text{Rec}}$the host;
13. P_{Rec} is a population rebuilt the kernel.

The CGAADT architecture shown in figure 9, the algorithm is divided into three functions:

- CGAADT-HOST;
- CGAADT-DEVICE (GPU);
- ANALYSIS-CGAADT.

The CGAADT algorithm starts from the creation of alphabets: gene, basis, and chromosome. The CGAADT-HOST function receives a so-called base B_{Host}.

This base is B_{Host} comprised of three other elementary bases, which form the basic morphology of the ECG signal, stand out: $B_{PointsWavesHost} \cup B_{NamesWavesHost} \cup B_{hostλ}$, which are transferred to the memory M_k, aims to determine the profile of the data area in the kernel (K) where each thread must on the GPU, from CGAADT-DEVICE function.

To trigger the K kernel, the CGAADT-DEVICE function initializes the parallelization of the database GD (grid blocks), from which is the grid blocks on the GPU. At this time, the basis $B_{PointsWavesHost} \cup B_{NamesWavesHost} \cup B_{hostλ}$ is converted into genes g_D, which represent a thread of sub-blocks formed by the sequence of elements B_{Host}belonging to the set AGFD. Then the genes g_D are grouped into clusters to form chromosomes C_D in the GPU, which complies with the conditions laid down ACFD.

These chromosomes C_D are grouped in sets GD to form a population and this representation will ensure fairness in the evaluation of the chromosomes that comprise the population P_{DFRAG} transferred to the global memory G_M. Each chromosome $C_D$is a patient as the database characteristics. For each patient, represented by the chromosome $c_{PATIENT0}$are assigned to the 12 (twelve) from the ECG leads, the defined set of chromosomes in the population is limited the number of patients in the existing database. These chromosomes $C_D$form the initial population P_{DFRAG0}will be transferred to the memory bank M_k, to be transferred to the kernel (K) and memory to boot the CGAADT algorithm. The CGAADT algorithm initializes its implementation as follows:

1. Initialize the function degree of gene adaptation is a function (equation 11) in the GPU;
2. Initialize the adaptive function adaptation$_D \leftarrow 0$. For every chromosome gene, select the dominant genes, which is a function like domi:$G \times G \rightarrow G$. Dominant gene is a function dominant of type:

$$domi(g_{D1},g_{D2}) = \begin{cases} g_{\lambda} & if\ (g_{D1}, g_{D2}) \notin same \\ g_{D1} & if(g_{D1},g_{D2}) \in same \wedge degree(g_{D1}) \geq degree(g_{D2}) \\ g_{D2} & if(g_{D1},g_{D2}) \in same \wedge degree(g_{D1}) < degree(g_{D2}) \end{cases} \quad (12)$$

3. Initialize the function $domi(g_{D1},g_{D2})$, whereas two genes g_{D1} and g_{D2} refer to the same feature, $\text{degree}_D(g_{D1}) \geq \text{degree}_D(g_{D2})$ namely if return g_{D1} , otherwise g_{D2};
4. Initialize the crossover function for all possible pairs of chromosomes C_D with the population P_{DFRAG} of more adapted chromosomes. Then select the dominant genes, forming all possible chromosomes with these dominant genes and include it in the population to carry the cross.
5. Initializes the mutation function, for all the chromosomes contained in the least adapted population, conduct exchanges of up to 50% of their genes, resulting in a more adapted chromosome $C_{\text{adaptation}}$ than the original chromosome C_D.

The CGAADT algorithm receives the population $P_{D\ RAG0}$ and read the environment A to submit it to the simulation of an evolutionary process, and returns a population $P_{D\ RA\mathbf{G}}$, which will be rebuilt by P_{Rec} the CPU after completion of the evolutionary process, according to the architecture in figure 9. Initializes the calculation of the adjustment of the chromosome C_D of the current population $P_{D\ RAG0}$.

The CGAADT algorithm selects the most appropriate chromosome $C_{\text{Adaptation1}}$. If the fittest chromosome is $C_{\text{Adaptation2}} \leftarrow C_{\text{Adaptation1}}$ while the process of $adaptation_D(C_{Adaptation2}) \geq$

$Adaptation_D(C_{Adaptation1})$ the algorithm selects a population of chromosomes adapted. Then the CGAADT performs the crossover operation in the population of more adapted chromosomes. The population of chromosomes not adapted, will be selected for the implementation of the mutation operator.

In this case, the chromosomes $C_{Adaptation1} \leftarrow C_{Adaptation2}$, chromosome new algorithm performs adaptation calculating and selecting $C_{Adaptation2}$. Thus, the algorithm performs CGAADT new reading to the environment A , to form the new population P_{DFRAG}=$P_{DFRAG_{CROSSOVER}} \cup P_{DFRAG_{MUTATION}} \cup P_{DFRAGt}$. This new population is added to the environment A to completion of the CGAADT algorithm.

The population of chromosomes $P_{PDFRAGt}$ is the chromosomes of the population P_{DFRAG0},P_{DFRAG1},P_{DFRAG2}, ...,$P_{DFRAGt-1}$,that best meet the requirements of the problem Rq_D. It is said then that the population $P_{PDFRAGt}$ evolved the population $P_{PDFRAG0}$. The preservation and death of the chromosomes of the current population $P_{PDFRAGt}$ crafted by CGAADT is driven by a unary predicate called $P_{cut\_off\_point}$.

This predicate belongs to the set problem requirements Rq_D on the GPU, which operates in conjunction with the SMs on the chromosome $P_{PDFRAGt}$.

The chromosomes that satisfy the predicate $P_{cut\_off\_point}$ will be part of the population $P_{PDFRAGt+1}$, while the other chromosomes of the population $P_{PDFRAGt}$ will die. The dead chromosomes can be retrieved from the taxonomy Tx_D of the chromosomes of the current population to prevent them from reappearing in the next iterations of the CGAADT function.

This restriction serves the understanding of Darwin's evolution[30], which does not contemplate the possibility of an extinct species appears again in another future time. Then CGAADT-ANALYSIS function, the population receives $P_{PDFRAGt}$ and guides the operator to rebuild the population P_{Rec} in the original data in the kernel format, which represents the algorithm the processing results involving genetic selection operations, crossover, mutation, reproduction, inclusion of offspring in the population and the waves of arrhythmias, atrial fibrillation and flutter.

4. Results

The acceleration of the CGAADT algorithm on the GPU / CUDA platform aims to partition the stratified chromosome representation (basis, gene, chromosome), during processing and cleaning of ECG signals. The methodology for developing the algorithm described in this paper, consists in dividing the chromosome stratification in two phases: the first phase is represented by the abstract term basis, composed of 12 (twelve) ECG leads, divided into unipolar and bipolar leads, identified by leads (DI, DII, DIII, Avr, Avf, Avl, V1, V2, V3, V4, V5, V6), a total of 100000 examples of data in arrhythmia database (original data) given in **Table 1**.

During the kernel filtering process on the device, the original database is partitioned into blocks by sets of fragmented representation of the genes distributed by the set of threads represented in each block in the chromosome formation at the beginning of the rolling process CGAADT. In the second phase, each chromosome has a patient related to the constant database in **Table 1.**

After running the CGAADT in the device, the population is rebuilt and downloaded from the constant memory to CPU, aiming at gathering results, which can be seen in figure 9, which represent examples of ECG leads. First signals (Figures 10 A and B) represent the abstract term basis (original database), before processing the CGAADT algorithm. In **Figure 10**(B) shows the population reconstructed after processing CGAADT, after cleaning the ECG signal.

The performance measure for assessing the adaptive filter will be evaluated by mean square error (equation 12), from all ECG leads, representing the sum of the differences between the estimated population of the genetic algorithm CGAADT and the real value of data, weighted by chromosome size.

$$MSE = e^2 = [B_{Hos} - P_{Rec}] \quad (13)$$

Patients	Frequency	Date	Age	Sex	PR	Smoker	SN
01	1000 Hz	01/10/90	81	F	IM	Not	10000
02	1000 Hz	17/10/90	58	F	IM/G/H	Yes	10000
03	1000 Hz	18/10/90	63	M	IM	Yes	10000
04	1000 Hz	23/10/90	69	M	IM/O/H	Not	10000
05	1000 Hz	24/10/90	74	M	IM	Not	10000
06	1000 Hz	29/10/90	62	F	IM	Yes	10000
07	1000 Hz	31/10/90	70	M	IM	Not	10000
08	1000 Hz	30/10/90	60	M	IM	Yes	10000
09	1000 Hz	06/11/90	66	M	IM	Yes	10000
10	1000 Hz	06/11/90	46	F	IM	Not	10000
∑							100000

Table 1. Characteristics of ECG signals. Sex (F) Female (M) Male. Precedent: (MI), Myocardial infarction (G) Gastritis, (H) Hypertension, (O) Obesity, (PR) Precedents, (SN) Sample Number. Source: Phisiobank[21] and Maciel[26].

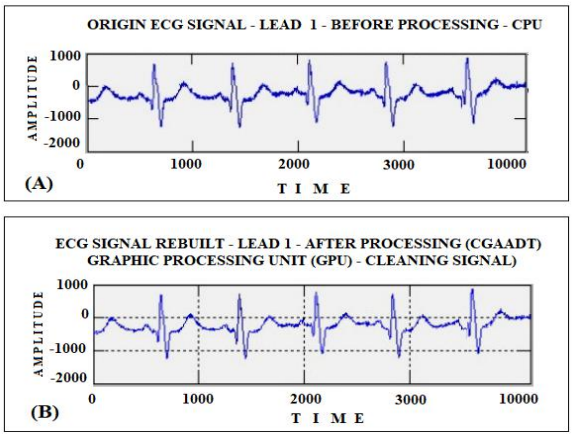

Figure 10. Processing and cleaning of ECG signals rebuilt into the CPU/GPU[26].

ALGORITHMS	P (%)	QRS (%)	T (%)	U (%)
WT	76,0	87,3	82,3	74,0
CA	-	94,3	-	-
ALS	-	99,68	-	-
KNN	-	99,81	-	-
FIR	-	99,82	-	-
CGAADT(GPU)	77,2	99,88	81,4	73,8

Table 2. Comparative map on the ECG wave detection process. Subtitle: WT (Wavelet Transform)[16], CA (Cluster Analysis)[7], ALS (Adaptive Lifting Scheme)[8], KNN (K-Nearest Neighbors)[14], FIR (Finite Impulse Response)[10-12].

The CGAADT (GPU) algorithm was run 12 (twelve) times for each ECG lead (D1, D2, D3, AVR, AVF, AVL, V1, V2, V3, V4, V5, V6), multiplied by the number of genetic operations of selection, crossover, reproduction, mutation and insertion of descendants in the population. The CGAADT algorithm was executed 60 (sixty) times for each patient exposed in table 1, totaling 600 (six hundred) executions of the CGAADT algorithm in GPU.

The CGAADT algorithm obtained 99.88% detection of the QRS complex in relation to the CA, ALS, KNN and FIR algorithms (**Table 2**), during the P, T and U wave detection process, the CGAADT is still advantageous in the detection Wave P (77.2%), Wave T (81.4%) and Wave U (73.8%) compared to Wavelet Transform (WT).

To perform all the databases (table 1), the GAADT (CPU) algorithm obtained a total processing time of 31:28:09 hours while in the new version of the CGAADT running in GPU it got a time of 00:09: 56 minutes to run all databases.

The CGAADT algorithm execution time is optimized when the genetic operators: selection, crossover, mutation, reproduction; occurs in the partitioning of the abstract base type between blocks in the grid, during the representation of fixed size chromosome and integration between the streams (SMs) from the features of the genetic operators.

When the crossover and mutation operations are performed individually, CGAADT processing time is fewer. When considering the unification of both operators into a reproduction environment, the computational effort of the GPU algorithm increased significantly. But even it becomes advantageous when compared to the computational effort of GAADT algorithm runs on conventional CPUs without performing partitioning of abstract base type between blocks in the grid, on the device.

The development of a compact genetic algorithm based on abstract data types follows the GAADT basic idea: using a representation of chromosome stratified into three awareness levels (gene, basis, and chromosome). The difference between these two algorithms lies primarily in the form of engineered genetic material as is structured in the hardware.

In this paper, the base abstract term is defined in the host, not fragmented, independent of chromosome formed in the device, only when the kernel trigger occurs, the base is fragmented into genes. The chromosome is a set of genes that are based in the fragmented structure of a grid of threads blocks, to CGAADT instantiation of the device, where the chromosome is a fixed size vector set on a given alphabet in the grid. With this fragmented perception, the chromosome is stratified into two levels represented in host and device, are parallel in the streaming environment, divided into threads units for scheduling in streaming.

The final population reconstruction is performed by the host, building genetic algorithms based on abstract data types becomes more runtime performance.

Furthermore, the spectrum of problems to which this algorithm may be adopted as a computational engine solution is higher, that is, the family of problems that can be solved by this mechanism is broader, particularly as regards the processing acceleration algorithm and collecting results. Such gain is due primarily to the structure and fragmentation of the base, before the formation of the chromosome. The genetic algorithm model presented in this article reduces the computational algorithm effort in processes that require high latency and originated by genetic operators.

5. Conclusions

The genetic algorithm CGAADT obtained 73.6% performance improvement compared with the algorithm developed Holland in GPU. The data examples given in processing and cleaning the ECG signals, presented similar results to traditional methods of adaptive filtering, distortion, not getting cardiac signals during the reconstruction of the database in the host CGAADT algorithm.

The liberation of representations uses of all the features from genetic operators contained in the compact genetic algorithm based on abstract data types of hardware, enables the choice of various chromosomes representations. About the algorithms approach adopted by the traditional model of Holland with binary fixed-size vector, which provides only a genetic feature of the proposed hardware operations views.

The use of a representation of the population with fixed size of chromosomes, pre-established by the base

fragmentation in the thread block scheduling and the streaming grid, minimizing the algorithm runtime, generated by the action of crossing, mutation and reproduction operations.

In the works developed in the GAADT software, the main disadvantage was related to the runtime interval and collect results due to the limitations in software that impossible better performance in this environment.

With the optimization algorithm CUDA, the new compact hardware version allows the use of this algorithm for the treatment of on-line applications for monitoring cardiac signals, dealing with complex problems reduced processing time.

The results obtained by CGAADT are the desired results for the issue at hand, it is the computational effort algorithm in hardware and software. The definition of a gene on chromosome weight function allows us to model the dependency relationships between the gene provided in the grid of blocks and the other genes of chromosome, thus ensuring the possibility of applying the CGAADT to problems such as epistasis. The behavior of crossing operations, ensures the transmission of relevant characteristics for the parent chromosome to chromosome child, during the fragmentation of bases in the device, thus leading to a local search among the thread blocks on the promising chromosomes.

The application of the mutation operation in the global memory on chromosomes not adapted prevents the genes relevant to the problem belonging to these chromosomes are lost and ensure the diversity of the population. The use of multiple devices GPUs, making the new approach to fragmentation base should be treated in future work.

Author Contributions

Andrilene Ferreira Maciel (Maciel A.): Currently is adjunct professor of the Federal Rural University of Pernambuco (UFRPE / CODAI)- Brazil, Ph.D. in Computer Science by Federal University of Pernambuco (UFPE) in year 2015. Master's in Computational Modeling of Knowledge by Federal University of Alagoas (UFAL) in year 2008 with Specialization in Information Management and Information Systems by Federal University of Pernambuco (UFPE), Graduation in Data Processing Technology / Information Systems in year 2003. Her research interests include Genetic Algorithm, Discovery of Knowledge in Database, Data Mining, Artificial Intelligence, Self-Organizing Maps, Artificial Neural Networks, Fuzzy Logic and Signal Processing.

Roberta Vilhena Vieira Lopes (Vieira R.V.L): Currently is adjunct professor of the Federal University of Pará (UFPA) and the Federal University of Alagoas (UFAL) – Brazil, Ph.D. in Computer Science by Federal University of Pernambuco (UFPE) in year 2003, Master's in Computer Science by Federal University of Paraiba (UFPB) in year 1997, Bachelor in Computer Science by Federal University of Pará (UFPA) in year 1994. Her research interests include Artificial Intelligence and Genetic Algorithm.

Conflict of Interest

No conflict of interest was reported by the authors.

Acknowledgments

My thanks to Roberta Vilhena Vieira for her collaboration and commitment to research on the genetic algorithm based on abstract data types.

References

1. Porto C. C. Doenças do Coração: prevenção e tratamento. Guanabara, 2005.
2. Mendis S. Global Status Report: on non-communicable diseases. WHO Library Cataloguing, 2014.
3. Oliveira W. J. Pedrosa L. C. Doenças do Coração: Diagnóstico e Tratamento. 1ª edição, Revinter, 536p, São Paulo, 2011.
4. Moreira D. A. R. Fibrilação Atrial. 2ª edição, Lemos Editorial, 311p. São Paulo, 2005.
5. Sanches P., Moffa P. Eletrocardiograma: Uma abordagem didática. 1ª edição, Rocca, 356 p, São Paulo, 2010.
6. Delbem A.C.B, Casseb M. V. G. at all. Programable logic design of a compact genetic algorithm for phasor estimation in real-time. Elsevier, 109-118, 2014.
7. Yeh Y., HongJhih L., Chiou C. W. Analyzing ecg for cardic arrhythmia using cluster analysis. expert systems with applications. Elsevier, page 1000:1010, 2012.
8. Jiang L., Ma L., Ji Z., Li Y., Yu H. Adaptive Lifting Scheme for ecg qrs complexes detection and its fpga implementation. 3rd International Conference on Biomedical Engineering and Informatics, IEEE, 2010.
9. Nianqiang L., Guoyi Z., Yongbing W. A preferable

method on digital filter in ecg signal processing based on fpga. Third International Symposium on Intelligent Information Technology and Security Informatics, IEEE, 2010.
10. Gao X. . Diagnosing Abnormal Electrocardiogram (ECG) via Deep Learning. IntechOpen, 2019. Disponível em https://www.intechopen.com/online-first/diagnosing-abnormalelectrocardiogram-ecg-via-deep-learning. Last acess: 15/06/2019.
11. Qiang L., Yao C., Yu H. Application of distributed fir filter based on fpga in the analyzing of ecg signal. International Conference on Intelligent System Design and Engineering Application, IEEE, 2010.
12. M. N. Iqbal; M. Bomhara, at. all. Real-time signal processing of data from an ECG. IEEE, Wrexham, UK, 2019.
13. Sato M. Y., Hasegawa N. Acceleration of genetic algorithms for sudoku solution on many-core processors. GECCO-11, Dublin, Ireland. ACM, 12-16., 2011.
14. Arun Khosla Indu Saini, Dilbag Singh. Qrs detection using k-nearest neighbor algorithm (knn) and evaluation on standard ecg databases. Journal of Advanced Research, Elsevier, 331-344., 2013.
15. Acharyaa. R. at all. Ecg beat classification using pca, lda, ica and discrete wavelet transform. Biomedical Signal Processing and Control, Elsevier, 437- 448., 2013.
16. Johannesen L., at all. A wavelet-based algorithm for delineation and classification of wave patterns in continuous holter ecg recordings. Computing in Cardiology, 37:979:982, Elsevier, 2010.
17. Faraji R. and Hamid R. An efficient crossover architecture for hardware parallel implementation of genetic algorithm. Neurocomputing 128, 316-327 p., Elsevier, 2014.
18. Song G., Long C. X. Z., Lua Y., Guoa M. Architecture-based design and optimization of genetic algorithms on multi-core and many-core systems. Future Generation Computer Systems 38, 75-91 p. Elsevier, 2014.
19. Holland. J. H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press., 1975.
20. Vieira R. V. V. Um Algoritmo Genético Baseado em Tipos Abstratos de Dados e sua especificação em Z, Tese de Doutorado, Universidade Federal de Pernambuco, 2003.
21. Physionet, MIT-BIH Arrhythmia Databases. Disponível em: http://www.physionet.org/physiobank/database/mitdb. Last acess: 28 de abril de 2019.
22. Cunha P. C. N. Modelo de eletrocardiógrafo portátil de baixo consumo. Dissertação de Mestrado em Modelagem Computacional de Conhecimento, Universidade Federal de Alagoas, 2012.
23. Melo B. R. P. An application of a new genetic algorithm to assist the detection of segments of an ecg signals. ISABEL' 11, ACM, Barcelona, Span, 2011.
24. Melo B. R. P. Melo. Um Sistema Adaptativo para Detecção de Ondas de Eletrocardiografia Dissertação de Mestrado do Mestrado em Modelagem Computacional de Conhecimento Dissertação de Mestrado, Universidade Federal de Alagoas, 2011.
25. Ramos V. W J. Uma aplicação do algoritmo genético baseado em tipos abstratos de dados ao problema de separação cega de fontes com não-linearidade posterior. dissertação de mestrado. Dissertação de Mestrado, Universidade Federal de Alagoas, 2011.
26. Maciel, A. Um filtro adaptativo de alto desempenho instanciado do algoritmo GAADT para o processamento de sinais de eletrocardiograma, Tese de Doutorado do Centro de Informática da Universidade Federal de Pernambuco, 2015.
27. NVIDIA CUDA C Programming Guide, Version 4.2,1999.
28. OpenCL Programming Guide for the CUDA Architecture, V 2.3, 2009.
29. Math Works - Softwares Matlab R2010, R2012a, R2012b, R2013a e R2014a. URL: http://www.mathworks.com. Último acesso em 20/07/2019.
30. Darwin, C. The Origin of Species. 1859. Disponível em : http://darwin-online.org.uk/converted/pdf/1861_OriginNY_F382.pdf. Last acess: 27/03/2019.