

Medical Diagnosis Using Adaptive Perceptive Particle Swarm Optimization and Its Hardware Realization using Field Programmable Gate Array

Shubhajit Roy Chowdhury · Dipankar Chakrabarti · Hiranmay Saha

Received: 12 June 2008 / Accepted: 14 August 2008 / Published online: 6 September 2008
© Springer Science + Business Media, LLC 2008

Abstract The paper proposes to develop a field programmable gate array (FPGA) based low cost, low power and high speed novel diagnostic system that can detect in absence of the physician the approaching critical condition of a patient at an early stage and is thus suitable for diagnosis of patients in the rural areas of developing countries where availability of physicians and availability of power is really scarce. The diagnostic system could be installed in health care centres of rural areas where patients can register themselves for periodic diagnoses and thereby detect potential health hazards at an early stage. Multiple pathophysiological parameters with different weights are involved in diagnosing a particular disease. A novel variation of particle swarm optimization called as adaptive perceptive particle swarm optimization has been proposed to determine the optimal weights of these pathophysiological parameters for a more accurate diagnosis. The FPGA based smart system has been applied for early detection of renal criticality of patients. For renal diagnosis, body mass index, glucose, urea, creatinine, systolic and diastolic blood pressures have been considered as pathophysiological parameters. The detection of approaching critical condition

of a patient by the instrument has also been validated with the standard Cockford Gault Equation to verify whether the patient is really approaching a critical condition or not. Using Bayesian analysis on the population of 80 patients under study an accuracy of up to 97.5% in renal diagnosis has been obtained.

Keywords Medical diagnosis · Criticality · Adaptive perceptive particle swarm optimization · Field programmable gate array

Introduction

Medical diagnosis is a complicated and judgmental process, based not only on medical knowledge derived from books and literatures and data obtained from various pathological tests, but also depends largely on experience, judgment and reasoning which essentially are the functions of human brain [1]. But, in many situations, availability of human brain for decision-making is scarce. Instruments, in those situations, play a major role in helping to reduce human suffering.

In third world countries, doctors are scarcely available in rural areas. A recent statistical data obtained from the Technical Report of the Indian Medical Society shows that 75% of qualified consulting doctors reside in urban areas and another 23% in semi-urban areas, and only about 2% of doctors reside in rural areas, where, unfortunately, nearly 78% of Indians reside [2]. This has created an unwarranted imbalance in patient–doctor ratio to more than 10,000 patients for one doctor in rural India [3]. The need of an inexpensive and portable equipment that can predict an imminent health hazard and red-alert the patients in rural sectors to contact doctor for necessary care, therefore, becomes very pertinent.

S. Roy Chowdhury · H. Saha (✉)
IC Design and Fabrication Centre,
Department of Electronics and Telecommunication Engineering,
Jadavpur University,
Kolkata 700032, India
e-mail: hsaha@juicentre.res.in

S. Roy Chowdhury
e-mail: shubhajit@juicentre.res.in

D. Chakrabarti
Om HealthNet Telemedicine Private Limited,
Kolkata 700055, India
e-mail: drdipankar_ohnet@yahoo.co.in

Over the past few decades, telecommunication has been used to transmit health related data of patients to remote locations [4–25]. The basic idea is to use some sort of e-mailing or videoconferencing mechanism to enable the rural patients communicate with the urban doctors at the referral centres. However, medical diagnosis through telecommunication facilities suffers from two fundamental drawbacks. Firstly, videoconferencing is too costly a technology to be implemented in the rural areas of third world countries. Secondly, with the rural masses supporting the bulk of the population and the overall patient to physician ratio being so poor (approximately 8000:1 [3]), the urban physicians find it almost impossible to cater the huge population of rural patients.

However, in the backdrop of scarcity of physicians especially in the rural areas, an instrument with some auto-decision making support can be used as a preventive device for early diagnosis of problems related to specified systems of patients' bodies. Moreover, as each doctor in these countries have to handle large number of patients, it may become handy for a doctor to keep track of the previous data of a particular patient—especially when preservation of data and documents, like investigation reports, is poor in rural backdrop.

Recently, generation of huge amounts of medical data as well as health care systems' needed for making appropriate medical decision from them easily and reliably have raised the interest in medical information processing and diagnostic systems. These systems can aid the physicians or the health care professionals in absence of the physicians to predict the pathophysiological state of a patient knowing the past pathophysiological data and red alert the physician or the health care professionals in case a critical condition of the patient occurs. Although considerable progress has been made in the field of medical information processing, computer/instrument aided diagnostic systems have not been developed as fast as other information systems due to certain problems.

One of the most critical problems that impede the development of diagnostic systems is the limited acceptance of these systems by the clinicians. Most of these systems run knowledge discovery algorithms. However, they do not satisfy the critical issues and limitations of medical applications in general. The earliest algorithm based on decision tree classifiers is found in packages like C4.5 [26]. Classical reasoning structures such as Bayesian Belief Networks lack flexibility and are weak in extracting information from the raw data, i.e., data mining [27–29].

However recent innovations have introduced novel algorithms that outperform the classical methods and are more likely to be accepted in the medical community. In particular Swarm Intelligence techniques such as particle swarm optimization (PSO) and ant colony optimization [30]

have recently been applied to rule extraction and data mining problems. Swarm intelligence is a relatively new category of stochastic, population-based optimization algorithms. These algorithms are closely related to evolutionary algorithms that are based on procedures that imitate natural evolution [31, 32]. Swarm intelligence algorithms draw inspiration from the collective behavior and emergent intelligence that arise in socially organized populations. Social and Swarm Intelligence algorithms can be used for retrieving information from the raw data or partially processed data [33]. Distributed Genetic Algorithm, Ant Colony Optimization and Particle Swarm Optimization are the most commonly used evolutionary algorithms that provide for flexible rule extraction where we are facing with incomplete and inaccurate datasets [34]. In particular PSO is a swarm intelligence algorithm that is used mainly for numerical optimization tasks. PSO gained increasing popularity in recent years due to its ability to solve efficiently and effectively a plethora of problems in science and engineering. PSO has been effectively used in a variety of data mining and optimization problems from power systems to composite beam structures and from medicine to operation research problems [35–48]. Such problems are characterized by discontinuities, lack of derivative information, noisy function values and disjoint search spaces [49, 50]. The dynamic essence of Swarm Intelligence provides flexibility and robustness.

However, the conventional particle swarm optimization suffers from some drawbacks. In a standard PSO, the further the particle is from the best position based on its own experience and its neighbor, the larger a change in velocity is to be made in order to return to that best position. The acceleration limits the trajectory of particle oscillation. The smaller the acceleration, the smoother the trajectory of the particle is. However, too small an acceleration can lead to slow convergence, whereas too large an acceleration drives the particles towards infinity. The updated velocity is limited by the maximum velocity to prevent particles from moving too fast in space [51]. Kaekawkamnerdpong and Bentley proposed a Perceptive Particle Swarm Optimization (PPSO) algorithm in [51] with the objective of further closely approaching the global optimum in the search space. However, the perception radius, number of sample points and the number of sampling directions are kept constant in PPSO. This has a serious drawback. If the number of sample points per direction and the number of sampling directions are kept sufficiently low, then the algorithm runs quite fast, but we may miss the global optimum position. On the other hand, increasing the number of sampling points per direction and the number of sampling directions, we may reach the global optimum very closely, but we shall have to pay considerably for the computation time of the algorithm.

With an intent of compromising between the two extremes, the current work proposes a novel variation of PPSO called adaptive PPSO (APPSO). In APPSO, depending upon the present position of a particle in the search space, its perception radius, and/or number of sampling directions and/or the number of sample points per direction may be varied. Since, one or more of three parameters may be varied, hence eight cases may arise. The perceptive particle swarm optimization proposed by Kaekawkamnerdpong is a special case of adaptive perceptive particle swarm optimization, in which all these three parameters are kept constant. The APPSO algorithm has been proposed with an objective of determining the optimum weights of pathophysiological parameters in the example of renal diagnosis problem. For diagnosis purposes, body mass index (BMI), glucose, urea, creatinine, systolic and diastolic blood pressures are considered as pathophysiological parameters. Since different pathophysiological parameters contribute to different degrees in different diseases, hence, weights of different pathophysiological in medical diagnosis problems should be found out instead of performing diagnosis by giving equal weights to all parameters. The optimal weights of these pathophysiological parameters in medical diagnosis have been determined using APPSO. Since, the final position of the particles (i.e. the optimal weights of the pathophysiological parameters) is not known in advance and has to be determined using an optimization algorithm, the calculation of the optimal weights of the pathophysiological parameters has been done using the proposed APPSO as an optimization tool. Moreover, swarm intelligence techniques use a tracking memory that accelerates its convergence to the best solution [52] and runs faster than that of the genetic algorithms (GA) [53]. The authors in their previous works have already established the prediction of future pathophysiological state of a patient using past pathophysiological data assuming equal weights for all medical parameters [54]. However, APPSO has been chosen as a tool to optimize the weights of the diagnosing parameters so that with a fewer set of readings the system becomes capable of predicting an approaching critical condition of the patient.

The whole system is implemented on a field programmable gate array (FPGA). The implementation of the diagnostic algorithm on a reconfigurable architecture makes it suitable for further modification of functional logic of the processor with minimum programming effort. The smart diagnostic system implemented in our work, has been applied to predict the approaching critical condition of a patient with confidence. The system being simple and user friendly could be operated by paramedical staff members who need not have to be physicians. The diagnostic system can be installed in health care centers and sub-centers of rural areas where the patients need to register for periodic

diagnoses. The system can alert the user of a potential health hazard at an early stage so that the patient can have adequate time to contact the physician in the referral center with the assistance of paramedical staff members. The whole system implemented on board consumes as low as 70 mW power so that a 7 V 12A-h battery can operate the system continuously for 50 days. Since a unit of the proposed diagnostic system can store large number of patient data, therefore it is not necessary to buy such units on a one unit per patient basis. The cost of each unit of the diagnostic system will be around 120 USD so that the government can easily afford to buy and install them at rural health care units. It is also possible for the paramedical staff members who belong to the middle class to buy such units and diagnose a large number of patients in rural areas at a nominal service charge.

For diagnosis purposes body mass index, glucose, urea, creatinine, systolic and diastolic blood pressure has been used as pathophysiological parameters. The patient data has been subjected to a process of fuzzification, inferencing and defuzzification with the aid of the designed and implemented fuzzy processor. The processor is capable of predicting the future pathophysiological state of a patient using the past pathophysiological data.

The paper is organized as follows. “[Adaptive perceptive particle swarm optimization](#)” focuses on APPSO. Next, APPSO and its application in medical diagnosis have been discussed. The application of APPSO in medical diagnosis has been discussed in “[Methodology for medical diagnosis using adaptive perceptive swarm organization](#)”. The realization of the diagnostic system on an FPGA based configurable hardware is discussed in “[FPGA based hardware are realization of the diagnostic system](#)”. The system has finally been tested with renal data of patients in “[Results and discussion](#)”. The reliability of the developed system has been tested by performing Bayesian analysis on the population under study. The detection of approaching critical condition of a patient by the developed system has also been validated with the standard Cockford Gault Equation to verify whether the patient is really approaching a critical condition or not. “[Results and discussion](#)” focuses on the different experiments being performed and the results obtained.

Adaptive perceptive particle swarm optimization

The proposed adaptive perceptive particle swarm optimization algorithm is relatively similar to the perceptive swarm optimization algorithm and the conventional particle swarm optimization algorithm. In conventional PSO, for an n -dimensional optimization problem, an n dimensional search space is considered. However, in PPSO and in the proposed

APPSO, the algorithm operates in $(n+1)$ dimensional search space. The added dimension represents the underlying performance of particles at their positions in n -dimensional space. As in PPSO algorithm, in APPSO also, the particles fly around $(n+1)$ dimensional search space. In effect, the particles fly over a physical fitness landscape observing its crests and trough from a far. Particles observe the search space within their perception ranges by sampling a fixed number of directions to observe and sampling a finite number of points along those directions. The particles attempt to observe the search space for landscape at several sampled distances from its position, in each direction. If the sampled point is within the landscape, the particle perceives the height of the landscape at that point. The particles can observe neighboring particles in their perception range. The particle randomly chooses the neighboring particles which will influence the particle to move towards them. The position of the chosen neighbor will be used as the local best position of the particle. However, unlike the PPSO algorithm, in APPSO algorithm, the perception radius and/or the spacing between the sample points along any direction within the perception radius and/or the number of sampling directions may be varied at each iteration depending upon whether the local best position of the particle at the current iteration improves or deteriorates the performance of the particle. If the local best position of the particle at the current iteration does improve the performance of the particle, not only is its personal best position updated in the next iteration, but also the spacing between the sample points along any direction within the perception radius is minimized and/or the number of sampling directions is increased and/or the perception radius is minimized so as to encourage more social interaction of the particles. Conversely, if the local best position of the particle at the current iteration deteriorates the performance of the particle, the spacing between the sample points along any direction within the perception radius is minimized and/or the number of sampling directions and/or perception radius is increased. Since one or more of the three parameters viz. perception radius, number of sample points along any direction within the perception radius and the number of sampling directions may be varied in APPSO, therefore there can be eight different types of APPSO algorithm numbered APPSO (1–8). The basic idea behind such modification is to explore the landscape more exhaustively near the local maxima so that the global maximum is very closely reached which means that the results will be more optimized in APPSO than in case of PPSO. The presence of neighboring particles influences the calculation of new velocity for the next iteration in the same way as the local social interaction in the conventional particle swarm optimization [51]. As in PPSO, in APPSO also the fitness function is the average of the height of the

landscape observed from all observation directions minus distance between the particle and the point of observation in the landscape. Figure 1 shows the difference between conventional particle swarm optimization, perceptive particle swarm optimization and the proposed adaptive perceptive particle swarm optimization.

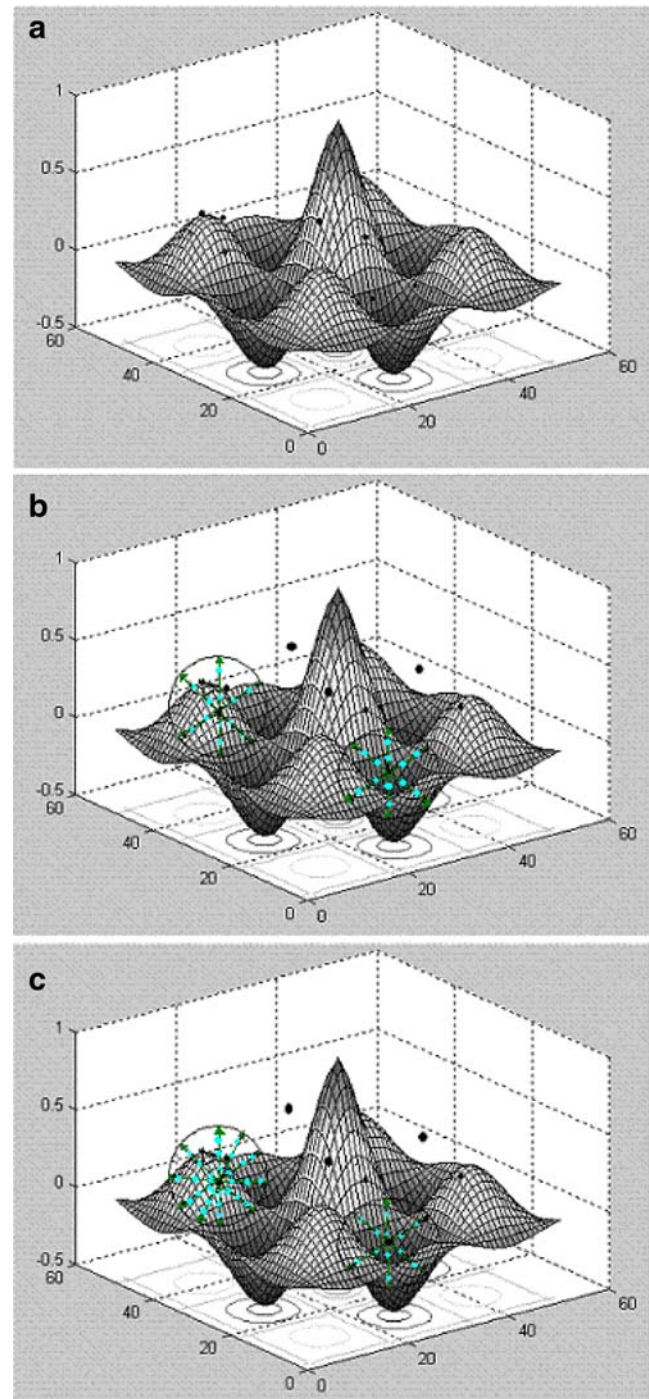


Fig. 1 Comparison between conventional **a** PSO, **b** PPSO and **c** APPSO

In Fig. 1a, the particles move about in the surface using the conventional PSO algorithm. In Fig. 1b and c, the particles are allowed to fly about in the space to observe the space and find out the optima of the landscape. However, in Fig. 1b, at each position of the particle, there are six directions of observation indicated by the arrows and two sampling points per direction. But in Fig. 1c, the particle near the trough has six sampling directions and two sample points per direction indicated by the blue dots on the arrows. But the particle near the crest has ten sampling directions and three sampling points per direction so as to explore the search space more exhaustively in search of any local optimum, which might possibly be missed out by PPSO algorithm.

Description of the APPSO algorithm

The algorithm starts with randomly initialized position and velocity of particles in an $n+1$ dimensional search space in case of an n dimensional optimization problem. The maximum and minimum values of perception radius, the number of observing directions and number of sample points along any observing direction and the maximum velocity of a particle are to be given as input. If the perception radius and/or the number of observing directions and/or the number of sample points per direction are fixed, then the fixed value has to be specified instead. The personal best position of a particle is set as the initial position of the particle. At each iteration, each particle randomly chooses the position of the neighboring particle. The local best position and velocity of the particles are updated accordingly and the fitness function is evaluated. If the present performance of any particle is better than its performance at its personal best position, then the personal best position of the particle is updated. If the perception radius and/or the number of observing directions and/or the number of sample points per direction are variable, then the spacing between the sample points along any direction within the perception radius is minimized and/or the number of sampling directions is increased and/or the perception radius is minimized so as to encourage more social interaction of the particles. On the other hand, if the local best position of the particle at the current iteration deteriorates the performance of the particle, the spacing between the sample points along any direction within the perception radius is minimized and/or the number of sampling directions and/or perception radius is increased. The iterations go on until the terminating condition is reached. The landscape under study in PPSO and APPSO algorithms is same as the landscape under study in the PSO algorithm. However, in PPSO and APPSO algorithms, the fitness function is the average of the height of the landscape observed from all observation directions minus distance

between the particle and point of observation in the landscape.

Different types of APPSO algorithm

In APPSO, depending upon the present position of a particle in the search space, its perception radius, and/or number of sampling directions and/or the number of sample points per direction may be varied. Since, one or more of three parameters may be varied, hence eight cases may arise. The eight variations are shown in Table 1.

As in evident from Table 1, the perceptive particle swarm optimization proposed by Kaekawkamnerdpong is a special case of APPSO1, in which all these three parameters are kept constant.

From the description of the algorithm itself, it is clear that there may be a slight degradation in performance with respect to the PPSO. However, at the cost of marginal loss in performance we can certainly gain in accurately reaching the global optimum position, which is particularly important in medical applications where accuracy is highly needed.

Methodology for medical diagnosis using adaptive perceptive particle swarm optimization

The present work proposes a methodology of medical diagnosis in absence of physician. For diagnosis, multiple pathophysiological parameters are involved. The present work uses the proposed APPSO algorithm for determining the weights of pathophysiological parameters in medical diagnosis. As an example of medical diagnosis, the specific case of renal diagnosis has been taken up. For renal diagnosis purposes, BMI, glucose, urea, creatinine, systolic blood pressure and diastolic blood pressure are chosen as pathophysiological parameters. However, the data from the

Table 1 Different types of adaptive perceptive particle swarm optimization algorithm

Algorithm	Perception radius	No. of directions	No. of sample points
APPSO1	Fixed	Fixed	Fixed
APPSO2	Fixed	Fixed	Variable
APPSO3	Fixed	Variable	Fixed
APPSO4	Fixed	Variable	Variable
APPSO5	Variable	Fixed	Fixed
APPSO6	Variable	Fixed	Variable
APPSO7	Variable	Variable	Fixed
APPSO8	Variable	Variable	Variable

patient cannot always be trusted as they are subjected to the quality and accuracy of measuring units and the skill of the technician. Moreover, based on a single data, it would be highly uncertain to make an accurate decision about the future pathophysiological state of the patient particularly in a chronic case [34]. So the patient data has been fuzzified with the objective of transformation of periodic measures into likelihoods that the pathophysiological parameter of the patient is high, low or moderate in comparison to a set of reference values [34]. As an example study, the present work comprises of analyzing the renal data of patient and predicting the future physiological state of a patient. From the height in feet and weight in kilograms, the BMI of a patient is computed by the system as given in [55]. Since doctors are more interested in knowing whether the pathophysiological risk parameters of patients are high, moderate or low, and also the trend of physiological parameters of patients, it would be more useful, to represent the pathophysiological risk parameters of patients as linguistic variable rather than ordinary variable and use fuzzy logic to build a predictive model, to predict the fuzzy set (low, moderate or high) in which the particular risk parameter of the patient, viz. B.M.I, glucose, urea, creatinine and blood pressure is to lie in the next reading of patient data. For this purpose, triangular and trapezoidal fuzzy operators have been used [54].

Fuzzification of patient data

The membership function has been determined in accordance with the ranges and tolerance limits set up by the World Health Organization. The plot of the membership functions is shown in Fig. 2 to illustrate the methodology.

Figure 2 depicts the membership functions used for fuzzy modeling pathophysiological data of patients. The trapezium to the extreme left correspond to the fuzzy set for low values of creatinine. The middle one corresponds to the fuzzy set for moderate values of creatinine and the trapezium to the extreme right correspond to the fuzzy set for high values of creatinine. It is obvious that all the low, moderate and high risk parameter ranges (herein modeled as fuzzy sets) of patient falls in the same universe of risk parameter values.

Algorithm for diagnosis

The algorithm for diagnosis accepts the patients' data periodically (e.g. at 10 days interval of time) and fuzzify them using the membership functions shown in Fig. 2. Using these membership function values, the algorithm computes the time- weighted mean of the membership functions of the patient's pathophysiological data collected over a number of time sequences which indicates the possibility that the next pathophysiological data will be low, moderate or high. The

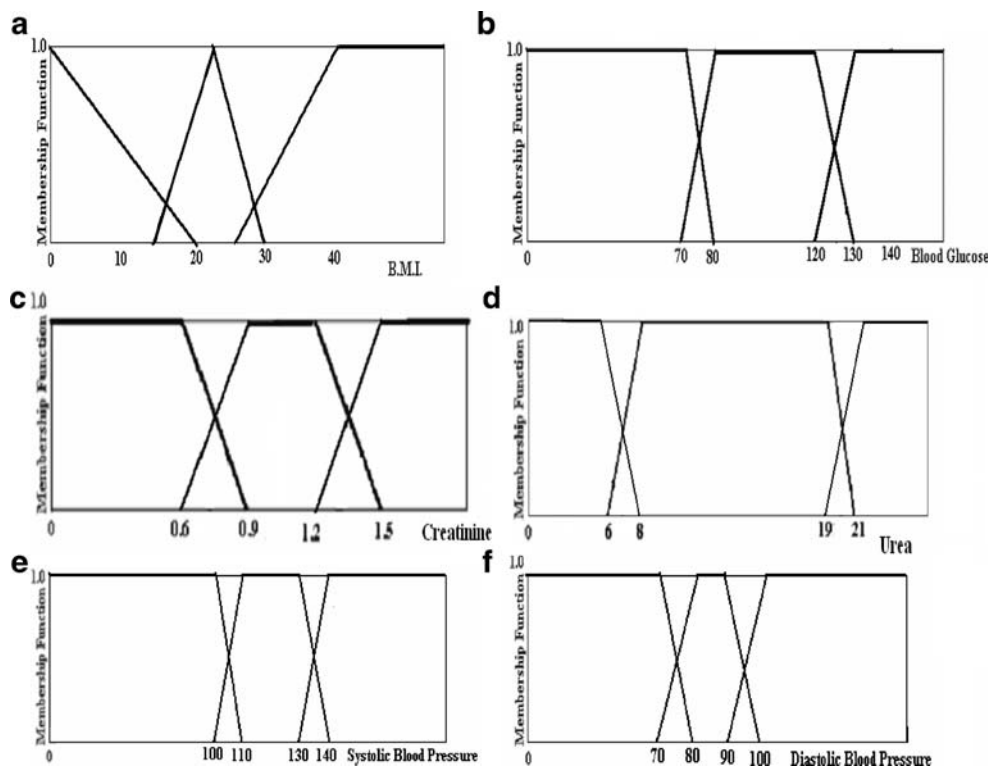


Fig. 2 Plots of the membership functions: **a** BMI **b** Glucose **c** Creatinine **d** Urea **e** Systolic Blood Pressure **f** Diastolic Blood Pressure

possibility that the next pathophysiological data will be low or moderate or high is computed as:

$$P_R(x) = \frac{\sum_{i=1}^n i\mu(x)}{\sum_{i=1}^n i} \quad (1)$$

where the summation is done from the time sequence number $i=1$ to n , n being the sequence number of the most recently taken data. $R \in \{\text{low, moderate, high}\}$. $x \in \{\text{b, g, u, c, s, d}\}$ accordingly as x represents BMI, glucose, urea, creatinine, systolic blood pressure and diastolic blood pressure respectively. $\mu(x)$ is $\mu_l(x)$, $\mu_m(x)$ or $\mu_h(x)$ accordingly as the membership function concerned refers to low, moderate or high fuzzy set respectively. The value of $P(x)$ corresponding to $\max(P_R(x))$ predicts the fuzzy set in which the next state input of a certain pathophysiological parameter is going to lie.

Inferencing

Inferencing involves giving a decision whether the patient is in normal condition or heading towards a moderately critical condition or a severely critical condition. Inferencing is done by taking the possible next state output of the diagnostic algorithm at different instants of time. For more precise inferencing purposes in renal diagnosis, the weighted sum of possibility values of low, moderate and high ranges of different pathophysiological parameters of medical diagnosis is taken. The diagnosing weights of the pathophysiological parameters have been determined using APPSO. We have determined the weights of diagnosis using all the eight possible cases of APPSO and finally have chosen the one that gives the best diagnostic accuracy. In order to compare the performance of the proposed APPSO algorithm in respect of accurately determining the optimal weights of the pathophysiological parameters, the weights of the pathophysiological parameters have been determined using other evolutionary algorithms like PSO and GA. Apart from the inferencing regarding the renal state of a patient, inferencing also involves giving specific decisions about specific diseases. Typical rules for inferencing include:

- R1: If (Weighted sum of possibility values of Body Mass Index, Glucose, Urea, Creatinine, Systolic Blood Pressure and Diastolic Blood Pressure in the high range exceeds that in the moderate range) then the (Renal condition of Patient is severe)
- R2: If (Body Mass Index is High) or (Glucose is High) or (Urea is High) or (Creatinine is High) or (Systolic Blood Pressure is High) or (Diastolic Blood Pressure is High) then the (Renal condition of Patient is moderately critical)

- R3: If (Glucose is High at time T_i) and (Glucose is Low at time T_j) and ($T_i \neq T_j$) then the (Patient can be suggested to go for Glycosylated Haemoglobin)
- R4: If (Body Mass Index is Moderate) and (Glucose is Moderate) and (Urea is Moderate) and (Creatinine is Moderate) and (Systolic Blood Pressure is Moderate) and (Diastolic Blood Pressure is Moderate) then the (Renal condition of Patient is normal) and so on.

Optimization of fuzzy inferences

The optimization of fuzzy inferences has been done based on the consideration that only part of the antecedents has a positive degree of truth and so only part of the rules have a positive degree of activation. So it is more optimal to identify the antecedents with a positive degree of truth. Only these antecedents are processed and the degree of truth is only calculated for active rules.

Optimization of the rule base

A typical fuzzy medical diagnostic application consists of groups of rules that share the common antecedent. This suggests not storing all the antecedents but only the terms which vary; the constant terms for each group are stored only once. In the actual implementation, two separate address spaces are used. One memory contains constant terms and the number of rules in the group. The other contains the other antecedents in each rule. This leads to a huge saving in memory space for rule storage.

The proposed methodology of medical diagnosis is summarized in the flowchart shown in Fig. 3:

Defuzzification and decision making

Defuzzification involves taking a crisp action based on the inference drawn. For defuzzification, the rule with the largest

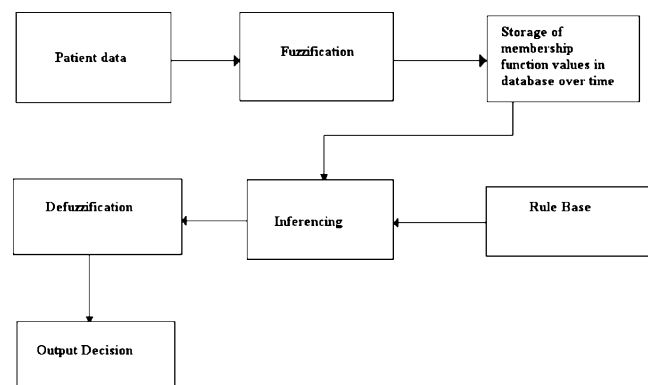


Fig. 3 Block diagram of the medical diagnosis process

firing strength is selected. The decision given by the smart agent can be implemented by glowing of an LED or by means of some output data. In our scheme, the approaching criticality of patients is indicated by the glowing of LEDs.

Determination of weights of pathophysiological parameters using APPSO algorithm

To illustrate the approach, the example of renal diagnosis problem is taken up. Our objective is to determine the diagnosing weights of BMI, glucose, urea, creatinine, systolic

blood pressure and diastolic blood pressure. Let x_b , x_g , x_u , x_c , x_s and x_d be the diagnosing weights of BMI, glucose, urea, creatinine, systolic blood pressure and diastolic blood pressure respectively which will be estimated using APPSO. As stated in the algorithm, when the weighted average of the possibility values of different pathophysiological parameters in the high range exceed the weighted average of the possibility values of different pathophysiological parameters in the moderate range, the patient has a possibility of attaining a critical condition. Hence, the patient attains a critical condition when the inequality:

$$(x_b P_h(b) + x_g P_h(g) + x_u P_h(u) + x_c P_h(c) + x_s P_h(s) + x_d P_h(d)) > (x_b P_m(b) + x_g P_m(g) + x_u P_m(u) + x_c P_m(c) + x_s P_m(s) + x_d P_m(d)) \text{ holds good, where } x_b, x_g, x_u, x_c, x_s, x_d > 0 \quad (2)$$

The basic idea is to make the diagnosing system predict a condition of criticality at the proper time sequence if the patient is really approaching a critical condition.

To determine the specific weights of the six diagnosing parameters using APPSO, we can consider a seven dimensional search space spanning over the set R , i.e. the position vector of a particle will be a point in R^7 . The landscape in the search space for our particular problem is defined by,

$$f(x_b, x_g, x_u, x_c, x_s, x_d) = \sum x_j (P_h(j) - P_m(j)) \quad (3)$$

where $j=b, g, c, u, s$ or d accordingly as the parameter concerned is BMI, glucose, urea, creatinine, systolic blood pressure or diastolic blood pressure respectively. The weighted sum of the differences between high and moderate values of the pathophysiological parameters determines the function that is used to define the landscape in the search space in the context of the present problem. As in PPSO algorithm [51], the fitness function defined in APPSO algorithm is the average of the height of the landscape observed from all observation directions minus distance between the particle and the point of observation in the landscape. The proposed APPSO algorithm is used to determine the weights of the pathophysiological parameters. The experimental parameters taken are discussed in “Weights of pathophysiological parameters”.

In order to determine the weights of the pathophysiological parameters of diagnosis, the weights of diagnosing parameters are determined at different instants of time and their average is taken. The weights are determined using all the eight possible cases of the APPSO algorithm as discussed in “Methodology for medical diagnosis using adaptive perceptive particle swarm optimization”. The weights of the diagnosing parameters are determined with half of the total set of data and testing the diagnosing property of the developed system is done with remaining half of the total set of data.

FPGA based hardware realization of the diagnostic system

The proposed system has been implemented on board using an Altera Cyclone family EP1C6Q240C8 FPGA chip for realizing the logic functionality. The proposed system could have also been implemented using software. But the main disadvantage of the software solution is that a powerful computer is to be used to run the software for achieving reasonable speed and accuracy. However, employing a powerful computer would be too costly a solution and would require a steady supply of electricity in rural sectors. The enormity of the cost of power could impede the implementation of the smart diagnostic system in the rural health care centres in the third world countries. The whole system implemented on board consumes as low as 70 mW power so that a 7 V 12A-h battery can operate the system continuously for 50 days. The main reason for a hardware based implementation is the need for an inexpensive portable diagnostic system. An ASIC based implementation could have given better performance in terms of speed and power dissipation. However, the main disadvantage of an ASIC based hardware is the high development cost and the low reconfigurability it allows for. The FPGA solution ensures that new changes in the proposed diagnostic algorithm can be mapped onto the hardware without having to make costly changes.

In order to realize the FPGA based diagnostic system, a VHDL model of the proposed system has been developed and finally mapped onto the EP1C6Q240C8 chip. The top level entity of the system comprises of 13 input ports. Of these ten input ports named as are used for reading input data in binary form. Amongst the other four inputs one is the clock input CLK, one is the reset input RST which is an asynchronous input, the third one is the control input PB which when on, the data is read from the input ports. There

are 14 output ports. Seven output ports named as DISP_SEG (0 to 6) are used to output the binary code for display on a LED 7 segment displays. DP is used for displaying the decimal point. Since, there are four seven segment displays for output in the final system; hence in order to select the appropriate seven segment display using a single 7 bit port, a 4 bit output code called SCAN (0 to 3) is used that selects the seven segment displays in the time shared mode. The change over takes place so fast as to give the illusion of a continuous display of four LED 7 segment displays. The behavioral modeling of the system has been done by considering the heart of the system as a finite state machine.

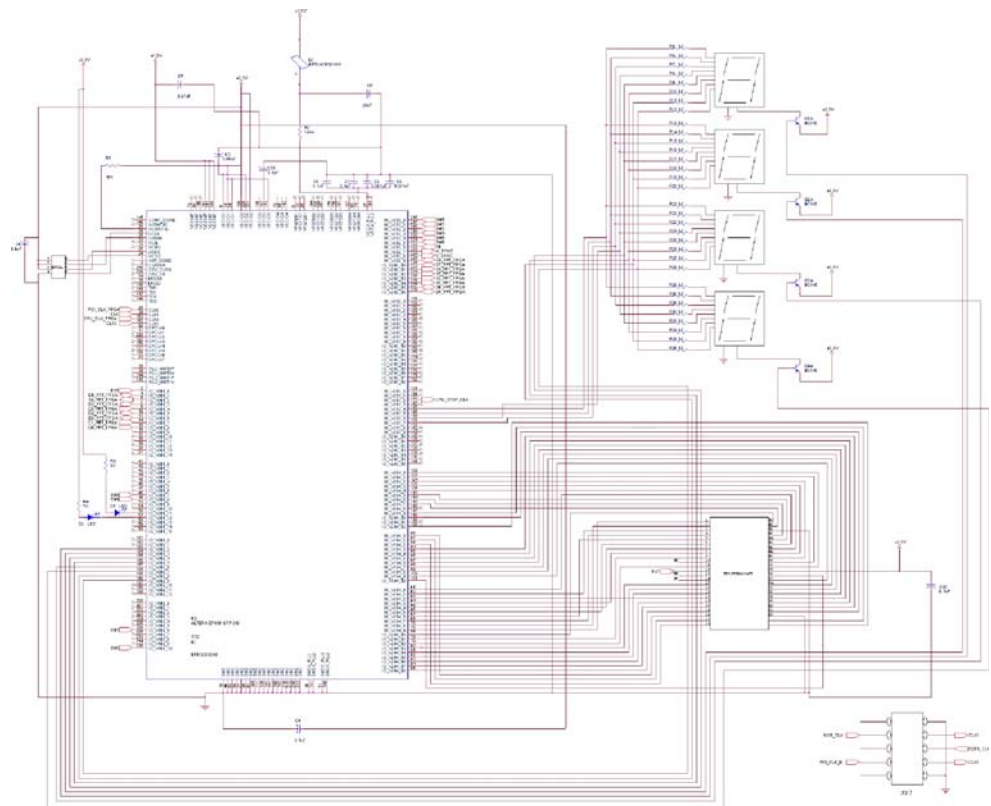
In order to map the VHDL model into an FPGA, a bit-stream pattern was generated using the VHDL model of the system, and finally downloaded on the FPGA chip using the JTAG interface. The bit-stream pattern is written in a .sof file (telmed.sof) that has been downloaded using Byte-Blaster II for JTAG configuration. The inputs have been given to the FPGA chip through push button switches. The FPGA chip receives a '0' input when each switch is pressed. The binary data entered through push button switch array have been converted into real numbers for computation using conversion-weights stored in an EPROM. Using these parameter values, the corresponding membership function values μ_L , μ_M , and μ_H are computed. μ_L , μ_M , and μ_H refers to the membership of a pathophysiological parameter value in the 'low', 'moderate' and 'high' fuzzy set respectively. Based on these membership function values, the possibilities the values of the different

pathophysiological parameters will be low, moderate or high has been computed by the system using the methodology described in "Methodology for medical diagnosis using adaptive perceptive particle swarm optimization". The maximum of these three possibilities at any instant of time suggests the possible next physiological state of the patient. The possibility values are also stored in a TC58FVB160AFT CMOS flash memory. The system predicts a condition of approaching criticality when the weighted sum of high possibilities exceeds the weighted sum of moderate possibilities. The output therapeutic decision has been displayed on LED 7-segment display. Figure 4 shows the circuit diagram of the FPGA based smart diagnostic system.

The heart of the circuit consists of the Altera Cyclone EPIC6Q240C8 FPGA chip in which the decision making logic is realized through programming. The FPGA is interfaced with an EPCS1 configuration PROM chip that stores the configuration bits for configuring the FPGA for the particular application. The patient data and the rules for inferencing are stored in a CMOS flash memory.

A typical fuzzy medical diagnostic application consists of groups of rules that share the common antecedent. This suggests not storing all the antecedents but only the terms, which vary; the constant terms for each group are stored only once. In the actual implementation, two separate address spaces are used. One memory contains constant terms and the number of rules in the group. The other contains the other antecedents in each rule. This leads to a huge saving in

Fig. 4 Circuit diagram of the FPGA based smart diagnostic system



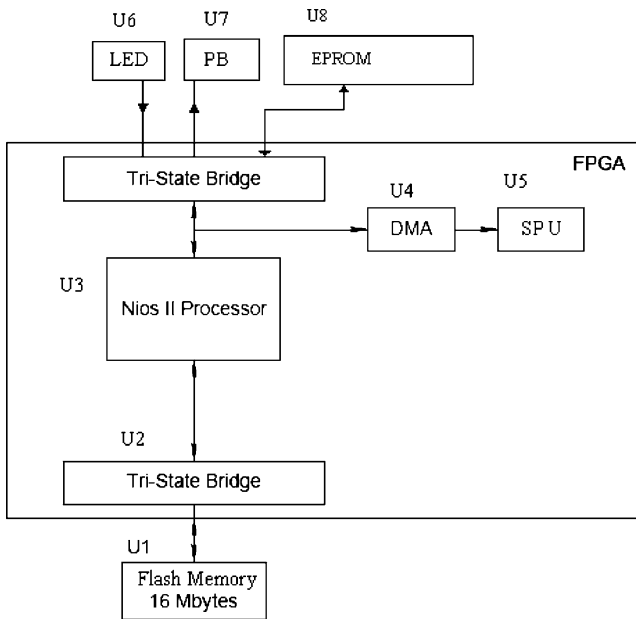


Fig. 5 Medical diagnostic system codesign on EP1C6Q240C8

memory space for rule storage. The rule memory can store a maximum of 256 rules. The number of memory locations can be sufficiently increased to store much more than 256 rules. There can be more than 256 rules in principle. However, for our particular application of medical diagnosis, we do not require more than 256 locations in the rule memory. The proposed diagnostic system does not calculate the degree of activation for all the rules, but only for active ones. This is achieved in the following way: As a group features the presence of constant terms, if their degree of truth is null, it is useless processing the rules in this group. The system can thus pass on to the rules in the next group.

The whole system is realized by configuring the FPGA as a Altera Nios II embedded soft core processor which handles the task of interfacing the peripherals with FPGA. The LED 7 segment displays are driven by BC 545 PNP transistors. The seven segment displays indicate the possibilities of low, moderate and high values of the different pathological parameters at the next physiological state of the patient. Since, there are four seven segment displays for output in the final system, and there is only one port available for display, hence a four bit output called SCAN (0 to 3) is used. Actually the different bit lines of the SCAN are connected to cathodes of different common cathode LED 7 segment displays so as to select the seven segment LED in time shared mode. The display codes corresponding to the seven-segment display has been stored in a ROM. The system can be reset at any point in time by a reset input which has been implemented using a push button switch. Two LEDs connected in the common anode mode indicate whether the condition of the patient is moderately critical or severely critical. The whole system is provided with a battery back-up

to provide a continuous power supply to the system so as to overcome the non-volatility of the FPGA chip.

For peripheral interfacing, a Nios II soft core processor has been realized in the FPGA. The architecture of the whole system including the Nios II soft core processor is shown in Fig. 5:

The main modules in the design are:

- U1: 2 MB of Flash Memory containing the patient data.
- U2: Tristate bridge
- U3: Nios II processor
- U4: DMA controller configured to feed the smart processing unit (U5) with patient data.
- U5: Smart Processing Unit that performs the task of decision making.
- U6: Four LED seven segment displays and two LED for display of output results.
- U7: 11 Push button switches for entering input data
- U8: EPROM that contains the look up table for storing the display codes and conversion weights of codes.

The algorithm implemented in VHDL is shown in the flowchart is shown in Fig. 6:

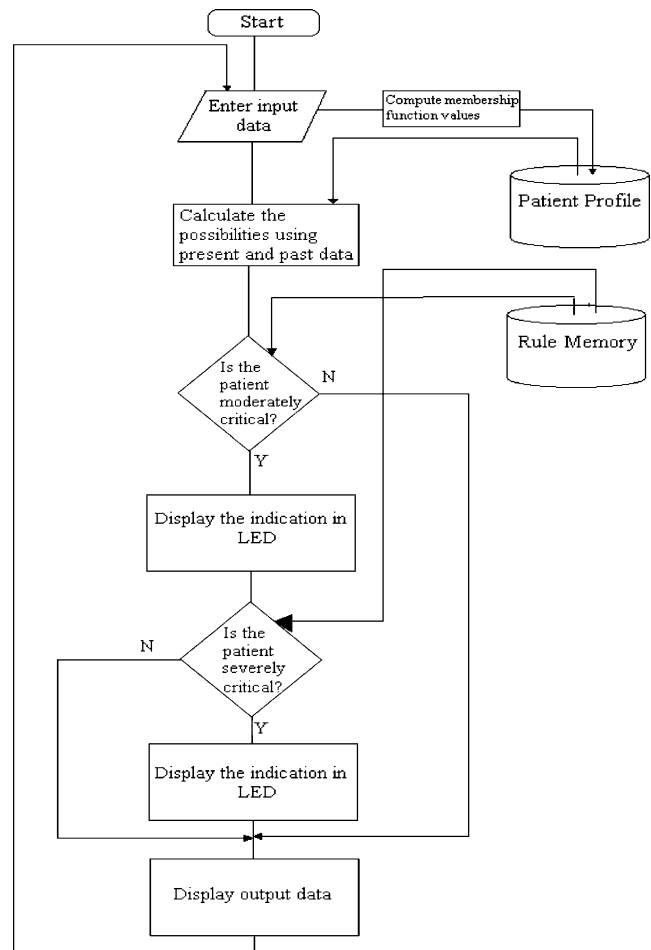


Fig. 6 Software algorithm of the system

The technology schematic of the smart agent as realized on the FPGA is shown Fig. 7.

The technology schematic shows how the look up tables (LUTs) in the FPGA chip have allocated and the connections between the different LUTs to realize the system

on the FPGA. Table 2 shows the resource utilization summary of the smart agent.

The resource utilization summary shows the number of different types of blocks realized on the FPGA for realizing the particular application. From Table 1, it is clear that only 13

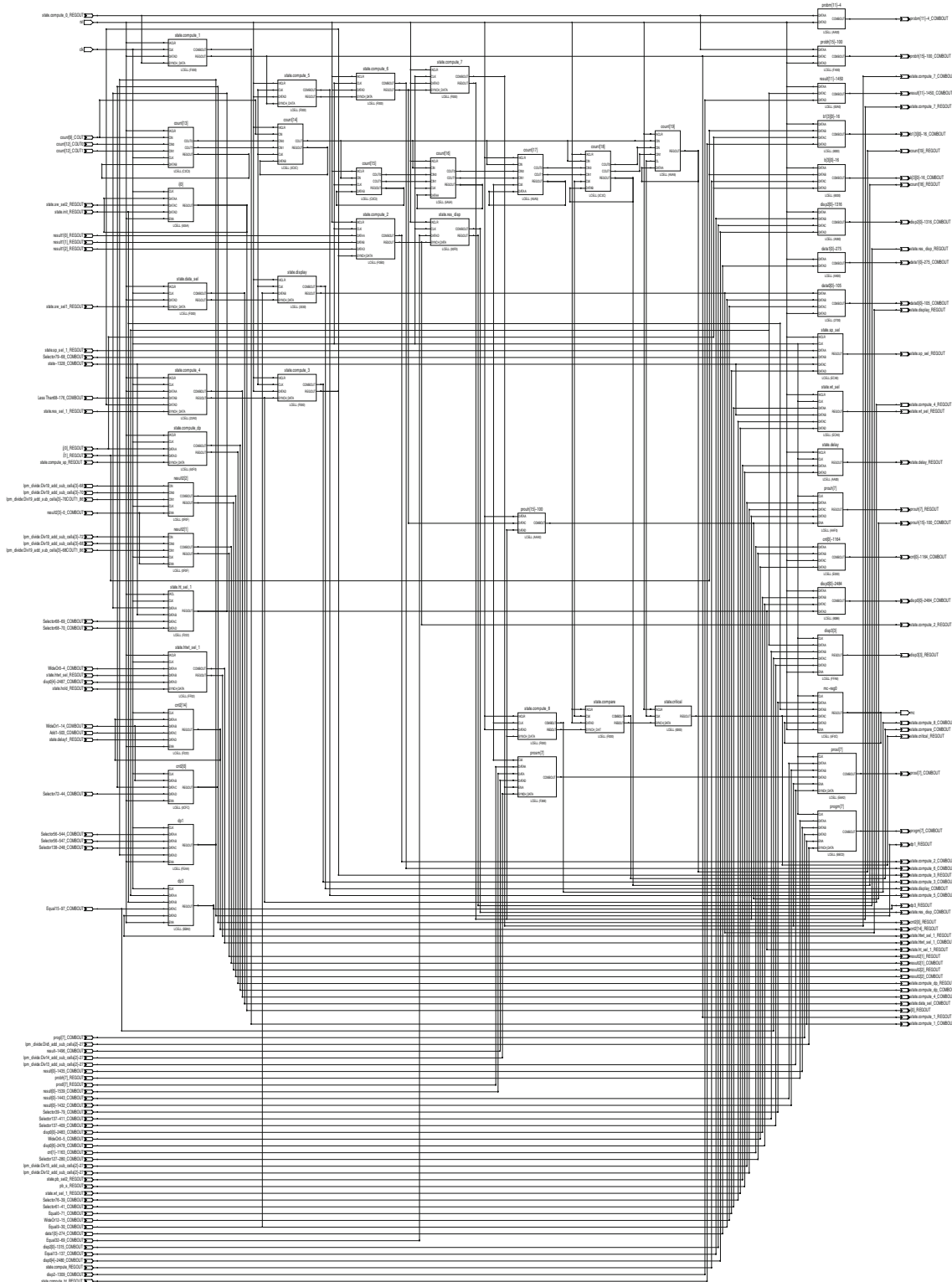


Fig. 7 Technology schematic of the FPGA based smart agent

Table 2 Resource utilization summary of the smart agent

Resource	Usage
Combinational logic elements with no register	4382
Registers	848
Combinational logic elements with a register	285
Total logic elements	5967 (out of 5980)
Total logic cells in carry chains	2354
I/O pins	27
Maximum fan out node	Clk
Maximum fan out	1164
Total fan out	15238
Average fan out	2.87

logic elements in the FPGA have been wasted in achieving an FPGA based implementation of the system. The FPGA based implementation of the system is very attractive owing to the fact that FPGAs are reconfigurable and becoming economical and faster day by day. The system has been tested with real life patient to find out the correlation between the decision being taken up by the physician and the decision being taken up by the smart agent.

Results and discussions

In order to determine the weights of the pathophysiological parameters, the APPSO algorithm is run using data of 40 patients from Chittaranjan National Medical College and Hospital. The FPGA based diagnostic system is then tested with the data of 40 patients to test the applicability of the system being implemented.

Weights of pathophysiological parameters

As discussed in “[Methodology for medical diagnosis using adaptive perceptive particle swarm optimization](#)”, eight variations of APPSO have been used to determine the

weights of the diagnosing parameters. As discussed in “[Adaptive perceptive particle swarm optimization](#)”, in the different variations of APPSO algorithms, one or more or none of the perception radius, number of sampling directions and the number of sample points per direction has been varied. If the perception radius is varied, as in APPSO (5–8) algorithms, the minimum perception radius is varied between 3 and 9 units; otherwise the perception radius is kept fixed at 6 units. If the number of sampling directions is varied as in APPSO (3, 4, 6, 8) algorithms, the number of sampling directions is varied between 4 and 12 units; otherwise the number of sampling directions is kept fixed at 8 units. Similarly, if the number of sample points per direction is varied as in APPSO (2, 4, 6, 8) algorithms, the number of sample points per direction is varied between 4 and 12; otherwise the number of sample points per direction is kept fixed at 8. To determine the weights of the parameters so that the system can clearly indicate whether the patient is tending towards a critical condition or not, the weights of the pathophysiological parameters are obtained separately from the possibility values of different pathophysiological parameters of 40 patients at different instants of time using APPSO (1–8), PSO and GA and then averaging is done in each case. The weights of the different pathophysiological parameters obtained by the four techniques are shown in Table 3:

These diagnosing weights are used by the system to make an inference about the current pathophysiological condition of the patient. It is very interesting to note that the parametric weights obtained through statistical calculations using APPSO exactly correspond to the prevailing medical judgment associated with these parameters.

Computation time of the algorithms

The running time of the proposed APPSO (1–8) algorithm on a Pentium IV machine has been determined. The performance of the proposed APPSO algorithm has been

Table 3 Weights of different pathophysiological parameters

Algorithm	Weights of parameters					
	BMI	Glucose	Urea	Creatinine	SBP	DBP
APPSO1 (PPSO)	22.18	57.03	71.01	96.57	35.36	80.78
APPSO2	22.23	56.99	71.01	96.52	35.41	80.85
APPSO3	22.17	57.04	71.05	96.59	35.48	80.86
APPSO4	22.13	57.21	71.38	96.95	35.34	80.41
APPSO5	22.28	56.85	70.98	96.47	35.44	80.87
APPSO6	22.34	56.84	70.87	96.42	35.49	80.92
APPSO7	22.24	56.97	71.03	96.54	35.43	80.84
APPSO8	22.26	57.23	71.37	96.61	35.45	80.89
PSO	22.24	56.58	70.84	96.05	35.32	81.43
GA	18.62	59.64	68.53	89.84	37.63	82.21

Table 4 Comparison of the running time of different algorithms

Algorithm	Running time (ms)
APPSO1 (PPSO)	0.68
APPSO2	0.66
APPSO3	0.67
APPSO4	0.69
APPSO5	0.64
APPSO6	0.62
APPSO7	0.65
APPSO8	0.63
PSO	0.65
GA	2.13

compared with PPSO, PSO and GA and is shown in Table 4.

The computing times of the algorithms have been determined by running the algorithms on a Pentium IV machine having a 2.8 GHz processor running Fedora 5.0. The running time of the algorithm is determined by using times() function in C. Analysis of Table 1 indicates that the running time of APPSO (1–8), PPSO and PSO are approximately the same, but the GA is much slower than the other three. However, the weights of the diagnosing parameters obtained through PSO are determined offline and these different values are only used in the FPGA based implementation of the system, which means the computing time of PSO algorithm does not affect the performance of the FPGA based hardware implementation of the system.

Results of patient data

Using the weights of the pathophysiological parameters obtained, the system is then tested with data of remaining 40 patients to ascertain the accuracy of diagnosis. To show the applicability of the system and the diagnostic algorithm, the data of a sample patient of age 42 years has been

analyzed and shown in the paper in Table 5. The data has been taken at 10 days interval of time. T1 refers to 0th day and T10 refers to the 90th day.

Using these data, the system computes the membership function values using the membership functions of the different pathophysiological parameters described in “FPGA based hardware realization of the diagnostic system”. The membership function values obtained at different instants of time is shown in Table 6.

In the above table, μ_L , μ_M and μ_H refers to the membership function values for low, moderate and high values of different pathophysiological parameters. From the above table, we come to know that the patient attains a critical condition when $\mu_H=1$ for all parameters. This happens at time T10.

Using these membership function data, the system computes the possibilities of low, moderate and high values of different pathophysiological parameters at different instants of time. Table 7 shows the possibility values of different parameters at different instants of time.

Using these possibility values, the system computes the weighted sum of the possibility values at different instants of time, called as criticality values to determine the criticality of the patient. The criticality values have been determined using all the eight APPSO, PSO and GA. Table 8 shows the criticality values determined using APPSO3.

Identically, criticality values are determined with all other algorithms [APPSO(1,2,4–8), PSO and GA]. In Table 8, C_L , C_M and C_H shows the criticality values for low moderate and high values of parameters. It is clear that at time T6, C_H exceeds C_M . Therefore, the system indicates a condition of approaching criticality at time T6 (at 50 days) much before the condition of criticality occurs (at time T10, viz. 90 days). This also elucidates that the system can be deployed in telemedicine environments in rural areas, where the health care professionals often provide support services in absence of the physician.

Table 5 Result of a sample patient of age 42 years

Time	Weight	BMI	Glucose	Creatinine	Systolic blood pressure	Diastolic blood pressure
T1	64.1	27.97	120	1.0	128	87
T2	66.2	28.31	125	1.1	131	88
T3	66.8	28.57	128	1.2	132	90
T4	67.5	28.87	127	1.3	136	94
T5	66.9	28.61	128	1.4	137	96
T6	67.8	29.00	128	1.4	138	98
T7	68.2	29.17	128	1.4	139	98
T8	69.5	29.73	129	1.4	140	97
T9	70.5	30.15	129	1.8	140	100
T10	70.6	30.62	131	2.4	143	101

Height of patient—5.0 ft

Table 6 Membership function values of different pathophysiological parameters

Time	Membership function values																	
	BMI			Glucose			Urea			Creatinine			SBP			DBP		
	μ_L	μ_M	μ_H	μ_L	μ_M	μ_H	μ_L	μ_M	μ_H	μ_L	μ_M	μ_H	μ_L	μ_M	μ_H	μ_L	μ_M	μ_H
T1	0.00	0.50	0.50	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
T2	0.00	0.17	0.83	0.00	0.50	0.50	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.90	0.10	0.00	1.00	0.00
T3	0.00	0.14	0.86	0.00	0.20	0.80	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.80	0.20	0.00	1.00	0.00
T4	0.00	0.11	0.89	0.00	0.30	0.70	0.00	0.95	0.05	0.00	0.67	0.33	0.00	0.40	0.60	0.00	0.60	0.40
T5	0.00	0.14	0.86	0.00	0.20	0.80	0.00	0.15	0.85	0.00	0.33	0.67	0.00	0.30	0.70	0.00	0.40	0.60
T6	0.00	0.10	0.90	0.00	0.20	0.80	0.00	0.20	0.80	0.00	0.33	0.67	0.00	0.20	0.80	0.00	0.20	0.80
T7	0.00	0.08	0.92	0.00	0.20	0.80	0.00	0.10	0.90	0.00	0.33	0.67	0.00	0.10	0.90	0.00	0.20	0.80
T8	0.00	0.03	0.97	0.00	0.10	0.90	0.00	0.05	0.95	0.00	0.33	0.67	0.00	0.00	1.00	0.00	0.30	0.70
T9	0.00	0.00	1.00	0.00	0.10	0.90	0.00	0.05	0.95	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
T10	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00

Validation of diagnosis by the FPGA based diagnostic system

In order to testify the fact whether the patient is heading towards a critical renal condition, the data set is verified using the Cockford Gault Equation which based on a single set of data at any point in time can indicate the glomerular filtration rate (GFR) that is an index of the renal condition of the patient.

The glomerular filtration rate of a patient at any instant of time is calculated on the basis of present inputs like age, weight and plasma creatinine according to Cockford Gault Equation [56]:

$$GFR = \frac{(140 - A) \times W}{72 \times Cr} \tag{4}$$

where A is the age, W is the weight and Cr refers to the plasma creatinine of the person. Depending on the value of

GFR, the level of risk is estimated according to the following grouping:

- GFR ≥ 90 => Stage 1 (lowest risk)
- 60 ≤ GFR ≤ 90 => Stage 2
- 30 ≤ GFR ≤ 60 => Stage 3
- 15 ≤ GFR ≤ 30 => Stage 4
- 0 ≤ GFR ≤ 15 => Stage 5 (highest risk).

Stage 5 indicates the early renal failure condition of a patient. If multiple patients have the same level of risk, priority is given to the patient with elevated blood pressure, blood glucose and blood urea levels.

For the 42 years old patient whose data is given above, the GFR is calculated and shown in Table 9:

The above table definitely indicates that the patient is heading towards a critical renal condition. The glomerular filtration rate is gradually decreasing and the patient has moved from stage 2 to stage 3 in the scale of risk levels

Table 7 Possibility values of different parameters at different instants of time

Time	Membership function values																	
	BMI			Glucose			Urea			Creatinine			S.B.P.			D.B.P.		
	P_L	P_M	P_H	P_L	P_M	P_H	P_L	P_M	P_H	P_L	P_M	P_H	P_L	P_M	P_H	P_L	P_M	P_H
T1	0.00	0.50	0.50	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
T2	0.00	0.18	0.82	0.00	0.67	0.33	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.93	0.07	0.00	1.00	0.00
T3	0.00	0.14	0.86	0.00	0.70	0.30	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.87	0.13	0.00	1.00	0.00
T4	0.00	0.14	0.86	0.00	0.54	0.46	0.00	0.98	0.02	0.00	0.87	0.13	0.00	0.68	0.32	0.00	0.84	0.16
T5	0.00	0.13	0.87	0.00	0.33	0.57	0.00	0.70	0.30	0.00	0.69	0.31	0.00	0.55	0.45	0.00	0.69	0.31
T6	0.00	0.12	0.88	0.00	0.29	0.71	0.00	0.56	0.44	0.00	0.59	0.41	0.00	0.45	0.55	0.00	0.55	0.45
T7	0.00	0.11	0.89	0.00	0.26	0.74	0.00	0.44	0.56	0.00	0.52	0.48	0.00	0.36	0.64	0.00	0.46	0.54
T8	0.00	0.10	0.90	0.00	0.23	0.77	0.00	0.36	0.64	0.00	0.48	0.52	0.00	0.28	0.72	0.00	0.42	0.58
T9	0.00	0.07	0.93	0.00	0.20	0.80	0.00	0.29	0.71	0.00	0.38	0.62	0.00	0.23	0.77	0.00	0.34	0.66
T10	0.00	0.06	0.94	0.00	0.16	0.84	0.00	0.24	0.76	0.00	0.31	0.69	0.00	0.18	0.82	0.00	0.28	0.72

Table 8 Criticality values of patient

Time	Criticality values		
	C _L	C _M	C _H
T1	0.00	352.10	17.19
T2	0.00	326.70	39.49
T3	0.00	328.76	40.79
T4	0.00	279.61	176.83
T5	0.00	213.23	178.29
T6	0.00	176.41	186.77
T7	0.00	148.72	214.46
T8	0.00	131.77	232.02
T9	0.00	105.22	257.27
T10	0.00	86.48	276.71

which definitely indicates the criticality of the condition is increasing. This is confirms the correctness of our diagnostic technique. Moreover, since in our diagnostic technique, analysis is not based on the data of a single instant of time, chances of incorrect decision due to noisy data being entered at a particular instant of time is minimized.

Estimation of reliability of the diagnostic system

Bayesian analysis has been carried out on the population under study to estimate the reliability of the system. In order to estimate the reliability of diagnosis, the definitions of statistical terms used in [57] have been used. As follows from the application of Bayes’ theorem, the predictive value of any diagnostic test is influenced by the prevalence among the tested population, and by the sensitivity and specificity of the test [58]. In our particular case, the total population under study was 80. However, the weights of the pathophysiological parameters for diagnosis have been obtained using the data of 40 patients. The system has been tested with the data of remaining 40 patients.

Let *A* be the number of patients where the diagnostic test yields a positive result and the patient really has a disease, *B* be the number of patients where the diagnostic test yields a positive result and the patient does not have a disease, *C* be the number of patients where the diagnostic test yields a negative result and the patient really has a disease and *D* be the number of patients where the diagnostic test yields a positive result and the patient does not have a disease.

Hence, $(A + B + C + D) = 40$.

Therefore,

$$\text{Sensitivity of diagnosis, } Se = \frac{A}{(A+C)}$$

$$\text{Specificity of diagnosis, } Sp = \frac{D}{(B+D)}$$

$$\text{False positive rate} = 1 - Sp = \frac{B}{(B+D)}$$

$$\text{False negative rate} = 1 - Se = \frac{C}{(A+C)}$$

$$\text{Accuracy of diagnosis} = \frac{(a+d)}{(a+b+c+d)} \times 100\%$$

The system has been tested for accuracy using the weights determined by the algorithms APSSO (1–8), PSO and GA. Apart from that, the system has been tested for accuracy of diagnosis using standard fuzzy inferencing techniques [59] like Mamdani method, Larsen Method, Tagaki, Sugeno and Kang method and Tsukamoto method. Table 10 shows a comparative study of sensitivity, specificity, 1-sensitivity (false negative rate), 1-specificity (false positive rate) and accuracy of diagnosis using the different algorithms for weighted diagnosis and fuzzy inferencing techniques.

Using the sensitivity and 1-specificity values, the receiver operating characteristics have been plotted as shown in Fig. 8.

From the receiver operating characteristics, we find a sharp peak at (1, 0.0371), which is also closest to the (1, 0) point on the receiver operating characteristic. This corresponds to the weighted diagnosis with weights determined using APPSO4 algorithm. Hence, we conclude that the APPSO4 is the most reliable one. The pseudo code of APPSO4 algorithm is shown in the Appendix.

Now Prevalence of disease, $P = \frac{(A+C)}{(A+B+C+D)} = 0.325$

Using Bayes’ theorem,

$$\text{Probability(disease | positive test result)} = P(d|+) = \frac{P \cdot Se}{P \cdot Se + (1-P) \cdot (1-Sp)} = 1.0000$$

$$\text{Probability(no disease | negative test result)} = P(d’|-) = \frac{(1-P) \cdot Sp}{(1-P) \cdot Sp + P \cdot (1-Se)} = 0.0000$$

These values of conditional probabilities give a measure of the reliability of the diagnostic system [1].

Performance analysis of the diagnostic system

The computing time for a single data set and one active group has been found to be equal to 0.24 μs. In the number of active group increases, then the computing time increases. In order to compare the delay of computation by the FPGA based hardware implementation with that of a

Table 9 Analysis of patient data based on Cockford Gault equation

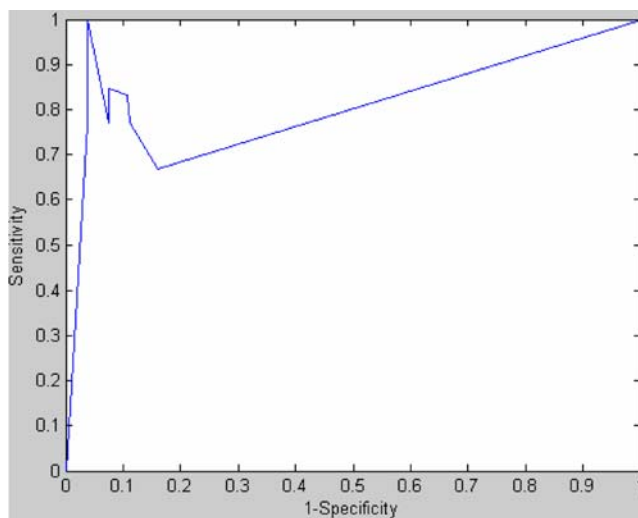
Time	Weight	Creatinine	GFR	Stage
T1	64.1	1.0	87.2	2
T2	66.2	1.1	81.9	2
T3	66.8	1.2	75.8	2
T4	67.5	1.3	70.7	2
T5	66.9	1.4	65.1	2
T6	67.8	1.4	65.9	2
T7	68.2	1.4	66.3	2
T8	69.5	1.4	67.5	2
T9	70.5	1.8	53.3	3
T10	70.6	2.4	40.1	3

Table 10 Comparative study of accuracy of medical diagnosis

Method	A	B	C	D	Sensitivity	Specificity	1-Sensitivity	1-Specificity	Accuracy (%)
APPSO1	12	1	1	26	0.9230	0.9629	0.0077	0.0371	95
APPSO2	11	1	2	26	0.8461	0.9629	0.1539	0.0371	92.5
APPSO3	12	1	1	26	0.9230	0.9629	0.0077	0.0371	95
APPSO4	13	1	0	26	1.0000	0.9629	0.0000	0.0371	97.5
APPSO5	11	2	2	25	0.8461	0.9259	0.1539	0.0741	90
APPSO6	10	2	3	25	0.7692	0.9259	0.2308	0.0741	87.5
APPSO7	12	2	1	25	0.9230	0.9259	0.0077	0.0371	92.5
APPSO8	11	2	2	25	0.8461	0.9259	0.1539	0.0371	90
PSO	11	1	2	26	0.8461	0.9629	0.1539	0.0371	92.5
GA	10	1	3	26	0.7692	0.9629	0.2308	0.0371	90
Mamdani Method	10	3	3	24	0.7692	0.8888	0.2308	0.1112	85
Larsen Method	10	4	5	21	0.6667	0.8400	0.3333	0.1600	77.5
Tsukamoto Method	10	2	3	25	0.7692	0.9259	0.2308	0.0741	87.5
Tagaki, Sugeno and Kang Method	10	3	2	25	0.8333	0.8928	0.1667	0.1072	87.5

software implementation, a sequential version of the algorithm is realized in *C* and studied on a general purpose computer (Pentium IV processor 2.0 GHz running Fedora 5.0). The actual running time of the sequential algorithm have been determined via clock ticks using times() function in *C*. The CPU time for running the sequential algorithm is found to be 0.06ms. This amount to a speed up of 343 using FPGA based implementation of the system. With an ARM 922T processor [60] of speed 200 MHz, the CPU time for running the sequential algorithm comes out to be equal to 2.5 μ s. This again amounts to a speed up of 10 using FPGA based implementation of the processor. The difference in delay of computation may stand out to be considerable when the system is redeployed for computationally intensive applications. Table 11 shows the comparative study of the different processors.

It is clear from the above table that the FPGA based fuzzy processor stands far superior to the other two

**Fig. 8** Receiver operating characteristics

processors. Moreover, assuming that there can be no more than 16 active rules at a time (which is true for our system), the processor is guaranteed to produce results within an interval of 1.92 μ s, which elucidates the real time behavior of the system. This interval of time is even less than minimum time needed to produce results on the ARM 922T processor

Conclusion

In this paper, an automated diagnostic system has been proposed to predict the approaching critical condition of a patient at an early stage. The weights of pathophysiological parameters used for medical diagnosis has been ascertained using a variant of particle swarm optimization called adaptive perceptive particle swarm optimization. The diagnostic system has been realized on an FPGA. The performance of the proposed diagnostic system has been compared with the state of the art processors and found to be better. As an example of medical diagnosis using the developed FPGA based diagnostic system, the problem of monitoring and predicting renal physiological condition of a patient has been taken up. For diagnosis purposes, BMI, glucose, urea, creatinine, systolic and diastolic blood pressures are considered as pathophysiological parameters. The optimal weights of these pathophysiological parame-

Table 11 Comparative study of the processors

Processor	Cost (USD)	Speed	Power dissipation (W)
FPGA based diagnostic system	120	5.0 MFLIPS	0.06
ARM 922T	150	0.41 MFLIPS	0.5
Pentium IV	200	0.02 MFLIPS	71.8

ters in medical diagnosis have been determined using adaptive perceptive particle swarm optimization. The correctness of diagnosis technique is verified by comparing its prediction with the indication of risk using Cockford Gault equation. The accuracy of the proposed diagnostic system has also been analyzed and found to be better than that obtained by using standard fuzzy inferencing techniques. However, for renal diagnosis, the proposed technique is much better because the analysis is not based on the data of a single instant of time; so that chances of incorrect decision due to noisy data being entered at a particular instant of time is minimized. Finally the reliability of the system has been studied using Bayesian analysis and the correctness of membership functions being used for fuzzification of patient data is verified using receiver operating characteristics.

Acknowledgements The authors would like to thank the Ministry of Communications and Information Technology, Government of India for providing the necessary fund to carry out the research work. The authors would like to thank Dr. Kaushik Chakraborty of Chittaranjan National Medical College and Hospital for his relentless support in providing the patient data. Thanks are also due to Dr. Chirasree Roy Chaudhuri of Bengal Engineering and Science University and Dr. Soumyabrata Roy Chaudhuri of Woodlands Hospital and Medical Research Centre for giving valuable suggestions regarding the validation of the system. The authors would also like to thank Prof. Amit Konar of the Department of Electronics and Telecommunication Engineering, Jadavpur University for his suggestions during the development of the system. Thanks are also due to Mr. Rajib Samadder for his support during the FPGA based implementation of the system.

Appendix

The pseudo-code of the APPSO4 algorithm is stated below.:

Algorithm APPSO4

Input: Randomly initialize position and velocity of particles in an $n+1$ dimensional search space in case of an n dimensional problem;

Output: Position of the approximate global optimum;

Begin

Initialize the perception radius, the maximum and minimum number of observing directions, maximum and minimum number of sample points along any observing direction and the maximum velocity of a particle;

/*This is also the minimum number of observing directions and minimum number of sampling points along any direction*/

Set personal best position of a particle as the initial position of the particle;

While terminating condition not reached do

Begin

For $I=1$ to number of particles

Randomly choose the position of the neighboring particle;

Update the local best position;

Update the velocity of particle;

Evaluate the fitness function;

If the present performance is better than the performance at personal best position then

Update personal best position of the particle;

Minimize the spacing between the sample points along any direction within limits;

Increase the number of sampling directions within limits;

Else if the present performance is worse than the performance at personal best position then

Increase the spacing between the sample points along any direction within limits;

Minimize the number of sampling directions within limits;

Else

Keep the spacing between the sample points and the number of sampling directions unaltered;

End if;

End if;

Increment I ;

End for;

End while;

End

References

1. Kukar, M., Transductive reliability estimation in medical diagnosis. *Artif. Intell. Med.* 29:81–106, 2003 doi:10.1016/S0933-3657(03)00043-5.
2. Indian Medical Society, Technical report of technical working group on telemedicine standardization. Technical Report, Indian Medical Society, May 2003.
3. Rural Community Development Program, http://www.rcdpnepal.com/india/india_health_sector.php
4. Eren, A., Subasi, A., and Coskun, O., A decision support system for telemedicine through the mobile telecommunications platform. *J. Med. Syst.* 32(1):31–35, 2008 doi:10.1007/s10916-007-9104-x.
5. Fidan, U., and Guler, N. F., A four channel biotelemetry system employing indoor. *J. Med. Syst.* 31(3):159–165, 2007 doi:10.1007/s10916-006-9029-9.
6. Walczak, S., A multiagent architecture for developing medical information retrieval agents. *J. Med. Syst.* 27(5):479–498, 2003 doi:10.1023/A:1025668124244.
7. Kuziemy, C. E., Laul, F., and Leung, F. C., A review on diffusion of personal digital assistants in healthcare. *J. Med. Syst.* 29(4):335–342, 2005 doi:10.1007/s10916-005-5893-y.
8. Hashiba, M., Matsuto, T., Arai, F., Yamakawa, T., and Azakawa, K., Accessing endoscopic images for remote conference and diagnosis using WWW server with a secure socket layer. *J. Med. Syst.* 24(6):333–338, 2000 doi:10.1023/A:1005596610165.
9. Druzovec, M., Welzer, T., and Brumen, B., Agent oriented approach to handling medical data. *J. Med. Syst.* 29(1):45–57, 2005 doi:10.1007/s10916-005-1103-1.

10. Neill, R. A., Mainous, A. G., Clark, J. R., and Hagen, M. D., The utility of electronic mail as a medium for patient-physician communication. *Arch. Fam. Med.* 3:268–271, 1994 doi:10.1001/archfam.3.3.268.
11. Halvorsen, P. A., and Kristiansen, I. S., Radiology services for remote communities: cost minimisation study of telemedicine. *Biom. J.* 312(3):1333–1336, 1996.
12. Fery-Lemonnier, E., Brayda, F., Charpentier, E., Couturon, I., Fay, A., and Souag, A., Transmission interhospitaliere d'images radiologiques pour la prise en charge des urgences neurochirurgicales Resultats de l'evaluation. Comité d'Evaluation et de Diffusion des Innovations Technologiques (CEDIT), Paris, 1996.
13. Bailes, J. E., Poole, C. C., Hutchison, W., Maroon, J. C., and Fukushima, T., Utilization and cost savings of a wide-area computer network for neurosurgical consultation. *J. Telemed.* 3:135–139, 1997.
14. Goh, K. Y. C., Lam, C. K., and Poon, W. S., The impact of teleradiology on the inter-hospital transfer of neurosurgical patients. *Br. J. Neurosurg.* 11:52–56, 1997 doi:10.1080/02688699746708.
15. Heautot, J. F., Gibaud, B., Catroux, B., Thoreux, P. H., Cordonnier, E., and Scarabin, J. M., Influence of the teleradiology technology (N-ISDN and ATM) on the inter-hospital management of neurosurgical patients. *Med. Inform. (Lond.)*. 24:121–134, 1999.
16. Hailey, D., and Jacobs, P., *Assessment of telehealth applications*. Alberta Heritage Foundation for Medical Research, Edmonton (Alta), 1997.
17. Wootton, R., Telemedicine: a cautious welcome. *J. Biomed.* 313:1375–1377, 1996.
18. Brecht, R. M., Gray, C. L., Peterson, C., and Youngblood, B., The University of Texas Medical Branch—Texas Department of Criminal Justice Telemedicine Project: findings from the first year of operation. *J. Telemed.* 2:25–35, 1996.
19. McCue, M. J., Mazmanian, P. E., Hampton, C., Marks, T. K., Fisher, E., and Parpart, F., The case of Powhatan Correctional Center/Virginia Department of Corrections and Virginia Commonwealth University/Medical College of Virginia. *J. Telemed.* 3(1):11–17, 1997.
20. Crump, W. J., and Tessen, R. J., Communication in integrated practice networks: using interactive video technology to build the medical office without walls. *Tex. Med. Inform.* 93:70–74, 1997.
21. House, A. M., and Roberts, J. M., Telemedicine in Canada. *Can. Med. Assoc. J.* 117(4):386–388, 1977.
22. Murphy, R. L., and Bird, K. T., Telediagnosis: a new community health resource; observations on the feasibility of telediagnosis based on 1000 patient transactions. *Am. J. Public Health.* 64(2):113–119, 1974.
23. Murphy, R. L., Fitzpatrick, T. B., Haynes, H. A., Bird, K. T., and Sheridan, T. B., Accuracy of dermatologic diagnosis by television. *Arch. Dermatol.* 105(6):833–835, 1972 doi:10.1001/archderm.105.6.833.
24. Watson, D. S., Telemedicine. *Med. J. Aust.* 151(2):62–66, 1989.
25. Davis, J. G., *Final report: Video requirements for remote medical diagnosis*. Southern Hospice Foundation, Inc., Houston, 1974.
26. Li, J., and Wong, L., Using rules to analyze biomedical data: comparison between C4.5 and PCL. Institute for Infocomm Research, Singapore, 2003 Technical Report.
27. Ghennad Rezaie, M., Soltanain-Zadeh, H., Siadat, M. R., and Elisevich, K. V., *Medical Data Mining using Particle Swarm Optimization for Temporal Lobe Epilepsy*. Proceedings of IEEE Congress on Evolutionary Computation, Vancouver Canada, IEEE Press, Vol. 1447, pp. 761–768, 2006 (Edited by: Garg G. Yen, Simon M. Lucas, Gary Fogel, Graham Kendall, Ralf Solomon, Byoung Tak Zhang, Carlos A. Coello, Thomas Philip Runarssen).
28. Larranaga, P., Learning Bayesian networks from data: some applications in biomedicine. Intelligent System Group, Department of Computer Science and Artificial Intelligence, University of Basque, Basque, 2002 technical report.
29. Kazi, J. I., Furness, P. N., Nicholson, M., Ahmed, E., Akhter, F., Naqvi, A. et al., Interinstitutional variation in the performance of Bayesian belief network for the diagnosis of acute renal graft rejection. *Transplant. Proc. Elsevier.* 31(8):31–52, 1999. Editor: Barry D. Kahan.
30. Keneddy, J., The behavior of particles. In: William Porto, V., Saravanan, N., Waagen, D. E., and Eiben, A. E. (Eds.), *Proceedings of Evolutionary Programming Conference* Vol. 8. Springer, London, pp. 32–37, 1998.
31. Bäck, T., Fogel, D., and Michalewicz, Z., *Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press, New York, 1997.
32. Schwefel, H.-P., *Evolution and optimum seeking*. Wiley, New York, 1995.
33. Kennedy, J., and Eberhart, R. C., Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks November 27–December 1, Perth, Australia, IEEE Service Centre, Piscataway, New Jersey, Vol. 4, pp. 1942–1948, 1995.*
34. Gott, M., *Telematics for health: the role of telemedicine in homes and communities*. Radcliffe Med. Press, Oxford, 1995.
35. Abido, M. A., Optimal design of power system stabilizers using particle swarm optimization. *IEEE Trans. Energ. Convers.* 17(3):406–413, 2002 doi:10.1109/TEC.2002.801992.
36. Agrafiotis, D. K., and Cedeno, W., Feature selection for structure–activity correlation using binary particle swarms. *J. Med. Chem.* 45(5):1098–1107, 2002 doi:10.1021/jm0104668.
37. Fourie, C., and Groenwold, A. A., The particle swarm optimization algorithm in size and shape optimization. *Struct. Multidisciplinary Optim.* 23:259–267, 2002 doi:10.1007/s00158-002-0188-0.
38. Ourique, C. O., Biscaia, E. C., and Carlos Pinto, J., The use of particle swarm optimization for dynamical analysis in chemical processes. *Comput. Chem. Eng.* 26:1783–1793, 2002 doi:10.1016/S0098-1354(02)00153-9.
39. Parsopoulos, K. E., and Vrahatis, M. N., Recent approaches to global optimization problems through particle swarm optimization. *Nat. Comput.* 1(2–3):235–306, 2002 doi:10.1023/A:1016568309421.
40. Parsopoulos, K. E., and Vrahatis, M. N., On the computation of all global minimizers through particle swarm optimization. *IEEE Trans. Evol. Comput.* 8(3):211–224, 2004 doi:10.1109/TEVC.2004.826076.
41. Parsopoulos, K. E., Papageorgiou, E. I., Groumpos, P. P., and Vrahatis, M. N., Evolutionary computation techniques for optimizing fuzzy cognitive maps in radiation therapy systems. *Lect. Notes Comput. Sci.* 3102:402–413, 2004.
42. Parsopoulos, K. E., and Vrahatis, M. N., Unified particle swarm optimization in dynamic environments. *Lect. Notes Comput. Sci.* 3449:590–599, 2005.
43. Ray, T., and Liew, K. M., A swarm metaphor for multiobjective design optimization. *Eng. Optim.* 34(2):141–153, 2002 doi:10.1080/03052150210915.
44. Parsopoulos, K. E., and Vrahatis, M. N., *UPSO: A unified particle swarm optimization scheme*. Proceedings of the International Conference of Computational Methods in Sciences and Engineering ICCMSE 2004, Lecture Series on Computer and Computational Sciences Vol. 1. VSP International Science Publishers, Zeist, The Netherlands, pp. 868–873, 2004.
45. Parsopoulos, K. E., and Vrahatis, M. N., Parameter selection and adaptation in Unified Particle Swarm Optimization. *Math. Comput. Model.* 46(1–2):198–213, 2007 doi:10.1016/j.mcm.2006.12.019.
46. Kathiravan, R., and Ganguli, R., Strength design of composite beam using gradient and particle swarm optimization. *Compos. Struct.* 81(4):471–479, 2007.
47. Suresh, S., Sujit, P. B., and Rao, A. K., Particle swarm optimization approach for multi-objective composite box-beam design. *Compos. Struct.* 81(4):598–605, 2007 doi:10.1016/j.compstruct.2006.09.007.

48. Feng, H., Chen, C., and Ye, F., Evolutionary fuzzy particle optimization vector quantization learning in image compression. *Expert Syst. Appl.* 32:213–222, 2007 doi:10.1016/j.eswa.2005.11.012.
49. Bonabeau, E., Dorigo, M., and Théraulaz, G., *From natural to artificial swarm intelligence*. Oxford University Press, New York, 1999.
50. Kennedy, J., and Eberhart, R. C., *Swarm Intelligence*, Morgan Kaufmann, 2001.
51. Kaewkamnerdpong, B., and Bentley, P., *Perceptive Particle Swarm Optimization: An investigation*. Proc. IEEE Symp. Swarm Intell. IEEE CS Press 1(1), 8–10 (2005)
52. Saldam, A., Ahmad, I., and Al-Madani, S., Particle swarm optimization for task assignment problem. *Microprocess. Microsyst.* 26:363–371, 2002 doi:10.1016/S0141-9331(02)00053-4.
53. Xiaohui, H., Eberhart, R. C., and Yuhui, S., *Particle Swarm with extended memory for multiobjective optimization*. Proc. IEEE Swarm Intell. Symp. Indianap. USAIEEE Press 1, 193–197 (2003)
54. Roy Chowdhury, S., Chakrabarti, D., and Saha, H., FPGA realization of a Smart Processing System for Clinical Diagnostic Applications using Pipelined Datapath Architectures. *Microprocess. Microsyst.* 32(2):107–120, 2008 doi:10.1016/j.micpro.2007.12.001.
55. Gibbs, A. L., and Braunwald, E., *Primary cardiology*. Saunders: Philadelphia, 1998.
56. Kasper, D. L., Braunwald, E., Hauser, S., Longo, D., Jameson, J. L., and Fauci, A. S., *Harrison's Principle of Internal Medicine*, 16th edition. McGraw Hill, New York, 2008.
57. Gibbs, A. L., and Braunwald, E., *Primary cardiology*. Saunders, Philadelphia, 1998.
58. Olona-Cabases, M., The probability of a correct diagnosis. In: Candell-Riera, J., and Ortega-Alcalde, D. (Eds.), *Nuclear Cardiology in Everyday Practice* 348–357, 1994.
59. Lee, K. H., *A first course on fuzzy theory and application*. Springer Verlag, Germany, 2005.
60. Fletcher, B. H., *FPGA embedded processors: revealing true system performance*. Proceedings of Embedded System Conference, Vol. 1, No. 1, pp 11–31, San Francisco, 2005