LOW POWER VLSI CIRCUITS WITH MAXIMUM SLEEP TIME AND MINIMUM NET CUT: A GENETIC ALGORITHM BASED BI-OBJECTIVE OPTIMIZATION ALGORITHM

Dr. D. Baswaraj, Mr. M. Ravi, Ms. P. Rupa, Ms. G. Swetha, CMR Institute of Technology Hyderabad

Abstract

This paper addresses the issues associated with the physical design of a very large-scale integration (VLSI). A major challenge to the physical design of a VLSI is to reduce the consumption of power by integrated circuit. Present study finds that some parts of the system of a circuit need to be deactivated at the idle time of processor in order to automatically reduce the power consumption of circuit. So, in order to attain the minimum power consumption level to the system, this proposes to increase the sleep time and to diminish the net cuts of it. These two objective functions are combined to develop an efficient normalized fitness function, thereby yielding a NP-hard model. Since genetic algorithm-based solutions are essentially global optimal, the VLSI circuit partitioning is performed with proposed efficient fitness function in modified genetic algorithm based bi-objective optimization technique. All computational parts along with input–output pads are converted into a hyper graph. While evaluating the cost function, the solution with low fitness is discarded .In MATLAB 20a, this technique applied on the net list files as used in ISPD'98 circuit benchmark suite with 20-30 nodes in each file. Significant improvement in the efficiency compared to initial partition of circuit in VLSI physical design can be achieved through the proposed algorithm.

Keywords: **C**ircuit partitioning, Sleep time, Net cut, Genetic algorithm.

1. INTRODUCTION

Recently ,VLSI chips are widely used everywhere in Engineering like Computers, Electronics, Automobiles, Voice and Data Communication networks . So it is now one of the tremendous growing industries. The main issues to the physical design of a VLSI is to reduce the average power consumption of integrated circuit. Researchers are trying to find out better algorithm and methodology for achieving better performance of VLSI circuit. Low powered integrated circuit will provide longer battery life of the modern electronic circuit like mobile, tab and laptop. Low powered circuit automatically reduce the heat dissipation and energy consumption on chip [1]. For this dynamic and sub threshold leakage power should be reduced in CMOS circuits. The average dynamic power estimation in complementary metal oxide semiconductor circuit (CMOS) can be formulated as:

$$P_{average} = ½ * F * V^2 * d* f \quad (1)$$

where, F is the load capacitance consists of gate and wire , V is the applied supply voltage, f is the frequency of operation and d is the activity density [2]. we need to reduce load capacitance (F) and switching activity(d) to minimize dynamic power and sub threshold power . Partitioning is a process to break up large complex circuit into smaller connected circuit unit.. Circuit partitioning is a non-polynomial (NP) hard problem. For achieving the improvement of partition quality of circuit genetic algorithm based solution plays important role. The proposed genetic algorithm based method might prove to be efficient to fulfill the current trend in the design of VLSI circuit . The circuit is considered as hyper graph and The first objective is to minimize the number of net cut size so that the number of inter-connections among the partition have been minimized. The load capacitance will be optimized after minimizing net cut among the partition . The second objective of this method is to maximize sleep time . this can be achieved if number of switching activity can be minimized. In a sleep mode at a particular time components or blocks have no activity. Overall power consumption of the system will be saved if the components are idle through some control signals [4] and power consumption will be automatically reduced for whole system after maximize sleep time and minimizing power loss in transmission [14]. Kernighan and Lin at bell telephone laboratories proposed first iterative algorithm for partitioning where they took random partitions on graph and after swapping pairs of nodes the number of edge cut had been minimized [3]. But local minimum is the main problem of this method. To the local minima another method proposed by Fiducca and Mattheyses was very fast [4]. In both of the methods, better fitness values had been obtained with a single point in the solution space. But, all of these methods led to the locations of false peak in multi-modal solution space [5]. Genetic algorithm based solution is used because it is possible to the search in a single point as well as multi point search .This exhibits higher degree of parallelism (large number of parallel points). Many researchers used random crossover points over the chromosome to justify the concepts of Goldberg [6-8]. An efficient multi objective hMetis partitioning for simultaneous cut size and circuit delay minimization was proposed [7] in achieving delay minimization. Based on the difference between the individual chromosomes , efficient solutions had been achieved by this method [9]. Mutation operators were emphasized in this method proposed by Yuen and Chow where revisiting was avoided so that run time complexity of the algorithm can be reduced [10]. A method was proposed by Jigang and Srikanthan that is efficient for partitioning of hardware and software. It improved power consumption of the system and reduced total running time of the program [11]. But, these methods are not are not powerful enough to choose better intelligent chromosome selection [12]. Arato et al., proposed a remarkable study where partitioning is being done using both Integer Linear Programming (ILP) and Genetic algorithm . He had shown that GA was better than ILP in achieving runtime of the system [13]. Prakash and Lal proposed a method recently which was an important methodology for multi-objective VLSI circuit partitioning using Particle Swarm Optimization (PSO) [14]. G.K. Chellamani , P. V. Chandramani proposed efficient method of optimization[32]. R.P.HYPERLINK "javascript:;" Guru, V. Vaithianathan proposed efficient vary recent VLSI circuit partitioning algorithm based on satin bowerbird optimization (SBO)[30].

The proposed GA based bi- objective algorithm is designed to achieve better solution by varying the of crossover probability . This proposed method has been used on some Standard benchmark circuit for finding the optimal solution which optimize the two objectives. MATLAB 2020a tool has been used to codes all the algorithms. Genetic algorithm based solutions are more advantageous because the search is done not only in a single point but also multi point search is possible.

2. PRELIMINARIES

2.1 Hyper graph : hyper graph H = (V, E) on a finite set of vertices (or nodes) V = { v_i : i ∈ [n] } where n ∈ N^*: [n] = { i : i ∈ N^* ∧ i <= n } is defined as a family of hyper edge E = $(e_j)_{j\in[p]}$ where each hyper edge is a non-empty subset of V and such that $U_{j\in[p]}$ [e_j = V]. It means that in a hyper graph, a hyper edge links one or more vertices [22]

figure 1: hyper graph with 9 virtices and 4 hyper edges

In figure 1, hyper graph consists of 9 vertices and 4 hyper edges which are in the following.
Vertices set= {v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9, }
Hyper edges={e_1,e_2,e_3,e_4}, e_1={v_1,v_2,$v_{3,v6}$},e_2={v6,v_7,v_8,v_9},e_3={v_5, v_8,v_9},$e_{4=}${ v_2,v_3,v_4,v_5}.

2.2 Genetic operators

Crossover: A **crossover point** on both parents chromosomes is picked randomly. Bits to the right of that **point** are swapped between the two parent chromosomes .After cross over new offspring are formed which carries **genetic** information from both of the parents [25].

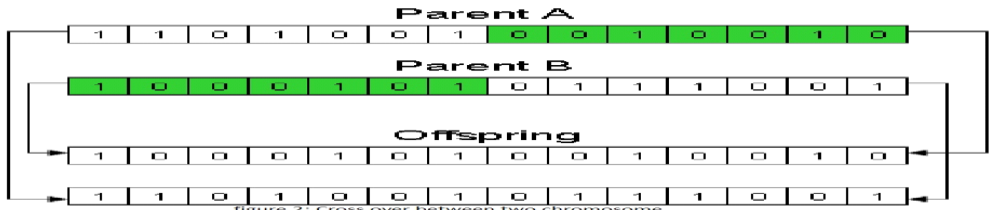

figure 2: Cross over between two chromosome

A point is chosen in the two parent chromosomes above (between the 7th and 8th genes), and copied for the offspring. After the chosen point, the remaining genes of the parents are swapped and joined to the opposite offspring.

Mutation: **Mutation operator** is used to maintain **genetic** diversity from one generation of a population of **genetic algorithm** chromosomes to the next. As Genetic algorithm is based on natural selection, better solution can be achieved using **mutation**. Mutation operator is used based on the mutation probabilities [28].

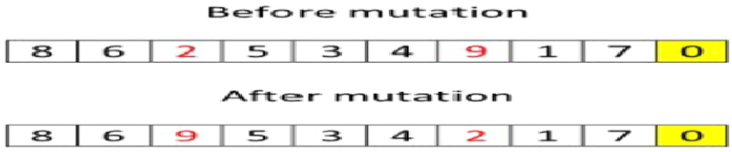

Figure 3. Mutation process

Selection: Selection is the process to select the better individual from the population after evaluating the fitness function. **We use the tournament selection in the proposed algorithm [**23].


figure 4: tournament selection

2.3. Net list: Net list is the input circuit to be partitioned. All the circuit information is a set of net list with ISPD'98 benchmark suite format . This can be considered as a hyper graph with vertices corresponding to cells (modules/ components/gates) and edges corresponding to signal nets [26]. Net list processing is done so as to convert the circuit net list in the form of chromosome[27].

3. PROBLEMS FORMULATION

3.1. Minimization of net cut: The circuit is considered as hyper graph which consist of set of modules and nets. Each net is connected with set of modules. If total number of modules is denoted as N and number of nets as P, So the circuit is composed as N Modules (functional components) where N= $\{n_1, n_2 \ldots n_m\}$ and P nets where P= $\{p_1, p_2 \ldots p_n\}$. The circuit has been divided into K partitions(blocks). The objective is to minimize the total number of cut nets. x_{is} has been defined as x_{is}=1, if the module i is in partition s (K_s) and y_{js}=0 otherwise. y_{js} has been defined as y_{js} = 1, if net j (**s**$_j$) has been completely absorbed in partition s (k_p) and y_{js} =0 otherwise. Then the objective is to minimize

F= Minimize ($\sum_{j=1}^{p} \sum_{s=1}^{k} y_{js}$) (2)

Subject to:
Module Placement Constraint:$\sum_{s=1}^{k} x_{js} = 1$
Net List Constraint: $y_{js} \leq x_{is}$, where $l \leq j \leq P$, $l \leq s \leq k$; i, j ∈ P
Constraints: $x_{is} \in \{0,1\}$, $1 \leq i \leq P$; $1 \leq s \leq k$
$y_{js} \in \{0,1\}$, $1 \leq j \leq P$; $1 \leq s \leq k$
This formula was applied on NP-hard problem with 0-1 linear integer programming problem [16-19, 29,31] . In the following example shown in figure 6, A hyper graph (circuit) consists of 6 modules 5 nets. Modules are v_1, v_2, v_3, v_4, v_5, v_6. And nets n_x^1 n_x^2, n_x^3 n_x^4 ,and n_x^6. This circuit has been

partitioned into 3 blocks. The blocks are P_1,P_2,P_3 which are shown in figure 6.. In this example net n_x^1 n_x^2, n_x^3 n_x^4 ,n_x^5,and n_x^6 are cut . so total number of net cuts is 11.

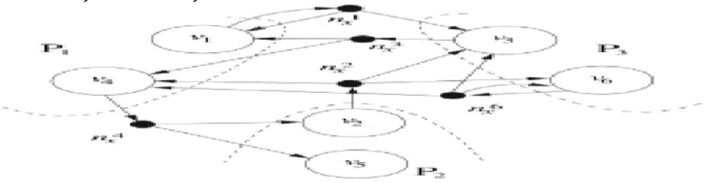

figure 6 : hyper graph with 6 modules and 6 nets which is partitioned in 3 blocks

3.2 Maximization of Sleep time

Maximization of sleep time which were shown on the studies conducted in the recent past [16], [17]. M is a set of m functional modules. Module m can be switched to sleep mode when module m is in idle state during the time interval T=(r,e) if e<r. If two intervals are given as $T_1=(r_1,e_1)$ and $T_2=(r_2,e_2)$, R is called as non-overlapping if $r_1 \geq e_2$ or $r_2 \geq e_1$. A non-overlapping interval set R_i for module m, is a set of interval during which module m could be set to sleep mode $R_i = \{T_{i1}, T_{i2} \ldots T_{im}\}$. S is the idle sets of all modules in M and was given as: $S = \{R_1, R_2 \ldots R_m\}$. An empty interval T_1 was denoted by (). It is assumed that T_1 covers T_2 if $r_1 \leq r_2 \leq e_2 \leq e_1$ or if T_2= (). The length of interval T, L (T) is defined as the intervals end point subtracted from the intervals starting point (e-r). Intersection of the two intervals T_1 and T_2 is denoted by $T_1 \cap T_2$ which is the longest interval that covered both T_1 and T_2. Intersection of two non-overlapping intervals set, $R_1$and R_2 is defined as:$R_1= \{T_{11}, T_{12} \ldots T_{1n}\}$ and $R_2= \{T_{21}, T_{22} \ldots\ldots T_{2n}\}$, $R_1 \cap R_2= \{T_1 \cap T_2 \mid T_1 € R_1, T_2 € R_2, T_1 \cap T_2 \neq ()\}$.Duration of non-overlapping intervals sets $R = (T_1, T_2 ... T_k\}$ is defined as $D(R)=\sum L(T_i)$. Given $S = \{R_1, R_2,\ldots R_m\}$. A(S) is defined as the intersection of all the non-overlapping intervals sets in S. $\{S_1, S_2,\ldots S_k\}$ is a k-partitioning of S if $\{S_1, S_2,\ldots S_k\}$ <S and $S_i \cap Sj$ = NULL and $S_1 \upsilon$, $S_2 \upsilon,\ldots \upsilon S_k$ is b-balanced if $|S_1| \geq b,|s_2| \geq b$, $|S_k| \geq b$ where |S| is cardinality of set S and equal to size of partition S. To define the objective function for maximizing the sleep time for k-partitioning problem, a gain G $(S_1, S_2 \ldots S_k)$ of a balanced k-partition is defined as follows:

G $(S_1, S_2 \ldots S_k)$ =f $(t_1, t_2 \ldots t_{p,}$ $sw_1+sw_2+sw_3+\ldots+sw_k)$ where t_i and sw_i are defined as: t_i= D (A (S_i)) (sleep time of partition) and sw_i=|A (S_i) | (number of switching's of partition S_i).It is noted that higher discrete overlapping of idle time meant greater number of switching and a more complicated control circuitry. Hence, the gain function G $(S_1, S_2 \ldots S_{p)}$ should be an increasing function of t_1 and a decreasing function of sw_1. For a p-partitioning problem the gain function that needed to be maximized was defined as: $f= t_1+t_2+\ldots+t_{p,}$ $-\beta$ $(sw_1+sw_2+sw_3+\ldots+sw_p)$ (3)

Parameter β controls relative significance of power savings (t_1) and the overhead terms (sw_1) and which depended on the available technology and on circuitry in modules m. Figure. 5.a, shows the activity profile of the modules.

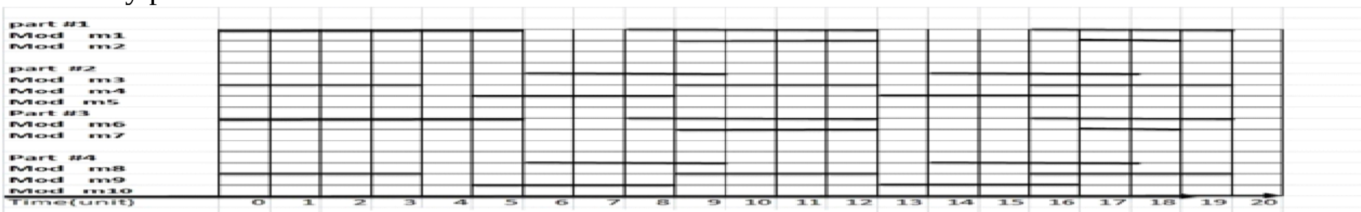

figure 5.a: Activity profile of each module

Activity profile of each module is shown with bold line in the Fig. 5.a(see figure 5.a) . Figure 5.b shows an example of overlap and switching time of four partitions part #1, part #2, part 3# and part #4. For partition part #1, overlapped time ={(8,12),(16,18)} So, t_1=6 switching time, sw_1=2; For partition part #2, overlapped time={(8,9)}, t_2=1 switching time, sw_2=1. For partition part #3, overlapped time ={(8,12),(16,18)}(see fig 5.b). So, t_3=6 switching time, sw_3=2. For partition part #4 overlapped time={(8,9),(15,16)}, t_4=2 switching time, sw_4=2. Total sleep time = $(t_1+t_2+t3+t4)$- β $(sw_1+sw_2+sw_{3+}$ $sw_{4)}$=(15-7)=8 as shown in the Figure 5.b.

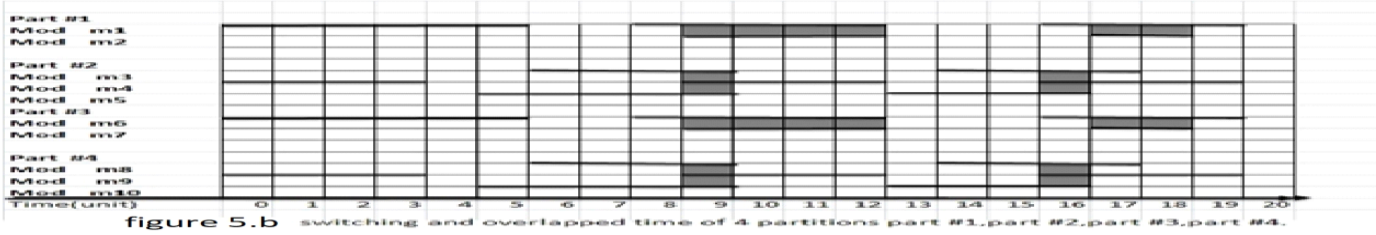

figure 5.b switching and overlapped time of 4 partitions part #1,part #2,part #3,part #4.

Figure 5.c, shows another example of overlap and switching time of four partitions four partitions part #1, part #2, part 3# and part #4. For partition part #1, overlapped time ={(8,9),(15,17)} So, t_1=3 switching time, sw_1=2; For partition part #2, overlapped time={(8,9)}, t_2=1 switching time, sw_2=1. For partition part #3, overlapped time ={(8,9),(15,17)} So, t_3=3 switching time, sw_3=2. For partition part #4 overlapped time={(8,9)} t_4=1 switching time, sw_4=1. Total sleep time = (t_1+t_2+t3+t4)- β (sw_1+sw_2+sw_{3+} $sw_{4)}$=8-6=2 . So we can say that partitions as in figure 5.b is more better than the partitions in figure 5.c as we want to maximize the sleep time in this study.

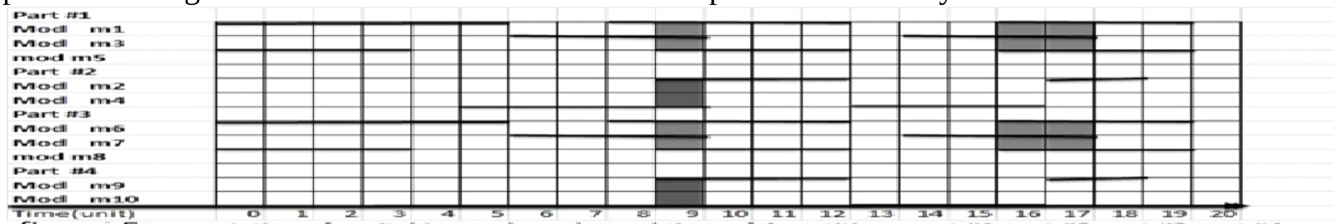

figure 5.c : variation of switching and overlapped time of 4 partitions part #1,part #2,part #3,part #4.

3.3 Dual objective function

The objectives are formulated separately and finally combined them into a single objective function

The first objective function f_1 = Minimize ($\sum_{j=1}^{s}\sum_{p=1}^{k} y_{jp}$)

The second objective function

f_2 **= Maximize**$\sum_{i=1}^{p}$ (ti-βsw_i)(maximizing sleep time) Where β is considered as 1 in this present study.

The combined function,

f = Minimize (γ_c * f_1 + γ_s *(1/(1+f_2)) (4)

Where γ_c and γ_s are cut factor and sleep factor respectively which controls the relative significance of cut nets versus sleep time in the objective function. Equal weight is given for γ_c and γ_s in this proposed study i.e γc= γ_s= 0.5. Factors of this two are summed to be unity (γ_c+ γ_s=1).

4. SOLUTION METHODOLOGY

The proposed methodology is genetic algorithm based bi-objective optimization to solve the objectives. Some parameters are taken as input to satisfy the objectives. It would help us to search for a optimized solution from the large set of possible solution. The proposed algorithms are designed to satisfy all the objectives which are formulated above.

A. Section 1

In this section a connectivity matrix has been constructed from the circuit description file (.NetD) in the ISPD 98 benchmark and a graph corresponding to the connectivity matrix has been formed . After that genetic algorithm based method has been proposed for a bi-objective optimization for satisfying all the objectives.

Proposed Algorithm for section 1.

Input: Input circuit description (.NetD of ISPD'98 benchmark). files as a text file .

Output: Connectivity matrix ,Net matrix with m by n where m is number of nets and n is number of modules and corresponding Graph.

Step 1: Prepare a text file from the circuit given description of a file.
Step 2: Convert the text file into tabular form and it is assigned to a variable aa.
Step 3. list all Source cell and connected cells by reading the table from beginning to end.
Step 4: Construct Adjacency matrix (connectivity matrix) from table aa.
4.1: Calculate the size of the table aa i.e number of rows and number of columns.

4.2: Assign k to the number of rows and l to the numbers of columns of the tables.
4.3: Read each row of the table from the beginning until end.
4.4: let i=0, j=0
4.5: do step 3.6 to 3.9 until i==k
4.6 do step 3.7 to 3.8 until j==l
4.7: If there is a connection between module V_i and module V_j then
Adjacency matrix [i, j] =1 Else
Adjacency matrix [i, j] =0
4.8: j=j+1
4.9: i=i+1;
Step 5. Construct a net matrix with m by n where m is the number of nets and n is the total number of modules.
Step 6. Construct a graph using the source modules and adjacency matrix.
Step7: End

B. Section 2

In this section the proposed genetic algorithm based method for bi-objective optimization has been applied on the graph which is formed as per section 1 .This method is used for satisfying all the objectives. In this section variation of cross over has been done and optimized results have been reported.

Proposed Algorithm for section 2:
Input: Output of algorithm 1(section 1)
P_{mt} ←Type of mutation, P_m ← Probability of mutation, P_{ct} ← type of cross over, P_c ← Probability of crossover
N ←Number of generations, final crossover probability ,Population size
$P_{activity}$ ← Activity profiles of module
Output: Set of optimal solution that satisfies the objectives
Step 1: **Partition the circuit**: Partition the circuit into k parts as it is k-way partitioning (some circuit components are in partition 1 and some circuit components are in partition 2 and .... some components are in partition k.
Step 2: **Encode the Chromosome**: The graph is traversed in BFS order and encode chromosome with partition number (from 1 to k since k-way partition is used) A chromosome is a sequence of partition number associated with vertex.
Step 3: **Initial population**: The initial population with size is created randomly that generates individuals in which the chromosome contained all partition numbers so that every node can appear exactly once.
Step 4: **Initialize with generation number, t=0 and use the genetic operator's** p_m, p_c, p_{mt}, p_{ct} whose values are defined above.
Step 5: do step 6 to step 11 until $p_c \leq 1$.
Step 6: **Evaluation**: Fitness functions are evaluated for each individual in P (t). and
Step 6.1: Calculate first objective that minimize the net cuts with the formula as in the following.

$$y(1)= \text{Minimize} \left(\sum_{j=1}^{s} \sum_{p=1}^{k} y_{jp} \right)$$

Step 6.2: Calculate second objective that maximize the sleep time with the formula as in the following.

$$F2= \textbf{Maximize} \sum_{i=1}^{p} (t_i-\beta sw_i)$$ (maximizing sleep time)

now we can convert it as

$$y(2)= 1/(1+F2).$$

As per the activity profile($P_{activity}$) for modules defined above t_i and sw_i are calculated.

Step 6.3: Evaluate the composite objective function (dual objectives) function as the following function

$$y = \text{minimize} (\gamma_c * y(1) + \gamma_s * y(2))$$

Taking γ_c =0.5 and γ_s =0.5 here.

Step 7: Form the new population

Step 8: if the new optimized solution is better than the previous generation, accept and record the solution

Step 9: Increase generation: t=t+1.

Step 10: if t=G (stopping criteria is met) go to step 11.

Else go to Step 6.

Step 10: Output the optimal solutions

Step 11: change the value of cross over probability Pc and go to step 5.

Step 12: plot fitness function value generation wise with the variation of cross over probability and mark optimized dual fitness value.

Step 13: Separate optimal net cuts and sleep time from dual fitness as per equation 2 and equation 3 .

Step 14. plot net cut value generation wise with the variation of cross over probability and mark optimized net cut value.

Step 15.plot sleep time value generation wise with the variation of cross over probability and mark optimized sleep time value.

Step 16: end

5. RESULT AND DISCUSSION

After applying the proposed method on a net list in the ISPD 98 bench mark file, the result has been prepared. Implementation of all the algorithms was done in MATLAB 20a. The method has been applied on the net list consisting of 20-30 nodes and 20-25 nets. All the programs have been executed on core-i3 (8th generation) machine with 6GB RAM and results have been reported accordingly. The procedure has been started with taking input file with extension .netD of ISPD 98 bench mark and equivalent hyper graph and graph has been constructed from the circuit description. After that, the hyper graph has been partitioned into 4 (four) equal blocks initially. Numbers of node are equally distributed in partitions so that the balanced criterion has been satisfied. The bi-objective fitness function optimize net cut and sleep time between partitions. Based on initial assumption with random partition so that the average number components distributions are equal. The values of net cut and sleep time were calculated and reported accordingly before optimization. The method has been applied one some net lists, the result has been shown with 50% weight has been given to net cut and 50% weight has been given to sleep time . After optimization the results of final fitness has been achieved and separated optimal net cut and sleep times have been shown . Graph view of the net list spp_N20_E25_R1_1944 have been shown in figure 7

.

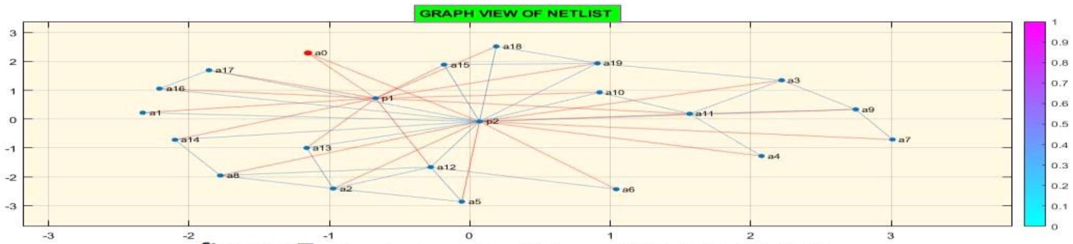

figure 7. Graph view of netlist spp_N20_E25_R1_1944

Generation wise Set of solutions i.e. dual fitness , net cut and sleep time value for the circuit spp_N20_E25_R1_1944 have been shown graphically from figure 8 to figure 10 and optimized fitness, optimized net cut and optimized sleep time has been marked accordingly. It has been clearly shown optimized Minimum net cut and sleep time in figure 9 and figure 10 respectively according to optimized fitness values which is shown in figure 8. It has been observed that the optimum result has been obtained with cross over probability **0.4**

figure 8: Dual fitness of net list spp_N20_E25_R1_1944

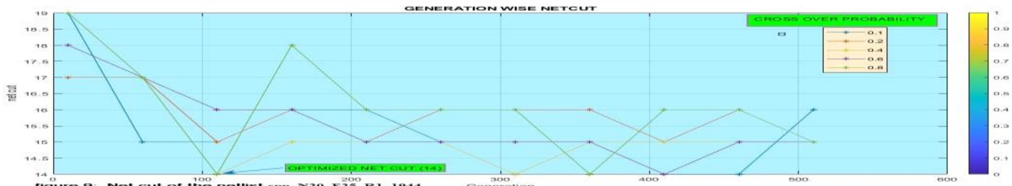

figure 9: Net cut of the netlist spp_N20_E25_R1_1944 Generation

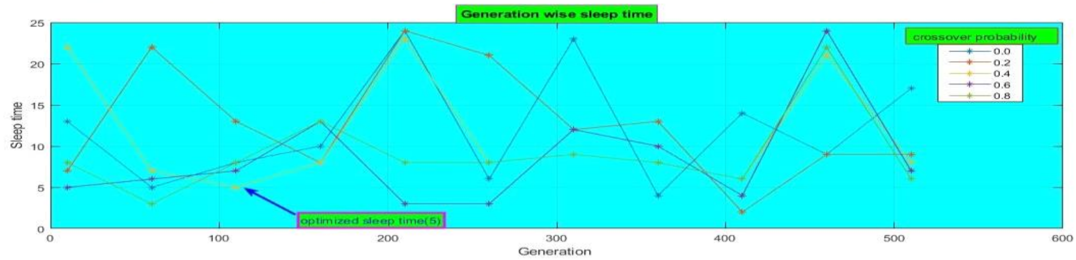

figure 10: Sleep time of net list spp_N20_E25_R1_1944

Generation wise average distance of the individual of the net list spp_N20_E25_R1_1944 during optimization has been clearly depicted in figure 11. if the average distance is small, the diversity is low and if the average distance between individuals is large, the diversity is high;. If the diversity is too high or too low, the genetic algorithm might not perform well [24].

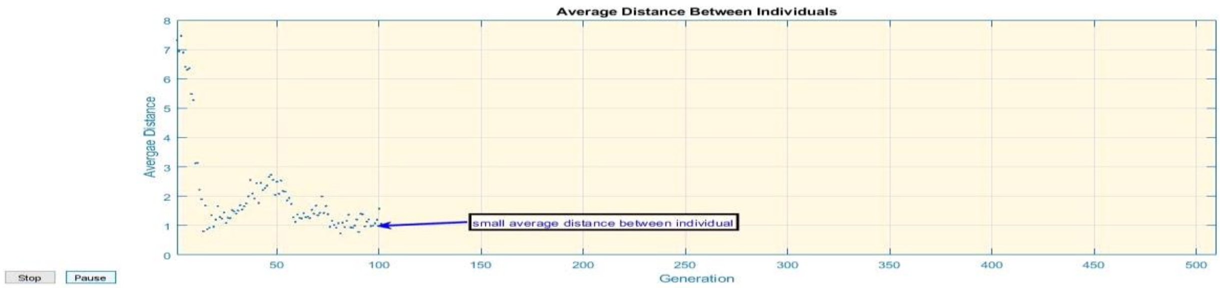

figure 11: Average distance of the individual of the netlist spp_N20_E25_R1_1944 shows low diversity of population

Generation wise result of some other benchmark circuit (net lists) are depicted clearly from figure 12.a to figure 20.c and optimized fitness, net cut and sleep time have been marked accordingly.

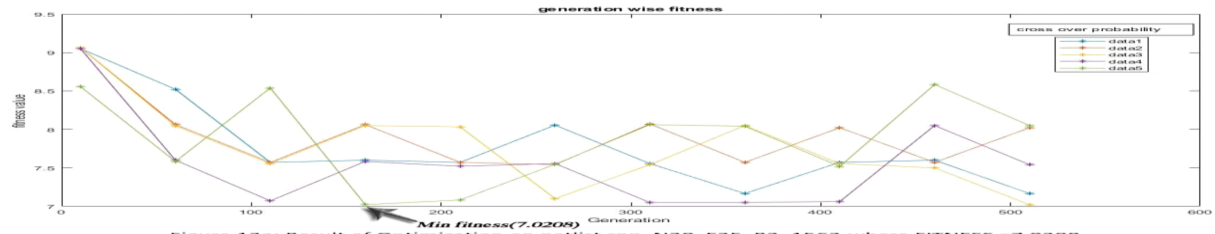

Figure 12a: Result of Optimization on netlist spp_N20_E25_R2_1562-where FITNESS =7.0208

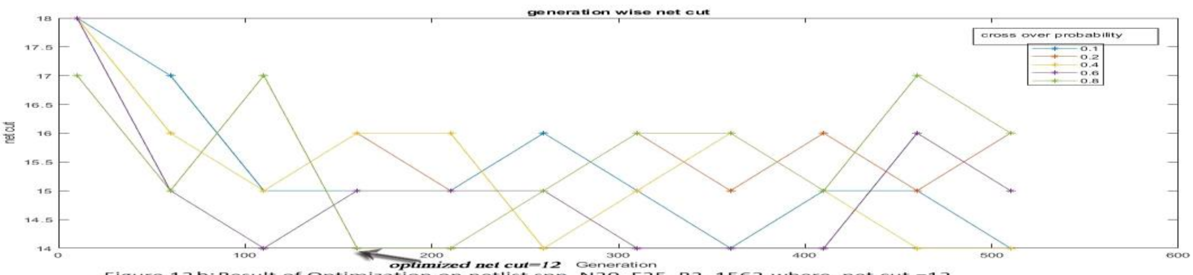

Figure 12 b: Result of Optimization on netlist spp_N20_E25_R2_1562-where net cut =12

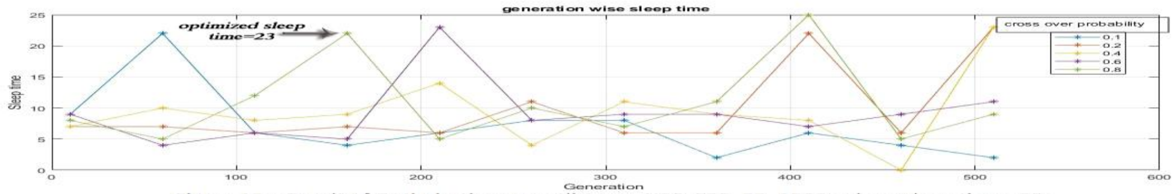

Figure 12c: Result of Optimization on netlist spp_N20_E25_R2_1562- where sleep time= 23..

In figures 12a,12b,12c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E25_R2_1562 . It has been clearly marked the optimum net cut (here 14) and optimum sleep time (here 23). The optimum result has been obtained with cross over probability **0.8**.

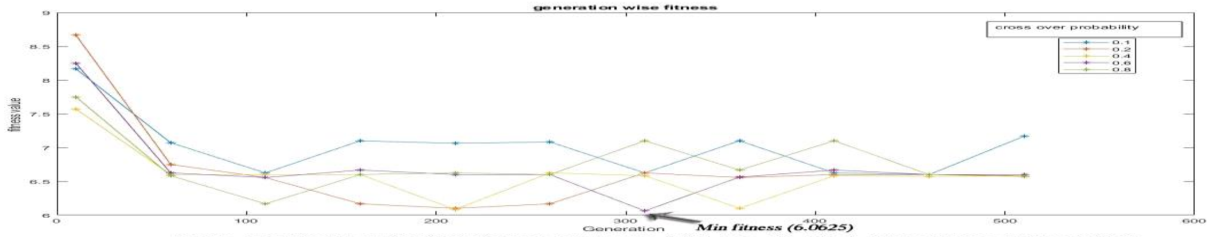

Figure 13a: Results of Optimization on netlist spp_N20_E25_R2_1917- where Fitness value =6.0625

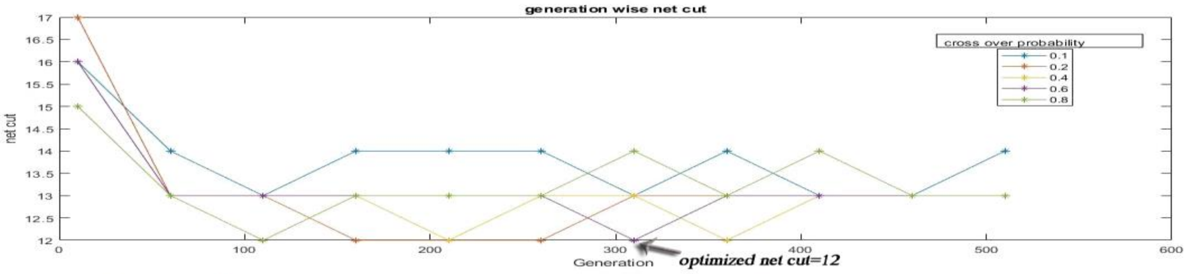

Figure 13b: Results of Optimization on netlist spp_N20_E25_R2_1917- where net cut 12,

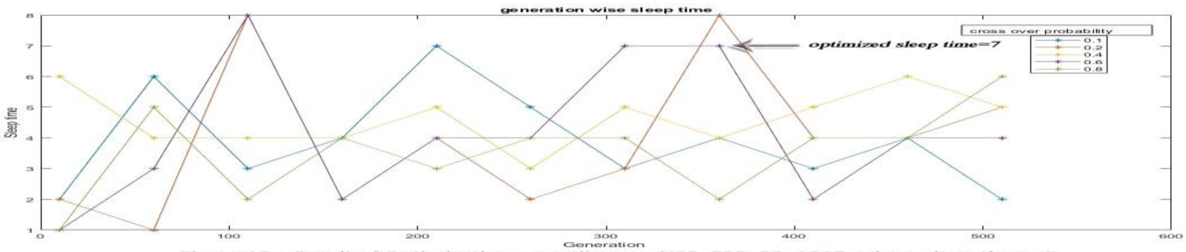

Figure 13c: Result of Optimization on netlist spp_N20_E25_R2_1917- where sleep time = 7

In figures 13a,13b,13c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E25_R2_2200. It has been clearly marked the optimum net cut (here 12) and optimum sleep time (here 7). It has been observed that the optimum result has been obtained with cross over probability **0.6**.

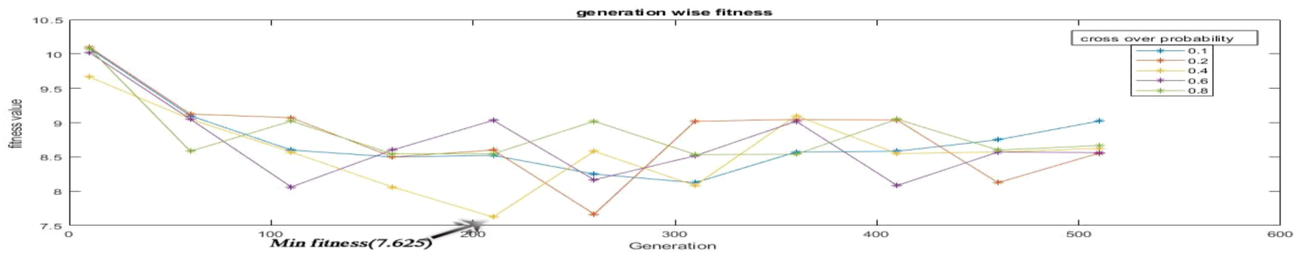

Figure 14a: Result of Optimization on netlist spp_N20_E25_R2_2200- where fitness value= 7.625

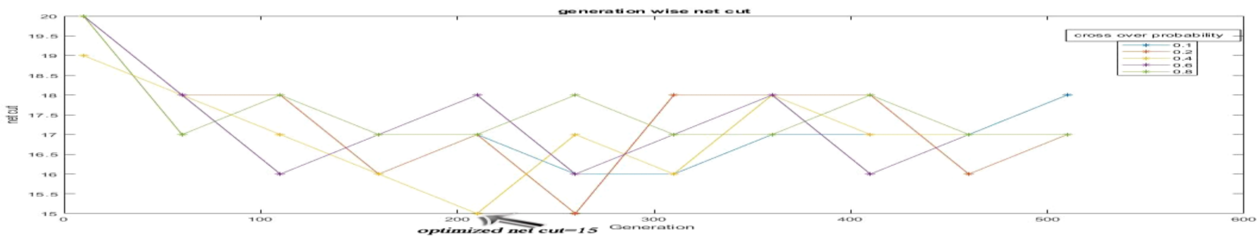

Figure 14b: Result of Optimization on netlist spp_N20_E25_R2_2200- where net cut =15

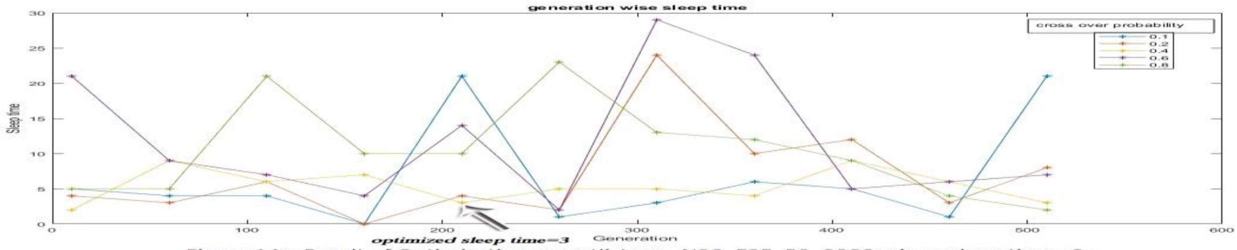

Figure 14c: Result of Optimization on netlist spp_N20_E25_R2_2200-where sleep time= 3..

In figures 14a,14b,14c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E25_R2_2200. It has been clearly marked the optimum net cut (here 15) and optimum sleep time (here 3). It has been observed that the optimum result has been obtained with cross over probability **0.4**.

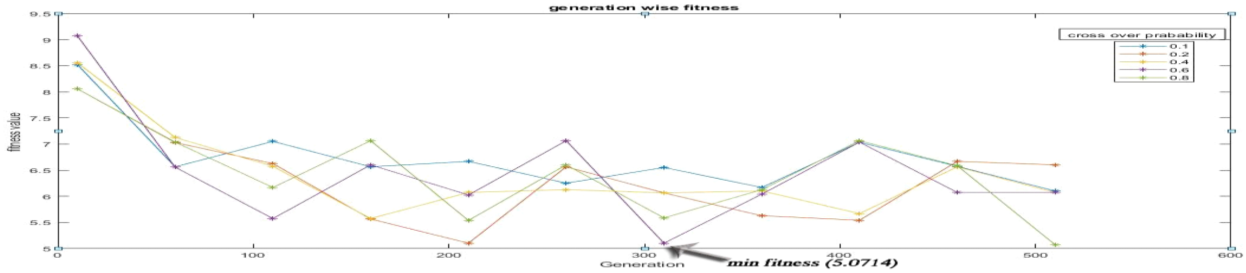

Figure 15a: Result of Optimization on netlist spp_N20_E25_R2_2391- where Fitness value= 5.0714

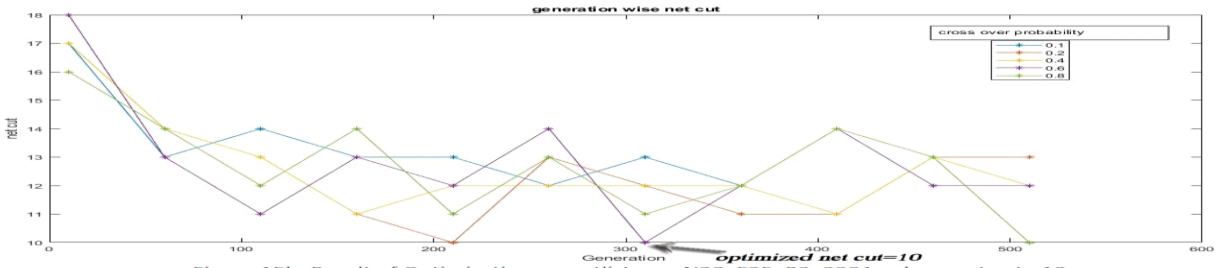

Figure 15b: Result of Optimization on netlist spp_N20_E25_R2_2391- where net cut =10

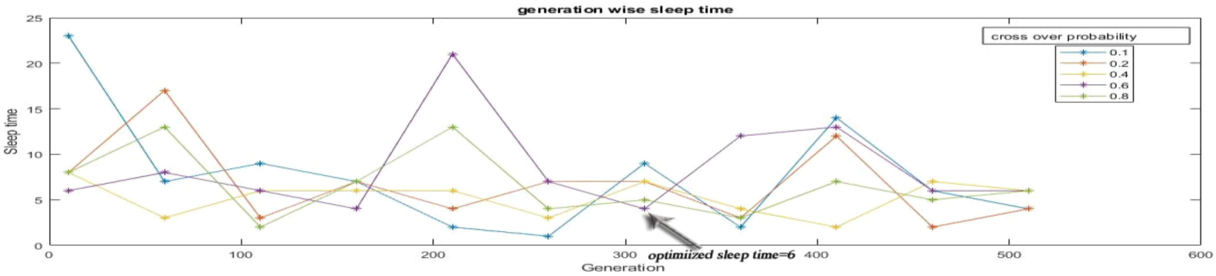

Figure 15c: Result of Optimization on netlist spp N20 E25 R2 2391- where sleep time= 6..

In figures 15a,15b,15c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E25_R2_2391. It has been clearly marked the optimum net cut (here 10) and optimum sleep time (here 6). It has been observed that the optimum result has been obtained with cross over probability **0.6**.

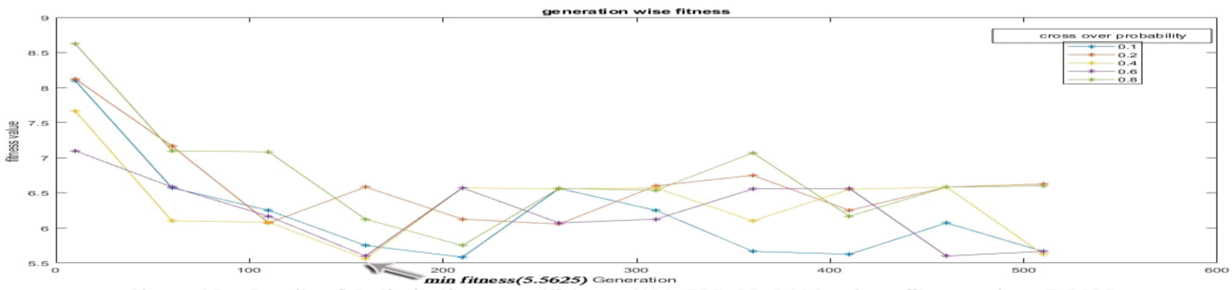

Figure 16a: Results of Optimization on netlist spp_N20_E25_R2_2481- where fitness value= 5.5625

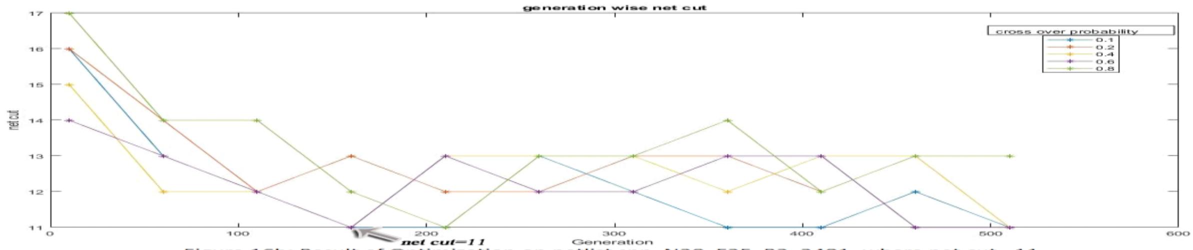

Figure 16b: Result of Optimization on netlist spp_N20_E25_R2_2481- where net cut =11

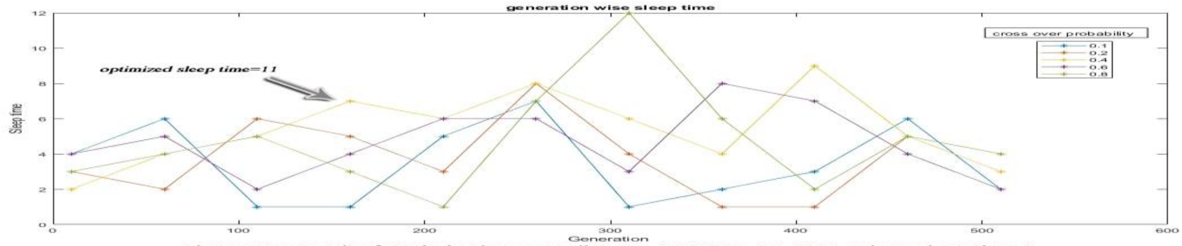

Figure 16c: Result of Optimization on netlist spp_N20_E25_R2_2481- where sleep time 7 ..

In figures 16a,16b,16c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E25_R2_2481. It has been clearly marked the optimum net cut (here 11) and optimum sleep time (here 7). It has been observed that the optimum result has been obtained with cross over probability **0.4**.

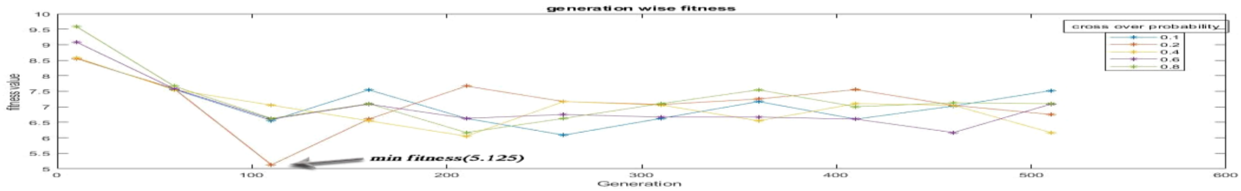

Figure 17a: Results of Optimization on netlist spp_N20_E25_R2_2844- where Fitness= 5.125

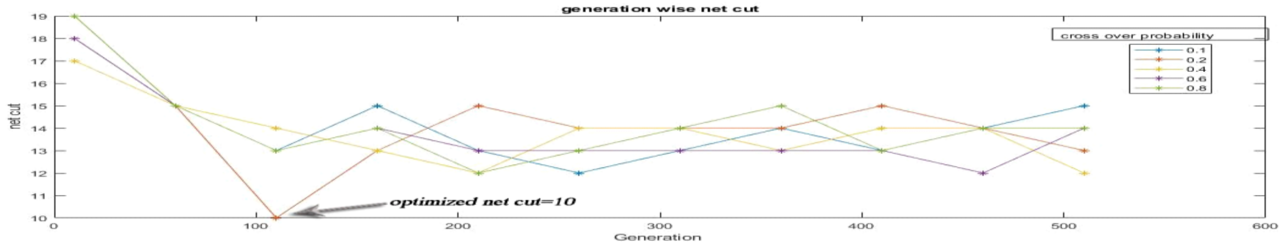

Figure 17b: Results of Optimization on netlist spp_N20_E25_R2_2844- where net cut =10

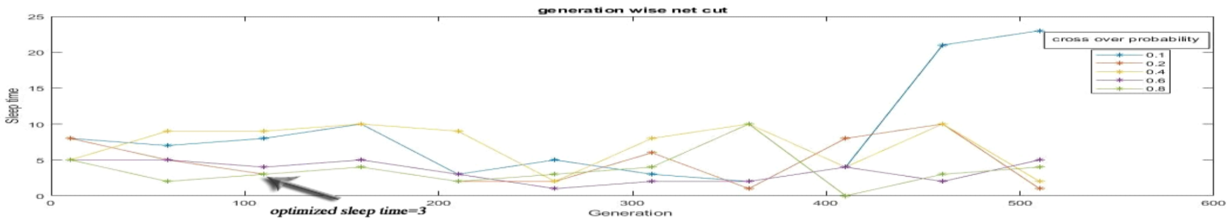

Figure 17c: Results of Optimization on netlist spp_N20_E25_R2_2844- where sleep time = **3**.

In figures 17a, 17b, 17c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E25_R2_2844. It has been clearly marked the optimum net cut (here 10) and optimum sleep time (here 3). It has been observed that the optimum result has been obtained with cross over probability **0.2**.

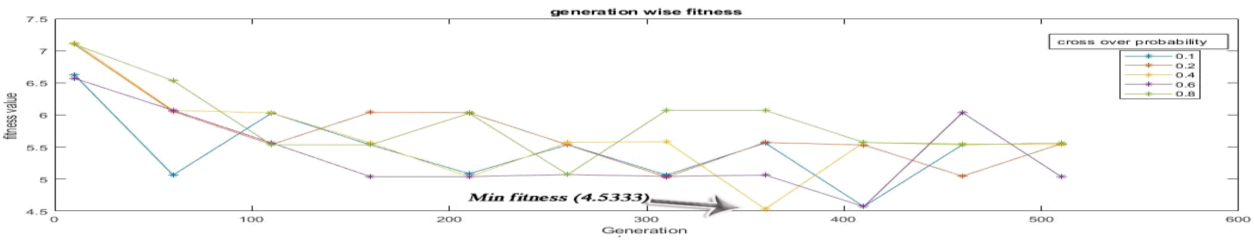

Figure 18a: Result of Optimization on netlist nspp_N20_E22_R2_1529- where fitness value= 4.5333

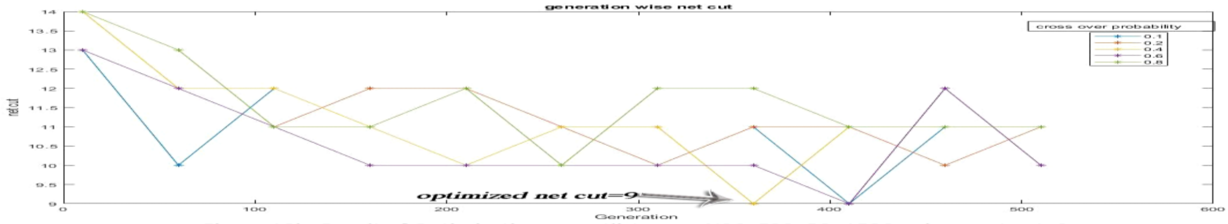

Figure 18b: Result of Optimization on netlist nspp_N20_E22_R2_1529- where net cut=9

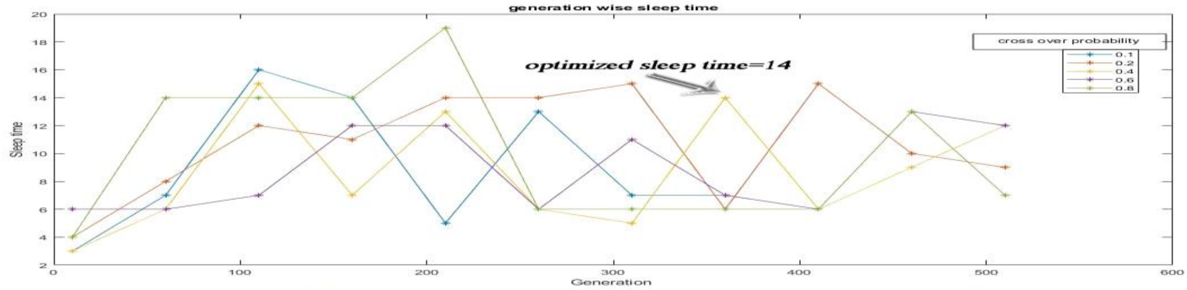

Figure 18c: Results of Optimization on netlist nspp_N20_E22_R2_1529-where sleep time =14 ..

In figures 18a,18b,18c show the result of dual fitness, net cut and sleep time of the net list nspp_N20_E22_R2_1529. It has been clearly marked the optimum net cut (here 9) and optimum sleep time (here 14). It has been observed that the optimum result has been obtained with cross over probability **0.4**.

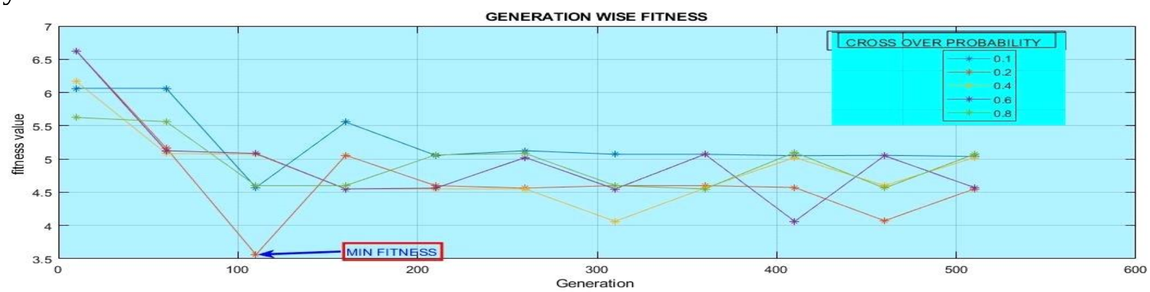

figure 19.a : Result of optimization on netlist spp_N20_E20_R2_942 where optimized fitness value = 3.563

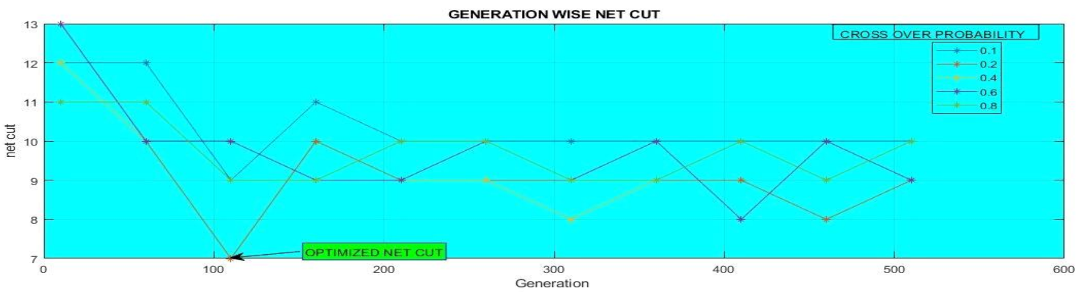

figure 19.b : Result of net cut on netlist spp_N20_E20_R2_942 where optimized net cut = 7

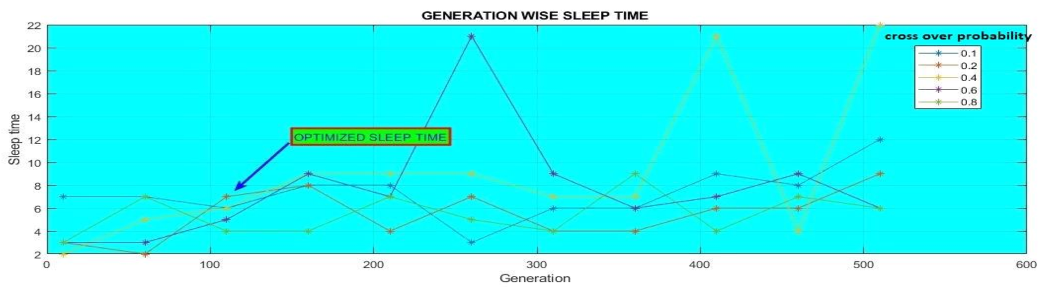

figure 19.c : Result of sleep time on netlist spp_N20_E20_R2_942 where optimized sleep time= 7

In figures 19a,19b,19c show the result of dual fitness, net cut and sleep time of the net list spp_N20_E20_R2_942. It has been clearly marked the optimum net cut (here 7) and optimum sleep time (here 7). It has been observed that the optimum result has been obtained with cross over probability **0.2**.

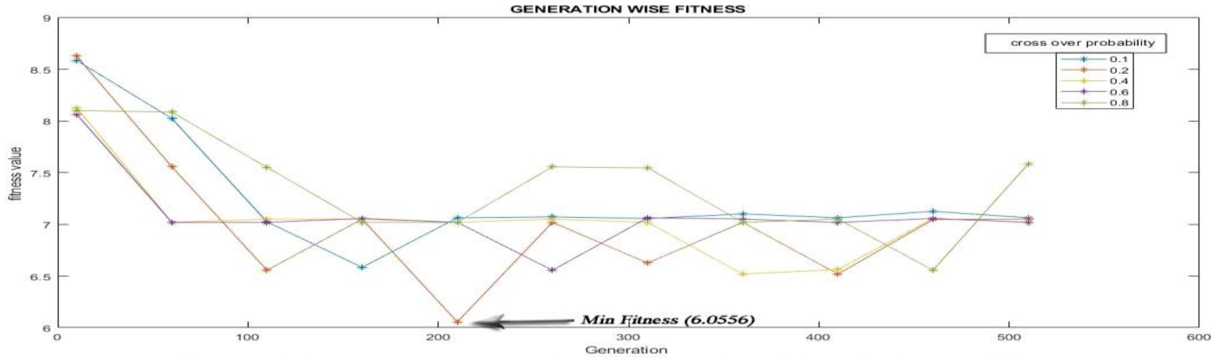

Figure 20 a: Result of Optimization on netlist spp_N20_E25_R2_1540- OPTIMIZED FITNESS 6.0556

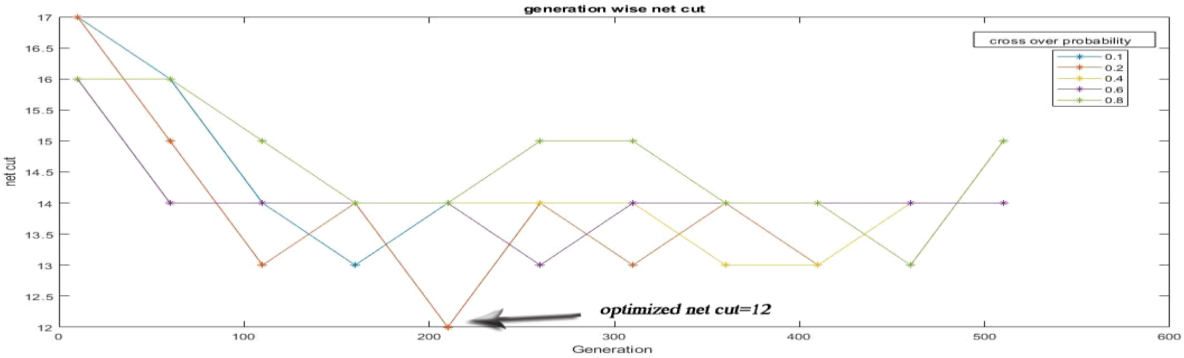

Figure 20 b: Result of Optimization on netlist spp_N20_E25_R2_1540- where net cut= 12

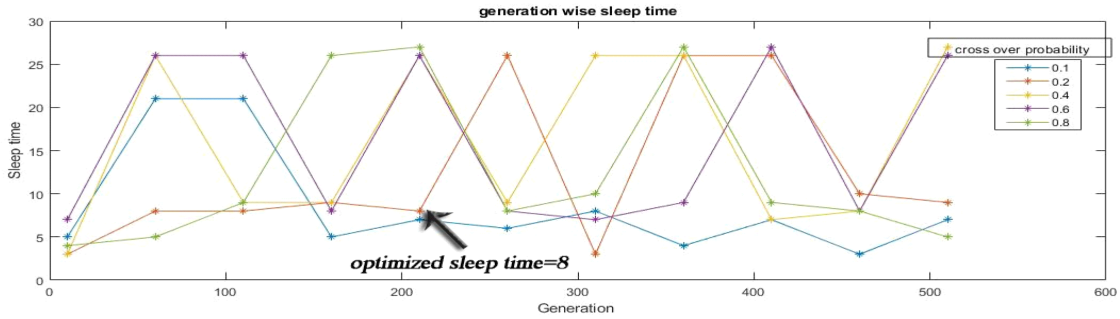

Figure 20: Result of Optimization on netlist spp_N20_E25_R2_1540- where sleep time= 8

In figures 20a,20b,20c show the result of dual fitness, net cut and sleep time of the net list. spp_N20_E25_R2_1540. It has been clearly marked the optimum net cut (here 12) and optimum sleep time (here 8). The optimum result has been obtained with cross over probability **0.2**.

All the information before and after optimization are shown in the table 1 and table 2. The table 1 shows the initial values before optimization where sleep weight: =0.5 and net weight =0.5.

Sl .No	**Circuit description**	**No.of Net**	**net cut**	**Sleep time**	**switching activity**	**sleep weight**	Net weight
1	spp_N20_E25_R2_1562	**25**	**21**	**8**	**20**	0.5	0.5
2	spp_N20_E25_R2_1917	**25**	**18**	**3**	**21**	0.5	0.5
3	spp_N20_E25_R2_1944	**25**	**21**	**2**	**17**	0.5	0.5
4	spp_N20_E25_R2_2200	**25**	24	**1**	16	0.5	0.5
5	spp_N20_E25_R2_2391	**25**	**19**	**2**	**18**	0.5	0.5
6	spp_N20_E25_R2_2481	**25**	**19**	**3**	**19**	0.5	0.5
7	spp_N20_E25_R2_2844	**25**	**20**	**1**	**18**	0.5	0.5
8	spp_N20_E22_R2_1529	**22**	**16**	**5**	**19**	0.5	0.5
9	spp_N20_E20_R2_942	**20**	**16**	**3**	**21**	0.5	0.5
10	spp_N20_E25_R2_1540	**25**	**18**	**4**	**16**	0.5	0.5

Table 1: Initial values which have been assigned before optimization.

Optimized result on various net lists are shown in Table 2. It has been observed that average improvement of this method are 41.98 % net cut, 161.75 % sleep time compared to initial partition of the circuit.

Results obtained after optimization:

							Improvement	
Sl.No	**Circuit description**	**optimized fitness**	**net cut**	**Sleep time**	**switching activity**	**cross over probability**	**net cut**	**Sleep time**
1	**Spp_N20_E25_R2_1562**	**7.021**	**12**	**23**	**11**	**0.8**	**42.86**	**187.5**
2	**Spp_N20_E25_R2_1917**	**6.063**	**12**	**7**	**17**	**0.6**	**33.33**	**133.33**
3	**Spp_N20_E25_R2_1944**	**7.023**	**14**	**5**	**14**	**0.4**	**33.33**	**150**
4	**Spp_N20_E25_R2_2200**	**7.625**	**15**	**3**	**12**	**0.4**	**37.50**	**200**
5	**Spp_N20_E25_R2_2391**	**5.071**	**10**	**6**	**15**	**0.6**	**47.37**	**200**
6	**Spp_N20_E25_R2_2481**	**5.563**	**11**	**7**	**15**	**0.4**	**42.11**	**133.33**
7	**Spp_N20_E25_R2_2844**	**5.125**	**10**	**3**	**14**	**0.2**	**50.00**	**200**
8	**Sspp_N20_E22_R2_1529**	**4.533**	**9**	**14**	**16**	**0.4**	**43.75**	**180**
9	**Spp_N20_E20_R2_942**	**3.563**	**7**	**7**	**14**	**0.2**	**56.25**	**133.33**
10	**Spp_N20_E25_R2_1540**	**6.056**	**12**	**8**	**12**	**0.2**	**33.33**	**100**
		Average improvement					**41.9831**	**161.75**

Table 2: Result obtained after optimization along with percentage of improvement.

6. CONCLUSION

Genetic algorithm based bi-objective method has been proposed to optimize net cut (minimization) and sleep time (maximization. This evolutionary algorithm for VLSI partitioning produces a significant improvement in result quality. The method is based on the principle of Darwin's theory. The main philosophy is followed to Holland [21]. Here the objectives are formulated separately and then combined into a composite objective function(dual). After optimization the optimized net cut and optimize sleep time has also been separated from the dual fitness value. This method of optimization has been applied on some net list with ISPD'98 benchmark suite. We have made multi-way partitioning of circuit and every chromosome is coded accordingly . The partitioning has been done in such a way that balanced criteria can be satisfied. After observing the average improvement of the percentage in the result , the method might be proven to be efficient . The probability mutation has been kept low and variation of cross over probability has been done to achieve good quality solution .This method improves the power consumption of the system which was our main goal of this study. As the GA is inherently discrete, it is able to encode the design variable into bits easily. One of the most important factors that determine the performance of this method's performance is the diversity of the population. Generation wise the average distance between individuals has been observed in every case and when it is found to be low, we can conclude that the optimal solution are not far from the current individual, that is an efficient measure of the diversity of a population. Finally ,we can conclude that this method will be able to achieve a faster convergence without compromising the quality of the optimum solutions.

REFERENCES

[1] H.B. Bakoglu. "Circuits, Interconnections, and Packaging for VLSI". Addison-Wesley Publishing Co., 1990.

[2] S. Devadas and S. Malik. "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits". In Design Automation Conference. IEEE, June 1995.

[3] B.W. Kernighan, S. Lin, "An efficient heuristic procedure for partitioning graphs", *The Bell System Technical Journal*, 1970, 49: 291-372. **DOI:** 10.1002/j.1538-7305.1970.tb01770.x

[4] C.M. Fiducca, R.M. Mattheyses, S. Areibi, "Mimetic algorithms for VLSI Physical Design: Implementation Issues". *Genetic and Evolutionary computation*, Proceedings of the 19th Design automation Conference, *IEEE Press*, 1992, 175-181.

[5] D.E. Goldberg, "Genetic algorithms in search, optimization and machine learning". *Pearson Education*, 2004.

[6] S. Areibi, "Mimetic algorithms for VLSI Physical Design: Implementation Issues". *Genetic and Evolutionary computation Conference, San Fransisco, California*, 2001, 140-145.

[7] C. Ababei, S. Navaratnasothie, K. Bazargan, G. Karypis, "Multi-objective circuit partitioning for cut-size and path-base delay minimization". *IEEE International Conference on Computer aided Design*, 2002.

[8] M. Palesi, T. Givargis, "Multi-objective design space exploration using genetic algorithms", *Proceedings of the 10th international symposium on Hardware/software codesign, ACM Press, Estes Park, Colorado,* 2002, 67-72.

[9] Z.Q. Chen, Y.F. Yin, "A new crossover operator for real-coded genetic algorithm with selecting breeding based on difference between individuals", *Natural Computation (ICNC), 2012 Eighth International Conference*, 2012, 644-648. **DOI:** 10.1109/ICNC.2012.6234556

[10] S.Y. Yuen, C.K. Chow, "A genetic algorithm that adaptively mutates and never revisits", *Evolutionary Computation*, 2009, 13: 454-472. **DOI:** 10.1109/TEVC.2008.2003008

[11] W. Jigang, T. Srikanthan, "Efficient algorithms for hardware/software partitioning to minimize hardware area", *Circuits and System*, 2006, IEEE Asia Pacific Conference, 1875-1878. **DOI:** 10.1109/APCCAS.2006.342205

[12] S.S. Gill, R. Chandel, A. Chandel, "Genetic algorithm-based approach to circuit partitioning", *International Journal of Computer and Electrical Engineering*, 2010, 2: 1793-1863. **DOI**: 10.7763/IJCEE.2010.V2.136

[13] P. Arato, S. Juhasz, Z.A. Mann, D. Papp, "Hardware-Software partitioning in embedded system design", International Conference on Complex, Intelligent and Software Intensive Systems, 2003, 197-202.

[14] A. Prakkash, R.K. Lal, "PSO: An approach to multi-objective VLSI Partitioning". *Proceedings of the 2nd International Conference on Innovations in Information, Embedded and Communication systems. IEEE*, 2015.

[15] N. Sherwani, "Algorithms for VLSI physical design automation". 3rd edition, Springer (India) Private limited, New Delhi, 2005.

[16] A.H. Farrahi, M. Sarrafzadeh, "System partitioning to maximize sleep time", Proceedings of the 1995 IEEE/ACM International Conference on computer Aided Design, San Jose, California, USA, 1995, 452-455. **DOI:** 10.1109/ICCAD.1995.480155

[17] P. Ghafari, E. Mirhard, M. Anis, S. Areibi, M. Elmary, "A low power partitioning methodology by maximizing sleep time and minimizing cut nets". *IWSOC, Bauf, Alberta, Canada*, 2005, 368-371. **DOI:** 10.1109/IWSOC.2005.15

[18] J.J. Cong, K.S. Leung, "Optimal wire sizing under Elmore delay model", *IEEE Transactions on Computer Aided Design of Integrated Circuits and System*, 1995, 14: 3. DOI: 10.1109/DAC.1996.545625.

[19] P. Zarkesh, J.A. Davis, J.D. Meindail, "Prediction of Net-Length Distribution for global interconnects", In a Heterogeneous system-on-a-chip. *IEEE Transaction on VlSA Systems*, 2000, 8: 6. **DOI:** 10.1109/92.902259

[20] K.P. Subbaraj, P. Sivasundari, S. Kumar, "An effective mimetic algorithm for VLSI partitioning problem", *International conference on ICTES Chennai India*, 2007, 667-670.

[21] J.H. Holland, Adaption in natural and artificial system: "An introductory analysis with applications to biology, control and artificial intelligence", *MIT press*, 1992. ISBN: 0262082136

[22] Ouvrard Brunet, Xavier. (2020). "Hypergraphs: an introduction and review".

[23]"Tournament selection", https://www.geeksforgeeks.org/tournament-selection-ga/

[24] "Population diversity", https://www.mathworks.com/help/gads/population-diversity.html#f14724)

[25] "Crossover operator", http://mnemstudio.org/genetic-algorithms-recombination.htm

[26] P. Mazumder, E.M. Rudnik, "Genetic Algorithms for VLSI Design, Layout and Test Automation", Pearson Education, 2003.

[27] S.S. Gill, R. Chandel, A. Chandel ," Genetic Algorithm Based Approach To Circuit Partitioning", International Journal of Computer and Electrical Engineering, Vol. 2, No. 2, April, 2010 1793-8163.

[28] Mutation operator : https://www.geeksforgeeks.org/genetic-algorithms/

[29] A.P. Sen, A. Prakash, P. K. Singh, R. K. Lal ," Multiobjective VLSI circuit partitioning using ACO for optimal Solution", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 4, Issue 5, May 2015.

[30] R. P. HYPERLINK "javascript:;" HYPERLINK "javascript:;"Guru, V. Vaithianathan "An efficient VLSI circuit partitioning algorithm based on satin bowerbird optimization (SBO)". *Journal of Computational Electronics* **19,** 1232–1248 (2020).

[31] S. Roy, " Genetic Algorithm Based Multiobjective Optimization for Very Large Scale Integration (VLSI) Circuit Partitioning", Vol.-7, Issue-1, Jan 2019.
DOI: **https://doi.org/10.26438/ijcse/v7i1.409417**.

[32] G.K. Chellamani , P. V. Chandramani, "An Optimized Methodical Energy Management System for Residential Consumers Considering Price-Driven Demand Response Using Satin Bowerbird Optimization", *Journal of Electrical Engineering HYPERLINK "https://link.springer.com/journal/42835"& HYPERLINK "https://link.springer.com/journal/42835" Technology* volume 15, pages 955–967(2020).