ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2016.0030

SIMULATED ANNEALING ALGORITHM FOR MODERN VLSI FLOORPLANNING PROBLEM

J. Jenifer[1], S. Anand[2] and Y. Levingstan[3]
[1,2]*Department of Electronics and Communication Engineering, V V College of Engineering, India*
E-mail: [1]jeniferjeba28@gmail.com, [2]anandmay22@gmail.com
[3]*Department of Electrical and Electronics Engineering, V V College of Engineering, India*
E-mail: [3]levingstan@vvcoe.org

Abstract

In floorplanning, our aim is to determine the relative locations of the blocks in the chip and the objective is to minimize the floorplan area, wirelength. Generally, there are so many strategies in VLSI floorplanning like area optimization, wirelength optimization, power optimization, temperature optimization and etc. This paper concentrates on area optimization. The goal of the physical design process is to design the VLSI chip with minimum area. The primary idea is to minimize the floorplan area by reshaping the blocks which are present inside the floorplan in order to attain the minimum area with less computational time. Proposed problem is redefined with an efficient meta-heuristic as Simulated Annealing algorithm which will provide optimal solution with less computation time. The proposed algorithm has been tested by using set of benchmarks of Microelectronics Centre of North Carolina (MCNC).The performance of the proposed algorithm is compared with other stochastic algorithms reported in the literature and is found to be efficient in producing floorplan with minimal area. The performance of the proposed algorithm seems to be better than the existing algorithms.

Keywords:

VLSI, Floorplanning, Optimization, Deadspace, Meta-heuristic, Simulated Annealing

1. INTRODUCTION

In VLSI, more than thousands of transistors are integrated into a single chip in order to fabricate an IC. It has two design phases. There are logical design and physical design. Here, physical design is the process of determining the physical location of active devices and interconnecting them inside the boundary of the VLSI chip. The purpose of VLSI physical design is to fix an abstract circuit description such as netlist, into silicon, creating a detailed geometric layout of the IC. In physical design, floorplanning determines the topology of the layout i.e. the relative positions of the blocks on the chip based on the interconnection requirements of the circuit and estimates for area.

Inputs for a floorplanning problem are,

- A set of n rectangular blocks $i = 1,2,...,n$ and its area A_i.
- Interconnection between the blocks i and j is C_{ij}.

Outputs are,

- No overlap between the blocks.
- Optimized coordinates of each block.

Primary objectives of floorplanning problem are,

- The optimized relative location.
- Minimizing the deadspace.

Generally, floorplanning is a two-step process [1]. There are topology generation and sizing. In the topology generation, the shape is not considered but based on the netlist such as interconnection information, the blocks are arranged in order to minimize the area. In sizing the aspect ratio *i.e.* the height width ratio of the blocks are adjusted in order to minimize the area. Earlier days, the topology generation is used. Here, based on the shape curve representation the blocks are arranged in order to minimize the area [2], [3]. Nowadays, another approach to solve the floorplan design problem is by simultaneously considering the interconnection information as well as the area and shape information. This approach starts with an initial floorplan and iteratively improves solutions by taking both interconnect and shape information into account until the convergence is reached or the runtime exceeds. Here, the optimization like area, wirelength, power and temperature optimization are achieved using any one of the meta-heuristic techniques. Some of the meta-heuristic techniques are Simulated Annealing algorithm, Genetic algorithm, PSO algorithm, Defer algorithm and etc. PSO algorithm is simple, easy to implement and provides high quality solutions but it requires more parameters to implement. Single objective optimization i.e. area optimization is achieved well using this algorithm [4] but it is not suitable for multi objective optimization. Here, there is no guarantee for global convergence.

In APTCG algorithm [5], the target area can be estimated before packing and also used to check whether the perturbation is beneficial to solve this floorplanning problem or not. This algorithm also used to optimize only the area. So, it's also not suitable for multi objective optimization. The half perimeter of a floorplan is minimized using lagrangian relaxation technique [6] and here there is no importance given to achieve the area optimization. Then, the multi objective optimizations such as both areas, wirelength optimization are achieved using variable order ant system [7] with a floorplan model namely corner list as well as Defer algorithm [8]. This defer algorithm provides the non-slicing floorplan by compacting slicing floorplan. Here, also there is no guarantee for global convergence. The multi objective optimization such as area, wirelength and temperature optimization are achieved using the bus driven algorithm [9]. Here the importance given to the temperature optimization. Then the shaping techniques are introduced in order to solve the VLSI floorplanning problem. The area of the floorplan is minimized by shaping the blocks using convex optimization [10] as well as lagrangian relaxation technique [11]. But in their problem formulation, the aspect ratio constraint on each block was ignored. So, the block shaping is not that much accurate. So, it is necessary to design an efficient algorithm that is specially formulated for fixed outline floorplanning and considers the aspect ratio constraints. Here, in this paper's problem formulation the aspect ratio is considered during the block reshaping. Generally, there are two reshaping techniques,

1) To minimize the floorplan height by increasing the width and decreasing the height of each block.
2) To minimize the floorplan width by increasing the height and decreasing the width of each block.

In this paper, the first reshaping technique as to minimize the floorplan height by increasing the width and decreasing the height of each block is used to reshape the block in order to minimize the floorplan area.

2. PROBLEM FORMULATION

In this design, Consider there are n blocks are present inside the floorplan. Additionally we include two vertices 0 and $n + 1$ in all the four sides of the floorplan that used to indicate the left most and right most boundary as well as bottom and top most boundary of the floorplan [12]. Both the 0 and $n + 1$ vertices be the dummy vertices that area, width, height as well as its coordinates also equal to zero. Then the floorplan height is denoted as y_{n+1}.and the floorplan width is denoted as x_{n+1}. Each block B_i inside the floorplan has width w_i and height h_i. The width w_i varies from $W_i^{\min}$ to $W_i^{\max}$. The height h_i varies from $H_i^{\min}$ to $H_i^{\max}$. Let x_i and y_i be the coordinates of the block B_i.

Objective

Minimize y_{n+1}

Subject to constraints

$$x_{n+1} \le W$$
$$x_j \ge x_i + w_i, \text{ for all } (i, j) \in G_h$$
$$y_j \ge y_i + h_i, \text{ for all } (i, j) \in G_v$$
$$W_i^{\min} \le w_i \le W_i^{\max}, \quad 1 \le i \le n$$
$$H_i^{\min} \le h_i \le H_i^{\max}, \quad 1 \le i \le n$$
$$w_i.h_i = A_i, \; 1 \le i \le n$$
$$w_0 = h_0 = 0, x_0 = y_0 = 0$$

In this problem formulation, the objective is to minimize the floorplan height and the constraint is to attain the minimization with a given fixed upper bound width of the floorplan. This minimization has to be done only reshaping the blocks in the floorplan. The cost function is formulated based on the weighted sum method and it is given as below,

$$C = x_1A + x_2W + x_3AR \tag{1}$$
$$x_3 = (x_1 + x_2) - 1 \tag{2}$$

where,

A - area; W - wirelength; AR - Aspect Ratio
x_1, x_2, x_3 - weight factor for area, wirelength and aspect ratio.

Aspect ratio means the height width ratio. Here, the aspect ratio is considered in order to reshape the block. Because in reshaping both the height and width is considered. x_1, x_2, x_3 values are decided based on the trial and error method and the importance of these objectives. So, here we set x_1 value as 0.5 and x_2 value as 0.2 and we get x_3 value as 0.3.

2.1 COST MEASURES

2.1.1 Area Calculation:

The blocks which are present inside the floorplan assume be the rectangular blocks. So the area of rectangular block is the product of width and height. For block *i*, the width is denoted as w_i, the height is denoted as h_i and the area is denoted as area A_i. So, A_i is given by,

$$A_i = w_i.h_i \tag{3}$$

2.1.2 Dead Space Calculation:

Dead space means the unused area in the layout. It is also called as white space. The dead space average [14] is denoted as Γ is calculated using the following formula,

$$\Gamma = \left\{ \frac{[\text{floorplan area - sum of modules area}]}{\text{floorplan area}} \right\} * 100 \tag{4}$$

Floorplan area is the product of width and height of the floorplan and the modules which are present inside the floorplan be a rectangular module. So, the area of each module is the product of its width and height.

2.1.3 HPWL Calculation:

In the physical design process, using routing only, the connection should be made between the blocks which are present inside the floorplan. It made the connections based on the netlist. Here, the blocks are connected using the wires. Wirelength be the length of the wire between two blocks.

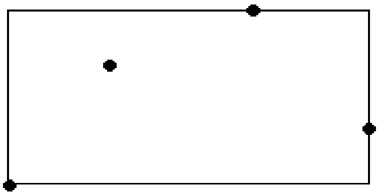


Fig.1. Manhattan bounding box

An example Manhattan bounding box of a four terminal net is shown in Fig.1. The MHPWL is the half of the perimeter length of the Manhattan bounding box [13]. The formula for calculating MHPWL is,

$$MHPWL(e) = \underset{v_i,v_j \in e^{|x_i - x_j|}}{MAX} + \underset{v_i,v_j \in e^{|y_i - y_j|}}{MAX} \tag{5}$$

3. PROPOSED WORK

Algorithm is a set rule or procedure to solve the problem. It provides the exact solution. An algorithm is an effective method that can be expressed within a finite amount of space and time. If the algorithm fails to provide the output means at that time, enter into the heuristic. Heuristic means rule of thumb and here the problem can be solved and provide the approximate solution. There are so many heuristic algorithms like Simulated Annealing algorithm, Genetic algorithm, PSO algorithm, Tabu Search algorithm and etc. Meta-heuristic is the high level procedure for selecting the heuristic in order to provide the good solution for the optimization problem. To start with Simulated Annealing (SA) algorithm studied for floorplanning.

Annealing is the process of heating the metal to high temperature and reduced to low temperature in order to change the physical and chemical properties of the metals. During the annealing process, atoms migrate in the crystal lattice and the number of dislocations decreases and changes done in its ductility and hardness. This annealing process simulated in a computer is called Simulated Annealing. SA is the physical process of heating a material and then slowly lowering the temperature according to a specific schedule. It is an iterative improvement algorithm. The other names for Simulated Annealing algorithm are Monte Carolo Annealing, Statistical Cooling, and Probabilistic Hill Climbing, Stochastic Relaxation, and Probabilistic Exchange algorithm.

It is the iterative improvement algorithm. Here, initially the high temperature set to the initial solution. Then, for every iteration the temperature should be reduced. For every iteration, the perturbation has to be done with the current solution. After that the cost function like area, wirelength, and dead space should be calculated. If the current cost function is better than neighbourhood solution means the current solution is taken as a best solution. If the solution is poor means at that time the best solution cannot be changed but there will be confusion as whether the poor solution enters into the next iteration or not that is decided with the help of the probability value.

$$P = e^{\Delta c/T} \tag{6}$$

where,

Δc = difference of the cost of the neighbouring state and current state

T = Current Temperature

For each iteration of annealing the temperature is reduced by a fixed ratio. The probability randomly set by the floor planner. If the probability value is equal to the generated random probability value means at that time the poor solution enter into the next iteration otherwise it is rejected. The flowchart for Simulated Annealing algorithm can be shown in below Fig.2.

The operations involved in each iteration are reshaping, perturbation and packing.

3.1 RESHAPING

There are two reshaping techniques.

1) To minimize the floorplan height by increasing the width and decreasing the height of each block.
2) To minimize the floorplan width by increasing the height and decreasing the width of each block.

In this works, the problem objective is to minimize the layout height. So, first the reshaping technique is applied. Here, the blocks which are present inside the layout's height should be decreased and width should be increased. After reshaping, the slack is applied to the blocks in order to determine the amount of changes on the blocks. There are two types of slack [12].

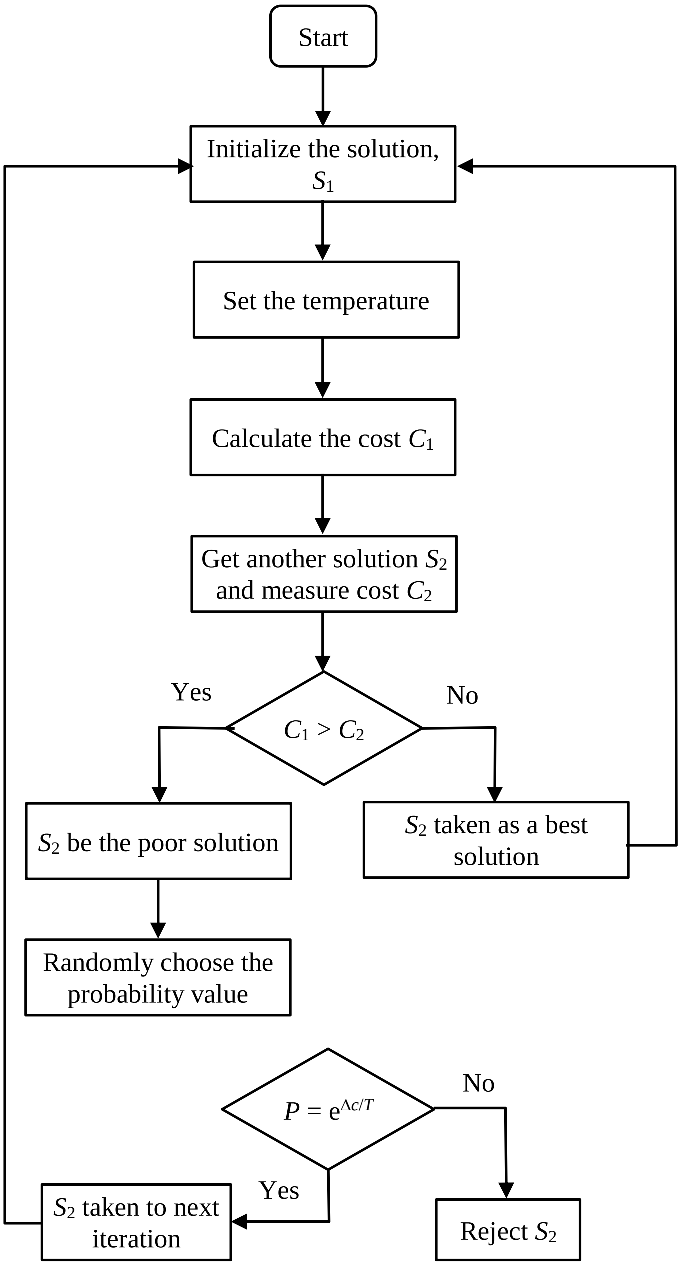
- Horizontal slack
- Vertical slack

If we are minimizing the layout height means at that time horizontal slack is applied to determine the amount of increase on width for each reshaped block. If we are minimizing the layout width means at that time the vertical slack is applied to calculate the amount of increase on height for each reshaped block. In this work, the first reshaping scheme is used.

3.1.1 Parameter Estimation:

The amount of increase on width is denoted as δ_i^h [12]. The formula for δ_i^h is,

$$\delta_i^h = \frac{\left(W_i^{\max} - w_i\right) S_i^h}{\underset{P \in P_i^h}{MAX}\left(\sum_{k \in P}\left(W_k^{\max} - w_k\right)\right)} \tag{7}$$



Start
Initialize the solution, S_1
Set the temperature
Calculate the cost C_1
Get another solution S_2 and measure cost C_2
Yes
No
$C_1 > C_2$
S_2 be the poor solution
S_2 taken as a best solution
Randomly choose the probability value
$P = e^{\Delta c/T}$
No
Yes
S_2 taken to next iteration
Reject S_2

Fig.2. Flowchart for SA algorithm

Here the term S_i^h is the horizontal slack. Horizontal slack is the distance between the block and the left most or right most boundary of the layout. The first term ($W_i^{\max}$ - w_i) can be obtained in constant time. Then S_i^h (i.e.), horizontal slack obtained in linear time. But the denominator term is obtained in the exponential time. So, the dynamic programming approach can be used here.

3.1.2 Dynamic Programming Approach:

Dynamic programming approach is a method for solving a complex problem by breaking it into simpler sub problem. Then each sub problem is solved and provides the solution. The final step is to combine all the solution in order to provide the final solution. In dynamic programming approach each sub problem is solved only once. So, the number of computation is minimized.

The characteristics of the dynamic programming approach are,

- The problem can be divided into stages.
- Each stage has a number of states.
- The decision at a stage updates the state of the stage into state for the next stage.
- Given the current state, the optimal decision for the remaining stages is independent of decisions made in previous state.
- There is a recursive relationship between the value of decision at a stage and the value of the optimum decisions at previous stages.

In order to calculate the amount of increase on width δ_i^h, the dynamic programming approach [12] has the following steps,

1) Apply topological sort algorithm on G_h.
2) Scan the sorted vertices from the source to the sink and calculate,

$$\underset{P\in P_i^{out}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)=\underset{P\in P_i^{in}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)+\left(W_i^{\max}-w_i\right) \tag{8}$$

3) Scan the sorted vertices from the sink to the source and calculate

$$\underset{P\in P_i^{out}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)=\underset{P\in P_i^{out}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)+\left(W_i^{\max}-w_i\right) \tag{9}$$

4) Finally calculate,

$$\underset{P\in P_i^{h}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)=\underset{P\in P_i^{in}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)+\underset{P\in P_i^{out}}{MAX}\left(\sum\nolimits_{k\in P}\left(W_k^{\max}-w_k\right)\right)-\left(W_i^{\max}-w_i\right) \tag{10}$$

where,

P_i^h denotes the set of paths in G_h going through block *i*.

P_i^{in} denotes the set of paths that start at the source and end at vertex v_i in G_h.

P_i^{out} denotes the set of paths that start at vertex v_i and end at the sink in G_h.

3.2 PERTURBATION

Perturbation is the process of rearranging the blocks which are present inside the layout. The perturbation has done in order to minimize the floorplan area. It performs three operations. There are move, rotation and swap.

The move operation is shown Fig.3, the blocks which are present inside the layout move from one location to another location. The rotate operation is shown in Fig.5; the block is rotate to minimize the floorplan area after that it placed in a proper location. The swap operation is shown in Fig.4; two blocks are interchanged together in order to minimize the floorplan area.

3.3 NON OVERLAPING CONSTRAINTS

The non-overlapping constraint is used, to describe the set of all positions and orientations that can be assigned to the first object so that intersection with the second one is empty. This is done using a dedicated branch & prune approach. Consider two blocks B_i and B_j. In the horizontal constraint graph, these two blocks are placed without any overlap means at that time the blocks satisfy the constraint as $x_j \ge x_i + w_{i.}$. In the vertical constraint graph i.e., in the *y* direction, these two blocks are placed without overlap means at that time the blocks satisfy the constraint as $y_j \ge y_i + h_i$. Here, (x_i , y_i) be the bottom left corner coordinate of block B_i and (x_j, y_j) be the bottom left corner coordinate of block B_j. w_i and h_i be the width and height of the block B_i.

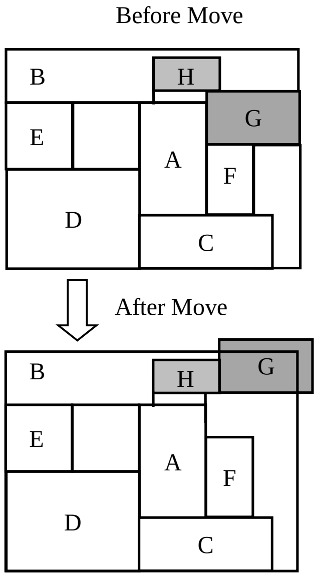


Fig.3. Move operation

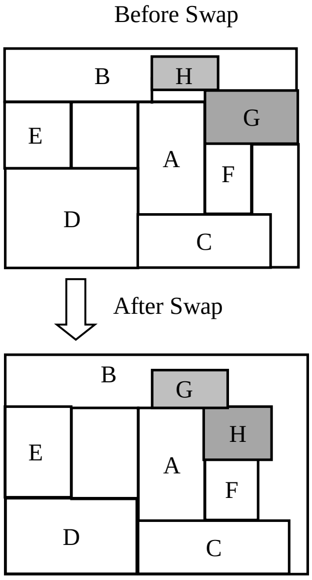


Fig.4. Swap operation

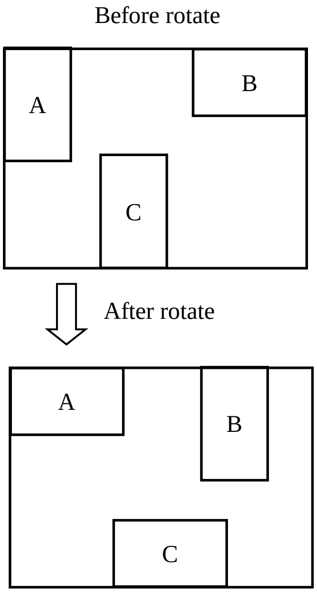


Fig.5. Rotate operation

4. RESULTS AND DISCUSSION

The proposed Simulated Annealing algorithm is implemented in C++ programming language. All the simulations are made on Intel Core i5, 2GB RAM.

4.1 BENCHMARK CIRCUITS

The proposed Simulated Annealing are tested with one of the benchmark circuits named as MCNC (Microelectronic Centre of North Carolina) and to find solutions to modern floorplanning problems with the fixed- outline constraint. The benchmark suite was released for design workshops in the early 1990's and is often referenced in the literature as the MCNC benchmarks. They were originally maintained by North Carolina's Microelectronics, Computing, and Networking centre, but are now located at the CAD Benchmarking Laboratory (CBL) at North Carolina State University. These MCNC benchmarks are standard problems in floorplanning, and the characteristics of the circuits are shown in Table.1. The benchmark circuits vary in size from 9 to 49 modules.

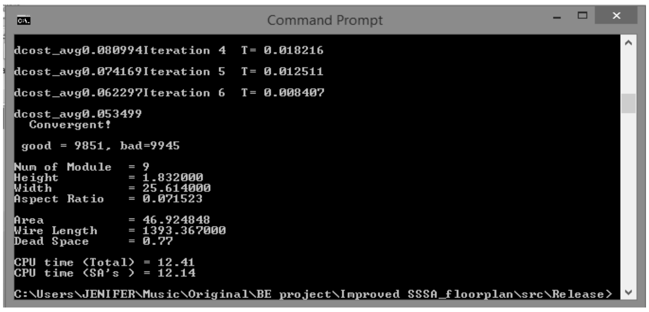
4.2 EXPERIMENTAL RESULTS

The performance of the proposed Simulated Annealing algorithm in terms of Area, Wire length (WL), Dead Space (DS) and CPU time (CPUT) obtained on the MCNC benchmark. Experiments were conducted by assigning three different sets of values to the weight factor x_1, x_2 and x_3 in the objective function. The weight factors are assigned as $x_1 = 0.5$, $x_2 = 0.2$ and $x_3 = 0.3$, giving different emphasis to area, wire length and aspect ratio. Here, the more importance given to the area. The Table.2 represent the proposed method results.

Table.1. Details of MCNC Benchmark

Benchmark	No. of modules	No. of nets	No. of IO pads	No. of pins
APTE	9	97	73	287
XEROX	10	203	2	698
HP	11	83	45	309
AMI33	33	123	42	522
AMI49	49	408	22	953

The benchmark file apte contains 9 modules. During the Simulated Annealing process, it takes 6 iterations to improve their performance. Here the total area minimized to 46.92, the aspect ratio minimized to 0.0715, the wirelength minimized to 1393.37 and the dead space minimized to 0.77 and it takes the CPU time as 12.41 seconds. The simulated result for "apte" file is shown in Fig.6.



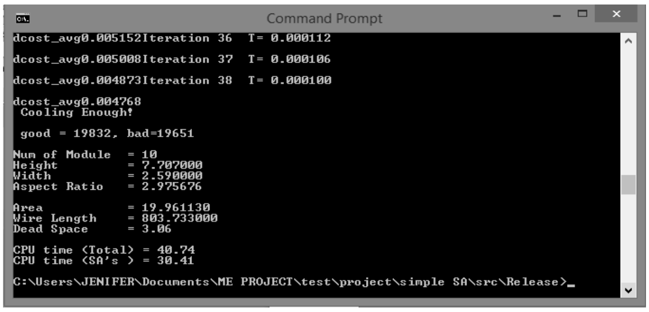
```
dcost_avg0.080994Iteration 4  T= 0.018216

dcost_avg0.074169Iteration 5  T= 0.012511

dcost_avg0.062297Iteration 6  T= 0.008407

dcost_avg0.053499
  Convergent!

 good = 9851, bad=9945

Num of Module   = 9
Height          = 1.832000
Width           = 25.614000
Aspect Ratio    = 0.071523

Area            = 46.924848
Wire Length     = 1393.367000
Dead Space      = 0.77

CPU time (Total) = 12.41
CPU time (SA's ) = 12.14

C:\Users\JENIFER\Music\Original\BE project\Improved SSSA_floorplan\src\Release>
```

Fig.6. Simulated result for "apte" file

The benchmark file xerox contains 10 modules. During the Simulated Annealing process, it takes 38 iterations to improve their performance. Here the total area minimized to 19.96, the aspect ratio minimized to 2.98, the wirelength minimized to 803.73 and the dead space minimized to 3.06 and it takes the CPU time as 40.74 seconds. The Simulated result for "Xerox" file is shown in Fig.7.

Table.2. Proposed method results when $x_1 = 0.5$, $x_2 = 0.2$, $x_3 = 0.3$

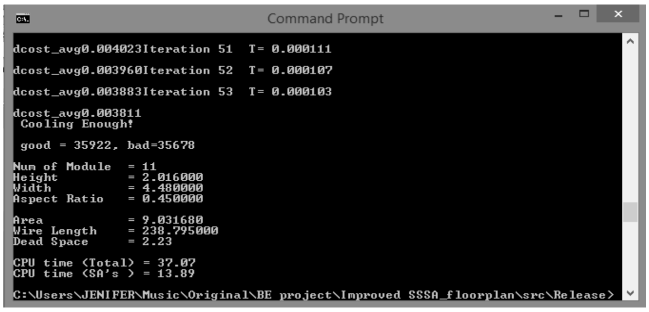
Benchmark	Iteration	Height (mm)	Width (mm)	Aspect Ratio	Area (mm^2)	Wire Length (mm)	Dead space (%)	CPU time (seconds)
apte	6	1.83	25.61	0.0715	46.92	1393.37	0.77	12.41
xerox	38	7.71	2.59	2.98	19.96	803.73	3.06	40.74
hp	53	2.02	4.48	0.45	9.03	238.79	2.23	37.03
ami33	36	0.64	1.84	0.35	1.18	112.09	2.46	83.23
ami49	30	9.98	3.61	2.76	36.05	2042.29	1.69	110.40



```
dcost_avg0.005152Iteration 36  T= 0.000112

dcost_avg0.005008Iteration 37  T= 0.000106

dcost_avg0.004873Iteration 38  T= 0.000100

dcost_avg0.004768
 Cooling Enough!

 good = 19832, bad=19651

Num of Module  = 10
Height         = 7.707000
Width          = 2.590000
Aspect Ratio   = 2.975676

Area           = 19.961130
Wire Length    = 803.733000
Dead Space     = 3.06

CPU time (Total) = 40.74
CPU time (SA's ) = 30.41

C:\Users\JENIFER\Documents\ME PROJECT\test\project\simple SA\src\Release>_
```

Fig.7. Simulated results for “Xerox” file

The benchmark file hp contains 11 modules. During the Simulated Annealing process, it takes 53 iterations to improve their performance. Here the total area minimized to 9.03, the aspect ratio minimized to 0.45, the wirelength minimized to 238.79 and the dead space minimized to 2.23 and it takes the CPU time as 37.03 seconds. The Simulated result for “hp” file is shown in Fig.8.

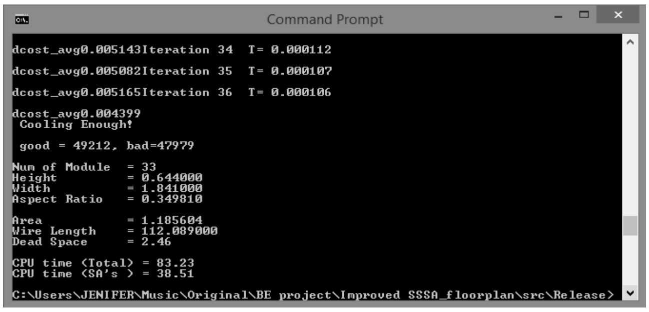


```
dcost_avg0.004023Iteration 51  T= 0.000111

dcost_avg0.003960Iteration 52  T= 0.000107

dcost_avg0.003883Iteration 53  T= 0.000103

dcost_avg0.003811
 Cooling Enough!

 good = 35922, bad=35678

Num of Module  = 11
Height         = 2.016000
Width          = 4.480000
Aspect Ratio   = 0.450000

Area           = 9.031680
Wire Length    = 238.795000
Dead Space     = 2.23

CPU time (Total) = 37.07
CPU time (SA's ) = 13.89

C:\Users\JENIFER\Music\Original\BE project\Improved SSSA_floorplan\src\Release>
```

Fig.8. Simulated results for “hp” file

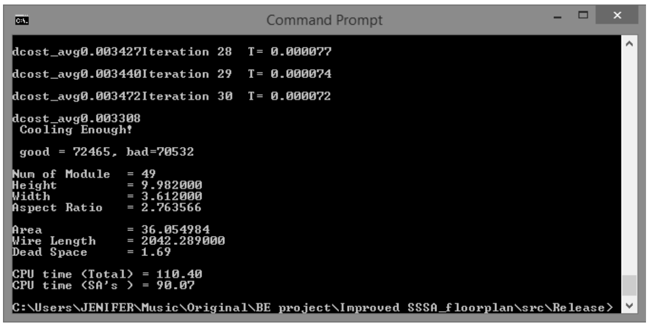
The benchmark file ami33 contains 33 modules. During the Simulated Annealing process, it takes 36 iterations to improve their performance. Here the total area minimized to 1.18, the aspect ratio minimized to 0.35, the wirelength minimized to 112.09 and the dead space minimized to 2.46 and it takes the CPU time as 83.23 seconds. The Simulated result for “ami33” file is shown in Fig.9.

The benchmark file ami49 contains 49 modules. During the Simulated Annealing process, it takes 30 iterations to improve their performance. Here the total area minimized to 36.05, the aspect ratio minimized to 2.76, the wirelength minimized to 2042.29 and the dead space minimized to 1.69 and it takes the CPU time as 110.40 seconds. The Simulated result for “ami49” file is shown in Fig.10.



```
dcost_avg0.005143Iteration 34  T= 0.000112

dcost_avg0.005082Iteration 35  T= 0.000107

dcost_avg0.005165Iteration 36  T= 0.000106

dcost_avg0.004399
 Cooling Enough!

 good = 49212, bad=47979

Num of Module  = 33
Height         = 0.644000
Width          = 1.841000
Aspect Ratio   = 0.349810

Area           = 1.185604
Wire Length    = 112.089000
Dead Space     = 2.46

CPU time (Total) = 83.23
CPU time (SA's ) = 38.51

C:\Users\JENIFER\Music\Original\BE project\Improved SSSA_floorplan\src\Release>
```

Fig.9. Simulated results for “ami33” file



```
dcost_avg0.003427Iteration 28  T= 0.000077

dcost_avg0.003440Iteration 29  T= 0.000074

dcost_avg0.003472Iteration 30  T= 0.000072

dcost_avg0.003308
 Cooling Enough!

 good = 72465, bad=70532

Num of Module  = 49
Height         = 9.982000
Width          = 3.612000
Aspect Ratio   = 2.763566

Area           = 36.054984
Wire Length    = 2042.289000
Dead Space     = 1.69

CPU time (Total) = 110.40
CPU time (SA's ) = 90.07

C:\Users\JENIFER\Music\Original\BE project\Improved SSSA_floorplan\src\Release>
```

Fig.10. Simulated results for “ami49” file

4.3 PERFORMANCE COMPARISION

The Table.3 shows that the area comparison for MCNC benchmark circuits. Here, the performance of the proposed Simulated Annealing in terms of area is compared with the other best methodologies reported in the literature, viz., Transitive Closure Graph (TCG) algorithm [15], Improved SSAA [14], DPSO algorithm [4] as well as VOAS [7].

Here, the TCG algorithm gives importance only to the area, so the weight of area is assigned as 1. In Improved SSAA, equal importance is given to both area and wirelength. So, the weights of both area and wirelength were considered as 0.5 and 0.5 respectively. In DPSO algorithm, the importance given to wirelength i.e. the weight of area is 0.4 and the weight of wirelength is 0.6. Similarly, in VOAS, the equal importance is given to both area and wirelength. So, the weights of both area and wirelength were 0.5 and 0.5 respectively. In this proposed work, the weight of area is 0.5, the weight of wirelength is 0.2 and the weight of aspect ratio is 0.3. Then the respective area of each benchmark file can be shown in below table. From the table it can be cleared that the results produced by the proposed algorithm produces better quality solutions than most of the other methods.

Table.3. Area comparison for MCNC benchmark circuits

Algorithm	**apte Area (mm^2)**	**xerox Area (mm^2)**	**hp Area (mm^2)**	**ami33 Area (mm^2)**	**ami49 Area (mm^2)**
TCG [15]	46.92	19.83	8.947	1.20	37.49
ISSAA [14]	48.47	20.42	9.40	1.26	37.76
DPSO [4]	47.31	20.2	9.5	1.28	38.8
VOAS [7]	47.1	20.3	9.46	1.20	37.8
SA	**46.92**	**19.96**	**9.03**	**1.18**	**36.05**

5. CONCLUSION

This work has deal with a problem in VLSI layout design named as floor planning. Objective of the research work is to design and develop efficient algorithm for the mentioned VLSI layout design process .This research work concerned with optimizing the physical properties of the VLSI circuit. The VLSI layout problem is proved to be NP hard. Main aim of the VLSI layout design is to reduce the chip size signal transmission delay of the chip. So that other parameter like power dissipation and durability of the integrated circuit can be concluded.

Due to inherent complexity of the VLSI problem it requires CAD tools to automate the design process and it enables huge research actives in area of VLSI CAD. In this paper, the floorplan area is minimized by reshaping the blocks which are present inside the floorplan. Here, the Meta heuristic techniques proposed to solve this modern VLSI floor planning problem. To start with Simulated Annealing algorithm that has been implemented and its performance evaluated using MCNC bench mark. This algorithm's performance is compared with TCG [15] algorithm, Improved SSAA [14] algorithm, Discrete PSO [4] algorithm and VOAS [7] as a result the area optimized well in this Simulated Annealing algorithm. It reduces the unused area (i.e.) dead space but it still needs proper tuning in design concern. Because it takes more iteration and more computational time in order to identify the best solution. So, as a future work, we aim to study other algorithms like Genetic algorithm, Particle Swarm Optimization algorithm, Cuckoo search algorithm to solve this modern VLSI floorplanning problem with less computational time along with the reshaping technique.

REFERENCES

[1] Charles J. Alpert, Dinesh P. Mehta and Sachin S. Sapatnekar, "*Handbook Algorithm for Physical Design Automation*", Auerbach Publications, pp. 1-1024, 2008.

[2] Larry Stockmeyer, "Optimal Orientations of Cells in Slicing Floorplan Designs", *Journal of Information and Control*, Vol. 57, No. 2-3, pp. 91-101, 1983.

[3] Weiping Shi, "An Optimal Algorithm for Area Minimization of Slicing Floorplan", *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 480-484, 1995.

[4] Guolong Chen, Wenzhong Guo and Yuzhong Chen, "A PSO-based Intelligent Decision Algorithm for VLSI Floorplanning", *Journal of Soft Computing A Fusion of Foundation, Methodologies and Applications*, Vol. 14, No. 12, pp. 1329-1337, 2010.

[5] Li Yiming, Li Yi, Zhou Mingtian, "Area Optimization in Floorplanning using AP-TCG", *Journal of Convergence Information Technology*, Vol. 5, No. 10, pp. 216-222, 2010.

[6] F.Y. Young, C.C.N. Chu, W.S. Luk and Y.S. Wong, "Handling Soft Modules in General Non-Slicing Floorplan using Lagrangian Relaxation", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 5, pp. 687-692, 2001.

[7] Chyi Shiang Hoo, Kanesan Jeevan , Velappa Ganapathy and Harikrishnan Ramiah , "Variable-Order Ant System for VLSI Multi Objective Floorplanning", *Applied Soft Computing*, Vol. 13, No. 7, pp. 3285-3297, 2013.

[8] Jackey Z. Yan and Chris Chu, "Defer: Deferred Decision Making Enabled Fixed-Outline Floorplanning Algorithm", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol. 29, No. 3, pp. 367-381, 2010.

[9] Po Hsun Wu and Tsung Yi Ho, "Bus-driven Floorplanning with Thermal Consideration", *Integration the VLSI Journal*, Vol. 16, No. 4, pp 369-381, 2013.

[10] Teng Sheng Moh, Tsu Shuan Chang and S.L. Hakimi, "Globally Optimal Floorplanning for a Layout Problem", *IEEE Transactions on Circuits System I Fundamental Theory and Applications*, Vol. 43, No. 5, pp. 713-720, 1996.

[11] Chuan Lin, Hai Zhou and Chris Chu, "A Revisit to Floorplan Optimization by Lagrangian Relaxation", *Proceeding of IEEE/ACM International Conference on Computer Aided Design*, pp. 164-171, 2006.

[12] Jackey Z. Yan and Chris Chu, "SDS: An Optimal Slack Driven Block Shaping Algorithm in Fixed Outline Floorplanning", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 32, No. 2, pp. 175-188, 2013.

[13] Tung Chieh Chen, Yi Lin Chuang, and Yao Wen Chang, "Effective Wire Models for X-Architecture Placement", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol. 27, No. 4, pp. 654-658, 2008.

[14] S. Anand, S. Saravanakumar and P. Subbaraj, "Customized Simulated Annealing based Decision Algorithms for Combinatorial Optimization in VLSI Floorplanninng Problem", *Computational Optimization and Application*, Vol. 30, No. 2, pp. 667-689, 2010.

[15] Jai-Ming Lin and Yao-Wen Chang, "TCG: A Transitive Closure Graph-based Representation for Non Slicing Floorplan", *Proceedings of Design Automation Conference*, pp. 764-769, 2001.