Simultaneous optimization of the area, wirelength and TSVs in a 3D IC design

Atul Prakash[1], Rajesh Kumar Lal[2]

[1,2]Department of Electronics & Communication Engineering, Birla Institute of Technology, Ranchi 835215, India.

E-mail: [1]prakashatul.uit@gmail.com; [2]rklal@bitmesra.ac.in

Abstract: The technology of a three-dimensional integrated circuit (3D-IC) is an emerging approach for improving performance. In comparison to a standard 2-D IC design, which arranges all of the devices on a single planar layer, a 3D-IC stacking of many tiers enables more devices to be placed close together, resulting in the significant area and wirelength reduction. Designing a 3D-IC introduces an extra parameter to be considered while assigning a layer to any circuit component where different layers are connected by Through Silicon Vias. In this paper, we have applied the Parallel-PSO approach to optimize the area, wirelength of the layout and the number of TSVs to connect the different layers simultaneously. The results are obtained and compared with the benchmark circuits available with MCNC and GSRC.

Keywords: Sequence pair (SP), Very large scale integrated circuit (VLSI), 3D-IC, TSVs PSO, Floorplanning, MCNC, GSRC

1 Introduction

A three-dimensional IC is an Integrated circuit manufactured by laying several vertical silicon wafers or die. The different layers are connected either by Through Silicon Vias (TSVs) or by Cu-Cu interconnects [1-5]. While the Cu-Cu interconnection with silicon generates sheer force between them, there is a high chance of failure when IC is heated and hence TSVs are widely used to connect these different layers. Recently 3-D integration has attracted researchers as it provides a higher device density as well as higher bandwidth. It is also possible to integrate heterogeneous technologies in a layered die stack structure, which counteracts system-on-chip integration. Other benefits of 3-D ICs include smaller footprints; lower interconnect delays, higher performance, as well as lower power consumption. 3-D IC changes the wirelength distribution from 2-D layout. Nets can be made shorter in 3-D layout, but TSVs are not free and therefore cannot be used at random. There are two approaches used to design 3-D ICs, via-first and via-last. The TSVs in the Via-first approach only interfere with the device layers [Fig: 1(a)], while in the Via-Last approach: the TSVs interact not only with the device layers but also with the metal layers [Fig: 1(b)].

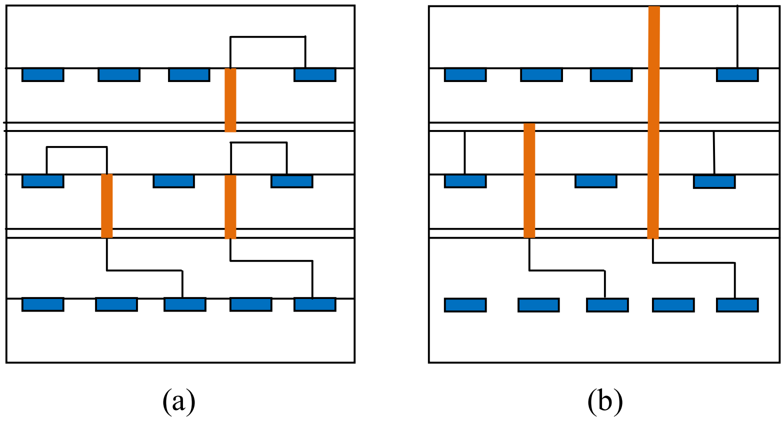

Figure 1: (a) Via-First TSVs & (b) Via-Last TSVs [6]

The inputs to 3D-ICs are:

1. A set of blocks with a specific shape and size,
2. A list of number of terminals on each block and,
3. The netlist describing the interconnections between these blocks.

2 Problem Formulation

We have focused our work in Two-Layer/Four-Layer 3D-IC, where we have partitioned the set of blocks in two/four different layers such that their area difference is as small as possible.

Let $B = \{b_1, b_2, \ldots, b_n\}$ be a set of 'n' modules. The module $b_i$could be represented by$(W_i,\ H_i), 1\ \leq i \leq n,\ where$ $W_i$is its width and $H_i$is its height.

Let $\eta = \{N_1, N_2, \ldots, N_m\}$ be the set nets, where 'm' denotes the total number of nets that link the blocks,

TakingL_i as the estimated length of net N_i,1 $\leq i \leq m$, The placement objective is to identify a group of rectangles represented by$R = \{r_1, r_2, \ldots, r_n\}$ for each block represented by set B such that,

1. Each block b_i can be placed either in the two layer
2. Each b_i block can be put in the rectangle r_i, which has the dimensions$(W_i,\ H_i)$.
3. No two modules overlap each other, that is $r_i \cap r_j = \Phi, 1 \leq i, j \leq n.$
4. The total area occupied by R is minimized
5. The total wirelength determined by $\sum_{i=1}^{m} L_i$, is minimized.
6. The total number of TSVs should be minimized.

2.1 Area Optimization

VLSI floorplanning consists solely of the arrangement of non-overlapping rectangles. The arrangement of blocks on a chip is divided into two types: slicing floorplan and non-slicing floorplan. The sequence-pair (SP) approach was used to investigate the non-slicing floorplan in this work. Sequence pair is a technique for packing bocks that use a pair of modules known as sequences. The ability to have a limited solution space is essential for successful optimization. Murata et. al. [7] has demonstrated that the SP's searching space results in an effective rectangular packing of the modules.

Tang et.al. [8] Proposed a method called Fast Longest Common Subsequence (fast LCS) to encode a sequence pair to its corresponding floorplan. The first order of sequence-pair S_1) is formed by arranging the lines drawn from the chip's southwest corner to its northeast corner in a linear fashion. These are non-intersecting, non-overlapping lines that each pass through one module. The second-order sequence pair (S_2) is obtained by drawing similar lines from the chip's southeast corner to its northwest corner.

Fast LCS is a quick and easy way to calculate LCS for a given sequence pair, where n represents the number of items and the weights is not limited to 1 or integers like LCS.

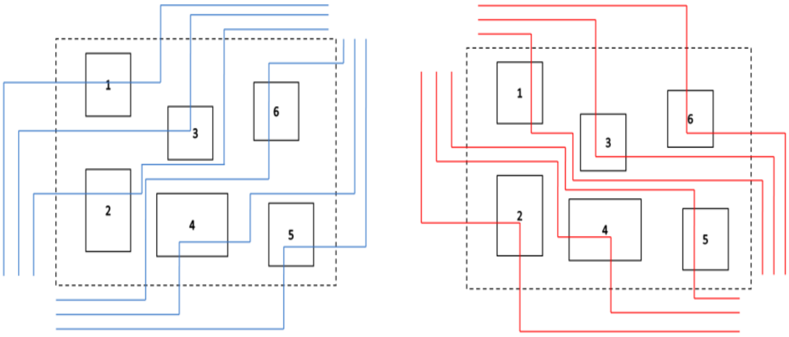

Figure 2: The sequence-pair for the specified placement is (132645, 245136)

If the blocks are1,2,3 ... n and the input sequence pair is (S_1, S_2),, then both S_1 and S_2 are permutations of $\{1,2, \ldots n\}$.The array $P(b)$, $b = 1,2, \ldots, n$ of block positions is used to store the coordinates of block b based on the their weight vector$w(b)$, which corresponds to the width or height of block b. The array $match(b),\ b = 1,2, \ldots., n$ is created to be $match[b].x = i$and$match[b].y = j\ if\ b = S_1[i] = S_2[j]$, the length array $L[1,2,3 \ldots., n]$ represents the length of candidates for the longest common subsequence. The following is the algorithm:

Pseudo code of Fast LCS [8]

```
Initialize  Match Array $\boldsymbol{match}$
Initialize Length Array $\boldsymbol{L}$ $\boldsymbol{with}$ $\mathbf{0}$
$\boldsymbol{for}\ i = 1\ to\ n$
$\boldsymbol{do}\ b = S_1[i]$
$P = match[b].y;$
$P[b] = L[p];$
$t = p[b] + w(b);$
$\boldsymbol{for}\ j\ = P\ to\ n$
$\boldsymbol{do}\ \boldsymbol{if}(t > L[j])$
$\boldsymbol{then}\ L[j] =\ t;$
$\boldsymbol{else}\ break;$
$return\ L[n]$
```

There are three distinct sorts of procedures that may be used to transform a sequence pair to another, which are as follows:

Op1: Swap the names of two modules in any of the two sequences.
Op2: Swap the names of two modules in each sequence.
Op3: Rotate a module as the third option.

To demonstrate the impact of a perturbation on a sequence pair, consider an example with seven modules of dimension as $(3{,}4), (2{,}5), (5{,}2), (3{,}2), (2{,}3), (3{,}4), (2{,}2)$ and the initial sequence pair (S_1, S_2) as $(6475312{,}7612534)$.

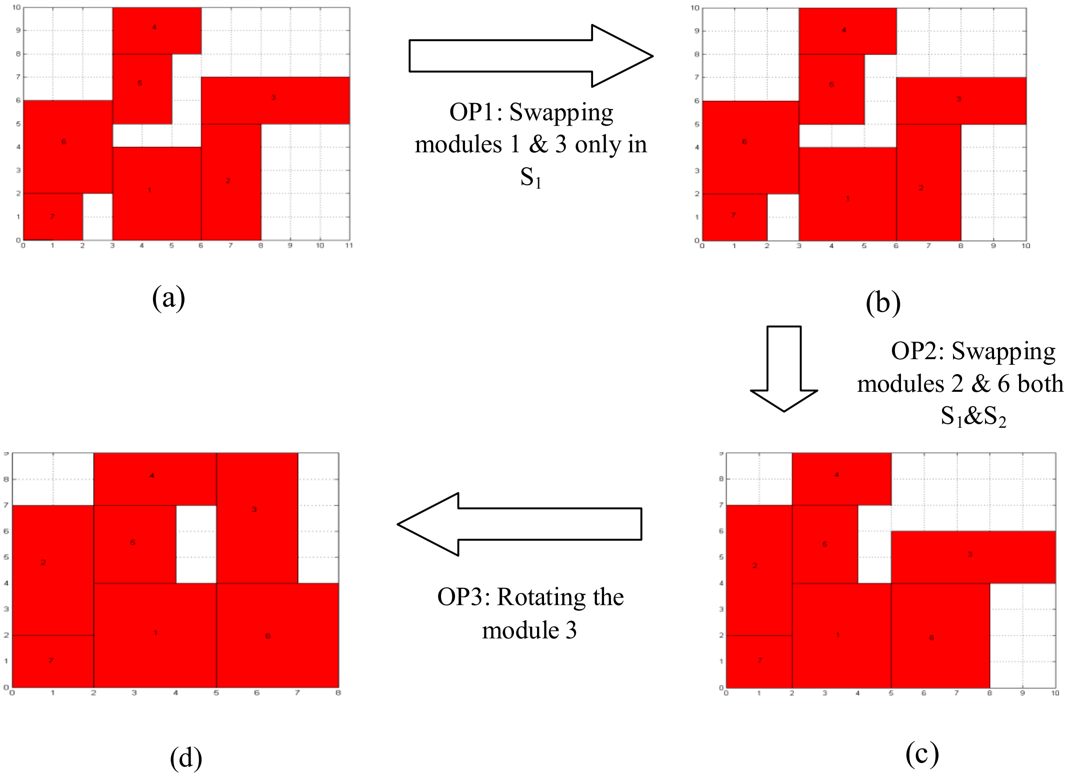
OP1: Swapping modules 1 & 3 only in S1
(a)
(b)
OP2: Swapping modules 2 & 6 both in S1&S2
OP3: Rotating the module 3
(d)
(c)

Figure 3: **A** Floorplan for the Initial sequence pair $(6475312{,}7612534)$. **b** Floorplan for the sequence pair$(6475132, 7612534)$ (after exchanging module 1 & 3 only in Sequence S_1). **c** Floorplan for the sequence-pair $(2475136, 7216534)$ (After exchanging module 2 and 6 in both SequencesS_1& S_2. **d** Floorplan for the sequence pair$(2475136, 7216534)$, when module 3 is rotated

Figure numbers 3 shows the effect of these operations. The floorplan's dimensions change from11×10 to 10×10 to 10×9 to8×9respectively. This example demonstrates how allowing these three valid operations on a sequence pair can significantly alter the floorplan area.

2.2 TSVs Optimization

The 3D-IC design problem involves division of the circuit netlist into multiple parts (in our case two or four parts) such that there are some connections between these parts. The number of edges in the two parts of the circuit is the number of TSVs in the 3D-IC and this number can be calculated as follows:

$$T = \sum_{i=1}^{k} \sum_{j=1}^{k} T_{ij} \ , (i \neq j) \tag{1}$$

Where i , j are the edge's vertices.

$$T = total\ number\ of\ TSVs$$

$$T_{ij} = \begin{cases} 1, & if\ i^{th}\ node\ in\ bottom\ laye\ has\ a\ connection\ with\ j^{th} \text{ node of top layer} \\ & 0, otherwise \end{cases}$$

The design problem in first stage is a partitioning problem where the netlist, say $V$is to partition into $V_1 \& V_2$, such that,

$$V_1 \cap V_2 = \emptyset \tag{2}$$

And,

$$V_1 \cup V_2 = V \tag{3}$$

As the problem involves bipartitioning of a circuit, so equality condition must be satisfied as

$$Number\ of\ nodes\ in\ partition\ 1 \cong number\ of\ nodes\ in\ partition\ 2 \tag{4}$$

2.3 Wirelength Estimation

In 3-D floorplanning, there is a high chance that all the terminals of a net may lie in multiple layers and hence the lateral wirelength calculation becomes necessary. Most of the works related to the calculation of lateral wirelength suggested using Half Perimeter Wirelength (HPWL), Wirelength of a net is determined by measuring half perimeter of the bounding box of all its terminals, assuming they are all in the same plane, as shown in figure 6.2 (a) [9].

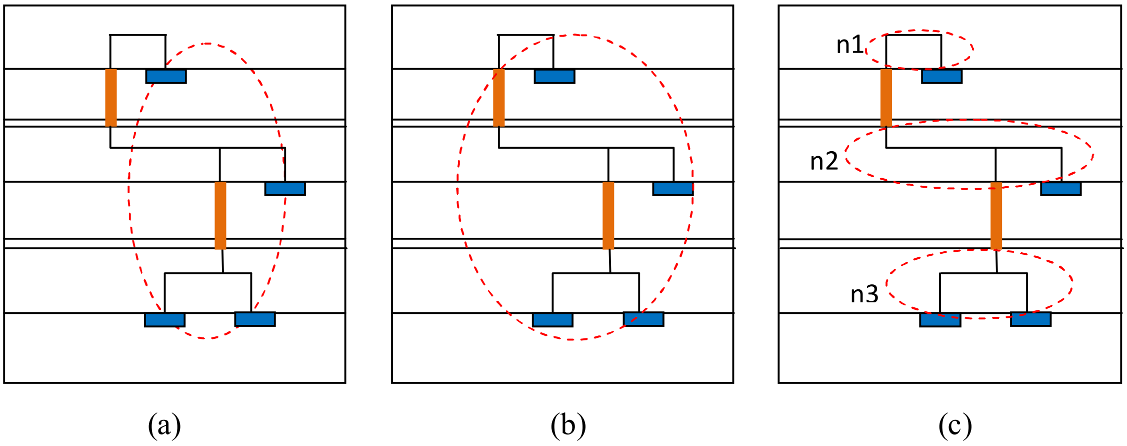

Figure 6.2: wirelength estimation models (a) Bounding box of all terminals of a net, (b) Bounding box of all terminals of a net and TSVs associated with them and (c) a net divided into subnets and summing up individual subnet wirelength

The drawback of this method is that it estimates the lateral wirelength without any information of TSVs locations. Although this is unavoidable as the floorplan do not take care of the TSV placements. In the other technique to estimate the wirelength, the bounding box is chosen such that it covers all the terminals of the net as well as the TSVs associated with that net as illustrated in figure 6.2(b). However, it underestimates the total wirelength when a net has terminals in multiple dies [9].

To overcome the drawbacks of these two methods, in our technique we have estimated the total lateral wirelength by first calculating wirelength on each individual dies then summing up them. Also in our proposed technique, the wirelength of a net (say n) on a particular die is calculated by the half perimeter wirelength of all its terminals on that die and any TSV of n in the same die or die above/below it. In figure 6.2(c), the lateral wirelength of a 3D net n is obtained by summing up the estimated wirelength of subsets n1, n2 & n3 [10].

Here we have bipartitioned the netlist hence the total wirelength will be calculated as the sum of wirelength of partition 0 (W_0), wirelength of partition 1 (W_1) and the wirelength due to TSVs (W_T),

Hence total wirelength $L = W_0 + W_1 + W_T$ (5)

To estimate the wirelength due to TSVs we have taken the TSV size as 3 μm as in [9093] unless otherwise specified.

Hence,
$W_T = T \times 3\ \mu m$ (6)
Where, T is the total number of TSVs
The TSV size is 3 μm as in [5] unless otherwise specified.

2.4 Combined Area, Wirelength and TSVs optimization

While the purpose of traditional 3D-IC design methods is to reduce the number of TSVs, integrating it with the area and wirelength reduction makes this work much harder. Designing multi-objective 3D-ICs is substantially more complicated. One of the most challenging aspects of multi-objective optimization is that there is no one best solution for every target in the solution space. Additionally, adopting an ideal solution for one goal may necessitate receiving a suboptimal result for another. Hence, it is difficult to define what constitutes a good answer. The formulation of a good solution is adopted as it appears in [11]:

1. Allow for more precise handling of the tradeoffs between goals.
2. Produce partitioning that is predictable.
3. Provide a method for dealing with objectives that correspond to amounts of various types.

In this case, a design technique is required that can optimize objective T, the number of TSVs, minimization of objective A, the total area occupied by the modules in two/four-layer and minimizing objective W, the total wirelength. However, the three objectives are dissimilar objectives, which means that optimizing T alone does not necessarily imply that A and $W$are optimized and vice-versa. That is why we adopt a combination-based formulation for multi-objective optimization: The combined objective will be a scalar combined metric C^c given by the following equation:

$$C^c = \lambda * \frac{A}{norm(A)} + (1-\lambda) * \frac{W}{norm(W)} + \mathrm{T} \quad (7)$$

Where,

$C^c = Combined\ objective\ function,$
$\lambda = weight\ given\ to\ area (0 \le \lambda \le 1)$,
$A = Total\ area\ occupied\ by\ the\ modules\ in\ different\ layers\ of\ 3D\ IC$
$L = Total\ wirelength\ of\ 3D\ IC$
$T = \ Total\ numbers\ of\ TSVs$
$norm(A) = normalized\ area$
$norm(L) = normalized\ wirelength$

Minimizing equation (1) seeks to calculate a 3D layout that is as close as possible to any of the best in terms of any beginning goal. The area weight may be used to traverse the distance between each objective's best solution locations, resulting in a predictable structure based on the area weight as well as fine-tuned management of the tradeoff between the three objectives.

3. The 3D-IC design technique

In the present work, a P-PSO algorithm [12] for the optimization of multimodal continuous functions is proposed. P-PSO is used for global optimization by updating particle locations to achieve quick convergence. To determine the layout in each of the two layers the sequence pair (SP) technique with LCS as described in chapter 4 has been used. Firstly, the given netlist has been bi-partitioned and then four different types of operations are allowed to perturb the bi-partitioned circuit and in the given sequence pair to another sequence pair listed as:

Op1: Swap two module names in all the partitions.
Op2: Swap two module names in only one sequence of each partition.
Op2: Swap two module names in both sequences of each partition.
Op3: Rotate a module in each partition.

To start with the designing of 3D-IC we first converted the information provided by the netlist into a matrix known as *adjacency matrix*, where column and row represents the nodes. Then we ha randomly bipartition the given netlist by calling the *initial_position* function. In the next step the total number of interconnections (TSVs) between different layers of 3D-IC is calculated. Total area (A) occupied along with the amount of wirelength required to connect different nodes are calculated. The combined objective function as given in equation 6.5 in initiated. The proposed parallel Particle Swarm Optimization is then applied to achieve the minimum of all the three parameters viz. Area, Number of TSVs and the Wirelength requirement. The steps for the proposed approach for the 3D design problem under consideration are presented as under:

<u>Algorithm for 3D IC design using P-PSO</u>

1. Start at the beginning of netlist and convert it into matrix form.
2. Bipartition the circuit into 0 and 1 partitions as
   $\sum_{i=0}^{L} l_i = \sum_{i=0}^{k} m_i + \sum_{j=0}^{k} n_j$ (8)
   $(total\ number\ of\ nodes\ (l_i)$
   Also, from (4),
   $Number\ of\ nodes\ in\ partition\ 1 \cong number\ of\ nodes\ in\ partition\ 2$
   $where\ l_i = m_i + n_j$
3. Calculate their TSV using
   $TSVs\ between\ partitions\ (T_{ij}) = \sum_{i=0}^{k} m_j \left(\sum_{j=0}^{k} n_j\right)$ (9)
4. Determine the position of modules in each partition and determine the corresponding area and wirelength using Sequence pair (SP) technique with the help of LCS. The total area and wirelength are as:
   $A = A_0 + A_1$ (10)
   $L = W_0 + W_1 + W_T$ (11)
   Where,
   $A_0 = Area\ of\ Partition\ 0$
   $A_1 = Area\ of\ Partition\ 1$
   $W_0 = wirelength\ of\ Partition\ 0$
   $W_1 = wirelength\ of\ Partition\ 1,$ and
   $W_T = wirelength\ due\ to\ TSVs$
5. Initialize the two different sets of PSO parameters parallelly with the same number of particles corresponding to each node of the sequence pair
   $NP, w1, w2, iter_{max}, c1i, c1f, c2i, and\ c2f(NP = number\ of\ particles)$
6. Correspondingly evaluate fitness function, C_{ij}, for all the particles using (6.5), taking weight $\lambda = 0.5$, (for 50% weight to Area and Wirelength objectives).
7. Randomly initialize position vector of each pair of particles x_{ij} $(i = 1,2, \ldots NP$ & $j = 1,2)$
8. Generate initial velocity vector v_{ij} $(i = 1,2, \ldots\ldots, NP$ & $j = 1,2)$ for each pair of particles
9. evaluate the fitness value of each pair of particles using the objective function (using equation 5)
10. set $pbes$t and $gbest$ in the swarm for both pair of particles
11. $\boldsymbol{while}\ iteration\ <\ iter_{max}$
12. update the inertia weight

```
        $w = (w_1 - w_2) \times \frac{(iter_{max} - iter)}{iter_{max}} + w_2$
update $C_1$& $C_2$
        $C_1 = (C_{1f} - C_{1i}) \times \frac{iter}{iter_{max}} + C_{1i}$
        $C_2 = (C_{2f} - C_{2i}) \times \frac{iter}{iter_{max}} + C_{2i}$
$\boldsymbol{for}\, i = 1{:}NP$
  $\boldsymbol{for} j = 1{:}2$
    update the velocity vector $v_{ij}$
    $\boldsymbol{v_{id}} = w \times v_{ij} + C_1 \times r_1^d \times (pbest_{ij} - x_{ij}) + C_2 \times r_2^d \times (gbest_{ij} - x_{ij})$
    update the position vector $x_{ij}$
    $x_{ij} = v_{ij} + x_{ij}$
    $\boldsymbol{do}$
      Op1: Swap two module names corresponding to $x_{i1}$&$x_{i2}$ in both the partitions and calculate the
           fitness function as $f1$.
      Op1: Swap the positions of the modules corresponding to $x_{i1}$& $x_{i2}$ in $S_1$ and calculate the fitness
           function as $f2$.
      Op2: Swap the positions of the modules corresponding to $x_{i1}$& $x_{i2}$ in both the sequence
           $S_1$&$S_2$ and calculate the fitness function as $f3$.
      Op3: Rotate the modules corresponding to $x_{i1}$& $x_{i2}$ and calculate the fitness function as $f4$
      Chose the best fitness among $f_1, f_2, f_3$ &$f_4$
    $\boldsymbol{if} x_{ij}$ is better than the $pbest_{ij}$
      Update $pbest_{ij} = x_{ij}$
    $\boldsymbol{end\ if}$
    if $x_{ij}$ is better than the $gbest_{ij}$
      Update $gbest_{ij} = x_{ij}$
    $\boldsymbol{end\ if}$
  $\boldsymbol{end\ for\ j}$
$\boldsymbol{end\ for\ i}$
$iteration = iteration + 1$
$\boldsymbol{end\ while}$
```

4. Parameters of the proposed algorithm

The values of the parameter $C_{1i}, C_{1f}, C_{2i}, C_{2f}, w_1$& w_2 are crucial since it ensures that the suggested PSO algorithm balances exploration and exploitation. $C_1$is a cognitive parameter that highlights personal best performance while the social parameter C_2 prioritizes the global best. Despite the lack of a well-defined procedure for selecting these numbers, several scholars have underlined the importance of maintaining C_1 and C_2 values so that $C_1 + C_2 = 4$, in which a good outcome is obtained in a different environment. To keep the total of cognitive and social factors at 4, we experimented with setting the values of C_{1i}, C_{1f}, C_{2i}, & C_{2f} to 0.5, 3.5, 3.5 & 0.5 respectively, which resulted in an improved solution.A linearly varying inertial weight of 0.1 to 1 is used. A smaller inertia weight value prioritizes the search in the local best's neighborhood, whereas a larger value global best promotes the search globally. As a result, early on in the process, local search is strong and as the operation progresses, global search becomes more sophisticated.

5. Experimental Results

We simulated the proposed 3-D floorplans with are, wirelength & TSV co-optimization. We used the MCNC & GSRC hard benchmark suites as our test cases. We have the unit in the MCNC & GSRC benchmarks to 1 μm. The TSV size is 3 μm as in [5] unless otherwise specified. The IO pad locations are assigned randomly. First, we tested our approach for two-layered 3D-IC then for a fair comparison the four-layered results were obtained and compared with [13, 5 and 9]; the comparison results are shown in table 6.1. The Tabu Search and Simulated Annealing techniques were used and designed to optimize the area and TSVs by [13]. In [5] authors applied the shuffling frog leaping method to tackle the optimization issue of 3D IC design in terms of area and TSVs count. SA [9] was used by to optimize the wirelength and TSVs. They presented their technique in two stages, stage one planned the hard macros and TSV-blocks at the same time. The wirelength is improved in stage two by reassigning signal TSVs. Further in Table 6.2 and Table 6.3 we have presented the optimization results of Area, Wirelength and TSVs simultaneously for two layered 3D IC design for MCNC and GSRC Benchmarks respectively.

Table 1: Characteristics of MCNC Benchmark circuits

Circuit	Modules	Nets	I/O Pads	Pins	Area(mm^2)
ami33	33	123	42	522	1.1564
ami49	49	408	22	953	35.4454

Table 2: Characteristics of GSRC Benchmark circuits

Circuit	Modules	Nets	Pins	Area(mm^2)
n50	50	485	1050	182962
n100	100	885	1873	179501
n200	200	1583	3599	175696
n300	300	1893	4358	273170

Table 3: 4-layered 3D IC Parameters optimization comparison of results on MCNC & GSRC Benchmark Circuits

Benchmark	**Algorithm**	**Tabu Search [13]**	**SFLP [5]**	**SA [9]**	**ours**	**% improvement over [13]**	**% improvement over [9]**
ami33	WL(μm)	----	----	45179	28750	----	36.36
	Avg. Area(μm^2)	573000	586000	----	339136	40.81	----
	TSV	116	108	141	159	-37.07	-12.76
ami49	WL(μm)	----	----	585804	365666		37.58
	Avg. Area(μm^2)	7481931	7481895	----	9167900	-22.53	----
	TSV	292	263	436	171	41.44	60.78
n30	WL(μm)	----	----	----	17906	----	----
	Avg. Area(μm^2)	----	----	----	55025	----	----
	TSV	----	----	----	310	----	----
n50	WL(μm)	----	----	----	24628	----	----
	Avg. Area(μm^2)	----	----	----	51622	----	----
	TSV	----	----	----	446	----	----
n100	WL(μm)	----	----	148748	53514	----	64.02
	Avg. Area(μm^2)	48170	48112	---	45087	6.40	----
	TSV	996	804	1171	742	25.50	36.64
n200	WL(μm)	----	----	291091	99728	----	65.74
	Avg. Area(μm^2)	50646	51097	----	45971	9.23	----
	TSV	2035	1468	2179	1542	24.23	29.23
n300	WL(μm)	----	----	391694	164364	----	58.04
	Avg. Area(μm^2)	76223	76478	----	71751	5.87	----
	TSV	2133	1823	2730	1793	15.94	34.32

Table: 4: 2-layered 3D IC Parameters optimization results for MCNC Benchmark Circuits

↓ **Benchmark**	**Parameters**	**Result**
Apte	Wirelength (μm)	20668
	Avg. Area(μm^2)	53191804
	TSV	9
Xerox	Wirelength (μm)	205399
	Avg. Area(μm^2)	19701250
	TSV	18
Hp	Wirelength (μm)	84308
	Avg. Area(μm^2)	8930936
	TSV	6
ami33	Wirelength (μm)	64992
	Avg. Area(μm^2)	1164240
	TSV	42
ami49	Wirelength (μm)	36812916
	Avg. Area(μm^2)	768986
	TSV	98

Table 5:2-layered 3D IC Parameters optimization results for GSRC Benchmark Circuits

↓ **Benchmark**	**Parameters**	**Result**
n10	Wirelength (μm)	15528
	Avg. Area(μm^2)	230468
	TSV	12
n30	Wirelength (μm)	54914
	Avg. Area(μm^2)	221028
	TSV	57
n50	Wirelength (μm)	90182
	Avg. Area(μm^2)	207208
	TSV	127
n100	Wirelength (μm)	156698
	Avg. Area(μm^2)	176599
	TSV	251
n200	Wirelength (μm)	270722
	Avg. Area(μm^2)	185008
	TSV	485
n300	Wirelength (μm)	429918
	Avg. Area(μm^2)	292008
	TSV	639

Table 6: Runtime (in seconds) comparison for 4-Layered 3D IC

Benchmark	Runtime (in seconds) SA [9]	Runtime (in seconds) (Proposed-PSO)	% Improvement
ami33	42.46	63.15	-48.73
ami49	184.63	126.33	31.58
n30	----	61.52	----
n50	----	114.35	----
n100	1306.39	306.65	76.53
n200	8237.10	1422.19	82.73
n300	21450.50	2793.42	86.98

Table 7: Runtime for 2-Layered 3D IC (in seconds)

Benchmark	Runtime (in seconds) (Proposed-PSO)
apte	7.58
xerox	8.59
hp	6.95
ami33	21.62
ami49	39.23
n10	4.29
n30	20.57
n50	36.54
n100	79.65
n200	396.12
n300	598.95

6. Summary

3D integrated circuits (3D-ICs) are a new technology that has a lot of promise. 3D-ICs have a tiny footprint and vertical linkages between dies, allowing for shorter wirelength between gates. As a result, they have lower connection latency and power consumption. The 3D Partitioning and Layer Assignment stage is the first of several in the design flow of 3D integrated circuits. This step is crucial since the outcome will have an impact on the performance of succeeding processes. This issue is NP-hard, much like other partitioning problems. The implementation of iterative heuristics [14] was used to handle this essential problem. When attempting to tackle this problem, several factors have been considered. Layer assignment, TSV reduction, wirelength optimization and area balance are some of these considerations. To do this objective, we have proposed Parallel PSO (P-PSO). The result obtained were compared and found to be giving better solutions when compared to other algorithms. As shown in table 3 as compared to SA [9] the average wirelength has an average improvement of 52.35% and the TSV count has an average improvement of 29.64%. Our results show an average improvement of 8.36% over the area occupied by the three-layered 3D IC and an average improvement of 14% over TSV count as compared to the results with Tabu Search [14].

References

[1] M. Jung, T. Song, Y. Peng and S. K. Lim, “Design Methodologies for Low-Power 3-D ICs With Advanced Tier Partitioning,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 25, no. 7, pp. 2109-2117, 2017.

[2] J. Lin and J. Yang, “Routability-Driven TSV-Aware Floorplanning Methodology for Fixed-Outline 3-D ICs,” *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 11, pp. 1856-1868, 2017.

[3] W. Liu, Y. Wang, G. Chen, Y. Ma, Y. Xie and H. Yang, “Whitespace-Aware TSV Arrangement in 3-D Clock Tree Synthesis,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 23, no. 9, pp. 1842-1853. 2015.

[4] T. Song, S. Panth, Y. Chae and S. K. Lim, “More Power Reduction With 3-Tier Logic-on-Logic 3-D ICs,” *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 35, no. 12, pp. 2056-2067, 2016.

[5] J. Kadambarajan and S. Pothiraj, “TSV Aware 3D IC Partitioning with Area Optimization,” *Arabian Journal for Science and Engineering (online)*, 2021.

[6] S. Banerjee, A. Ratna and S. Roy, “Satisfiability modulo theory based methodology for floorplanning in VLSI circuits,” in proc. *Sixth International Symposium on Embedded Computing and System Design (ISED)*, Patna, India, pp. 91-95, Dec. 2016.

[7] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, “VLSI module placement based on rectangle-packing by the sequence-pair,” in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 12, pp. 1518-1524, 1996.

[8] Xiaoping Tang, Ruiqi Tian and D. F. Wong, “Fast evaluation of sequence pair in block placement by longest common subsequence computation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 12, pp. 1406-1413, 2001.

[9] M. Tsai, T. Wang and T. Hwang, “Through-Silicon Via Planning in 3-D Floorplanning,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 8, pp. 1448-1457, Aug. 2011,

[10] C. Li, W. Mak and T. Wang, “Fast Fixed-Outline 3-D IC Floorplanning With TSV Co-Placement,” in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 3, pp. 523-532, 2013.

[11] Sivaranjani, P., Senthil Kumar, “A. Thermal-Aware Non-slicing VLSI Floorplanning Using a Smart Decision-Making PSO-GA Based Hybrid Algorithm,” *Circuits Syst Signal Process* vol. 34, pp. 3521–3542, 2015.

[12] Atul Prakash & R. K. Lal, “Floorplanning for Area Optimization Using Parallel Particle Swarm Optimization and Sequence Pair,” in *Wireless Personal Communication,* vol. 118, pp. 323-342, 2021.

[13] Sadiq M. Sait, Feras Chikh Oughali, Mohammed Al-Asli, “Design partitioning and layer assignment for 3D integrated circuits using tabu search and simulated annealing,” *Journal of Applied Research and Technology*, Volume 14, Issue 1, pp. 67-76, 2016.

[14] M. E. Ekpanyapong and S. K. Lim, “Simultaneous delay and power optimization for multi-level partitioning and floorplanning with retiming,” in Proc. *the Int. Symp. on Circuits and Systems (ISCAS 2004)*, Vancouver, Canada, pp. 1-9, May 2003.

[15] MCNC/GSRC Benchmark, Retrieved from http://vlsicad.cs.binghamton.edu/benchmarks.html.