

Improved Harmony Search Algorithm for Nonslicing Floor Planning of VLSI Chip with Fixed-Outline Constraint

K.Raveendra¹, A V Kiranmai²

Assistant Professor, Department of ECE, S V College of Engineering, Tirupati, A.P, India^{1,2}

ABSTRACT: Floor planning is an important step in the physical design process of Very Large Scale Integrated (VLSI) chips. It is used to estimate the chip area and wire length prior to the real placement of digital blocks and their interconnections. VLSI floor planning is an optimization problem; many optimization techniques were adopted in the literature. In this work, music inspired Improved Harmony Search (IHS) algorithm is proposed for the fixed-outline constrained no slicing floor planning, with the aim of reducing the total chip area. IHS draws its inspiration from the musical improvisation process of searching for a perfect state of harmony. Initially B*-tree is used to generate the primary floorplan for the given rectangular hard modules and HIS algorithm is applied to obtain an optimal solution for the efficient floorplan. The experimental results of the IHS are obtained using the MCNC benchmark circuits. The IHS algorithm provides 2.4% reduction in chip area on an average than HS algorithm.

KEYWORDS: Improved Harmony Search (IHS) - Very large scale integration (VLSI) - Floorplanning – Nonslicing - fixed-outline – Microelectronic Center of North Carolina (MCNC)

I. INTRODUCTION

Floorplanning provides early feedback that evaluates architectural decisions and estimates delay and congestion caused by wiring. Each block consists of several hundreds or thousands of cells that perform a specific operation. The blocks are of rectangular shape with different aspect ratios. The blocks can be classified into two types based on their shape flexibility. They are hard blocks and soft blocks. Hard block has fixed width and height whereas soft block's width and height can be varied as long as its aspect ratio is within the given range and its area is fixed. The aspect ratio of a block defined as the ratio between the height and the width of a block. To optimize the area of the chip, hard blocks are rotated then the width and height of soft blocks are modified without affecting the total area of the block. The classical floorplanning clock packing to minimize the total chip area, but modern floorplanning methods could be devised as a fixed outline floorplanning. There are two types of floorplanning methods used in electronic design automation. They are slicing floorplan and nonslicing floorplan. In slicing floorplan, the whole block area is first partitioned into two slices of equal or unequal sizes using either a horizontal or vertical line then the individual blocks are again partitioned into by using either horizontal or vertical lines. This process continues until all the blocks are separated. This process of partitioning the block area is called slicing floorplan. The slicing floorplan is represented by a binary tree structure known as slicing tree. The leaf nodes of the slicing tree are the blocks of the design. The other nodes are either a vertical partition represented by V or a horizontal partition represented by H. If a floorplan is obtained with no recursive through cuts, then it is called as nonslicing floorplan.

Different techniques are used to represent the nonslicing structure of a floorplan. They are sequence pair [1], bounded slicing grid [2], O-tree [3] and B*-tree [4] representations. In general, nonslicing floorplan methods give optimal layout than the slicing floorplan. Optimization is the act of achieving the best possible result under given conditions. The objective of any optimization algorithm is to minimize or maximize the objective function. An optimization algorithm is a procedure which is executed iteratively by comparing various solutions till an optimum or a satisfactory solution is found. Optimization algorithms are mostly used in all engineering problems. VLSI floorplanning is a NP-hard problem which is to be optimized; many optimization techniques were used in the literature. Harmony Search (HS) algorithm is one of the optimization technique based on the musical improvisation process. It has been successfully applied in the field of VLSI floorplanning [5]. It does not need any earlier domain knowledge, such as the gradient information of the objective function. It requires fewer mathematical requirements and does not entail initial value settings of the decision variables. In this paper, Improved Harmony Search (IHS) algorithm, a variation of HS is proposed for fixed-outline nonslicing floorplanning. The rest of this paper is organized as follows: Section II presents The related work in the field of VLSI floorplanning. Section III explains the proposed methodology. Section IV elaborates on the experimental setup and discussion of obtained results. Section V concludes the paper with possible scope for further explorations.

II. RELATED WORK

Researchers focused on optimization algorithms for floorplanning in both slicing and nonslicing structures. A widely used global search method for VLSI floorplanning problems is genetic algorithm (GA). For nonslicing VLSI floorplanning, a GA has also been presented by Gwe and Lim[6]. Since the encoding scheme does not capture any topological information of the floorplans, the performance of the GA was not satisfactory. Tabu search algorithm is applied to solve module placement problem in [7]. Particle swarm optimization (PSO) was introduced into the floorplanning problem to find the potential optimal placement solution [8]. A new heuristic method was proposed that applied hybrid simulated annealing (HSA) to represent a nonslicing floorplan with an objective function by restricting the area and wirelength [9]. A sequence pair approach to represent nonslicing floorplans with a smaller search space was introduced in [10]. In [11], the authors proposed a coevolutionary multi objective particle swarm optimization (CMOPSO) algorithm to solve a VLSI floor planning problem which was a multi objective combinatorial optimization and has been proved to be a NP-hard problem. An approach based on iterative prototypes optimization with evolved improvement (POEMS) algorithm was proposed in [12]. It used a GA for local search and adopted a nonslicing structure B*-tree for the placement of rectangular modules. A VOAS (Variable – Order Ant System) for area optimization based on the ant algorithms was proposed in [13]. A smart decision-making PSO-GA based hybrid method for thermal aware no slicing VLSI floor planning was proposed in [14]. VLSI floor planning is a NP-hard problem. The solution space will increase exponentially with the growth of circuits scale, thus it is difficult to find the optimal solution by exploring the global solution space.

III. PROPOSED METHODOLOGY

A. HARMONY SEARCH ALGORITHM

The Harmony Search (HS) method is a meta-heuristic optimization algorithm proposed by Geem et al [15]. It mimics a musical improvisation process in which the musicians in an orchestra/band try to find a perfect state of harmony through musical improvisations. Each musician (decision variable) has a different instrument whose pitch range corresponds to a decision variable's value range. A solution vector, also called an improvisation, at certain iteration corresponds to the musical harmony at a particular period, and the objective function corresponds to the audience's aesthetics. New improvisations are based on earlier remembered good ones which are stored in the data structure called the Harmony Memory (HM). A new solution is improvised by using three rules. They are (a) Play what he/she exactly knows (memory consideration) (b) Play by slightly adjusts the pitch Play a new composition (random selection). So, the perfect harmony means the global or nearglobal solution.

The main control parameters of HS algorithm are harmony memory (HM), Harmony Memory Size (HMS), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR), and Bandwidth (BW). Here, HM is a memory location where all the solution vectors are stored; HMCR and PAR are parameters that are used to improve the solution vector.

B. IMPROVED HARMONY SEARCH ALGORITHM

HS has good performance but one drawback it holds is the single harmony memory for optimization process. To improve the speed of the search process, we propose an IHS algorithm for VLSI floorplanning. In IHS, two harmony memories are used. Both memories are initialized with HMS randomly generated solutions. New improvisations are based on the past remembered good ones which are stored in the HM. Since two harmony memories are used in this method, the value of a design variable can be selected from the values stored in both the HM with a probability HMCR. Instead of HMS initial solutions, by using the proposed method $2 \times \text{HMS}$ solutions are generated. From that, HMS best solutions are taken into consideration. The Figure 1 shows the flowchart of proposed IHS.

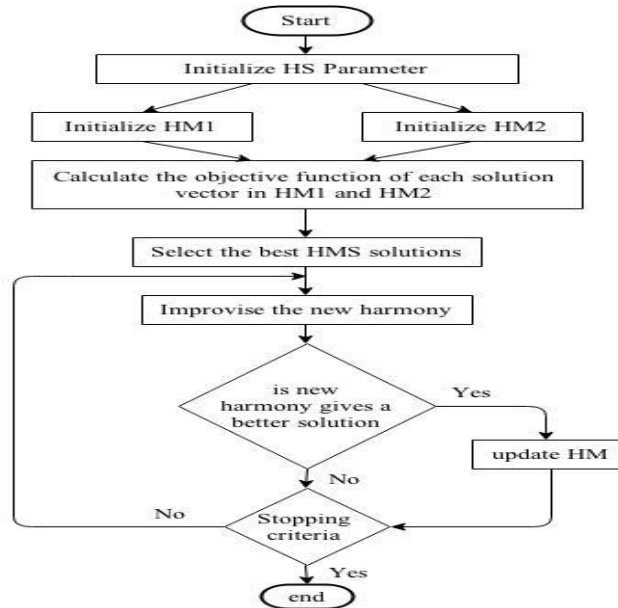


Figure 1. Flowchart of IHS algorithm.

C. FITNESS FUNCTION EVALUATION

For the IHS floorplanning, each musician corresponds to a possible solution. The main objective of the floorplanning is to minimize the total chip area and wire length. The main objective of the floorplanning is to minimize the total chip area and wire length. The formula for calculating the total area of the chip that contains 'i' modules is given in (1).

$$Area = \sum_{i=1}^n (length(i) * width(i)) \quad (1)$$

In this work, the HMS value is taken as 30, 50 and 100. The HMCR value is taken as 0.9. The proposed algorithm is run for 100, 500 and 1000 iterations. PAR and BW values are taken as 0.3 and 0.01. Here α and β values are treated as the weighting factors. In this work α value is taken as 0.8 and β value is taken as 0.2. Table I shows the comparison of the performance of HS and IHS algorithms. Table II compares the performance of the proposed work with other existing work which utilizes B* tree representation.

The optimum result for the apte circuit is obtained when HMS value is 50 and 100. When HMS value is 100 the minimum area is achieved for ami33 circuit. For hp circuit, the minimum area is obtained in 1000th iteration with HMS value as 50. The minimum area for xerox circuit is attained when HMS value is 30 and 100. From the results it is observed that the proposed IHS algorithm gives better results than HS based floorplan optimization algorithm. Figure 6 to Figure 9 show the performance of the HS and IHS algorithm in floorplanning with different HMS values and different iterations.

TABLE I. PERFORMANCE OF THE PROPOSED METHOD

Circuit Name	HMS	HS (Area in mm ²)			IHS (Area in mm ²)		
		Number of Iterations			Number of Iterations		
		100	500	1000	100	500	1000
apte	30	48.95	48.47	48.43	48.43	48.43	48.21
	50	48.47	48.43	48.21	48.43	48.05	48.05
	100	48.47	48.47	48.43	48.21	48.21	48.05
ami33	30	1.52	1.38	1.35	1.35	1.35	1.28
	50	1.835	1.65	1.35	1.28	1.28	1.28
	100	1.75	1.65	1.30	1.28	1.28	1.24
hp	30	10.50	10.16	10.05	10.54	10.50	10.08
	50	10.98	10.79	10.54	10.03	10.03	9.852
	100	10.63	10.26	10.03	10.03	10.03	10.03
xerox	30	20.97	20.64	20.64	20.64	20.42	20.04
	50	21.86	21.86	20.64	20.64	20.42	20.14
	100	21.86	20.97	20.64	20.44	20.42	20.04

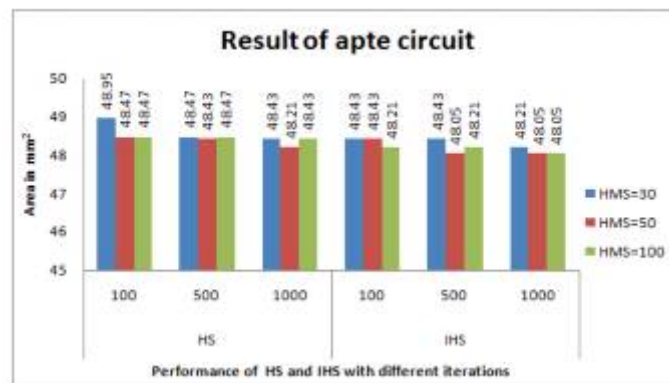


Figure2. Result of apte circuit.

V. CONCLUSION AND FUTURE WORK

VLSI floorplanning is a NP-hard combinatorial optimization problem. To solve this problem in an efficient way, IHS algorithm is proposed in which two harmony memories are used. Hence the initial solutions generated are doubled and the best HMS initial solutions are considered for further process which improves the optimization process. The simulation results of the MCNC benchmark circuits have good and reasonable solution for the nonslicing placement of hard modules within fixed outline constraints. From the results it is inferred that the performance of the proposed IHS method is better than the HS algorithm. When number of iteration increases the total area of the floorplan gets reduced. The HMS value does not influence the performance of the algorithm, so the value of HMS should be identified through empirical analysis. Currently, the IHS algorithm-based framework considers blocks with two dimensions only. Blocks with three-dimension (3D floorplanning) can be used for further work.

REFERENCES

- [1] H. Murata, K. Fujiyoshi, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 12, pp. 1518-1524, 1996.
- [2] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," Proceedings of IEEE/ACM International Conference on Computer Aided Design, pp. 484-491, 1996.
- [3] P.-N. Guo, C.-K. Cheng, and T. Yoshimura, "An O-Tree Representation of Non-Slicing Floorplan and Its Applications," Proceedings of the 36th Annual ACM/IEEE Design Automation Conference, pp. 268-273, 1999.
- [4] Y.-C Chang, Y.-W Chang, G.-M Wu, and S.-W Wu, "B*-Trees: A New Representation for Non-Slicing Floorplans," ACM, 2000.
- [5] K.Sivasubramanian, and K.B.Jayanthi, "Music-Inspired Harmony Search Algorithm for Fixed-Outline Non-Slicing VLSI Floorplanning," World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol. 9, no.6, 2015.
- [6] B. Gwee, and M. Lim, "A GA with heuristic based decode for IC floorplanning," Integr., VLSI J., vol. 28, no. 2, pp. 157-172, 1999.
- [7] N. Xu, X.-L. Hon, S.-Q Dong, and H.-B. Yu, "TSCSP: Tabu Search Algorithm for VLSI Module Placement Based on the Clustering Sequence-Pair," IEEE, 2003.