International Journal of Engineering & Technology, 7 (2.33) (2018) 755-758

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Performance analysis of VLSI floor planning based on artificial bee colony algorithm

GokulAnand K. R [1] *, Rajalakshmi N. R [1], Saravanan K [2], VenkatachalamK [1]

[1] Department of Electronics and Communication K.S.R Institute for Engineering and Technology Tiruchengode, Tamilnadu, India.
[2] Department of Computer Science and Engineering Anna University Regional Campus Tirunelveli, Tamilnadu, India
**Corresponding author E-mail: vlsi.gokulanand@gmail.com*

Abstract

Floorplanning has a great significance in the field of automating the physical design of integrated circuits like VLSI chips, which are manufactured to perform high end operations. The Style of placing the integrated circuits in a floorplan seems to be a paramount. Because, the circuits must be closely placed without overlap between them. A good floorplanner always checks the credibility of the floorplan by its area, output, dependency and potential of placement. In order to get a feasible solution in floorplanning, the designers choose Meta heuristic optimization algorithms to get an optimized solution. This paper chooses Artificial Bee Colony Optimization algorithm to obtain a feasible floor planning in VLSI circuit placement. The main objective of the paper is to minimize the area and interconnect in order to achieve balanced floorplan Experimental results obtained using MCNC benchmark circuits shows that the proposed algorithm is efficient in its optimization.

Keywords: *Floor Planning; Corner List; Block Placement; Optimization; Artificial Bee Colony.*

1. Introduction

In designer's point of view, floorplanning is a process of placing circuit modules of arbitrary sizes and dimensions on a given layout area. The common objectives of floorplanning design include the minimization of total chip area and the minimization of interconnection wire length. For many years, floorplanning has been considered as an important problem in IC design. The entire circuit is partitioned into several sub-circuits and it further partitioned into smaller sub-circuits, until they are small enough to be handled. Each sub-circuit is called as block. From the layout point of view, each block corresponds to a rectangle with certain specific height, width and aspect ratio. From the part of computation, floorplanning in VLSI seems to be a hard non-polynomial problem due to nature of placement. The solution space for the problem say the placement will increase exponentially with the increase in scaling of circuits, thus making it difficult to explore an optimal solution within a local search space. In order to get optimal solution and not to get stuck in local search space, we go for global search space using population based optimization algorithm. Module block placement study has been explored in detail since the 2000s [1-2]. The representation for arrangement of modules is based on the flexibility in its arrangement. Pei et al. explained the tree representation for the placement of modules [1]. For flexible and vibrant placement, the modules should be placed without any placement restrictions. In this regard, cornerlist placement proves to be efficient as it has the dynamic ability to place closer to the next module without overlap. The paper by Hong et al., proved the efficiency of cornerlist placement over other approaches [3].

In optimization, a countable number of metaheuristic algorithms are opted by VLSI floorplanners such as Genetic Algorithm (GA), Evolutionary Algorithm (EA) and many more due to its flexibility and easiness to be adapted in a varied floorplan models. Now a day, Swarm intelligence gives promising solutions than other optimizations. It is a metaheuristic approach based on the behaviors of insects or animals. Karaboga et al. stated the performance of artificial bee colony with other algorithms [4]. Further, the performance of algorithms can enhance by combination of meta heuristic algorithms. Sivaranjaniet al., stated the performance of two optimization algorithms (PSO & FF algorithms) for non-sliceable floorplanning [5]. In combination of two or more algorithms, the solution from one algorithm acts as an initial solution to other one and future, the solution gets optimized by next algorithm to get highly optimized solution.

2. Existing work

Initially, Corner Block List helps to model floorplan. Here, the transition from floorplan representation to modeling it as a layout is done in a linear runtime O[n]. Corner List's computational complexity and search space are compared to Corner Sequence and Moving Block Sequence. The modules can be placed, rotated from their respective positions to new positions to obtain a feasible solution in placement. Hong et al., proposed in their paper about effective placement of modules by insertion and deletion of modules [6]. In spite of search space and time complexity of Corner List being same as Corner Sequence, Corner List works more on corners than Corner Sequence of modules. Corner Sequence considers only the corners bounded by the floorplan contour. Hoo et al., has stated in the paper about search for corners while placing the modules [8].

In modern optimization approach to solve NP (non-polynomial) problems, we go for optimization algorithms which are taken from the inspired behavior of insects and animals commonly known as Swarm intelligence. This intelligence helps us to develop an algorithm and explore it such as Particle Swarm Optimization (PSO) algorithm, Ant colony Optimization algorithm, Flock intelligence of bird's algorithm and many more [9]. Now-a-day, new algorithms

seem to be a promising one.Lalin et al., explained about successful implementation of Bat algorithm for solving floorplanning problems in design of VLSI circuits [10]. Bai explained about its heuristic optimization method using swarm intelligence [11]. This intelligence helps in floorplan modeling.
chen et al., stated about the methodology of applying PSO algorithm to floorplan modeling[12]. Ant colony algorithm goes one step forward in area of optimization by its problem solving approach. Hoo et al., has stated about the ant colony optimization for floorplanning [8].

3. Proposed model for placement

In this proposed model, Corner Block Placement method is adapted to make a good placement for the modules. Corner list allow the modules to choose any of the corners (top or bottom) available for the placement of modules.

a) Layout Model Using Corner List

The Module's top leftmost corner and bottom rightmost corner are taken as positions for placement of next successive module in floorplan. The corners of each individual modules act as a subset of the entire corner points in the floorplan layout. The floorplan layout is plotted in X-Y axis. The corners should satisfy the condition that X-extreme-bottom ≥ X-extreme-left and Y-extreme-left ≥Y-extreme-bottom.

b) Corner List (CL)-placement

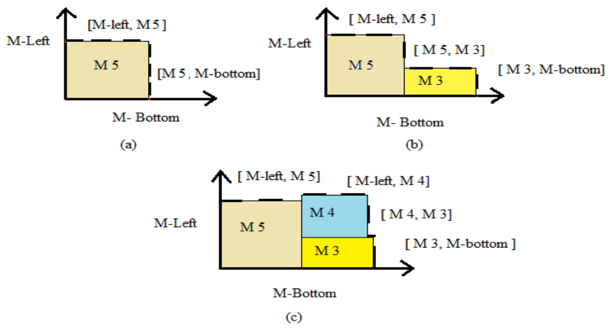

Fig. 1: Examples of Corner List-Layout Transformation Where Modules Are Placed in the Sequence of (A)–(C) (Dotted Lines Represent the Contour Plot).

We have corners known to beModule M-leftandModule M-bottom.Here, Module M5 occupies first in the row to the edge of the contour at the bottom corner.By corner update, two corners namely [M-left, M5] and [M5, M-bottom] are generated, as shown in Figure.1.a. Subsequently, Module M3 is placed in the corner of M5, M-bottom,counting it to three rising corners, that are[M-left, M5], [M5, M3] and [M3, M-bottom]. When third module placement occurs, Module M4 is located to the edge of the corner [M5, M3].Here, a newlyplotted contour and upgradedcornerlist are generated, as shown in Figure.1.c.The corner [M-left, M4] is spotted out before choosing the corner[M5, M4]. It is all about choosy nature of cornerlistwhich give priority to the corners thatsatisfythe floorplan layout. This cycle continues until the last module is placed in the layout design.

c) Corner List's Corner Update

The corner coordinates of the modules are crucial as the algorithm need to identify the coordinates before the next module is placed. The new placement is on its previous module's desired corner. When the coordinates are found erroneous, it lead to misalignment of placing modules in a desired space. This is eventually followed by overlapping of modules or occurrence of gap between the modules. Hence, periodic up gradation of corners along with their coordinates help the floorplanners to avoid overlaps or gaps in the layout.

4. Proposed model for optimization

The paper proposes an algorithm to solve the problem in floorplan design using bee intelligence. Karaboga et al., given in paper [14] a comparative study on this bee algorithm with other algorithms.

a) Artificial Bee Colony Algorithm

In biological eco-system, beesserve as an effective swarm insects that works in collaboration with themselves to explore food and protect their colony. By nature, bees are special insects that has natural ability to explore and exploit its surrounding to maximum.This qualityhelps the bees to spotlight the food source in a shortest distance thus guiding more foraging or searching bees to follow the same way for the search of food. This efficient exploration of food source by searching beesis a collective work of entire bee community. The combined intelligence of swarm of bees hold three important features: sources of food, employed search bees and finally, unemployed search bees and also denotes two working models of bee's behavior: one is recruiting bees for exploring the source of food and other is discarding the exploited food source. The three important features are discussed below:

1) Sources of food: A search bee examines several characters connected to the source of food in order to select it. The factors for selection of source depends on its distance to colony, nectar quality, richness in energy and challenges in exploring the energy from source
2) Employed Search Bees: An employed search bee is associated to a particular source of food and its colony.It carries information about this particular source of food. It spreads it as a message to onlooker bees in the colony. Onlooker bees always look out for a new opportunities trying to explore the message supplied by the search bees. The information in the message includes the duration of the path, security and percentage of gain from the source.
3) Unemployed Search Bees: A un employed search bees can either act as an onlooker bee trying to explore the source through message from the employed bees or it can be a scout bee which searches for environmental threats randomly around the hive.

Sharing of information within the bees help to achieve a stable communication among them and it make the bees to find a good quality of food sources. The foraging bees in the hive resemble a dance called waggling dance. This waggle dance informs the scout and onlooker bees to employ the dancing bees. The ultimate goal of bees is to find a profitable food source. There are two possible ways for a bee:

1) It could be a scout bee and begins its search for food around the nest.
2) It would be a recruited bee after looking its waggling dances in the search of new location for food.

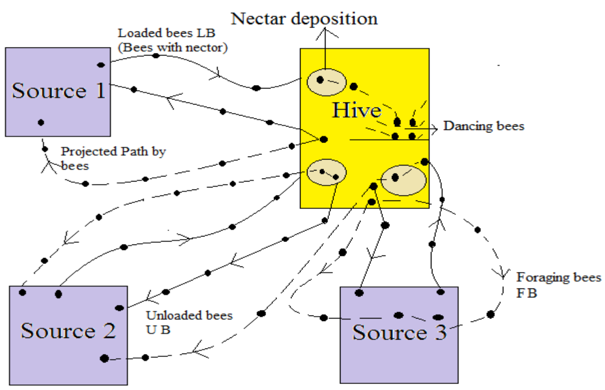

Fig. 2: Working Behavior of Bees.

After exploration of search space for the source of food, bee utilizes the source by memorizing the path of the source through its ability of foraging. Thus, the bee turns up into an employed forager. The function of foraging bee is to collect the nectar and return it safely to the hive. After depositing the nectar, the bee follows below given methods:

i) The bee may turn to be a foraging bee as soon as the food source is discarded.
ii) The bee may perform wangling dance around the hive for informing its idea for new search to other ones.

5. Algorithmic operation

In this algorithm, the position of source of food denotes a feasible solution to the problem that need to be optimized. Here, the total count of onlookers or employed bees are found to be equal to number of feasible solutions obtained from the population. The algorithm develops a random distribution in the initialization of population say 'P' in a finite number of solutions (FN). Here, FN denotes the number of onlooker or employed bees .The solution in each cycle is represented as S_i (i=1 to FN) and denoted as vectors in dimensional space 'D'. The 'D' defines the count of measuring parameters in optimization. First step is to initialize the algorithm and next step is making the positions (solutions) to undergo a cycle of repetition say cycle C = (1 to HCN) where HCN denotes highest cycle number in search operations done by onlookers, employed and scout bees. As soon as the search operation starts, bees convey the message of their source of food say nectar and its location to onlookers. Usually, the onlookers will evaluate the message about nectar and if the valve of nectar seems to be greater when compared to the preceding one, the bee upgrades the newly spotted location to its memory and erases past location. The steps involved in the operation of algorithm are stated as below:

1) First step is population initialization.
2) Now, place the source of food available to the employed bees.
3) Place the onlookers to the source of food available to the employed bees.
4) Place the onlookers to the source of food based upon quantity of nectar available.
5) Explore new sources of food chain through scout bees.
6) Periodically upgrade the best source of food explored so far.
7) Repetition of above steps must be done till desired result is obtained.

a) Search Operation

To this model, at every stage at least one scout bee is employed to search for latest source of food.The selection of onlookers and employed bees are based on their count to one another. Every process in search has three steps to be followed: First is to send bees that are employed for search of food source and evaluate their nectar amount. Next step is to share the nectar information to onlookers and to evaluate the food source areas available. Final step is to determine the bees that scout randomly and find new sources of food. These three steps are done continuously throughout the cycle for a determined number of times called highest cycle number (HCN) or till the algorithm terminates.

b) Objective Function

In Artificial Bee Colony algorithm, the objective function can be defined according to the problem undertaken for optimization. Here, the objective function governs the work of placing the modules in an optimized area. The expression is given as below

$$A_{ij} = S_{ij} + \phi_{ij}(S_{ij} - S_{kj}) \quad (1)$$

Here,

$S_{i\_1 j\_1} = 0$,

S_{ij} = Total Area,
ϕ_{ij} = Random (1,-1),
S_{kj} = Total area of the Modules,
A_{ij} = New Area.
Where
k = (1 to FN) and j $\in$ (1 to D) both are randomly chosen indexes. Even though denoting k in the algorithm in a random fashion, it must be varied from i. ϕ_{ij} denotes random range from [1,-1] and it resists the generation of nearby source of food around S_{ij} and compares the positions of food through onlookers. The equation (1) says that the difference in parameter S_{ij} and S_{kj} starts to decrease resulting in decrease of perturbation on S_{ij}. This makes the process of search to approach a optimum result in a global search space. Thus, the step size is greatly decreased.

c) Fitness Function

An onlooker discovers particular source of food based on its probabilistic number assigned to that particular source of food. Here, P_i is determined through the below given expression

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \quad (2)$$

Where fit $_i$ refers to fitness value of solution i. and its proportionality is measured with amount of nectar in position i. Here, FN is taken as the number of sources of food that are matched to the count of onlooker or employed bees.

d) Pseudo-Code:

1) Let the population for the solutions be S_i where
2) i=1 to FN.
3) Examine the obtained population.
4) Total counts of cycle = No. of iterations.
5) Repeat it.
6) Newly obtained solutions A_i are produced by employed bees through equation (1).
7) The greedy selection method is chosen by employed bees for selection and placement.
8) The probabilistic values p_i for feasible solutions are calculated using S_i by equation (2).
9) Depending on $p_i$valves, the newly obtained solutions A_i are generated for onlookers from selected S_i solutions.
10) Now the greedy selection process is applied to the onlooker bees for further placement.
11) Finally, the abandoned solutions are determined by scout bee and replaced by randomly generated new solution S_i.
12) Further solutions are obtained by multiple iterations i.e. cycle = cycle + 1.
13) Repeat until cycle gets completed i.e. Highest Cycle Number (HCN).
The best solution achieved so far is memorized.

e) Experimental Results

Consider the standard MCNC benchmark circuits AMI33, AMI49 and XEROX. Every benchmark circuits are assigned with standard number of modules, nets, i/o pad and pins [3]. The characteristic features of standard MCNC benchmark circuits are stated in the table below

Table 1: Characteristics of Standard Benchmark Circuits

CIRCUIT	MODULES	NETS	I/O PAD	PINS
AMI33	33	123	42	522
AMI49	49	408	22	953
XEROX	10	203	2	698

We implement this algorithm in C programming language on a standard PC running under Intel i3 processor with512MB RAM memory. The algorithm gives an iterative and deterministic solution. Here, performance of this algorithm is examined as a comparative study with other placement algorithms [7] and the results obtained are fruitful.

Table 2: Comparison among HPSO-FF, HAS, PGHA, SA with B*-Tree, PSO with B*-Tree, O-Tree, Enhanced O-Tree with Our ABC Algorithm with Corner List

Algorithm	Ami33 Area (mm^2)	Ami49 Area (mm^2)	Xerox Area (mm^2)
ABC with Corner list	1.19	38.97	20.05
HPSOFF	1.23	38.53	20.10
HSA	1.21	37.80	20.89

PGHA	1.24	38.60	20.20
PSO with B*-tree	1.28	41.01	19.45
SA with B*-tree	1.36	43.34	20.47
Enhanced O-tree	1.29	39.92	20.42
O-tree	1.34	45.5	23.80

The algorithm holds good for two benchmark circuits such as AMI33 and Xerox. Following figures [3-4] & [5] illustrate the placements of the benchmark circuits.

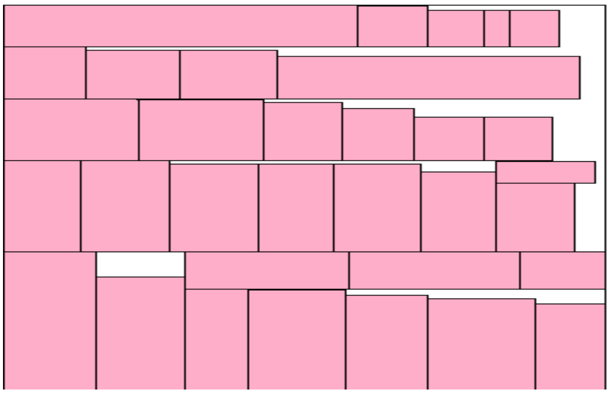

Fig. 3: Placement of AMI33.

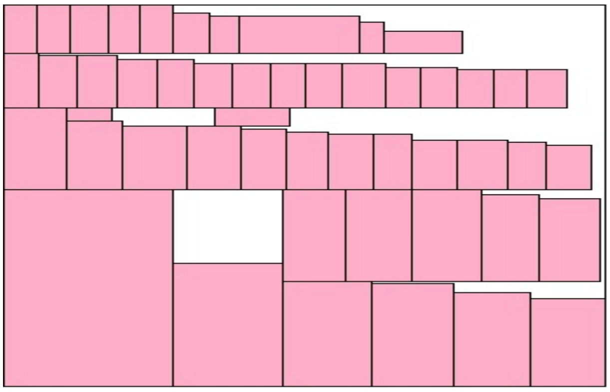

Fig. 4: Placement of Ami49.

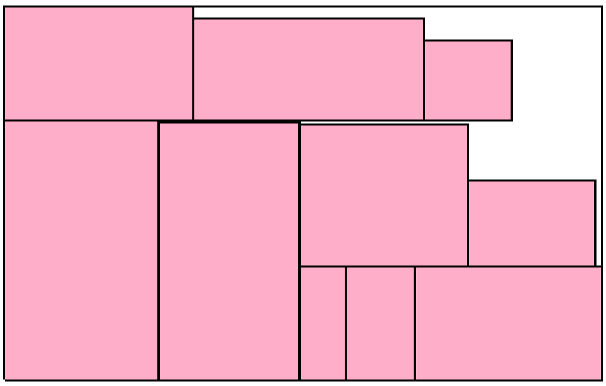

Fig. 5: Placement of Xerox.

6. Conclusion

In this paper, the algorithm finds a feasible location for placement and it can be modified and updated at any time of operation for locations of the modules say either bottom-left or top-right. Here, the dead space of ami33 module is reduced to 4% and the results of other benchmark circuits in table 2 are also quite promising one when compared with other algorithms.

Acknowledgement

The authors would like to thank Sivaranjani. P [2], [3] andChyi-ShiangHoo [5] for technical discussions regarding placements and optimization.

References

[1] Pei-NingGuo, Toshihiko Takahashi, Chung-Kuan a Chengand Takeshi Yoshimura (2001) “Floor planning Using a Tree Represen-tation” *IEEE Transactions On Computer-Aided Design of Inte-grated Circuits and Systems, vol. 20, no. 2.

[2] Yun-Chih Chang, Yao-Wen Chang, Guang-Ming Wu In addition, Shu-Wei Wu (2000) “B*-Trees: A New Represen--tation for Non-Slicing Floor plans” *IEEE Design Automation Conference: 458-463.

[3] Xianlong Hong, Gang Huang, YiciCai, JiangchunGuSheqin Dong and Chung-Kuan Cheng (2004) “Cornerblock list- An efficient and effective topological representation of non-slicing floorplan” *IEEE Transactions on Circuits and Systems -II: Express Briefs, vol. 51, no. 5.

[4] DervisKaraboga and BahriyeBasturk (2008) “On the Performance of artificial bee colony (ABC) algorithm” *European Journal of Applied Soft Computing 8687-697.

[5] Sivaranjani. P and A. Senthil Kumar."Hybrid Particle Swarm Optimization Firefly algorithm (HPSOFF) non-slicing VLSI floorplanning."Journalof Intelligent & Fuzzy Systems32, no.1 (2017):661-669.

[6] XianlongHong, SheqinDong, GangHuang, YiciCai, Chung-Kuan Cheng and Jun Gu (2004) “Corner Block List Representation and Its Application to Floor plant Optimization”*IEEE Transactions on Circuits AndSystems -II: Express Briefs, vol. 51, no. 5.

[7] Sivaranjani.P andA.SenthilKumar.”Thermal-aware non-slicing vlsi-floorplanning using a smart decision- making Pso-Ga based hybrid algorithm.” Circuits, Systems,and Signal Processing, 2015 – Springer.

[8] Chyi-ShiangHoo, KanesanJeevan, VelappaGanapathy and HarikrishnanRamiah (2013) “Variable-Order Ant System for VLSI multi objective floor planning” European Journal of Applied Soft Computing 13, 3285–3297.

[9] Yang, Xin-She.Nature-inspiredoptimization algorithms.Elsevier, 2014.

[10] LalinL.Laudis, ShilpaShyam, Caroline Jemila, Suresh. “MultiObjectiveBatAlgorithmforCombinational optimization in VLSI.” ICSCC2017, 7-8, Procedia Computer Science 125 (2018); 840-846.

[11] Qinghai Bai (2010) “Analysis of Particle Swarm Optimization Algorithm” *IEEE Transactions on Computer and Information Science vol. 3, no. 1.Guolongchen, WenzhongGuo and Yuzhongchen (2010) “A PSO-based intelligent decision algorithm for VLSI

[12] floorplanning” *IEEE Transactions on Soft Computing, vol.14, no. 12.

[13] DervisKaraboga and BahriyeBasturk (2007) “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm” *Springer Science and Business Media B.V.

[14] DervisKaraboga and Bahriye Akay (2009) “A comparative study of Artificial Bee Colony algorithm” European Journal of Applied Mathematics and Computation 214 108–132.