JOURNAL OF PURE AND APPLIED MICROBIOLOGY, Aug. 2015. Vol. **9**, p. 71-77
*SPECIAL ISSUE ON RECENT RESEARCH CHALLENGES IN BIO-MEDICAL APPLICATIONS*

# Evolutionary Algorithmical Approach for VLSI - Placement Problem using Signal Processing

**R. Varatharajan[1], Angeline Peace Preethi[2] and K.G. Revathi[3]**

[1]Department of ECE, Sri lakshmi Ammal Engineering College, Chennai, India.
[2]Anna University, Chennai, India.
[3]Department of ECE,DMI College of Engineering, Chennai, India.

(Received: 03 February 2015; accepted: 01 May 2015)

**Physical layout automation is an important in VLSI's field. With the advancement of semiconductor technology, VLSI is coming to VDSM (Very Deep Sub Micrometer), and the scale of the random logic IC circuits goes towards million gates. Physical design is the process of determining the physical location of active devices and interconnecting them inside the boundary of the VLSI chip the earliest and the most critical stage in VLSI layout design is the placement. The background is the rectangle packing problem: given a set of rectangular modules of arbitrary sizes, place them without overlap on a plane within a rectangle of minimum area. The VLSI placement problem is to place the object in the fixed area of die with out overlap and with some cost constrain. Such as the wire length and area of the die. The wire length and the area optimization is the major task in the physical design. We first introduced about the major technique involved in the algorithm.**

**Key words:** Placement problems, Memetic algorithm, Wire length minimization, Area minimization.

The task of the very large scale integration (VLSI) placement is to assign exact location to various circuit components within the chip area. It involves number of objective such as wire length, area of the die, timing and power. Placement treats the shapes of all blocks as fixed; i.e., it only determines the location of each block on the chip. The variables are the xy locations of the blocks; most blocks are standard cells the y-locations of cells are restricted to standard- cell rows. Placement instance sizes range in to the tens of millions and will continue to increase. Placement is usually divided in to two steps: global placement and detailed placement[1]. Global placement assigns blocks to certain sub regions of the chip with out determining the exact location of each component within its sub region. As a result, the blocks may still overlap. Detailed placement starts from the result of global placement, removes all overlap between blocks, and further optimizes the design. Placement objectives include the estimated total wire length needed to connect blocks in nets, the maximum expected wiring congestion in subsequent routing, and the timing performances of the circuit. To solve such large scale mixed size VLSI placement problem. Many algorithms are used to solve the placement problem in VLSI among these algorithm memetic algorithm is used to solve these placement problem consider here.

The placement algorithms are classified into constructive and iterative improvement methods[12]. The constructive algorithm starts with the empty set and builds up the partition by adding one element at a time. These algorithms are faster but the quality of result is not good. An iterative algorithm starts with initial placements and repeatedly modifies it. These algorithms give good result but it takes long time.

* To whom all correspondence should be addressed.
E-mail: varathu21@yahoo.com

The placement problems are, the wire length[2] and area of the die[3], routability, power minimization and delay[2]. Out of these mention problem the area minimization and the wire length minimization are the most critical part. For the area and wire length optimization a modern placer needs to handle the large-scale design with millions of object, heterogeneous object with different sizes and various constrained placement such as preplaced blocks and chip density. The traditional approach in placement is to construct an initial solution by using constructive heuristic algorithms. A final solution is then produced by using iterative improvement techniques where a modification is usually accepted if a reduction in cost occurs, otherwise it is rejected. The solution generated by constructive algorithms may be far from optimal. Thus an iterative improvement algorithm is performed next to improve the solution the computation time of such algorithm is also large.

For many combinatorial optimization problems, no effective algorithms capable of finding guaranteed optimum solutions in short time are available. Therefore, powerful heuristics have been developed that deliver the guarantee optimum solution, but have shown to be highly effective in many test cases. A special class of heuristics is investigated. The algorithms under consideration are called memetic algorithms, which are - roughly speaking - hybrids of evolutionary algorithms and problem-specific search algorithms, such as greedy heuristics and local search.

Memetic algorithms (MAs) are evolutionary algorithm (EAs) that apply the local search process to refine the individual MAs include a broad of metaheuristics[7, 4, 11] This method is based on the population of agents and proved to be of practical success in a variety of problem domains. We can be sure that MAs constitute one of the most successful approaches for combinatorial optimization in genera, and for the approximate solution of NP optimization problems.

The reminder of this paper is organized as follows section 2 gives the memetic algorithm flow. Section 3 involves the equation involves in the placement problem. Section 4 experiment result for the problem consider.

**Memetic algorithms**

Memetic Algorithms are class of stochastic global search heuristics in which Evolutionary Algorithm based approaches are combined with problem-specific solvers. Later local search heuristics techniques are implemented. This hybridisation is to either accelerate or to discover good solution from the population. Where the evolution alone would take long time to discover or to reach the solution. Memetic Algorithms use heuristic local searches either approximate method or exact method to get the local refine solution from the population.

There are several reasons for the hybridising evolutionary algorithms with local searchers some of them are mention below.

1. Complex problems are decomposed to different sub problems could be better solved by different methods
2. The hybridisation of evolutionary algorithm with local search algorithm result in fine tuning or repairing the best solution(s) produced by the evolutionary algorithm. The powerful local searcher introduced diversity into the solution.
3. The sub problems are deal by the various operators for example cross over and mutation or by the local searcher. Thus the search process is done with in the search space region.
4. In some cases the available exact or approximate methods for the sub problems are include with the evolutionary algorithm.
5. The heuristic or local search used to repair the solutions found by the evolutionary algorithm. The heuristic or local search do the same in the Memetic algorithm such heuristic are called Meteheuristic algorithm.

**Memetic algorithm flow**

Memetic algorithms try to simulate cultural evolution rather than the biological one, as evolutionary algorithm do. They are a combination of population based global optimization and heuristic local search. First individuals are initialized randomly. Starting with each single individual, one local search is performed. After that, the individuals start to interact either competitively (in selection form) or by cooperatively (in recombination form) these two steps are repeated until the stopping criterion is met.

From this context of above we can know that memetic algorithm as two level optimization algorithm; the top level algorithm is an evolutionary algorithm otherwise a population based algorithm

and at the bottom levels a single individual optimizer like hill climbing or simulated annealing or some other local search algorithm.

**Evolutionary Algorithms + Local Search = Memetic Algorithms**

Combining global and local search algorithm used for many hybrid optimization approaches. Memetic algorithm (MAs) are Evolutionary Algorithm (EAs) along with that we apply some sort of local search algorithm to further improve the fitness of individual in the population. Memetic Algorithm have been shown to be very effective in solving many hard combinatorial optimization problems[7]. The approach combines a hierarchical design technique, Genetic Algorithms, constructive algorithms and advance local search to solve the VLSI layout in various steps like partitioning and placement.

The efficient optimization algorithm used to solve hard problems usually employs a hybrid of at least two techniques to find a near optimal solution to problem consider. The main motivation for this hybridisation is to increase the efficiency that is to get the good quality solution in specified time.

The above figure identifies the Memetic algorithm template. The structure is the basic evolutionary algorithm structure the black mark place on the particular block is to mention to hybridisation is take place in that block. Each of the black mark provides a opportunity for hybridisation. For example the initial population could be seed with solutions arising from sophisticated problem for specific heuristics. The cross over and mutation operator could be enhanced with domain specific and representation specific constrain as to provide better search ability to Evolutionary Algorithm. More over local search could be applied to any or all of the intermediate set of solution for example offspring set. The more problem specific knowledge is incorporated with in a Memetic algorithm the better it will perform. The most popular form of hybridisations is to apply one or more phase of local search, based on some probability parameter, to individual members of population in each generation.

**Memetic Algorithm for Circuit Placement**

The placement is the arranging the circuit components in layout. In general-cell and standard-cell placement, we want to position the components of the circuit such that the layout area is minimized. The area measure used here comprises the area taken up by the circuit components as well as the area needed for the wiring the circuit components. The placement problem has the dual flavour of a two dimensional packing problem and connection cost optimization problem. The packing problem is concerned with fitting a number of cells of different sizes and shapes tightly into a rectangle. The connection cost optimization aims at minimizing the amount of wire necessary.

The proposed Memetic Algorithm for circuit placement[7] is based on the Genetic

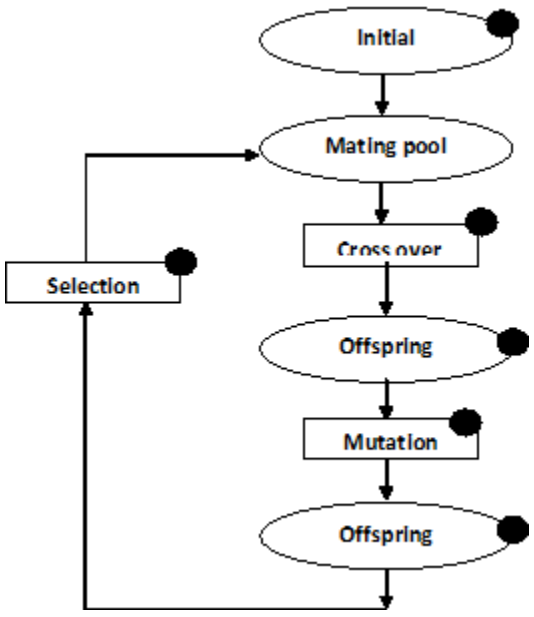


**Fig.1.** Algorithm Template

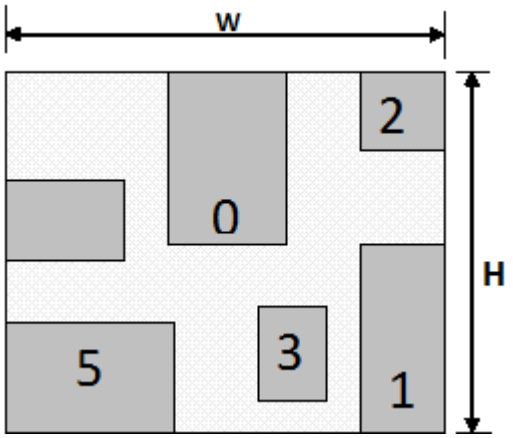


**Fig. 2.** Six block placement

Algorithm[8,9]. In each generation, a Tile-based local search heuristic is performed on part of the population to improve their fitness. The local search may introduced in genetic algorithm is to offspring population, it may be in cross over or else where. For the circuit placement problem, the pure Genetic Algorithm is combined with Tile-based local search in three different ways, referred to as performing local search on part of the population: (i) before the crossover (ii)after the crossover (iii) before and after the crossover[7].

1. Encode solution space for placement
2. Set population size, max-gen, generation=0;
   Set crossover-rate, mutation-rate;
3. Generate the initial population
4. Evaluate the initial population
5. While (generation < max-gen)
Apply genetic algorithm
Apply tile-based local search to population
End while
6. Return the best solution in current population

A Memetic algorithm for the circuit placement is purely based on the Genetic algorithm. From the above figure we can under stand that the memetic algorithm for the circuit placement is start with the initial random population generation before that we have to set the size of the population , maximum number of generation, cross over rate, and mutation rate. Next we apply the Genetic algorithm; the genetic algorithm start with the selection process, the selection process is based on the fitness function. Using this fitness function was selecting the chromosome for the cross over. Cross over is the base point for the next generation population. The cross over technique used in Genetic algorithm is, one point cross over, two point cross over, cut and splice cross over , uniform cross over and half uniform cross over. After the cross over the mutation process, is to maintain the genetic diversity from one generation to next generation. In this mutation the genetic sequence will be changed from its original by generating a random variable for each bit sequence. After the mutation accepting offspring is placed in the new population for further iteration. The next step in memetic algorithm for circuit placement is to apply the local search algorithm in between this genetic algorithm as told before the local search is apply in three ways they are mention below

(i) Before the crossover
(ii) After the crossover
(iii) Before and after the crossover.

**Placement problem description**

**Area Estimation**

Given a set of module $M_1$, $M_2$,…..$M_n$ and set of n interconnects $N_1$,$N_2$……$N_n$ the objective of the placement is to obtain the non overlapping package of all the modules which achieves some optimization objective such as minimizing the area of package[3], the interconnection length as show in the below figure

**Horizontal constrain**

If *(-X,Y) =(. . . a. . .b. . .,a. . . b. . .)* Block *b* is is at right side of the block *a*

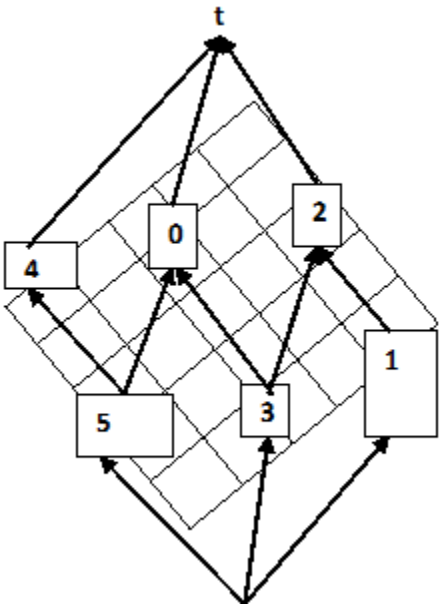


**Fig.3.** Example for vertical constrain graph (VGH)

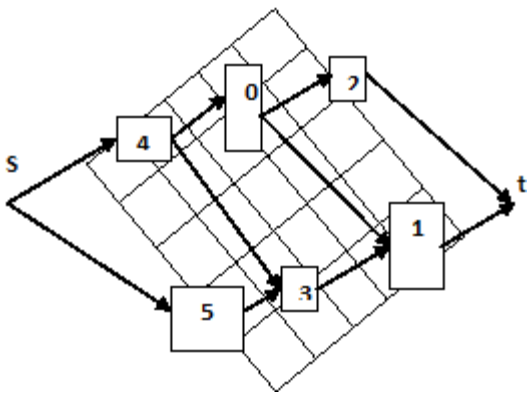


**Fig. 4.** Example for horizontal constrain graph

**Vertical constrain**

If *(X,Y) =(. . . a. . .b. . .,b .. . a. . .)* Block *b* is at the below side of the block *a*

Based on "left of' constraint of (X, Y), a directed and vertex-weighted graph $G_H$ (V, E) (V: vertex set, E: edge set), called the horizontal-constraint graph (HCG) is constructed as follows:

(1) V= $\{S_h\}$ U $\{t_h\}$ U $\{b_i\ i = 1, \ldots, M\}$,

Where $b_i$ correspond to the block

$S_h$ is the source node representing the left boundary

$t_h$ is the target node representing the right boundary

(2) E = $\{(S_h,b_i)\ i = 1,\ldots,M\}$ U $\{(b_i,t_h)\ i = 1,\ldots,M\}$ U $\{(b_i,b_j)\ \ldots\ b_i \ldots\ b_j\}$

If existing, edge $(b_i,b_{i+1})$ edge $(b_{i+1},b_{i+2})$ and edge $(b_i,b_{i+2})$ then $(b_i,b_{i+2})$ omitted

3) Vertex Weight equals the width of the block $b_i$, but zero for $S_h$ and $t_h$ similarly the vertical constrain graph (VGH) as show in the below figure

Vertical constrain graph $G_v(V,E)$ is constructed using "above" constrain and the height of the each block. As for the example show in the above figure. The corresponding constrain graph $G_h(V,E)$ and $G_v(V,E)$ are as show in the figures. Both $G_h(V,E)$ and $G_v(V,E)$ are vertex weighted acyclic graph so longest path algorithm can be applied to find the x and y coordinates of each block. The coordinates of the block coordinate of the lower left corner of the block.

**Wire length estimation**

We are addressing the problem of VLSI standard Cell placement with the objectives of minimizing wire length[2], power consumption, and timing performance (delay), while considering the layout width as a constraint.

**Wire length measurement and Cost**

In VLSI placement the cells present in the module are connected by wire. The estimation of wire length[2] required for connection is calculated by the formula

$$\text{Wire}_{\text{length}} = \text{“}\sum_{I > j} w_{i,j}\left((x_i - x_j)^2 + (y_i - y_j)^2\right)$$

Where

$W_{I,j}$ weight of the connection between cell x and y

$(x_i - x_j)$ distance in X direction

$(y_i - y_j)$ distance in Y direction

Inter connect wire length of each net in the circuit is estimate during Steiner tree and then total wire length is computed by adding the individual estimates

$$\text{Cost}_{\text{wire}} = \text{“}\sum_{i \in M} l_i$$

Where l is the wire length estimation for net i and M denotes total number of nets in circuit.

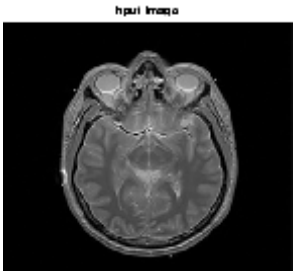


**Fig. 5.** Input image for Ami49 circuit

**Fig. 6.** Out put image for Ami49 circuit


**Fig. 7.** Input image for apte circuit

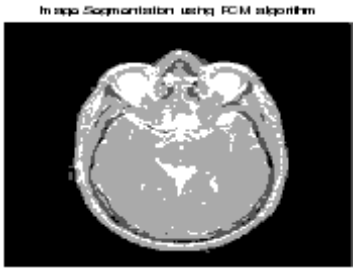


**Fig. 8.** Output image for apte circuit

**Table 1.** Performance statistics for MCNC circuits

| Performance statistics for five MCNC Benchmark circuits | | | | |
|---|---|---|---|---|
| Circuit | Cells | Nets | Final wire length(mm) | Final chip area($mm^2$) |
| apte | 9 | 97 | 590.6 | 61.8 |
| xerox | 10 | 203 | 1038 | 32.6 |
| hp | 11 | 83 | 365 | 42.9 |
| Ami33 | 33 | 123 | 278.5 | 1.23 |
| Ami49 | 49 | 408 | 2077 | ——— |

**Experimental result**

The Memetic algorithm for VLSI cell placement problem was tested on real life circuits chosen from a benchmark suite that for design work shop in early 1990's and it is often referring to as MCNC benchmark. They were originally released and maintain by North Carolina's Microelectronics, computing and networking centre and now it changes to CAD Benchmarking Laboratory (CBL).

All results of our Memetic algorithm presented in this section were obtain by implementation of Memetic algorithm for placement here first the initial population is generate for that evaluate the fitness function based o that fitness selection of parents for the cross over after this process the normal mutation and inversion operation are take place . in addition to this process for each subpopulation local search is apply to refine the fitness of each individual to get the most optimal solution

We implementing our algorithm in c language as (genetic algorithm + local search) , and the experiment was executed on the Intel Pentium processor (3.1GHz, 512 RAM) machine running windows xp . The memetic algorithm (genetic algorithm + local search) is embedded in our algorithm the block placement by implementing our algorithm results are show in below table

## CONCLUSION

The hybrid approach for the combinatorial placement problem is memetic algorithm, this memetic algorithm may also called as hybrid genetic algorithm, here the hybridisation is genetic algorithm with some local search technique. First the memetic algorithm start with the genetic algorithm it find the global minimum solution to further refine the individual and to get the local minimum we introduce a local search

The main future of the approach introduced here, in comparison with other approaches, is the manner in which block flexibility is treated. During the iteration several implementation such as wire length calculation and area estimation are consider and some shape functions are introduced for the block. The common hill climbers in genetic algorithm for combinatorial optimization problem randomly explore the solutions neighbouring the candidate solution encode in the current individual and accept those with better fitness. In hybrid approaches, local search techniques explore the solution space close to the sample points by applying specialized heuristics. When including problem specific knowledge during creation of individual, like in our approach, it is possible to identify unfavourable or redundant partial solutions and consider only the most promising ones. Therefore, each individual in our hybrid genetic algorithms encodes a set of high quality solutions, the best of which is a local optimum.

This paper explored the memetic algorithm strategies for a multi objective VLSI cell placement. Result produced by implementing memetic algorithm is most optimal solution compare to other set of algorithm used for VLSI cell placement. In this paper wire length estimation and area estimation are consider. For future work: width cost, delay cost, power cost and power minimization.

## ACKNOWLEDGEMENT

It is to note here that this topic-specific article for the easy reference of VLSI Placement Problem, optionally being used together with the all the problems in VLSI Physical Design. Author Varatharajan, I thank the management of sri lakshmi ammal engineering college and bharath university,Chennai, India to conduct his research in this area.

## REFERENCES

1. Varatharajan.R" memetic programming approach for floorplanning applications" August 2012 ,*international journal of information engineering and electronic business*, 2012; **4**(4).

2. Chen1, and Yao-WenChang1'2 Tung-Chieh Chen2, Jiang1, Hsin-Chen Zhe-Wei "challenges and solution in modern VLSI placement" 2007 *IEEE* 1-4244-0583-1/07.
3. Sadiq M. Sait, Mustafa Imran Ali, and Ali Mustafa Zaidi "Evaluating Parallel Simulated Evolution Strategies for VLSI Cell Placement" in 1-4244-0054-6/06 ©2006 *IEEE*.
4. Ning Xu,Feng Huang ,Zhonghua Jiang "A Fast Algorithm for VLSI Building Block Placement" in IMACS Multiconference on "Computational Engineering in Systems Applications"(CESA), October 4-6, 2006, Beijing, China.
5. Natalio Krasnogor, and Jim Smith "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy and Design issues" *IEEE Transactionson Evolutionary Computation*, Vol.A, No.B, CCC200D.
6. Bo Hu and Malgorzata Marek-Sadowska "Multi level Fixed-Point-Addition-Based VLSI Placement" *IEEE Transactionson Computer Aided Design Of Integrated Circuits And Systems*, 2005; **24**(8).
7. Ulrich Brenner and Jens Vygen "Legalizing a Placement with Minimum Total Movement" *IEEE Transactionson Computer Aided Design Of Integrated Circuits And systems*, 2004; **23**(12).
8. Shawki Areibi and Zhen Yang "Effective Memetic Algorithms for VLSI Design = Genetic Algorithms+ Local Search+ Multi-Level Clustering" in 2003 by the Massachusetts Institute of Technology Evolutionary Computation.
9. Fred Glover and Gary A. Kochenberger, Handbook of Metaheuristics, kluwer academic publishers, New York, Boston, Dordrecht, London, Moscow @ 2003.
10. Elizabeth M.Rudick, Pinaki Mazumder "Genetic Algorithm for VLSI Design Layout and Test Automation" Pearson Edition @ 1999.
11. Sabih H.Gerez "Algorithm for VLSI Design Automation" Jhon Wiley student Edition @ 1998.
12. Volker Schnecke and Oliver Vornberger "Hybrid Genetic Algorithms for Constrained Placement Problems" in IEEE 1998, appeared in *IEEE Transactions on Evolutionary Computation*. 1997; **1**(4): pp.266–277.
13. K. Shahookar and P. Mazumder, Department Of Electrical Engineering And Computer Science, University Of Michigan, Ann Arbor, Michigan 48109, Vlsi Cell Placement Techniques, *ACM Computing Surveys*, 1991; **23**(2).