Solving Standard Cell Placement Problem Using an Evolutionary Approach

Jobanpreet Kaur
Thapar University, Patiala, India

Maninder Kaur
Assistant Professor, Thapar University, Patiala, India

ABSTRACT
The paper presents an evolutionary algorithm for solving standard cell placement problem**(EASCP)**. The proposed algorithm follows a variant of genetic paradigm in which the population candidates after limited trials are replaced with the fresh ones. The algorithm explores the solution space and memorizes the best solutions of every generation thus exploiting the best solutions. The experimental results present a layout with better wirelength by the EASCP in comparison to the basic Genetic Algorithm (GA).

Keywords
Standard Cell Placement, Evolutionary Approach, Genetic Algorithm, Crossover

1. INTRODUCTION
In the present era, the designs of VLSI chips have become more complexand the increase in complexity of the chips brings a big challenge to the phases of VLSI design cycle. The VLSI design flow begins with the specification of a circuit, followed by various phases (functional design, logical design, circuit design, physical design and fabrication) [1]. The physical design stage includes partitioning, placement, routing and compaction. The emphasis of the proposed work is on the standard cell placement. The placement problem is to place the cell modules on a fixed size chip such that no modules overlap with each other and certain cost metrics are optimized. For a given circuit consisting of a set of cell modules and a net list giving the interconnections between these modules, the standard cell placement problem is to arrange a layout that would indicate the positions of the modules in parallel rows such that all the nets are interconnected using wires and the total layout area is minimized.

Although there are a number of objectives in standard cell placement problem such as routability, low power, time delay, performance, the proposed research work is focused on the main objective i.e. minimizing wire length.Placement is a very major stage in physical design process that has been widely studied by the researchers for many decades [2], [3], [4], [5],[6]. Due to large scale designs with millions of transistors on a chip that makes the layout more complicated, the modern placement problem has become very complex. Therefore, efficient and effective placement techniques are required to meet the fast-paced nature of VLSI CAD design.
Various metaheuristic algorithms were devised by the researchers to solve this NP hard problem. Some of the algorithms include genetic algorithms [9], simulated annealing [8], tabu search [7] and various hybrid algorithms [10] [11]. Genetic Algorithms are one of the efficient algorithms for cell placement. These algorithms work on a population of solutions called chromosomes that represent a solution to the problem. There is an objective function called a fitness function thatevaluates each individual's fitness value that would measure the goodness of the solution. In every single iteration, the genetic operators namely selection, crossover, inversion and mutation are applied that breeds a population of individuals. A new generation is evolved from the existing population with new solutions. The fitness of the populations gets improved over a number of generations [12]. Genetic Algorithms are multi-point heuristics that are less likely to get stuck in local optima than most other optimization techniques [13].

2. PROBLEM FORMULATION
Given circuit consisting of standard cell modules and a netlist interconnecting the terminals of these cells, the Standard Cell Placement problem is arranging a layout indicating the positions of the modules, such that the estimated wire length and the layout area are minimized and other given constraints are satisfied. The inputs to the problem include module description with sizes and terminal locations and the netlist describing the interconnections between the cells. The output list contains a list of x- and y- coordinates of thecell modules [14].
Graph theory can be successfully used to model VLSI physical design problems including Standard Cell Placement [1]. A circuit is represented by a hypergraph G (V, E) where the vertex set V = $\{v_1, v_2,\ldots,v_n\}$ represents the set of cells to be placed and the edge set E = $\{e_1,e_2,\ldots,e_n\}$ represents the set of nets connecting the cells. Each edge $e_j$is an ordered pair of vertices with a non-negative weight $w_j$assigned to it. The placement problem is assigning all cells of the circuit to the locations in the chip such that no two cells overlap. Each cell *i*is assigned to a location (x_i, y_i) on a 2-D plane. Minimizing the wire-length is approximately equivalent to minimizing the total chip area for the standard cell layout [9],therefore, the total cost of a placement layout, denoted by F (x, y), can be estimated by the sum of wire length over all nets [15].

$$\emptyset(x,y) = \sum_{1\le i<j\le N} w_{ij}[(x_i - x_j)^2 + (y_i - y_j)^2]$$

where (x_i , y_i) denotes the location of cell *i*; $w_{ij}$is a non-negative weight of the edge connecting cell *i*and cell *j*. The above formulation can be rewritten in matrix form as:

$$\emptyset(x,y) = \frac{1}{2}x^T Cx + d_x^T x + \frac{1}{2}y^T Cy + d_y^T y + t$$

Vectors *x* and *y* represent the coordinates of the N cells; matrix *C* is the Hessian matrix; vectors $d_x^T$and $d_y^T$and the constant term *t* result from the contributions of the fixed cells.The solutions are evaluated using the fitness function F:

$$F = \frac{1}{\sum_{i=1}^{n} HPWL_i}$$

where$HPWL_i$is the estimated wire-length of the net *i*and *n* is the number of nets. HPWL is Half Perimeter Wire Length method used to approximately estimate the total wire length since detailed routing is not available at the placement stage.

3. THE EASCP ALGORITHM

The work proposes a variant of an evolutionary approach for solving standard cell placement(EASCP) problem where a trial value is associated with every solution in the population. The trial value of an individual is modified based on its capability to generate a better offspring. The following describes basic components of the algorithm.
In the given circuit, let N be the total number of cell modules each having a defined cell width, channel height and cell height that are to be placed on a fixed size chip. Let Nnets define the total number of nets in the circuit such that a net weight is associated with every net. The output will include an arrangement of the cell modules giving the individual cell positions(x and y co-ordinates). For the algorithm, P defines the total number of population (even number of population) in a single generation (gen) such that the number of generations cannot exceed the maximum number of generations i.e. MAXGEN.

Pseudo code for the algorithm:

*Step 1.*Randomly generate an initial population of size P with a set of feasible solutions.
*Step 2.*Read the input files (netlist files and the cell library files) and assign the parameter values to the variables. The cell positions for the initial population are assigned by considering the parameters of cell library files.
*Step 3.*Calculate the fitness value of each solution in the population using the Half Perimeter Wirelength (HPWL) method since detailed routing is not available at the placement stage. The perimeter (p_i) of a net *i* is calculated using the sum of horizontal ($span_{xi}$) and vertical spans ($span_{yi}$) of net *i*'s bounding box [15]. The fitness function is the inverse wirelength. More the fitness value less is the wirelength. The fitness value is maximized in the algorithm.

$$\sum_{i \in Nets} HPWL_i = \sum_{i \in Nets} \frac{p_i}{2} = \sum_{i \in Nets} [span_{xi} + span_{yi}]$$

$$F = \frac{1}{HPWL_i}$$

*Step 4.*Pair the individual chromosomes in the population with each other such that order crossover operation is applied on random points to generate two offsprings.
Step 5. Mutate the generated offsprings by inverting the cell positions at random points. The crossover and mutation operations are applied on the whole population at once to generate a new population from the existing population.
*Step 6.*Assign the fitness value and the cell positions for the generated offsprings using the parameters cell height and cell width. The two most fitted individuals among the parent and the child solutions make up to the population whereas the rest two are discarded.
Step 7. At the end of every generation, the best solution in the generation is memorized and stored in the repository and before incrementing the generation, the check for duplicate individuals in the population is carried out. In case of duplicity, solutions are randomly generated iteratively until population is filled with distinct solutions.
Step 8. After a population is generated, the trial values will be changed for a set of solutions. If a parent solution produces an offspring with a worse fitness value, its trial value will be incremented by 1 and the parent solution will become the part of the next generation population. For the vice-versa case, the trial value will remain unchanged giving a chance to the child solution to be inserted in the next population.
Step 9. After every generation, the trial value of the individuals will be evaluated. If the trial value of a particular individual exceeds a predefined limit, then that individual is discarded from the population and is replaced by a randomly generated distinct solution in the population.
Step10. After a repeated set of generations, the best solution amongst the best solutions of every generation is chosen as the final solution from the repository.

4. EXPERIMENTAL RESULTS

The algorithm is tested on sample circuits.The work is carried out by writing code in Turbo C on Windows 7 platform and running on an Intel(R) Core(TM)2 Duo (2.10 GHz) machine with 2 GB memory, using crossover rate,Rc = 0.66, mutation rate, Rm = 0.01 and population size 10. The simulation was carried on six test circuits with a maximum nodes of 15 and maximum nets 13. The results of the experiments are reported in Table 1. The optimum values of the fitness values (inverse of wirelength) in table 1 shows a variation between the two algorithms. The EASCP shows better results in terms of the quality of solutions. The CPU time for the two algorithms is comparable for circuits having larger nodes. The decrease in time is due to the reason that the individuals with limited trial values are discarded and are not explored repeatedly. The comparison of the performance of the two algorithms in terms of fitness values is shown in the form of column chart in figure 1. Figure 1 depicts clearly that the EASCP gives better average wirelength in comparison with the basic GA producing better placement results but the CPU time for EASCP is comparable for circuits with large number of nodes.

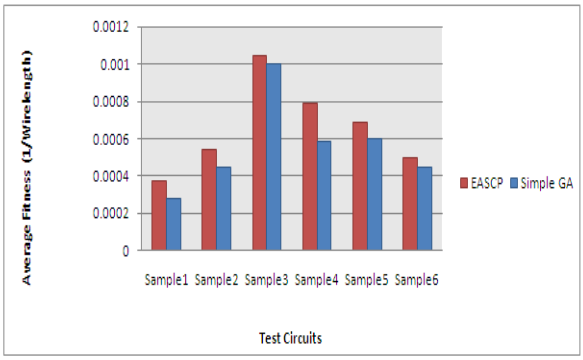

Fig 1: Comparative Results of Simple GA and EASCP Based on Fitness Value (1/Wirelength)

5. CONCLUSIONS

The work presents an approach for standard cell placement problem combining features of evolutionary approach with the limited trials of the solutions that further breeds the next population based on the trial

values, hence avoiding the algorithm getting trapped in local convergence and also exploiting the solutions of every generations. The EASCP produces better results in comparison to the basic GA in terms of the wirelength when experimented on a set of test circuits.

The authors are very thankful to Dr. Kanwaljeet Singh, Punjabi University, Patiala,without whose knowledge and assistance, this study would have been unsuccessful. The authors also extend deep gratitude towards Professor Hardeep Singh for his valuable support and guidance.

6. ACKNOWLEDGEMENTS

Table 1: Comparison of Basic genetic Algorithm and the Proposed Evolutionary Algorithm based on the minimum and average fitness and CPU time (in seconds)

Circuit	No. of Nodes	No. of Nets	Basic Genetic Algorithm			Proposed Evolutionary Algorithm		
			Maximum Fitness	Average Fitness	CPU	Maximum Fitness	Average Fitness	CPU
Sample1	10	10	0.000315	0.000275	0.22	0.000560	0.000370	0.28
Sample2	14	10	0.000564	0.000448	0.20	0.000782	0.000540	0.21
Sample3	12	7	0.001091	0.001000	0.17	0.001182	0.001046	0.19
Sample4	15	13	0.000724	0.000587	0.26	0.000965	0.000787	0.26
Sample5	10	6	0.000659	0.000597	0.25	0.000861	0.000688	0.27
Sample6	12	10	0.000491	0.000443	0.18	0.000516	0.000494	0.21

7. REFERENCES

[1] Sherwani, Naveed A. Algorithms for VLSI Physcial Design Automation. Kluwer Academic Publishers, 1998.
[2] Pan, Min, NatarajanViswanathan, and Chris Chu. "An efficient and effective detailed placement algorithm."Computer-Aided Design, 2005.ICCAD-2005.IEEE/ACM International Conference on.IEEE, 2005.
[3] Huang, Dennis J-H., and Andrew B. Kahng. "Partitioning-based standard-cell global placement with an exact objective."Proceedings of the 1997 international symposium on Physical design.ACM, 1997.
[4] Hur, Sung Woo, and John Lillis. "Mongrel: hybrid techniques for standard cell placement." Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design. IEEE Press, 2000.
[5] Kim, Myung-Chul, Dong-Jin Lee, and Igor L. Markov. "SimPL: An effective placement algorithm." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 31.1 (2012): 50-60.
[6]Cho, Hwan Gue, and C. M. Kyung. "A heuristic standard cell placement algorithm using constrained multistage graph model."Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 7.11 (1988): 1205-1214.
[7]Sait, Sadiq M., Habib Youssef, Aiman H. El-Maleh, and Mahmood R. Minhas. "Iterative heuristics for multiobjective VLSI standard cell placement." In Neural Networks, 2001.Proceedings.IJCNN'01. International Joint Conference on, vol. 3, pp. 2224-2229. IEEE, 2001.
[8] Sun, Wern-Jieh, and Carl Sechen. "Efficient and effective placement for very large circuits."Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 14.3 (1995): 349-359.
[9] Shahookar, Khushro, and Pinaki Mazumder. "Gasp: a genetic algorithm for standard cell placement." Proceedings of the conference on European design automation. IEEE Computer Society Press, 1990.
[10] Minhas, Mahmood R. "An evolutionary algorithm for low power VLSI cell placement." Circuits and Systems, 2003 IEEE 46th Midwest Symposium on.Vol. 3.IEEE, 2003.
[11] Bunglowala, Aaquil, B. M. Singhi, and Ajay Verma. "Optimization of Hybrid and Local Search Algorithms for Standard Cell Placement in VLSI Design."Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09.International Conference on.IEEE, 2009.
[12]Goldberg, D. E. "'Genetic Algorithms in Search, Optimization & Machine Learning', Addison- Wesley Publishing Company, Inc., 1989."
[13] Reeves, Colin R., and Jonathan E. Rowe. Genetic algorithms-principles and perspectives: a guide to GA theory. Vol. 20.Kluwer Academic Pub, 2002.
[14] Mazumder, Pinaki, and Elizabeth M. Rudnick. Genetic algorithms for VLSI design, layout & test automation. Prentice Hall PTR, 1999.
[15] University of Guelph. School of Engineering, and Zhen Yang. Area/congestion-driven Placement for VLSI Circuit Layout.University of Guelph, 2003.
[16] Manik Sharma. "Role and Working of Genetic Algorithm in Computer Science". International Journal of Computer Applications and Information Technology (IJCAIT). Vol. II, Issue I, 2013