I.J. Information Engineering and Electronic Business, 2012, 4, 39-45
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2012.04.06

Memetic Programming Approach for Floorplanning Applications

Dr. R. Varatharajan,
Department of ECE S.M.K Fomra inst. Of tech Chennai.
varathu21@yahoo.com
Muthu Senthil
Department of CSE S.M.K Fomra inst.tech Chennai.
bmssen@gmail.com
Dr. Perumal sankar
Research Director Cape Institute of Tech Kanyakumari
spsankar2004@yahoo.com

Abstract **—** Floorplanning is a very crucial step in modern VLSI design. It dominates the top level spatial structure of a chip and initially optimizes the interconnections. Thus a good floorplan solution among circuit modules definitely has a positive impact on the placement, Routing and even manufacturing. In this paper the classical floorplanning that usually handles only block packing to minimize silicon rate, so modern floorplanning could be formulated as a fixed outline floorplanning. It uses some algorithms such as B-TREE representation, simulated annealing and adaptive fast simulated annealing, comparing above three algorithms the better efficient solution came from adaptive fast simulated annealing, it's leads to faster and more stable convergence to the desired floorplan solutions, but the results are not an optimal solution, to get an optimal solution it's necessary to choose effective algorithm. Combining global and local search is a strategy used by many optimization approaches. Memetic algorithm is an evolutionary algorithm that includes one or more local search phases within its evolutionary cycle. The algorithm combines a hierarchical design technique, genetic algorithms, constructive techniques and advanced local search to solve VLSI floorplanning problem.

Index terms **—** Floorplan problem, memetic algorithm, local search, area, delay, layout optimization

I. INTRODUCTION

Floorplanning has become a very crucial step in modern very large scale integration (VLSI) designs. As the start of physical design flow, floorplanning not only determines the top-level spatial structure of a chip, but also initially optimizes the interconnections. Thus, a good floorplan solution among circuit modules definitely has a positive impact on the placement, routing, and even manufacturing. In the nanometer scale era, the ever-increasing complexity of integrated circuits (ICs) promotes the prevalence of hierarchical design. However, as pointed out by Kahng [1], classical outline-free floorplanning [2] cannot satisfy such requirements of modern designs. In contrast with this, fixed-outline floorplanning enabling the hierarchical framework is preferred by modern application specific integrated circuit designs. Nevertheless, fixed-outline floorplanning has been shown to be much more difficult, compared with classical outline-free floorplanning, even without considering wirelength optimization [7].

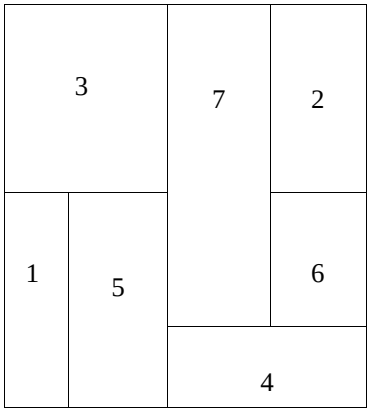
A common strategy for blocks floorplanning is to determine in the first phase and then the relative location of the blocks to each other based on connection-cost criteria. In the second step, block sizing is performed with the goal of minimizing the overall chip area and the location of each block is finalized [1]. Simulated annealing (SA) has been considered a good tool for complex nonlinear optimization problems. The technique has been widely

applied to a variety of problems. As an optimization technique, Genetic Algorithms simultaneously examine and manipulate a set of possible solutions. The power of GA's comes from the fact that the technique is robust, and can deal successfully with a wide range of problem areas, including those which are difficult for other methods to solve.

Genetic Algorithms are not well suited for fine-tuning structures which are close to optimal solutions. Incorporation of local improvement operators into the recombination step of a Genetic Algorithm is essential if a competitive Genetic Algorithm is desired. MAs are evolutionary algorithms (EAs) that apply a separate local search process to refine individuals (i.e.) improve their fitness by hill-climbing. Under different contexts and situations, MAs are also known as hybrid EAs, genetic local searchers. Combining global and local search is a strategy used by many successful global optimization approaches, and MAs have in fact been recognized as a powerful algorithmic paradigm for evolutionary computing. In particular, the relative advantage of MAs over EAs is quite consistent on complex search spaces.

II.PROBLEM FORMULATION

Generally the floorplanning problems are such as size, Chip area, and total wire length, delay of critical path, routability, noise, and heat dissipation. The modern floorplanning typically needs to pack blocks within a fixed die (outline) and additionally consider the packing with block positions as well as the interconnect constraints. The modern floorplanning problem is categories as Fixed-outline floorplanning. A module B is a rectangle of height hB, width wB, and area AB. A super-module consists of several modules, also called a sub-floorplan. A floorplan for n modules consists of an enveloping rectangle R subdivided by horizontal lines and vertical lines into n non-overlapping rectangles such that each rectangle must be large enough to accommodate the module assigned to it. In the given problem, we are given a set of hard modules and an outline-constraint is provided. The modules in the given Fixed-Outline (denoted as FO) have freedom to move while the modules outside the FO are infeasible in the final floorplan. A feasible packing is a packing in the first quadrant such that all the modules inside FO are not duplicate and overlapping. The objective is to construct a feasible floorplan R such that the total area of the floorplan R is minimized and simultaneously satisfy fixed-outline constraint. A slicing floorplan is represented by the slicing structure which can be obtained by recursively cutting a rectangle into two parts by either a vertical line or a horizontal line. As shown in Fig.2,a slicing floorplan can be represented as a slicing tree, that is, every leaf corresponds to a basic module and is marked by a number from 1 to n, and every internal node is labeled by a + or a *, corresponding to a horizontal or a vertical cut, respectively. Traversing the slicing tree in post-order, we obtain a Polish expression of length 2n - 1 for the slicing floorplan. A wheel is a non-slicing floorplan of five modules, which cannot be obtained by recursively cutting a rectangle into two parts by either a vertical line or a horizontal line Fig.3. Although slicing floorplans can be sub-optimal if compared to general floorplans, empirical evidence shows that slicing floorplans can be quite efficient in packing modules tightly. It has been proved mathematically and it is achieved for packing slicing floorplans tightly [2].



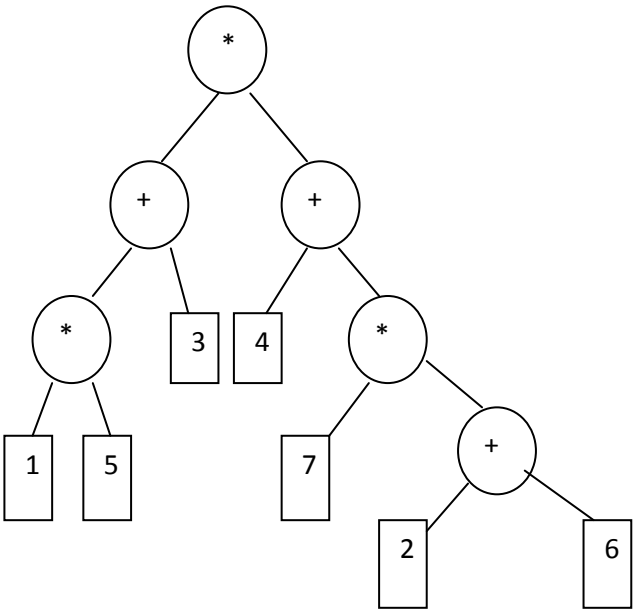


Fig 1.Polish expression

III.METHODOLOGY

3.1 Genetic algorithm approach

The genetic algorithm is based upon Darwinian evolution theory. The genetic algorithm is modeled on a relatively simple interpretation of the evolutionary process; however, it has proven to a reliable and powerful optimization technique in a wide variety of applications. Holland [17] in 1975 was first proposed the use of genetic algorithms for problem solving. Goldberg [8] was also pioneers in the area of applying genetic processes to optimization. As an optimization technique, genetic algorithm simultaneously examines and manipulates a set of possible solution. Over the past twenty years numerous application and adaptation of genetic algorithms have appeared in the literature. During each iteration of the algorithm, the processes of selection, reproduction and mutation each take place in order to produce the next generation of solution. Genetic Algorithm begins with a randomly selected population of chromosomes represented by strings. The GA uses the current population of strings to create a new population such that the strings in the new generation are on average better than those in current population (the selection depends on their fitness value). The selection process determines which string in the current will be used to create the next generation. The crossover process determines the actual form of the string in the next generation. Here two of the selected parents are paired. A fixed small mutation probability is set at the start of the algorithm. This crossover and mutation processes ensures that the GA can explore new features that may not be in the population yet. It makes the entire search space reachable, despite the finite population size.

1. Encode solution space

2. (a) Set pop_size, max_gen, gen=0
   (b) set cross_rate, mutate_rate;

3. initialize population

4. while max_gen ≥ gen
   evaluate fitness
   for (i=1 to pop_size)
   select (mate1,mate2)
   if (rnd(0,1) ≤ cross_rate)
   child = crossover(mate1,mate2)
   if (rnd(0,1) ≤ mutate_rate)
   child = mutation();
   repair child if necessary
   end for
   add offspring to new generation
   Gen=gen+1
   End while

5. return best chromosomes

Fig 2.Genetic Algorithm

3.2 Memetic algorithm approach

The genetic algorithm is not well suited for fine-tuning structures which are close to optimal solution [7]. The memetic algorithms [15] can be viewed as a marriage between a population-based global technique and a local search made by each of the individuals. They are a special kind of genetic algorithms with a local hill climbing. Like genetic algorithms, memetic Algorithms are a population-based approach. They have shown that they are orders of magnitude faster

than traditional genetic Algorithms for some problem domains. In a memetic algorithm the population is initialized at random or using a heuristic. Then, each individual makes local search to improve its fitness. To form a new population for the next generation, higher quality individuals are selected. The selection phase is identical inform to that used in the classical genetic algorithm selection phase. Once two parents have been selected, their chromosomes are combined and the classical operators of crossover are applied to generate new individuals. The latter are enhanced using a local search technique. The role of local search in memetic algorithms is to locate the local optimum more efficiently then the genetic algorithm. Figure 3 explains the generic implementation of memetic algorithm.

1. Encode solution space

2. (a) set pop_size, max_gen, gen=0;
   (b) set cross_rate, mutate_rate;

3. initialize population

4. while(gen < gensize)
   apply generic GA
   apply local search
   end while
   apply final local search to best chromosome

Fig 3.Memetic Algorithm

3.3 Crosscover Operator

The Traditional crossover operator used in GA may produce infeasible solutions for the standard cell placement problem, therefore a crossover operator called Order crossover is considered. Partially matched crossover (PMX) may be viewed as a crossover of permutations that guarantees that all positions are found exactly once in each offspring, i.e. both offspring receive a full complement of genes, followed by the corresponding filling in of alleles from their parents.

3.4 Mutation Operator

Each offspring is mutated with a probability equal to the mutation rate. The mutation operator mutates an individual by interchanging randomly selected pair of cells without changing the x-coordinate and row number.

3.5 Local search

In this work we have hybridized the genetic algorithm template with the SA method. The SA method is impregnated within GA, between the crossover and mutation operations, to improve all the solutions obtained after the crossover operation and before subjected to mutation operation.

3.6 Simulated Annealing

SA is very simple technique for State Space Search Problem. It can start from any state. And it is always move to a neighbor with the min cost (assume minimization problem). It can stop when all neighbors have a higher cost than the current state.

IV.MULTITHREADING

Multithreading (MT) is a technique that allows one program to do multiple tasks concurrently. The basic concept of multithreaded programming has existed in research and development labs for several decades. Co-routine systems such as Concurrent Pascal and InterLisp's Spaghetti stacks were in use in the mid-70s and dealt with many of the same issues. Ada's tasks are a language based construct that maps directly onto threads (so directly, in fact, that current. Ada compilers implement tasks with threads). Burroughs shipped a commercial mainframe OS with co-routine style threads as early as 1960. The threading models we describe are strictly software models that can be implemented on any general-purpose hardware. Much research is directed toward creating better hardware that would be uniquely suited for threaded programming

V.EXPERIMENTAL RESULT

To test the effectiveness of proposed Fixed-outline floorplanning algorithm, set the maximum percentages of dead space to 15% and10%.The expected aspect ratios R * of the floorplans are chosen from the range with interval 0.5. Experiments were performed on a 1.6-GHz Intel Pentium4 PC using the GSRC benchmark circuit n100.The results were averaged for 50 runs for each aspect ratio. We compared with FASA based on the same platform. We have tested MA with polish expression floorplan representations, polish Table I lists the average success rates for FASA and the MA. Proposed method obtained 100% success rates of fitting into the given fixed outlines for all runs with dead space Γ =15% and Γ=10%.Incontrast, the success rates when Γ =10% for MA,FASA were 30.3%, 65.5%, and 99.4% respectively. The dramatic differences reveal the effectiveness of our approach. Also, this proposed method required the least running time on average.

TABLE I COMPARISON OF WIRELENGTH UNDER FIXED-OUTLINE CONSTRAINT FOR n100, n200, & n300 WITH ASPECT RATIO R=1, 2, 3, 4

Circuit	Aspect ratio	Fast –sa		Memetic	
		Wire (mm)	Time (Sec)	Wire (mm)	Time (sec)
n 100	1	33.40	30	30.06	24
	2	34.45	30	34.40	24
	3	36.48	31	35.40	26
	4	36.90	31	32.74	27
n 200	1	63.50	174	59.34	148
	2	62.74	172	59.84	154
	3	63.34	178	61.54	155
	4	66.31	180	63.72	161
n 300	1	77.05	399	71.23	361
	2	77.50	385	73.45	358
	3	81.66	390	79.87	371
	4	88.47	394	83.16	372
Comparison		1.04	1.11	0.99	1.0

VI.CONCLUSION

The proposed algorithm for modern Floorplanning problems with Fixed-outline is based on the new Memetic algorithm. Experimental results have shown that MA leads to faster and stable convergence to desired Floorplan solutions. For fixed-outline floorplanning, the new cost function considering the aspect-ratio penalty drives MA more efficiently to find floorplans inside the given chip outline. The experimental results on the fixed-outline Floorplanning have shown the efficiency and effectiveness of our Floorplanning algorithms; for those applications, our results outperform the related recent Works by large margins. Research along this direction is ongoing.

REFERENCES

[1] Hameem shanavas, "Evolutionary algorithmical approach for vlsi floorplanning problem" in *international journal of computer theory and engineering*,2009.

[2] A. B. Kahng, "Classical floorplanning harmful?," in *Proc. Int. Symp.Phys. Design*, 2000, pp. 207–213.

[3] R. H. J. M. Otten, "Efficient floorplan optimization," in *Proc. Int. Conf.Comput. Design*, 1983, pp. 499–502.

[4] S. N. Adya and I. L. Markov, "Fixed-outline floorplanning through better local search," in *Proc. Int. Conf. Comput. Design*, 2001, pp. 328–334.

[5] H. H. Chan, S. N. Adya, and I. L. Markov, "Are floorplan representations important in digital design?," in *Proc. ACM Int. Symp. Physical Design*, San Francisco, CA, Apr. 2005, pp. 129–136.

[6]Jackey Z,yan and chris chu " Deferred decision making enabled fixed outline floorplanning algorithm*"IEEE Transcation on computer aided design of integrated circuits and systems*,march 2010

[7]B. Kahng, "Classical floorplanning harmful?," in *Proc. ACM Int. Symp.Physical Design*, San Diego, CA, Apr. 2000, pp. 207–213.

[8]S. N. Adya and I. L. Markov, "Fixed- outline floorplanning through better local search," *in Proc. IEEE Int. Conf. Computer Design, Austin,*TX, 2001, pp. 328–334.

[9] .Chang-Tzu Lin, De Sheng Chen, Yiwen.Wang, "An Efficient Genetic Algorithm for Slicing Floorplan Area Optimization"*Proceedings of the International Symposium on Circuits And Systems*,pp. 879-882, 2002.

[10] Changq Tzu Lin, De Sheng Chen, Yiwen Wangs, Hsin-Hsien Ho "Modern Floorplanning with Abutment and Fixed-Outline Constraints", *Proceedings of the International Symposium on Circuits and Systems*, pp. 879-882, 2002.

[11] S. Kirpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science, vol.* 220, no. 4598, pp. 671–680, May 1983.

[12] J. H. Y. Law and E. F. Y. Young, "Multi-bend bus driven floorplanning," *in Proc. ACM Int. Symp. Physical Design*, San Francisco, CA, Apr. 2005,pp. 113–120.

[13] C.-T. Lin, D.-S. Chen, and Y.-W. Wang, "Robust fixed-outline floorplanning through evolutionary search," *in Proc. IEEE/ACM Asia andSouth Pacific Design Automation Conf.*, Yokohama, Japan, Jan. 2004,pp. 42–44.

[14] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectanglepacking based module placement," *in Proc.IEEE/ACM Int. Conf.Computer-Aided Design*, San Jose, CA, Nov. 1995, pp. 472–479.

[15] *Parquet: Fixed-Outline Floorplanner*. [Online]. Available: http://vlsicad.eecs.umich.edu/BK/parquet/

[16] F. Rafiq, M. Chrzanowska-Jeske, H. H. Yang, and N. Sherwani, "Busbased integrated floorplanning*," in Proc. IEEE Int. Symp. Circuits and Systems*, Phoenix-Scottsdale, AZ, May 2002, pp. 875–878.

[17] S. M. Sait and H. Youssef, *VLSI Physical Design Automation: Theory and Practice*. Singapore: World Scientific, 1999.

[18] C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf placement and routing package*," IEEE J. Solid-State Circuits, vol. SSC*-20, no. 2,pp. 510–522, Apr. 1985.

[19] Stephen Coe, Shawki Areibi, Medhat Moussa "A Hardware Memetic Accelerator for VLSI Circuit Partitioning" *University of Guelph,School of Engineering*, Guelph, Canada, 2003.

[20] T.C.Chen andY.W.Chang, "Modern floor planning based on fast simulated annealing", *in Procz ACM Int Symp Physical Design San Francisco, CA*, Apr.2005, and pp 104–112.

[21] .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, March 2005.

[22] Natalio Krasnogor, Alberto Aragon and Joaquin Pacheco. "MEMETIC ALGORITHMS", School of Computer Science and I.T. University of Nottingham. England 2005.

[23] Tung-Chieh Chen, Student Member IEEE, and Yao Wen Chang,"Modern floor planning based on B-tree and fast simulated annealing"*Member IEEE transactions on computer-aided design of integrated circuits and systems*, vol.25, April 2006.

[24] Maolin Tang, Member IEEE, and XinYao, Fellow IEEE "A Memetic Algorithm for VLSI Floor planning". *IEEE transactions on systems, man, and cyber net*, vol.37, no.1, february2007.

[25] The MCNC Benchmark Problems for VLSI Floorplanning. [Online]. Available: http://www.mcnc.org.

Dr.R.Varatharajan, is an Associate professor in SMK FOMRA institute of technology, Chennai. He has completed his Bachelor Degree in Electronics and Communication (2006), Masters in VLSI Design (2008), Ph.D in Bharath University (2011),Chennai, India. He has published many journals and attended many Conferences in National and International Level. His research area is VLSI Physical Design.

Dr.S.Perumal Sankar, is the research director of Research Director Cape institute of technology. He has completed his PhD from Anna University, Chennai. He has published more than 20 journals in International Level. He is the Chief Coordinator for many Government Projects. His research areas are Optical MEMS, Digital image processing, Communication Systems.

