

A Genetic Approach to Standard Cell placement using Various Genetic Operators

Rini Mahajan
Lecturer CSE
I.E.T. Bhaddal, Ropar

Amit Saxena
Scientist "SC"
SCL Mohali

Baljit Singh Khehra
Ass. Prof. CSE
BBSBEC Fatehgarh Sahib

ABSTRACT

Genetic algorithm (GA) is a powerful optimization algorithm, which starts with an initial set of random configurations and uses a process similar to biological evolution to improve upon them [1]. As we know that there is a progression towards miniaturization. This paper describes the Genetic Algorithm for standard cell placement on a VLSI Chip for minimization of chip area. Unlike the other placement algorithms that apply transformations on the physical layout, the genetic algorithm applies transformations on the chromosomal representation of the physical layout.

Categories and Subject Descriptors

Optimization Technique-Genetic Algorithm – every Genetic Algorithm uses initial population. Components of initial population are called parents. Genetic operators like crossover, mutation are applied on the parents and new generation is created whose components are called offspring (child).

General Terms

Algorithms, Performance, Design, Documentation.

Keywords

Population, Genes, chromosomes, crossover, mutation

1. INTRODUCTION

1.1 Introduction to GA

Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated; multiple individuals are selected from the current population based on their fitness, and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum

number of generations has been produced, or a satisfactory fitness level has been reached for the population. [3].

The Genetic algorithm operates in a number of iterations, in an attempt to improve upon the trial solutions. In each iteration or "generation", several probabilistic operators are applied to the trial solutions with a view to creating better solutions. In each *generariorz*, the individuals in the current population are *evaluated* using some measure of fitness. A number of genetic operators are applied to the parents to generate new individuals, called offspring; by combining the features of both parents. The three genetic operators commonly used by the genetic algorithm are crossover, mutation, and inversion. The offspring are next evaluated, and a new generation is formed by selecting some of the parents and offspring and rejecting others so as to keep the population size constant.

1.2 Introduction to VLSI Design

VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas. The design of VLSI (very large scale integration) is a process of making consecutive specification, functional design, physical design, and fabrication. Macro-cell layout generation is a task in the physical design cycle. The circuit is partitioned and the components are grouped in functional units, the macro-cells. These cells can be described as rectangular blocks with terminals (pins) along their borders. These terminals have to be connected by single nets, along which power or signals (e.g. clock ticks) are transmitted between the various units of the chip. A net can connect two or more terminals, and some nets must be routed to pads at the outer border of the layout, since they are involved in the I/O of the chip. The layout defines the positions of the cell [4].

When VLSI Design is performed, some factors play very important role that are Floor planning, Placement, Area, speed, Power Consumption.

Floor planning -It is easy to deal with layout when structural details at the lowest abstraction are available, one knows the exact number of transistors in the circuit and the way they are interconnected. When this structural information is not available one can estimate the area to be occupied by various sub blocks and together with a precise or estimated interconnection pattern, and

try to allocate distinct region of integrated circuit to specific sub blocks. This process is called floor planning.

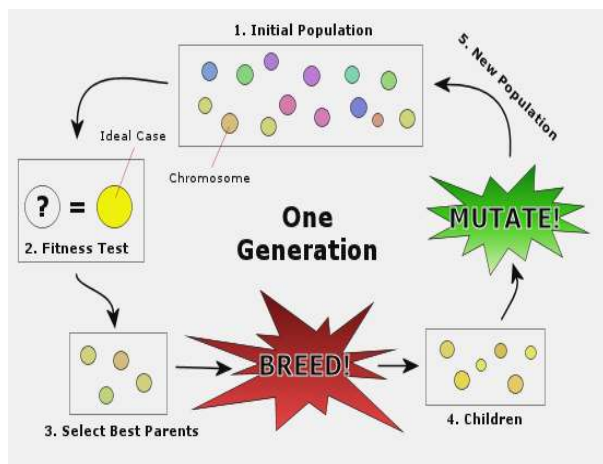
Placement - Placement problem is a problem of determining the relative position of each cell within the target cell architecture.

Area, Speed and Power Consumption – Speed and power consumption is related to the area. So as we are able to accommodate the components into less area, it will increase the speed and reduce the power consumption of the device.

The major concern of this paper is chip area minimization. Unlike the other placement algorithm that applies transformations on the physical layout; the genetic algorithm applies transformation on the chromosomal representation of the physical layout. The algorithm works on a set of configurations constituting a constant size population. The transformations are performed through crossover operators that generate new configurations assimilating the characteristics of the pair of configurations existing in the current population. Mutation and Inversion operators are also used to increase the diversity of the population, and avoid convergence at local optima. Due to the simultaneous optimization of a large population of configurations, there is a logical concurrency in the search of the solution space which makes genetic Algorithm an extremely efficient optimizer.

2. GENERALISED ALGORITHM

1. Randomly generate an initial population $M(0)$
2. Compute the fitness $f(t)$ for each individual (cells) in the current population $M(t)$
3. Define selection probabilities $p(m)$ for each individual m in $M(t)$ so that $p(m)$ is proportional to $f(t)$
4. Generate $M(t+1)$ by probabilistically selecting individuals from $M(t)$ to produce offspring via genetic operators.
5. Repeat step 2 until satisfying solution is obtained.



2.1 Algorithm Description

1. Population Generation

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the *search space*). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

In VLSI, the initial population of cells is generated in the search space representing some encoding scheme considering here the slot Identification Number (ID). Each cell is represented by the English alphabets and every cell (chromosome) has some parameter representing X, Y and Z co-ordinates. X represents the width of the cell represents the height of the cell and Z represents the serial number of the cells. These are the parameter set of the cell representing solution in search space.

2. Selection of Population

In every evolutionary step (iteration step), known as a generation, the chromosomes in the current population are decoded and evaluated according to some predefined quality criterion, referred to as the Fitness function where fitter solutions (as measured by a fitness function) are more likely to be selected. Some selection methods are Elicit Selection, Fitness-proportionate selection, Roulette wheel selection, Tournament Selection, Rank Selection etc.

in VLSI, every cells(chromosomes) in the population generated are decoded and evaluated according to the Fitness Function (considering Area and number of interconnections) where cells which has better fitter solutions are more likely to be selected. Although, there are a number of variants, the basic mechanism consists of Fitness-proportionate Selection.

This paper describes a selection method in which the cell which is having Maximum area will be selected first and so on. and every cell will be crossed with cells having next higher, next lower and equal area.

3. Fitness Function

Fitness function is the reciprocal of the sum of the bounding rectangle Area.

$$f = \frac{1}{\sum_{\text{nets}} x(i) W_H(i) + y(i) W_V(i)}$$

Where f is the fitness,

$x(i)$ and $y(i)$ are the horizontal and vertical spans of net i ,

and $W_H(i)$ and $W_V(i)$ are the horizontal and vertical weights.

Fitness Function of every cell (chromosome) is calculated which represents the area of each cell. The cell which has maximum area placed first in the chip, i.e. the cell having minimum fitness

function is placed first so that GA does maximization so to achieve an optimal solution for the search space.

4. Crossover

Crossover selects genes from parent chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent (| is the crossover point).

Chromosome 1	11011 00100110110
Chromosome 2	11011 11000011110
Offspring 1	11011 11000011110
Offspring 2	11011 00100110110

There are other ways how to make crossover. Crossover can be rather complicated and very depends on encoding of the encoding of chromosome. Specific crossover made for a specific problem can improve performance of the genetic algorithm.

This paper describes a crossover technique which will forward all the components of parent1 to child and only those components of parent1 will be forwarded which are not present in the parent1.

5. Mutation

After a crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can then be following:

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

This paper describes a mutation technique which will exchange the cells in the placement when two cells are having same area but the cell which is placed first is having less number of interconnection than the cell placed later.

3. GENETIC ALGORITHM FOR CELL PLACEMENT:

1. Generate initial population.
2. for I = 1 to number of components do
 - 2.1 Arrange population in sorted order.
 - 2.2 Selection-select the parent which is having highest area first in the (remaining) elements. Perform crossing with the elements which are having next higher, next lower and equal area.
 - 2.3 Crossing- forwards all the components of parent1 to child and only those components of parent1 will be forwarded which are not present in the parent1.
 - 2.4 Mutation- mutations will exchange the cells in the placement when two cells are having same area but the cell which is placed first is having less number of interconnection than the cell placed later.
4. Final combinations of the cells will be generated which will have minimum area; any of the chosen combination can be final solution.

4. CONCLUSION

The genetic algorithm that was originally invented by Holland for adaptive searching in AI [2] has been employed in this research for the placement of cells on a VLSI Chip. The genetic algorithm uses three powerful operators to guide its search through the solution space concurrently, by considering a set of configurations at a time. The placement problem is represented in the form of a genetic code, which is progressively refined, and improved by the operators. This is a major deviation from the conventional placement algorithms that directly apply the transformations to the physical layout. However, this feature of the genetic algorithm is also a potential problem, and unless clever representation is devised, the algorithm may prove inefficient. In this paper, the cell placement problem has been represented as an effective genetic algorithm, and crossover operator have been applied to generate new configurations. Mutation has also been applied to obtain a highly efficient placement algorithm which requires less iteration to converge to a high quality placement.

5. REFERENCES

[1] Goldberg, D.E. (1989): Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA.

[2] J. H. Holland *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan, 1972.

[3] Shahookar, K.; Mazumder, P.”A genetic approach to standard cell placement using meta-genetic parameter optimization” in *Proc. IEEE Int. Conf. on Computer-Aided Design of Integrated Circuits and Systems* pp. 500 - 511 , 1990.

[4] Youssef, H.; Sait, S.M.; Nassar, K.; Benten, M.S.T.” Performance driven standard-cell placement using the genetic algorithm” Fifth Great Lakes Symposium on VLSI pp. 124 – 127, 1995.