Determination of Circuit Bi-Partition in VLSI using Ant Colony Optimization Approach

Manoj Arora
Research Scholar ,Department of Electronics and Communication Engineering
SVU ,Gajraula (AMROHA) UP

Girish Chander Lall
Department of Electronics and Communication Engineering
HCTM, Kaithal,Haryana, India

Abstract - Circuit Partitioning is an important task in VLSI design and its applications. This paper discusses a new approach for Partitioning in VLSI design which is based on the Swarm Intelligence. Swarm Intelligence is a new technique of Artificial Intelligence which is totally inspired by real world insects. Ant Colony Optimization (ACO) approach is dealt in this paper. It is similar to Evolutionary algorithm which focuses on the Engineering aspect of development and applications. This paper out-lines the Ant Colony Optimization technique for Partitioning in VLSI design. The proposed algorithm is expected to be superior both in terms of quality and consistency of the solutions obtained. The objective of this paper is to introduce the swarm algorithm as global technique for optimization of any problem.

Key words:- Partitioning, Very Large Scale Integration, Ant Colony Optimization

I. INTRODUCTION

During the last decade, the complexity and size of circuits have been rapidly increasing, placing a stressing demand on Industry for faster and more efficient techniques for VLSI physical design automation. A typical VLSI design cycle may be represented by the flow chart shown in Figure 1. The figure shows the various physical steps with emphasis on partitioning. Efficient designing of any complex system necessitates decomposition of the same into a set of smaller subsystems. Subsequently each subsystem can be designed independently to speed up the design process. The process decomposition is called Partitioning. A computer system is comprised of tens of millions of transistors. It is partitioned into several smaller blocks for facilitation of the design process Partitioning (CP) is an important task in VLSI design applications. The partitioning of a system into a group of PCBs is called System level partitioning. The partitioning of a PCB into chips is called Board level partitioning, while partitioning of a chip into smaller sub circuits called Chip level partitioning. In VLSI design applications, partitioning algorithms are used to achieve various objectives such as Circuit Layout, Circuit Packaging and Circuit Simulation.

Circuit Layout: A class of placement algorithms, min-cut placement, is based on repeated partitioning of a given network, so as to minimize the size of the cut set at each stage, where the cut set is the set of net that connects two partitions. At each partitioning stage, the chip area is also partitioned, e.g., alternately in the vertical and horizontal directions, and each block of the network is assigned to one region on the chip. This process is repeated several times until each block consists of only one cell. The resulting of cells on chip gives the final layout.

Circuit Packaging: Semiconductor technology places restrictions on the total number of components that can be placed on the single semiconductor chip. Large circuits are partitioned into smaller sub circuits that can be fabricated on separated chips. Circuit partitioning algorithm are used to obtain such sub-circuits, with a goal of minimizing the cut set, which determines the number of pins required on each chip. This technique is gaining renewed importance these days for Field Programmable Gate Arrays (FPGAs), which automatic software for partitioning and mapping large circuits on several FPGA chips for rapid prototyping.

Circuit Simulation: Partition has also been used to split a circuit into two smaller sub circuits which can be simulated independently. The results are combined to study the performance of the overall circuit. This speeds up the simulation process by several times and is used in relaxation-based circuit simulators. This process is also used for simulating circuits on microprocessors.

Optimization is a decision theory which uses any scientific, mathematical or logical means regarding the optimized results in different operations. Partitioning algorithms are broadly divided into two classes: constructive and iterative improvement. Constructive algorithm may start from empty initial partitions while Iterative improvement algorithm, in contrast to constructive techniques, start with an initial partitioning, accomplished through some user defined method or randomly. On the basis of processes the partitioning algorithms can be classified as Group migration algorithms and Evolution based algorithms Evolution and simulated annealing type algorithm belongs to the probability and iterative class of algorithm.

A Introduction to Swarm Intelligence –

Swarm Intelligence is a new technique of Artificial Intelligence which is inspired by real world insects. There comes a lot of swarms under this category the social insects such as ants, termites, bees, and wasps and by swarming, flocking, herding, and shoaling phenomena in vertebrates. It is similar to Evolutionary algorithm which focuses on the Engineering aspect of development and applications. Swarm intelligence is an innovative computational and behavioral metaphor for solving distributed problems that originally took its inspiration from the biological. Beni & Wang introduced the expression "Swarm Intelligence" in 1989, in the context of cellular robotic systems. Two of the most successful swarm intelligence techniques currently in existence are Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). PSO is a global minimization technique for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. ACO is another optimization technique which is totally inspired by Real Ants. ACO algorithm requires the combinatorial optimization problems to be represented by hyper graphs.
This Paper is organized as follows: In Section II, Ant colony optimization (ACO) approach is briefly mentioned and characteristics related to tests are discussed. The Problem formulation related to VLSI design is proposed in Section III. The various steps of ACO algorithm used are briefly described in Section IV. Experimentation is done on section V. The Conclusion and future scope is described in Section VI.

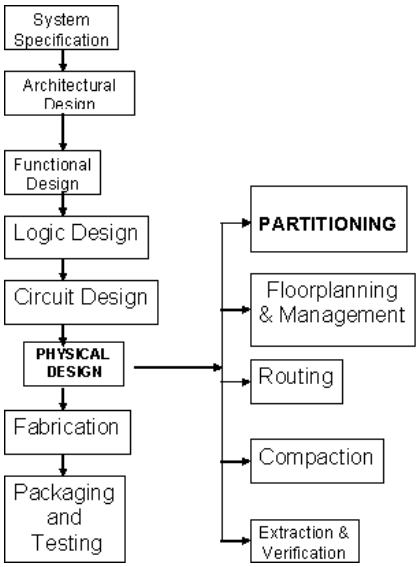

Figure 1. A simple VLSI Design cycle.

II. ANT COLONY OPTIMIZATION APPROACH

Recently, a new entrant to the family of evolutionary algorithms, the ant colony optimization (ACO) method, has been proposed. An intuitive description of the foraging behavior of ants is shown in Fig. 2. Initially, three ants leave their nest in random directions to search for food. As they wander around, they deposit certain amount of pheromone trails, which will evaporate slowly but are detectable by other ants. Now assume *Ant* 1 finds a food source. It will pick up some food and return to the nest by following its own pheromone trail, laying additional pheromone on the same path while *Ant* 2 and *Ant* 3 are still wandering randomly. When the next group of ants leaves their nest to search for food, they detect twice as much pheromone on *Path* 1 than on *Path* 2 and *Path* 3, assuming the

evaporation of pheromone is negligible. Since the probability for a path to be followed is proportional to its pheromone value, more ants will follow *Path* 1 in this second round of search for food. In this way, the ants can establish the optimized path from their colony to the feeding sources.
ACO is a metaheuristic optimization algorithm that can be used to find approximate solutions to difficult combinatorial optimization problems. Ant algorithm is multi agent systems in which the behavior of the each single agent is called artificial ant, is inspired by the behaviour of real ants. This work has been lead by Marco Dorigo at the Politecnico Di Milano. Ant Colony Optimization have numbers of components which includes:-

- A colony of ants, a mechanism to generate and to activate ants, pheromone evaluation mechanism, a daemon action and a stopping condition

Ant Colony Optimization is a general-purpose heuristic algorithm, which can be used to solve diverse combinatorial optimization problems. The Ant colony optimization (ACO) has the following desirable characteristics:-

- It is versatile, in that it can be applied to similar versions of the same problem.
- It is robust and general purpose.
- It is a population-based heuristic.

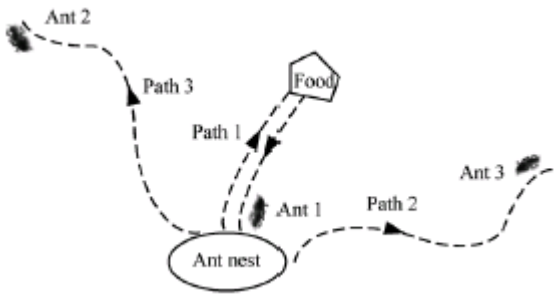

Figure 2. - Intuitive description of foraging behavior of ants.

III. PROBLEM FORMULATION

There are three distinct classes of VLSI Physical design layout problem such as Partitioning, Placement and Routing. Proper problem formulation for circuit partitioning is very important for effective partitioning, which in turn depends on the representation of the circuit as input to the partitioning algorithm. The partitioning problem can be expressed more naturally in Graph theoretic terms, with the components as nodes and the wires connecting them as edges. The weights of the nodes represent the size of corresponding component, and the weights of edges represent the number of wires connecting the component. The problem input consists of a hyper graph G= (V, E), where V = {v1,v2,….,v n} be a set of vertices and E = { e1,e2…..,en} be a set of hyper edges. The partitioning problem at any level or design style deals with one or more of the following parameters:-

- Interconnections between partitions.
- Delay due to the partitioning.
- Number of terminals.
- Area of each partition.
- Numbers of partitions.

Partitioning problem can be formulated as a bi partitioning or a multi-partitioning problem. Bi partitioning divides a graph into two nonempty sub graphs while minimizing the number or weight of the edges cut by the partition. In addition to this, multiway partitioning is decomposing the given larger system into more than two smaller sub-systems. i.e. for a k-way multi- partitioning, circuit is partitioned into N_1, N2…….N_k subsystems with the objective to minimize the interconnections between the partitions. An area constraint is often imposed on each sub-system so that all the sub-partitions have almost equal number of components. As a generalization of ratio cut

partitioning, various alternative objectives have been proposed for partitioning to combine cut size and the balance criterion in a single objective.

Ant colony optimization is a population-based search technique for the solution of combinatorial optimization problems. ACO algorithm requires the problem to be represented by a graph, consisting of a finite number of Nodes and Links between nodes. Any parameter like cost, path etc. may be associated with each link. The modeling of partitioning problems into hyper graphs allows us to represent the circuit partitioning problems completely as hyper graph partitioning problem. The proposed ACO algorithm is going to deal with Interconnections between partitions as one of the Parameter. The number of interconnections at any level of partitioning has to be minimized. Minimization of number of interconnections between partitions is called Min Cut Problem. Reducing the interconnections not only reduces the delay but also reduces the interface between the partitions making it easier for independent design and fabrication. The minimization of the cut is very important objective function for partitioning algorithm for any level or any style of design.

IV. THE ACO ALGORITHM

ACO is a didactic tool to explain the basic mechanisms underlying ACO algorithm. The algorithm adapts the real ant's behaviours to the solution of the shortest path problems on graphs. The problems are generally characterized by following:-

- a set of finite constraints.
- a solution.
- a cost function.
- a finite set of components and possible transition.

Following is the details on how to implement ACO approach on various problems.

Step 1: Ant's Path Searching Behaviour

Each ant builds, starting from the source node, a solution to the problem by applying a step by step decision policy at each node, local information stored on the node itself or on its outgoing arcs is read (sensed) by the ant and used in a stochastic way to decide which node to move to next. At the beginning of the search process, a constant amount of pheromone (e.g., $\tau_{ij} = 1$) is assigned to all the arcs. When located at a node i an ant k use the pheromone trails to compute the probability of choosing j as next node: -

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}}{\sum_{j \in N_i^k} \tau_{ij}^{\alpha}}, & if \quad j \in N_i^k; \\ 0, if j \notin N_i^k \end{cases} \qquad (1)$$

Where N_i^k is set of nodes concerned to i w.r.t. ant k and α is a constant (normally equal to 2) used to amplify the influence of pheromone concentration.

In ACO the neighborhood of a node i contains all the nodes directly connected to node "i" in the graph, except for the predecessor of node i .in this way the ants avoid returning to the same node they visited immediately before node i. An ant repeatedly hops from node to node using this decision policy until it eventually reaches the destination node .due to differences among the ants' paths, the time step at which ants reach the destination node may differ from ant to ant.

Step 2: Path Retracing and Pheromone Update

When an ant reaches the destination node, the ant switches from the forward mode to the backward mode and then retraces step by step the same path backward to the source node. An additional feature is that, before starting the return trip, an ant eliminates the loops it has built while searching for its destination node.

During this return travel to the source the k th ant deposits an amount $\Delta\tau^k$ of pheromone on arcs it has visited. In particular if ant k is in the backward mode and it traverses the arc (i, j), it changes the pheromones values as follows:-

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k \qquad (2)$$

where $\Delta\tau^k$ is increment in pheromone quantity = $1/L^k$.

By this rule an ant using the arc-connecting node i to node j increase the probability that forthcoming ants will use the same arc in the future. The values of $\Delta\tau^k$ can be constant or function of the path length-the shorter the path the more pheromone is deposited by an ant.

Step 3: Pheromones Trail Evaporation

In the last step, for each edge in the graph, evaporate pheromone trails with exponential speed. Pheromone trail evaporation can be seen as an exploration mechanism that avoids quick convergence of all the ants towards a sub-optimal path. In ACO, pheromone trails are evaporated by applying the following equations to all the arcs:-

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \quad (3)$$

where ρ is evaporation constant $\rho \in (0,1]$.

Step 4: Termination Condition

The program stops when the follow in termination condition apply:-

- A maximum number of algorithm iteration has been reached.
- If end of termination node occurs.

The algorithm discussed in this section is stochastic and population based search algorithm. The ACO algorithm overviewed in the paper has been applied to many combinational optimization problem defined over discrete space. The Traveling Salesman Problem (TSP) is an extensively studied problem in the literature and for a long time has attracted a considerable amount of research effort. The experimental study of TSP network using ACO algorithm defined above result in good performance.

V. EXPERIMENTATION

Adders of various bits are used as test cases for this work. Table 1 shows the characteristics of ten circuits for testing. The first columns indicate the circuit to be partitioned. The size of circuit varies from 06 nodes to 111 nodes. Table 2 presents results based on algorithm discussed in previous section. In Table 2, best minimum cut is reported with 100 to 2000 numbers of iterations and 50 to 700 numbers of ants on simulation. Figure 3 to 5 shows the simulation results for Add 10, Add 12 and Add 16.

Table 1 Circuit Characteristic

Test Case	Number of Nodes	Number of Edges
Add1	6	8
Add2	13	23
Add3	20	38
Add4	27	53
Add5	34	68
Add6	41	83
Add8	55	113
Add10	69	143
Add12	83	173
Add16	111	233

Table 2 Results of Various Adders

Test Case	Number of Nodes	Number of Edges	Min-Cut
Add1	6	8	4
Add2	13	23	8
Add3	20	38	8
Add4	27	53	10
Add5	34	68	18
Add6	41	83	21
Add8	55	113	24
Add10	69	143	29
Add12	83	173	37
Add16	111	233	47

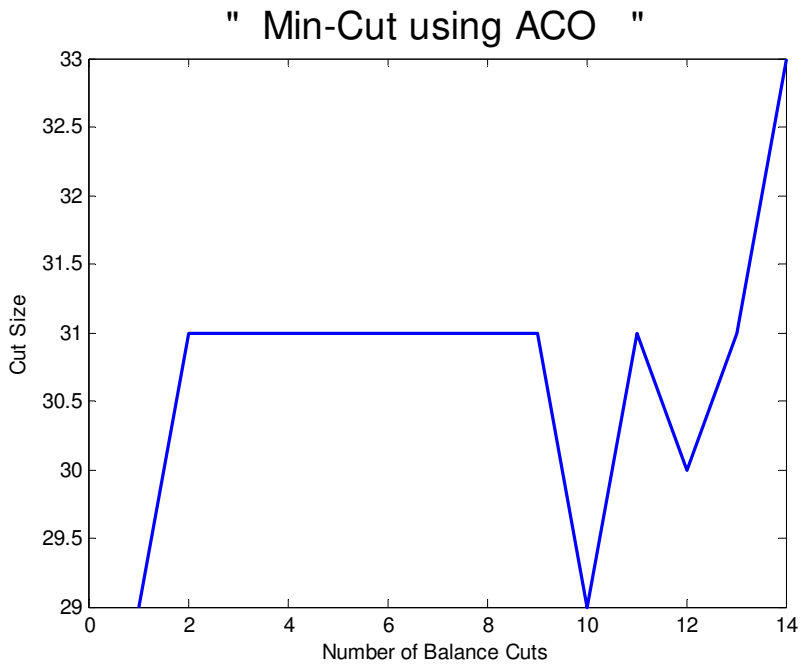

Figure 3. Response of Add10

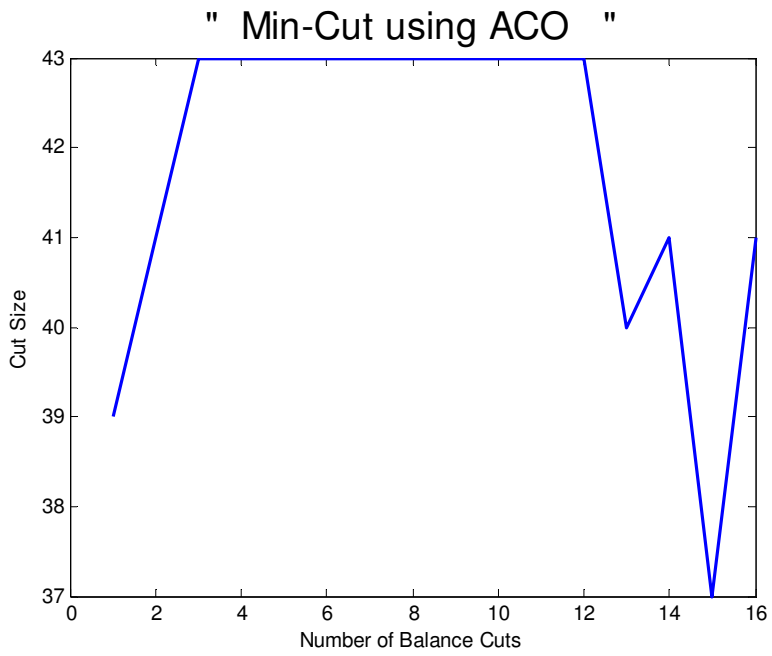

Figure 4. Response of Add12

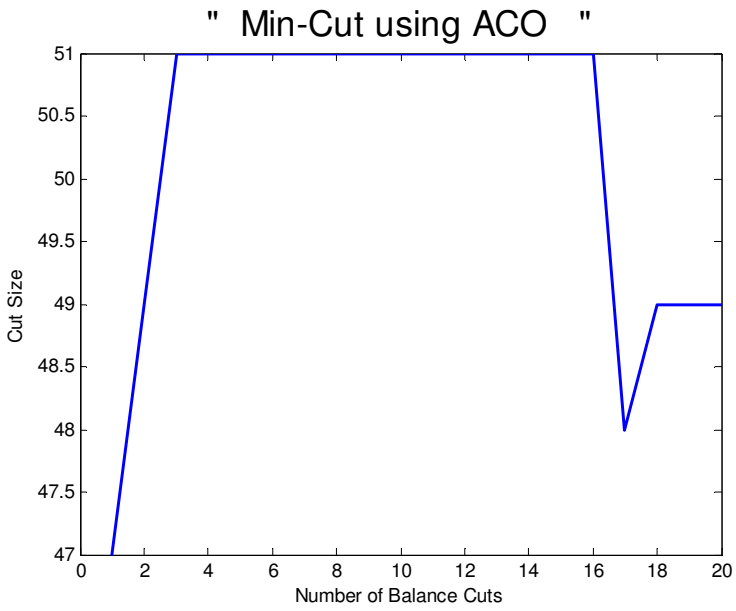

Figure 5. Response of Add16

VI. CONCLUSION & FUTURE DIRECTIONS

The Problem of circuit partitioning is a process of separating the sets of circuits elements into two or more subsets such that connectivity between the weighted edges across the blocks can be minimized. A number of deterministic and probabilistic approaches have been reviewed. This paper provides an overview of the standard ACO algorithm for partitioning in VLSI Design. It is thoroughly desired that a good meta-heuristic be developed to solve multiple criteria facility layout problem, which tries to optimize both qualitative and quantitative factors. Based on this efficient algorithm on large number of standard Benchmark circuits, superior results may be achieved. This work is

an endeavor in this direction and future of this technique seems to be brighter. The objective of this paper is to introduce the ACO algorithm as global technique for optimization of Partitioning in VLSI problem.

REFERENCES

[1] Isaacs Jason , Watkin Robert & Petrone Joe (2003) "*Ant Colony Systems* " International Conference on Military & Aerospace Program Logic Device (2003), pp. 32-46.

[2] Kumar Shakti (2006) "*Application of Ant Colony Optimization",* Workshop on Applied Soft Computing (SOCO-2006) , Jagadhri,India, pp.152-159.

[3] Bonabean Eric & Meyer Christopher (May,2001)"S*warm Intelligence : A Whole way to think* " Harvard Business Review(May 2001) , pp. 92-98.

[4] Tarasewich Peter & R. McMullen Patrick (Aug. 2002)"*Swarm Intelligence*" Communication of the ACM (Aug. 2002) , pp. 62-67.

[5] R. Chandrasekharam, S. Subhramanian & S. Chaudhury (Sept. 1993)*" Genetic algorithm for node partitioning problem and applications in VLSI design"* IEEE Proceedings, pp. 255-260.

[6] Pinaki Mazumder & E.M.Rudnik (2003) "*Genetic Algorithm for VLSI Designs layout and test Automation"* Pearson Education, Low price Edition, Delhi.

[7] Naveed A. Sherwani (2005) " *Algorithm for VLSI physical design Automation"* Springer (India) Pvt. Ltd., New Delhi.

[8] Young-Su Kwon & Chong-Min Kyung (2005)" *An Algorithm for Circuit Partitioning Into Multiple FPGAs Using Time-Multiplexed, Off-Chip, Multicasting Interconnection Architecture"* IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol. 13, No. 7, July 2005.

[9] Gaganpreet Kaur & S.S.Gill (2007) " *A comparative review of classical and evolutionary approaches for VLSI circuit Partitioning"* International Conference on Intelligence systems and Networking (IISN-07), pp. 112-118.

[10] Pawan Dahiya, J.S.Saini & Shakti Kumar (2007)" *Evolutionary Trends in VLSI Design"* International Conference on Intelligence systems and Networking (IISN-07), pp. 20-25.

[11] Kalapi Roy-Neogi & Carl Sechen (1995)" *Multiple FPGA part with performance optimization"* Proceeding of 3rd International symposium on FPGA ' 95, pp. 1-7.

[12] S.Areibi (2001)" *Recursive and Flat partitioning for VLSI circuit design"* Proceeding of 13 th International Conference on Microelectronics, pp. 71-78.

[13] J Tada, R Egawa, K Kawai, H Kobayashi, G Goto, "*A middle grain circuit partitioning strategyfor 3-D integrated floating point multipliers*"IEEE International conference 3D System Integration 2011, pp1-6

[14] T Kato, K Inoue, F Koga, Y Kotani, "*Parallel Analysis Method for power Electronic System by Circuit Partitioning,*" IEEE 13th Workshop on Control and Modeling for Power Electronics 2012,pp 1-8

[15] K.A.Sumithradevi, M.N.Vijayalakshmi, "Optimization of circuit partitioning in VLSI through classification algorithms" IEEE Region 10 Conference TENCON 2010, pp-1554-1557.