

# Hybrid Algorithm PSO and SA in Achieving Partitioning Optimization for VLSI Applications

Shikha Arora<sup>1</sup>

Department of ECE, DAVIET, JALANDHAR

Neeru malhotra<sup>1</sup>

Department of ECE, DAVIET, JALANDHAR

Rajan Vohra<sup>2</sup>

Department of Electronics Technology, G.N.D.U, AMRITSAR

Rajdeep Singh<sup>3</sup>

Department of ECE, CEM, KAPURTHALA

**Abstract:** This paper includes a new partitioning algorithm for circuit bi-partitioning, used for the reduction of the number of interconnections between elements of VLSI circuit. In this paper, the hybrid PSO and SA algorithm for the bi-partitioning problem is proposed. PSO employs a collaborative population-based search, which is inspired by the social behavior of bird flocking. It combines local search (by self experience) and global search (by neighboring experience), possessing high search efficiency. SA employs certain probability to avoid becoming trapped in a local optimum and the search process can be controlled by the cooling schedule. Experimental result shows that the developed hybrid PSO and SA algorithm can consistently produce the better fitness value and the time required is less than the other algorithms of optimization.[1-10]

**Keywords-** Partitioning, Simulated Annealing Algorithm, Particle Swarm Optimization Algorithm, Netlist.

## I. INTRODUCTION

IN the contemporary VLSI circuits the length of interconnections between elements significantly affects their performance. As the throughput of modern circuits is growing, the importance of shortening the interconnections between their elements is increasing. Although a lot of research effort has been made on development of partitioning methods of VLSI technologies there is still a big demand for new investigation and improvement in this area. The advancement in VLSI semiconductor technology has led to a phenomenal development in electronics industry, leading to more chip complexity and higher integration, increased design sizes and huge chip estate being occupied by interconnects, which leads to increased delay. Circuit partitioning is an important step in VLSI physical design and involves the division of a circuit into smaller parts for ease of design and layout. The main objective of circuit partitioning can include minimization of number of interconnections between the partitions, minimization of delay between partitions, power consumption, optimization and area optimization

VLSI circuit partitioning is a vital part of physical design stage. The essence of circuit partitioning is to divide the circuit into a number of sub circuits with minimum interconnections between them. In this paper the techniques of PSO and SA in solving bi-partitioning problem is evaluated. Multiple partitions can be obtained by applying the method on obtained partitions. Efficient easily applied algorithms for optimal clustering to minimize delay in digital networks have been developed by Lawler et al.[1] Kernighan and Lin[2] propose a heuristic for two way partitioning which is the first interactive algorithm based on swapping of vertices. A more practical model based on hypergraphs is proposed, but was inefficient due to time complexity.[3] A new data structure bucket list for cell gains and proposed cell move with

better time complexity is proposed. Krishnamurthy modified to introduce the concept of look ahead to choose the cell move. Various multiway partitioning algorithms are proposed[4] by modifying and developing appropriate data structures, top down clustering and iterative primal-dual approach, dual intersection graph representation and ration cut metric. Ariebi and Vanneli [5] describe the application of Tabu search to circuit partitioning problem.[6]

The different objectives that may satisfied by partitioning are:

- 1) Minimization of number of cuts
- 2) Minimization of delay due to partitioning
- 3) Reduce the fabrication cost
- 4) Limit on number of terminals is decided by maximum no of terminals available on PCB connector.[7]

From the literature review it is found that various researchers have applied numerous optimization techniques for the partitioning optimization problems with mixed result. In the present work two excellent optimization methods of PSO and SA have been applied to above problem

## II. PROBLEM FORMULATION

### 2. PSO & SA

#### 2.1 Basic PSO Algorithm

The concept of PSO roots from the social behaviour of organisms such as bird flocking and fishing schooling. Through cooperation between individuals, the group often can achieve their goal efficiently and effectively. PSO simulates this social behaviour as an optimization tool to solve some optimization problems. In a PSO system, each particle having two properties of position and velocity represents a candidate solution to the expressed by the objective function. In the iteration, the objective function is calculated to establish the fitness value of each particle using position as input. Fitness value determines which position is better. Each particle flies in the search space with a velocity that is dynamically adjusted based on its own flying experience and its companions' flying experience. In other word, every particle will utilize both the present best position information of its own (*Pbest*) and the global best position information (*gbest*) that the swarm has searched up-to-now to change its velocity and thus arrives in the new position.

PSO can be described mathematically as follows. Suppose that the search space is of d-dimension and the number of particles is n. The *i*th particle is represented by a d-dimension vector  $X_i=(x_{i1},x_{i2},\dots,x_{iD})$

$P_{best_i} = (p_1, p_2, \dots, p_D)$  denotes the best position searched by the  $i$ th particle and the  $g_{best} = (g_1, g_2, \dots, g_n)$  is the best position searched by the whole swarm up-to-now.

Each particle updates its velocity and position according to the following equations.

$$V_{id} = w \times V_{id} + c_1 \times \text{rand}() (p_{best_{id}} - X_{id}) + c_2 \times \text{rand}() (g_{best_{id}} - X_{id}) \dots\dots(1)$$

$$X_{id} = X_{id} + V_{id} \dots\dots(2)$$

where  $w$  is the inertia coefficient which is a chosen constant in interval  $[0,1]$ ;  $c_1, c_2$  are two acceleration constants;  $\text{rand}()$  is random value in interval  $[0,1]$ . The velocities of particles are restricted in interval  $[V_{min}, V_{max}]$ . If the resulting value is smaller than  $V_{min}$ , one element of the velocity vector is set to  $V_{min}$ ; if the resulting value is greater than  $V_{max}$ , one element of velocity vector is set to  $V_{max}$ .

The method described above is suitable for problems of continuous quantities. However, it can't be applied directly to problems of discrete quantities. Wang et al. defined the "swapping operator" for solving TSP problem based on PSO. However, the drawback of being easily trapped into local minima still exists. The hybrid PSo and SA algorithm was proposed to overcome this shortcoming.

### 2.2 Simulated Annealing

The term annealing derives from the physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure. Simulated Annealing (SA) is the optimization method to stochastically simulate this physical process of annealing on the computers. In SA, the simulation proceeds by randomly generating a solution and then determining its acceptance in certain probability. A temperature parameter is used to determine this probability. In the basic algorithm of SA, three operations bear an important role: generate, accept and cool.

The general operation modifies a current solution  $X$  and generates a next solution  $X'$  using a probability distribution  $G(X, X')$ .

The accept operation is a judgment to decide whether to accept the modification or not. The acceptance of modification is determined from a difference  $\Delta E (=E'-E)$  of a current energy  $E=f(x)$  and a modified energy  $E'=f(X')$ , and a temperature parameter  $T$ . If the  $\Delta E \leq 0$ , the modification is accepted. In case  $\Delta E > 0$ , the modification is accepted at certain probability. This algorithm can be defined as:

$$P_{accept} = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ \exp(-\Delta E / T) & \text{otherwise} \end{cases} \dots\dots\dots(3)$$

The temperature  $T$  is the important parameter to control the acceptance of the modified solution. At the beginning of the simulation, both the temperature and the acceptance levels are high. As the simulation proceeds and the temperature decreases, solutions that have the bigger fitness values.

Cooling is the operation to generate the temperature of the next state  $T_{k+1}$  from the temperature of the current state  $T_k$ .

The simulation begins with the initial solution  $x$  with energy of  $E$  and the initial temperature  $T$ . The

solution is then randomly modified to  $x'$  with energy of  $E'$ . The acceptance of the modification is calculated from the difference of energy,  $\Delta E (= E' - E)$ , and the temperature  $T_k$ . If accepted, the solution  $x'$  becomes the starting point of the next step. These operations are repeated long enough at each temperature for the system to reach a steady state, or equilibrium. When reaching the steady state at  $T_k$ , the temperature is cooled to  $T_{k+1}$  and the simulation is repeated until reaching steady state again. Users define the terminal criterion, such as the number of evaluations. The simulation is concluded when the temperature becomes low enough and a terminal criterion is met. The temperature can be updated by the following method

$$T_k = \alpha \cdot T_{k-1} \dots\dots\dots(4)$$

The problem involves dividing the circuit net list into two subsets. The objective function captures the interconnection information. The mathematical representation of the objective function is given as

$$F = \sum_{i=0}^k \frac{1}{M} \dots\dots\dots(5)$$

F= Fitness Function  
M= Min Cut

### 2.3 The Hybrid Algorithm Based on PSO and SA

- (1) iter ← 0, cpt ← 0, Initialize swarm size particles
- (2) stop criterion ← maximum number of function evaluations or Optimal solution is not attained
- (3) while Not stop criterion do
- (4) for each particle  $i \leftarrow 1$  to swarm size do
- (5) evaluate (particle(i)) if the fitness value is better than the best fitness value (cbest) in history then
- (6) Update current value as the new cbest.
- (7) end
- (8) end
- (9) Choose the particle with the best fitness value in the neighborhood (gbest)
- (10) for each particle  $i \leftarrow 1$  to swarm size do
- (11) Update particle velocity according to Equation (1)
- (12) Enforce velocity bounds
- (13) Update particle position according to Equation (2)
- (14) Enforce particle bounds
- (15) end
- (16) if there is no improvement of global best solution then
- (17) cpt ← cpt + 1
- (18) end
- (19) Update global best solution
- (20) cpt ← 0
- (21) if cpt = K then
- (22) cpt ← 0
- (23) //Apply SA to global best solution
- (24) iterSA ← 0, Initialize T
- (25) current solution ← global best solution

- (26) current cost ← evaluate(current solution)
- (27) while Not SA stop criterion do
- (28) while inner-loop stop criterion do
- (29) Neighbor ← generate(current solution)
- (30) Neighbor cost ← evaluate(Neighbor)
- (31) if Accept(current cost, Neighbor cost, T) then
- (32) current solution ← Neighbor
- (33) current cost ← Neighbor cost
- (34) end
- (35) iterSA ← iterSA + 1
- (36) Update (global best solution)
- (37) end
- (38) Update(T) according to Equation(3)
- (39) Update (SA stop criterion)
- (40) end
- (41) end
- (42) iter ← iter + 1, Update (stop criterion)
- (43) end

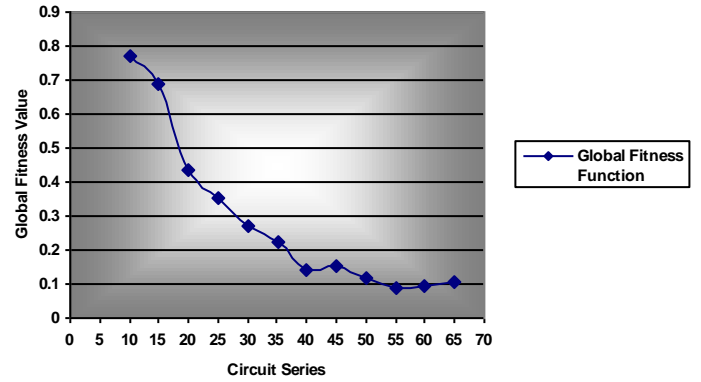
III. EXPERIMENTAL RESULT

The Proposed Algorithm is tested on 11 netlists to demonstrate the effect of iteration by using hybrid PSO and SA algorithm on partitioning. The result of partitioning with PSO and SA is given in table 1. Here the no of particles are taken as 5.

TABLE 3.1  
RESULT OF PARTITIONING WITH HYBRID PSO AND SA ON NETLIST

Circuit Series	No of Node	No of Files	Global Fitness by PSO and SA
SPP N-10 Series	10	483	0.771428568
SPP N-15 Series	15	184	0.68721014
SPP N-20 Series	20	121	0.434891767
SPP N-25 Series	25	107	0.352136176
SPP N-30 Series	30	52	0.267880135
SPP N-35 Series	35	31	0.223147255
SPP N-40 Series	40	41	0.142679579
SPP N-45 Series	45	28	0.153256176
SPP N-50 Series	50	24	0.117887013
SPP N-55 Series	55	20	0.090323572
SPP N-60 Series	60	9	0.091669922
SPP N-65 Series	65	7	0.10805404

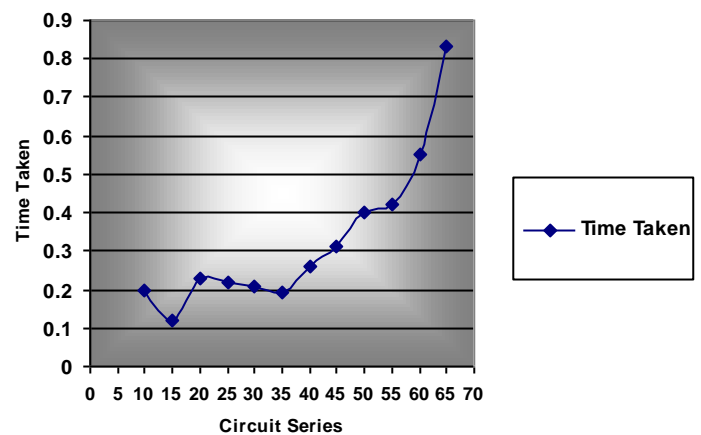
As seen from Table I, average results obtained by hybrid PSO and SA based partitioner are consistently better than these obtained by GA based partitioner for all partitioning instances overall size ranges.



Plot 3.1. Fitness Value of Different series Circuit

TABLE 3.2  
AVERAGE RUNTIMES FOR DIFFERENT CIRCUIT NETLIST SERIES.

Circuit Series	No of Node	No of Files	Time Taken (ms)
SPP N-10 Series	10	483	0.20
SPP N-15 Series	15	184	0.12
SPP N-20 Series	20	121	0.23
SPP N-25 Series	25	107	0.22
SPP N-30 Series	30	52	0.21
SPP N-35 Series	35	31	0.19
SPP N-40 Series	40	41	0.26
SPP N-45 Series	45	28	0.31
SPP N-50 Series	50	24	0.40
SPP N-55 Series	55	20	0.42
SPP N-60 Series	60	9	0.55
SPP N-65 Series	65	7	0.83



Plot 3.2. Time Taken of Different series Circuit

IV. CONCLUSION

Hybrid PSO and SA algorithm is applied to VLSI circuit partitioning problem. While this algorithm has been successfully applied to this problem, hybrid PSO and SA outperforms GA by an average of 32 percent over all test instances. Plot 3.1 shows that the fitness function of different net list is greater that the fitness function obtained by the GA or PSO or SA. Hybrid PSO gives better fitness function that all other optimization techniques. Plot 3.2 gives the timing relation, which shows that the time taken by the circuit series

is less as compared to other algorithms used for the optimization purpose.

#### REFERENCES

- [1] Eugene L. Lawler, Karl N. Levitt, and James Turner, "Module Clustering to minimize delay in Digital Networks", IEEE Transactions on Computers, Vol. C-18 , No.1,pp47-57,Jan, 1969.
- [2] B.W. Kerhingham , S. Lin, "An efficient heuristic procedure for partitioning graphs",Bell System Tech. Journal , Vol. 49, pp 291 – 307, Feb,1970..
- [3] D.G. Schweikert and B.W. Kernighan, "A proper model for the partitioning of electrical circuits," Proc. ACM/IEEE Design Automation Workshop, pp. 57-62, 1972
- [4] C.M. Fiduccia and Mattheyses, "A Linear time heuristic for improving network partitions" , Proc. 19th IEEE Design and Automation Conference, IEEE Press, Piscataway, NJ, USA , pp 175-182, 1982..
- [5] Shawki Areibi and Anthony Vannelli, "Circuit partitioning using a tabu search approach", IEEE International Symposium on Circuits and Systems , Chicago, Illinois,pp 1643-1646, March,1993
- [6]S. Areibi, "Recursive and flat partitioning for VLSI circuit design," The 13th International Conference on microelectronics, Rabat, Morocco, pp. 237-240, 2001.
- [7] Sandeep Singh Gill, Dr. Rajeevan Chandel, Dr. Ashwani Chandel (2010) 'Genetic Algorithm Based Approach to Circuit Partitioning' IEEE Vol. 2, No. 2, April, 2010
- [8] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization :An Overview," Swarm Intelligence Journal, vol.1, No.1, pp. 33-57, 2007
- [9] G. Wang, W. Gang, and R. Kastner, "Application Partitioning on programmable platforms using Ant Colony Optimization," Journal of Embedded Computing, vol. 2, Issue 1, pp.119-136, 2006
- [10]Sandeep Singh Gill, Rajeevan Chandel, and Ashwani Chandel (2009) 'Comparative study of Ant Colony and Genetic Algorithms for VLSI circuit partitioning'