International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 7, July 2014

A NOVEL APPROACH ON PSO IN VLSI ROUTING

K.Madhavi, Assistant Professor
Department of ECE, Dadi Institute of Engineering & Technology

***Abstract* -** The performance of very large scale integration (VLSI) circuits predominantly depends on routing of interconnected circuits. The chief problems in the design of VLSI layouts are wire sizing, buffer sizing and buffer insertion. This technique exits to improve power dissipation, area usage, noise and time delay. The interconnect interruption can be improved by choosing proper buffer locations along the routing path. A stochastic based Particle Swarm Optimization system is used now to optimize buffer locations to find the shortest path and also simultaneously minimize the congestion. The performance is compared with Particle Swarm Optimization based VLSI routing. The results obtained shows the proposed methodology has a good potential in VLSI routing and can be further extended in future.

***Keyword* -** Buffer insertion, Minimal Spanning Tree (MST), Routing, Particle Swarm Optimization (PSO).

I. INTRODUCTION

With rapid advancement of Very Large Scale Integrated (VLSI) circuit fabrication technology, size of the deep submicron and nanometer design has become a predominant factor in circuit performance and reliability.

Numerous techniques are developed to reduce interconnect delay; buffer insertion has been proven methodology among them. Therefore the performance of VLSI circuit is largely depending upon wire routing and buffer insertion through the path. The routing is a complex problem. Thus optimization algorithmsgive best solutions for the given routing problem with buffer insertion.

To solve this multifaceted problem,Hai Zhou [8] proposed novel algorithms to solve this problem comprehensively. An evolutionary computation technique, Particle Swarm Optimization (PSO) is used by Chen Dong [13] and his only shortest path without buffer insertion. Nasir Ayob etal [8] attempted the same problem including buffer insertion along with routing optimization using PSO.

In this paper,an Particle Swarm Optimization (PSO) is used to solve the complex interconnect delay with buffer insertion .In section II, brief outline of PSO algorithm is given.In section III, modeling in PSO is presented followed by results in section IV.At the end, conclusion is given in section V.

II.PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhant [8] proposed an approach called "Particle Swarm Optimization" which was inspired on the choreography of bird flock. It is a population based search algorithm that exploits a population of individuals to probe promising regions of the search space. In the context, the population is called a swarm, and individuals are called particles. Each particle moves with an adaptable velocity with in the search space and retains in its memory the best position it ever encountered.

Considering an D-dimensional search space, andi^{th}particle is associated with the position attribute $X_i = [x_{i,1}, x_{i,2}, \dots\dots . x_{i,D}]$ and velocity attribute $V_i = [v_{i,1}, v_{i,2}, \dots\dots . v_{i,D}]$. The best position encountered by the i^{th} particle is denoted as$P_i = [p_{i,1}, p_{i,2}, \dots\dots . p_{i,D}]$. Assume $g$to be index of the particle that attained the best position found by all particles in the swarm. The swarm is manipulated in the same form resembling the following equations

$$V_{id}(t+1) = w.V_{id}(t) + c_1.rand.(p_{id}(t) - X_{id}(t)) + c_2.rand.(p_{gd}(t) - X_{id}(t)) \quad \dots (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \dots (2)$$

Where $i = 1,2, \dots\dots Np$ is the particles index, $d = 1,2, \dots\dots, D$ is the dimension index and $t = 1,2, ..$ indicates the iteration number. The variable C_1 and C_2 are positive constants, which are referred to as cognitive and social parameters, respectively and $rand$ is a function which generates a random number that is uniformly distributed within the interval $\{0,1\}$. The variable w is a parameter called inertia weight, which plays the role of balancing the global and local searches. It is positive linear function of iteration, given as

ISSN: 2278 – 909X

$$w = w_{last} - \frac{(w_{last} - w_{start}) * iteration}{\max imum \quad iteration} \dots (3)$$

III. MODELING IN PSO

Initialize the Particle Swarm. Initialize m particles with random position and velocity inside the search space. Calculate the Fitness Value of each particle. Each particle represents a minimal spanning tree (MST), and the cost of the MST is the fitness value of this particle. The fitness value is calculated by Prim algorithm [7]. Compare the calculation of each particle with the global best previous values among the population pg: if the current value is better than the previous global best values, then set the current value to be the global best value. Calculate the self-adaptive Inertia weight for every particle: Put a particle Fitness Value and its' swarm population into equation (3), the particle's Inertia Weight wk would be get. Update the velocity and position of particles by using equation (1) and (2). Check the termination condition (a good enough position or the maximum number of iterations is reached). If fulfilled, the run is terminated.

IV.RESULTS

Swarm size is taken of 200 is used. The maximum number of iterations is set as 100. The acceleration constants are set as: $c_1 = c_2 = 2$. r_1 and r_2 are random numbers, their value are between 0 and 1. The inertia weight should be set in the range of [0.4 to 0.95] from Shi and Eberhart [4] for PSO's good performance. We set the inertia weight by equation (3) applying to VLSI problem. **Experiment 1:** There are 12 cells generated randomly in this experiment, and the coordinates are shown in Table 1.

TABLE 1. Experiment 1: Coordinates of 12 cells

NO.	**1**	**2**	**3**	**4**	**5**	**6**
X	36.97	36.99	53.66	67.56	67.55	63.22
Y	75.53	52.88	57.87	79.36	52.24	41.05
NO	**6**	**9**	**4**	**8**	**12**	**13**
X	81.67	36.23	20.28	43.11	30.73	59.77
Y	37.83	39.45	42.77	33.39	16.25	18.35

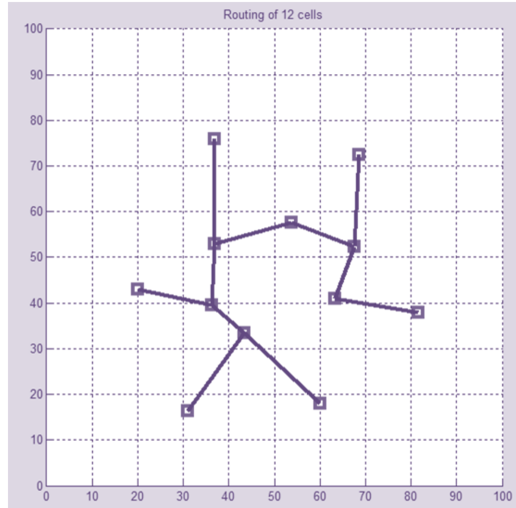

Figure 1: Optimal solution found by PSO for Experiment 1.

Experiment II. There are 21 cells generated randomly in this experiment, and the coordinates are shown in Table II.

TABLE II. Experiment II: Coordinates of 21 cells.

No	**1**	**2**	**3**	**4**	**5**	**6**	**7**
X	13.87	13.65	15.06	15.29	21.49	31.43	48.39
Y	14.77	32.6	48.68	45.99	74.12	84.36	85.53
No	**8**	**9**	**10**	**12**	**12**	**13**	**14**
X	66.81	78.22	92.84	79.39	63.69	88.16	56.58
Y	77.87	95.23	28.74	78.86	34.24	47.81	53.07

ISSN: 2278 – 909X

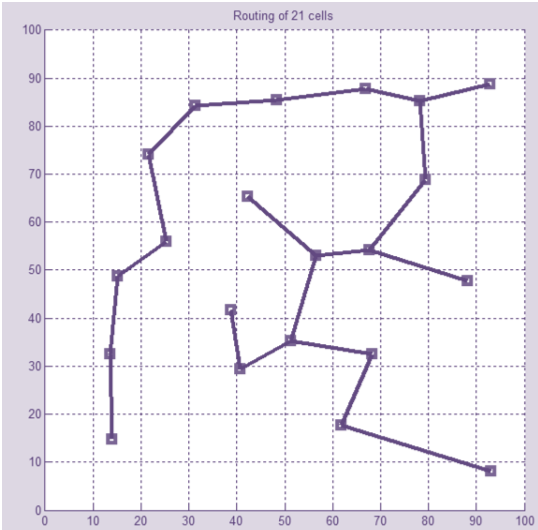

Figure 2: Optimal solution found by PSO for 21 cells.

V. Conclusion

The complex routing problem in VLSI has been solved using bio-inspired computing techniques. Particle Swarm Optimization algorithm is successfully applied to obtain optimum paths in VLSI routing. This paper presents an application of PSO for VLSI routing. The experiments have been carried out to demonstrate the feasibility of PSO implement VLSI routing. And algorithm also shows good results in VLSI routing optimization.

REFERENCES

[1]. Simultaneous Routing and Buffer Insertion with Restrictions on Buffer Locations. Hai Zhou, D. F. Wong, I-Min Liu, and Adnan Aziz,2000.

[2]. T. H. Cormen, C. E. Leiserson, and R. H. Rivest, *Introduction to Algorithms* Cambridge, MA: MIT Press, 1989.

[3]. J. Lillis, C. K. Cheng, and T. T. Lin, “Optimal and efficient buffer insertion and wire sizing,” in *Proc. Custom Integrated Circuits Conf.*, 1995, pp. 259–262.

[4]. Blum C, Sampels M. An ant colony optimization algorithm for shop scheduling problems. J Math Modeling Algorithms 2004;3(3):285–308.

[5]. Corry P, Kozan E. Ant colony optimization for machine layout problems. Compute Optim Appl 2004;28(3):287–310.

[6]. Kennedy and R. C. Eberhart, “Swarm Intelligence”. San Mateo, CA: Morgan Kaufmann, 2001.

[7]. Dorigo M, Blum C. Ant colony optimization theory: A survey. Theoret Compute Sci 2005;344(2–3):243–78.

[8]. A Particle Swarm Optimization approach for routing in VLSI by Nasir Ayob et al.

ISSN: 2278 – 909X