Ant System for Routing in FPGA

Pawan Kumar Dahiya
A.P., ECE Dept.,
Deenbandhu Chhotu Ram
University of Sci. & Tech., Murthal
(Haryana), INDIA

J.S. Saini
EE Dept.,
Deenbandhu Chhotu Ram
University of Sci. & Tech., Murthal
(Haryana), INDIA

Shakti Kumar
ECE Dept.
Baddi University of Emerging
Science & Technology, Baddi
(H.P.), INDIA

ABSTRACT
Routing of arbitrary placed blocks has been a long prevailing objective in any circuit in VLSI. In FPGA, the routing problem becomes more complex due to its fixed routing resources. An efficient routing algorithm tries to reduce the lengths of critical-path nets and also the congestion in the channel to improve the performance of the circuit. This paper presents an Ant System based approach, based on the intelligent behavior of ants, for the routing problem in FPGA. It is observed that the results after some iterations, converge towards the optimal solution at a better rate than other comparable techniques.

Keywords
Field Programmable Gate Array (FPGA), Application Specific Integrated Circuits (ASIC), Routing, Ant Colony Optimization (ACO).

1. INTRODUCTION
Field Programmable Gate Array (FPGA) is one of the the fastest growing sectors of the silicon industry in the recent years. FPGAs are usually slower than their application-specific integrated circuit (ASIC) counterparts, cannot handle as complex a design and draw more power (for any given semiconductor process). But their advantages include a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. They are mostly replacing ASICs where time to market is critical, or where a small production volume wouldn't justify the cost of an ASIC. Moreover, they have also spawned new applications and research interests, from rapid prototyping of VLSI to general-purpose hardware acceleration for numerical applications. FPGA is a semiconductor device having programmable logic components called 'logic blocks', and programmable interconnects. The Logic blocks can be programmed to perform the function of basic logic gates or more complex combinational functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. It is called as field-programmable due to the fact that the logic blocks and interconnects can be programmed by the customer or designer, after the FPGA is manufactured.

FPGAs are flexible and reusable high-density circuits that can be easily reconfigured by the designer, enabling the VLSI design / validation / simulation cycle (as shown in Fig.1) to be performed more quickly and cheaply. These advantages make FPGA an obvious choice for designers for their application.As FPGAs are becoming important implementation platforms for digital circuits. One of the necessary requirements to effectively utilize the FPGA's resources is an efficient placement and routing mechanism. System level ASIC designers are turning to FPGAs for design verification to take advantage of their low cost and fast prototyping [1]. The entire process necessary to implement a circuit in an FPGA consists of the following steps [8, 20]:

- *Logic optimization*: Multi-level minimization of the Boolean equations is done to optimize area and/or delay.
- *Technology mapping*: Boolean equations are transformed into a circuit of FPGA logic blocks involving optimization of the total number of logic blocks required (area optimization) or the number of logic blocks in time-critical paths (delay optimization).
- *Placement*: Specific location for each logic block in the FPGA is located to minimize the total length of interconnects.
- *Routing*: FPGA's routing resources are connected with the logic blocks distributed inside the FPGA by the placement tool.

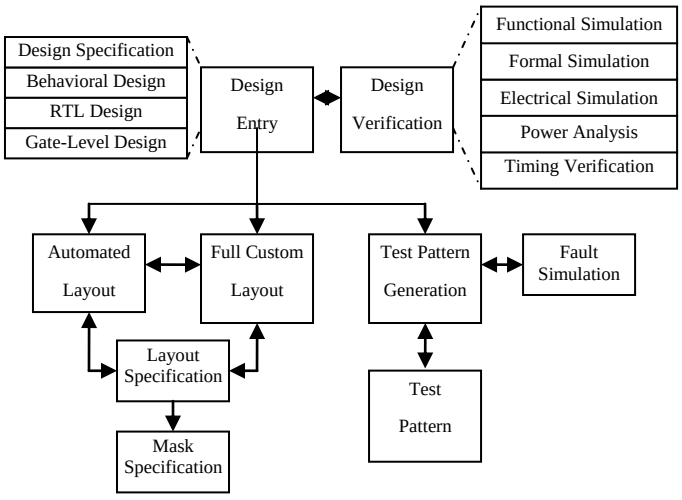


Fig. 1. VLSI System Design Model [16].

Since FPGA utilization typically does not exceed 80%, considerable flexibility remains onboard the FPGA for optimizing the routing. For example, we may wish to reduce signal propagation delay through critical paths by using the most direct interconnections (i.e., shortest paths), where a secondary criterion is to minimize overall wire length in order to reduce circuit capacitance and conserve routing resources.

Routing is an important step of the process as most of the FPGA's area is devoted to the interconnects; and the interconnection delays are greater than the logic delays of the designed circuit. Therefore, an efficient routing algorithm tries to reduce the total wiring area and the lengths of critical-path nets to improve the performance of the circuit. It is an identified NP-hard problem. Thus it is impossible to find optimal solutions in practice and various heuristics are used to obtain a near optimal solution. Obtaining good solutions to the NP-hard problems arising in this domain is crucial for the efficient & optimum utilization of the resources available in an FPGA [7].

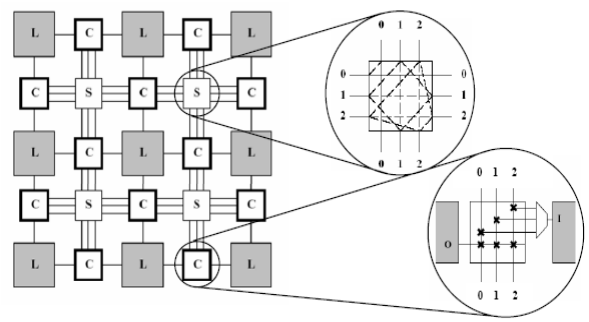


Fig. 2. FPGA Model [2]

FPGA routing is a very complex combinatorial problem. In order to make it manageable, the routing problem is usually solved using the two-stage method of global routing followed by detailed routing. The goals of these two stages are, balancing the channel densities of all routing channels and assigning nets to specify the tracks and internal switches, respectively.

The Global Optimization methods can be classified into two main categories: deterministic and probabilistic methods. The deterministic methods since suffer from the curse of local optimality, they have to resort to some application of heuristics, such as modifying the trajectory (trajectory methods) or adding penalties (penalty-based methods), to escape from the local minima. On the other hand, probabilistic methods rely on probabilistic judgments to determine whether or not search should depart from the neighborhood of a local minimum.

It has been a natural desire for men to expect greater & greater functionality, speed throughput and intelligence from his machines. One of the paradigms to bestow the machines with the aforementioned metrics of performance to empower the machines to be able to evolve just as the living species has evolved. A new optimization technique called Ant colony Optimization (ACO) has emerged very recently and has allied to evolutionary algorithms based on the collective behavior and intelligence of ants [13]. It is applicable to complex combinatorial optimization problems.

This contribution of the paper is as under:

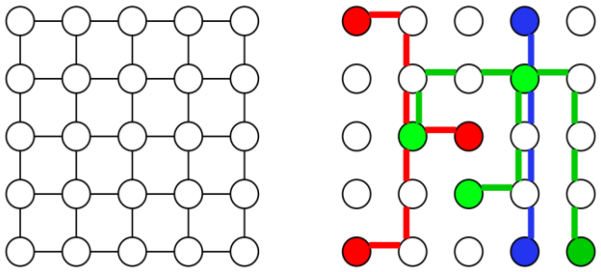
- The issues in routing of FPGA are detailed with all affecting parameters.
- A solution of routing problem in FPGA by means of AS, a typical ACO algorithms, is presented.
- The results of AS are also compared with the results of GAs for same problem.

The rest of the paper is organized as under:

The routing problem is detailed in section 2. Section 3, details about the issues arising while dealing with the routing problem. The ACO is detailed in section 4. Section 5 details the ant system and its applicability for routing problem in FPGA. The results are discussed in section 6. The paper is concluded and future directions are presented in section 7.

2. ROUTING PROBLEM

It is just obvious that FPGA routing is a complex problem and even though historically it has been underestimated by VLSI designers (speculating that fixed routing resources should make the routing easier), it has been the entirely to the contrary. The fixed routing resources rather make routing in FPGA a much harder problem since multiple and all constraints have to be satisfied to successfully route the design [2].



(a) FPGA before routing (b) FPGA after routing

Fig. 3. Global Routing [7]

Most approaches to FPGA routing have been based on the divide and conquer approach, in which the routing has been split in two phases, a global route that space out the track requirement throughout the FPGA and a detail router that does the actual assignment of routing resources. Among these two phases, the detail router is a much harder problem as it has to consider in depth the architecture of the FPGA. The details are as under:

2.1 Global Routing

The global router performs a coarse route to determine, for each connection, the minimum distance path through routing channels that it has to go through. If the net to be routed has more than two terminals the global router will break the net into a set of two-terminal connections and route each set independently.

Once all connections have been coarse routed, the solution is optimized by ripping up and re-routing each connection a small number of times. After that, the final solution is passed to the detailed router.

2.2 Detailed Routing

The detail router determines, for each two point connection, the specific wiring segments to use in the routing channel assigned by the global router. To do this, the detail routing algorithms construct a directed graph from the routing resources to represent the available connections between wires, C blocks, S blocks and logic blocks within the FPGA as shown in Fig. 2.

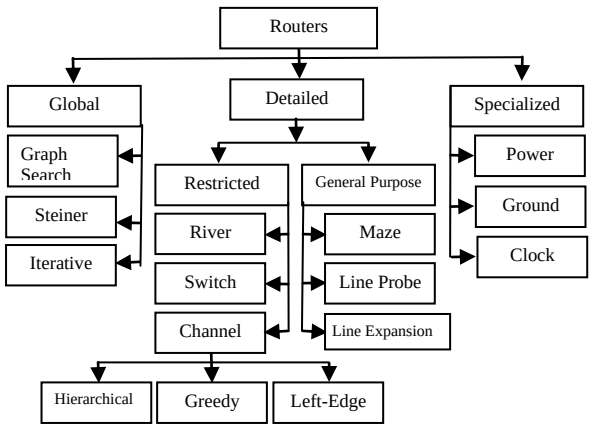


Fig. 4. Classical Routing Techniques.

The search performed on this directed graph is usually based on Dijkstra's algorithm [9] to find the shortest path between two nodes. The paths are labeled according to a cost function that takes into account the usage of each wire segment and the distance of the interconnecting points. The distance is estimated by calculating the wire length in the bounding box of the interconnecting points using a Manhattan metric. Most of the routers relax the bounding box constraints and allow searching for possible solutions in the surrounding routing channels of the bounding box. This is done to avoid subsequent iterations of ripping out and re-routing if the solution lies on the nearby space outside of the bounding box.

In a simultaneous place-and-route algorithm, often global routing is blended with the cell placement algorithm in an intertwined fashion. The detail of various classical routing techniques [7] is shown in Fig. 4.

3. ISSUES IN ROUTING

The main issues [18] which need to be tackled in the detailed & global routing are as follows:

3.1 Wire-Length

As discussed earlier, minimizing the length has long been the prevailing objective in VLSI routing, since a minimum-length interconnection occupies the minimum amount of area and has minimum overall capacitance and resistance. Although recent advances of integrated circuit technology into the deep-submicron realm have introduced additional routing objective functions, minimizing length remains the most important objective for non-critical nets and in physically small instances [2].

3.2 Delay

As VLSI technology scales to smaller feature sizes and larger layout areas, propagation delay increasingly dominates delay through switching devices. In performance-driven routing, one seeks to control the propagation delay between the source and a specified set of sinks.

3.3 Congestion

The global router considers for each connection multiple ways of routing it and chooses the one that passes through the least congested routing channels. By keeping track of the usage of each routing channel, congestion is avoided; and the principal objective of the global router, balancing the usage of the routing channels, is achieved.

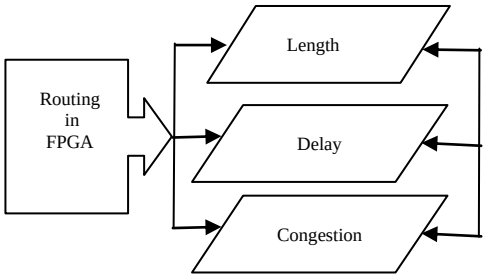


Fig. 5. Issues in Routing [16, 17]

It is discernible from the above discussion that keeping track of optimality is very difficult to achieve by subsuming all of these constraints. It is an NP-hard problem and needs to be tackled by some meta-heuristics only.

4. ANT COLONY OPTIMIZATION

Ant colony optimization (ACO) is a paradigm for designing meta-heuristic algorithms [6, 10, 11, 12, 13, 19], inspired by the intelligence of real ants, for finding solutions to combinatorial optimization problems. Ants as individuals are unsophisticated living beings. However, their collective behavior is an intelligent one. The ants do not possess abilities such as memory of past actions and knowledge about the distance to other locations. In nature, an individual ant is unable to communicate or effectively hunt for food, but as a group, ants possess the ability to solve complex problems and successfully find and collect food for their colony. The adoption of the strategies of ants adds another dimension to computational domain.

The ants communicate using a chemical substance called pheromone. As an ant travels, it deposits a constant amount of pheromone that other ants can follow. Each ant moves in a somewhat random fashion, but when an ant encounters a pheromone trail, it must decide whether to follow it. If it follows the trail, the ant's own pheromone reinforces the existing trail, and the increase in pheromone increases the probability of the next ant selecting the path. Therefore, the more ants travel on a path, the more attractive the path becomes for subsequent ants. Additionally, an ant using a short route to a food source will return to the nest sooner and therefore, mark its path twice, before other ants return. This directly influences the selection probability for the next ant leaving the nest. Over time, as more ants are able to complete the shorter route, pheromone accumulates faster on shorter paths and longer paths are less reinforced and ultimately abandoned.

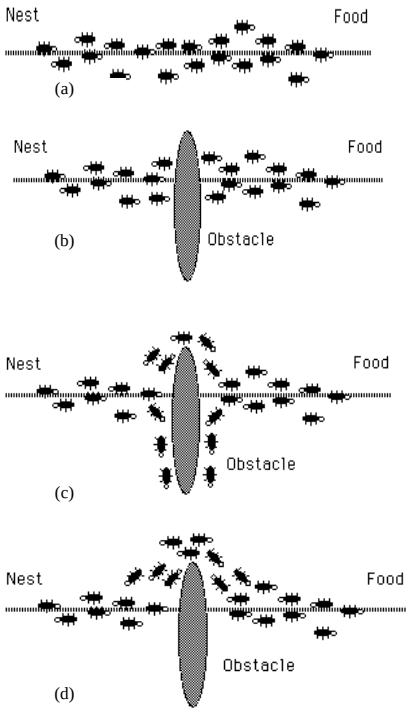


Fig. 6: An example with real ants. (a) Ants follow a path between points Nest and Food; (b) An obstacle is interposed; (c) ants can choose to go around it following one of the two different paths with equal probability. (d) On the shorter path more pheromone is laid down early.

Ant colony optimization algorithms have been used to produce near-optimal solutions to the traveling salesman problem [14, 15]. They have an advantage over simulated

annealing and genetic algorithm approaches when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

5. ANT SYSTEM

Ant system (AS) was the first ACO algorithm proposed in the literature. Its main characteristic is that the pheromone values are updated by all the ants that have completed the tour.

5.1 Route construction in Ant System

In ant system, an ant *k* will move from node *i* to node *j* with probability

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{\ell \in N_i^k} [\tau_{il}]^{\alpha}.[\eta_{il}]^{\beta}}, & if\ j = N_i^k \\ 0, & otherwise \end{cases}$$

where, τ_{ij} is the amount of pheromone on arc *i, j*; α & β are parameters to control the influence of τ_{ij}; $\eta_{ij} = 1/d_{ij}$ is a value depending upon the heuristic bias & it is available a priori; N_i^k is the feasible neighborhood of ant *k* when being at city *i*, i.e., the set of cities that ant *k* has not visited yet.

5.2 Pheromone Update in Ant System

In ACO algorithms [13, 14, 15, 21], one can find different types of pheromone updates. The pheromone update consists of two parts. First, a *pheromone evaporation*, which uniformly decreases all the pheromone values, is performed. From a practical point of view, pheromone evaporation is needed to avoid a too rapid convergence of the algorithm towards a sub-optimal region. It implements a sort of a useful form of forgetting, favoring the exploration of new areas in the search space. Then, one or more solutions from the current and/or from earlier iterations are used to increase the values of pheromone trail parameters on solution components that are part of these solutions. As an example, we outline in the following the pheromone update rule that was used in Ant System (AS). The pheromone evaporation is implemented by:

$$\tau_{ij} = (1-\rho).\ \tau_{ij}$$

where, $0 < \rho \leq 1$ is the pheromone evaporation rate. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

and $\Delta\tau_{ij}^{k}$ is the amount of pheromone ant *k* deposits on the arcs it has visited, typically given by:

$$\Delta\tau_{ij}^{k} = Q / L^{k} \quad if\ (i, j) \in T^{k}\ else\ 0.$$

where L^k is the cost of the k^{th} ant's tour (typically length) and T^k is the tour done by ant *k*, *Q* is a heuristic bias parameter depending upon the best cost factor.

5.3 Ant System Procedure

The pseudo-code for the aforesaid actions of ants can be phrased as follows:

```
procedure ACO_Routine
    while(not_termination)
        generateSolutions( )
        pheromoneUpdate( )
        daemonActions( )
    end while
end procedure
```

The AS is an identified technique for shortest path algorithm and traveling salesman problem [11], routing problems in networks [12] etc. It thus holds the potential to counter the delay related issues which arise due to longish data-paths.

5.4 AS for routing problem

As discussed earlier, Routing is an NP-hard problem that is generally separated in to two phases using the divide and conquer paradigm [8]: a global routing that balances the densities of all routing channels, and a detailed routing that assigns specific wiring segments for each connection. These two phases avoid congestion and optimize the performance of the circuit, making sure that all nets are routed minimizing wire-length and capacitance on the path. By running both algorithms, a complete routing solution can be created [17].

A router must determine paths for all connections in a net-list description while minimizing wire length and area, and satisfying resource constraints. Typically, a router attempts to minimize the wire length when the output net of one component connects to several other components by using a tree structure. Both the placement and routing problems are known to be NP-hard problems. There has been a lot of work on optimization for placement and routing, including genetic algorithms for both placement [1, 2, 3] and routing [2, 5, 10]. These problems are almost always solved independently, using various methods to estimate wire length during placement. A global routing algorithm for FPGAs is described in [4]. For ACO based routing, the following assumptions [16] are made in the present work:

- The distances between the CLBs (Configurable Logic Blocks) and IOBs (Input Output Blocks) are taken in terms of the normalized units.
- Congestion of the channels is not considered for routing.
- All channels are of equal capacity.

5.5 Fitness Function

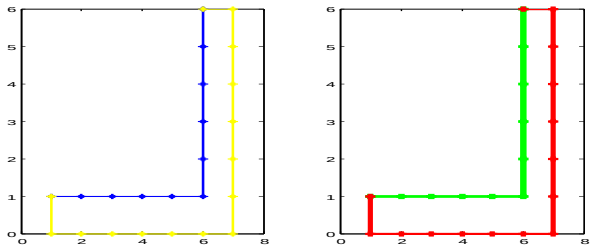
In order to compare the different candidate solutions AS is implemented with a common fitness function defined as under:

$$Fitness = [(row\ j - row\ i) + (column\ j - column\ i)]$$

Where *row i & column j* indicate the coordinates of the initial and final instances in FPGA.

6. RESULTS AND COMPARISON

In our simulation experiment, 100 ants are randomly initialized for routing problem of CPU (Central Processing Unit) circuit to be targeted to Xilinx FPGA (XC3S50).



(a) Route before pheromone update (b) Routeafterpheromone update

Fig. 7. The result of Ant System

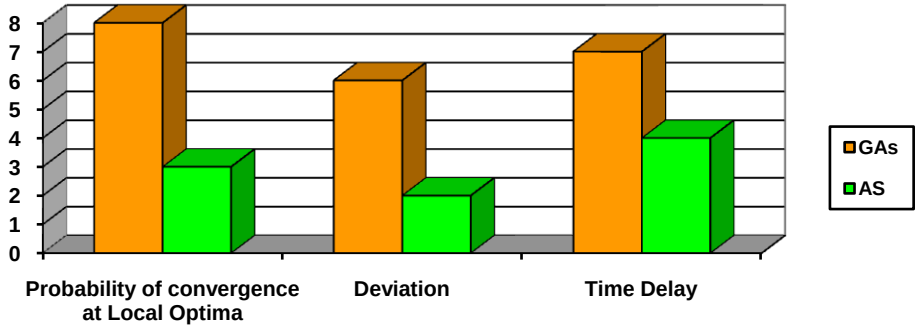


Fig. 8. Comparison of Ant System with Genetic Algorithms in terms of probability of convergence at local optima, deviation from optimum result and time delay

The results are compared after 100 iterations. The optimized paths for Ant System are shown in Fig.7. The left part of Fig.7 shows the two alternative paths. While the right part of Fig. 7 shows that there is difference in the quantity of pheromone (indicated through the colour /darkness) on these paths after 100 iterations. This clearly indicates the higher deposition and lesser evaporation of pheromone on shorter path. The shorter one is having more quantity of pheromone than the other one.

The results of AS are also compared with the that of GAs [7] in terms of probability of convergence at local optima, deviation from optimum results and time delay in the execution of the algorithm. These are tabulated in Fig. 8. In the passing, it is appropriate here to mention that, the result of AS is better than the GAs for the above said benchmark problem.

7. CONCLUSION AND FUTURE DITRECTIONS

After analyzing the specific result as above, it is obvious that AS is an effective optimization tool based on the collective behavior of ants for optimization of routing in FPGA. It has the potential of outperforming the traditional optimization techniques (by incorporating the features of animals such as collective intelligence, knowledge, experience, etc.) for complex combinatorial problems. Although this technique is still in its nascent stage of implementation for complex combinatorial optimization problems, yet it has already begun to show the promise of providing better results than its parallel techniques for some benchmark problems.

8. REFERENCES

[1] K. Roy-Neogi and C. Sechen, "Multiple FPGA Partitioning with Performance Optimization," *Proceedings of the 3rd International ACM Symposium on Field-Programmable Gate Arrays*, pp. 1-7, 1995.

[2] Pinaki Mazumdar and E.M. Rudnik, *Genetic Algorithm for VLSI Design, Layout and Test Automation*. Delhi: Pearson Education, 2003.

[3] P. Thomson and J.F. Miller, "Optimisation Techniques based on the use of Genetic Algorithms (GAS) for Logic Implementation on FPGAs," *The Institution of Electrical Engineers*, pp. 411-414, 1994.

[4] G.K. Venayagamoorthy and V.G. Gudise, "Swarm Intelligence for Digital Circuits Implementation on Field Programmable Gate Arrays Platforms," *Proceedings of the NASA/DoD Conference on Evolvable Hardware (EH'04)*, pp. 1-4, 2004.

[5] Yee Leung, Guo Li, and Zong-Ben Xu, "A Genetic Algorithm for the Multiple Destination Routing Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 2, No. 4, pp. 150-161, 1998.

[6] M. Nakao and K. Izumi, "Circuit Designs and Fabrication of Swarm-Intelligence LSIs based on Modeling Foraging Behaviors of Ants," *The 47th IEEE International Midwest Symposium on Circuits and Systems, IEEE (*0-7803-8346-X/04), pp. 97-100, 2004.

[7] Shakti, Kumar, J.S. Saini and Pawan Dahiya, "GAs for Routing in VLSI," *Proceedings of International Conference on Intelligent System and Networks*, ISTK, Jagadhri, pp. 20-25, 2008.

[8] D. G. Prado and M. Ciesielski, "A Tutorial on FPGA Routing," *Dept. of Electrical and Computer Engg., University of Massachussets*, pp.1-16.

[9] Russell G. Tessier , "Fast Place and Route Approaches for FPGAs," *Ph.D. thesis, Massachussets Institue of Technology*, 1999.

[10] V.K. Jayaraman et.al., "Ant colony framework for optimal design and scheduling of batch plants" *Computers and Chemical Engineering-24*, pp. 1901-1912, 2000.

[11] J. E. Bell and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Advanced Engineering Informatics-18*, pp. 41–48, 2004.

[12] K. M. Sim and W. H. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *IEEE Transactions on Systems, Man, And Cybernetics—Part A: Systems and Humans*, Vol. 33, No. 5, pp. 560-572, 2003.

[13] Marco Dorigo et.al., "Ant system: optimization by a colony of cooperating agents" *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, Vol. 26, No. 1, pp. 29-31, 1996.

[14] Marco Dorigo et.al., "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, *I (*1), pp. 53-66, 1997.

[15] Marco Dorigo and Thomas Stutzle, *Ant Colony Optimization*. New Delhi: Prentice-Hall of India, 2005.

[16] Pawan Dahiya, J. S. Saini, and Shakti Kumar, "PSO for Routing Problem in VLSI," *International Conference on Intelligent Systems & Networks (ISN-2008)*, 2008.

[17] Pawan Dahiya, J. S. Saini, and Shakti Kumar, "Evolutionary Trends in VLSI Design," *Proc. of International Conference on Intelligent Systems & Networks (IISN-2007*), 2007.

[18] Pawan Kumar Dahiya, "Recent Trends in Evolutionary Computation," Ph.D. dissertation, ECE Dept., M. D. Univ., Rohtak, Haryana, India, 2011.

[19] Vineet Chaudhary, Parveen Kumar Yadav, and Pawan Kumar Dahiya, "Evaluation of cost optimization for travelling salesman problem (TSP) by comparing various ant colony optimization (ACO) algorithms," *International Journal of Computational Engineering & Management, Vol. 15, No. 5, pp. 38-44, 2012.*

[20] Daniel Gomez-Prado Maciej Ciesielski*, "*A tutorial on FPGArouting,"Availableat:*http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.9313&rep=rep1&type=pdf.*

[21] John E. Bella, Patrick R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Advanced Engineering Informatics 18*, pp. 41–48,2004.