

Cuckoo Search Optimization Algorithm based Hardware Task Placement and Routing in CAD of FPGAs Design Flow

B. Premalatha* and S. Umamaheswari

Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore – 641014, Tamil Nadu, India; premalathaece.cit@gmail.com, umamaheswari.cit@gmail.com

Abstract

In this paper, Cuckoo search optimization algorithm based placement of tasks in FPGAs is introduced and was compared with PSO, ARPSO and GSA. Both single and multiobjective constraints based fitness function is formulated. The simulation result shows that CSO outperforms the others. By tuning the design parameters of CSO algorithm, the FPGA Placement process performs better.

Keywords: FPGA Placement, Optimization, Single and Multiobjective Constraints, Wirelength and Area Minimization

1. Introduction

¹Design Process steps in FPGA based system design involves logic synthesis, technology mapping, placement and routing. PaR is the crucial step in CAD flow of VLSI design process steps that affects the overall performance of design. For proficient use of FPGAs, the CAD tools with excellent algorithm to execute the design process steps in well-organized way. In this paper, we concentrate on introducing such optimization algorithm for placement process with objective to minimize the wire length, area and both. As the placement process directly affects the routing phase, the cost function in this paper includes the routing information also. PaR is an NP-Complete problem which can be solved by using optimization algorithms. ²Simulated annealing is one of the heuristics developed to solve the placement problem and acts as VPR Tool in industries. Many optimization algorithms such as Genetic algorithm, Particle Swarm Optimization, Firefly algorithm etc., have also been developed to solve the placement Problem. In this paper we also concentrate on double constrained task placement using CSO, GSA, ARPSO algorithm and simulation results are compared

with PSO. The rest of the paper is organized as follows: Section II includes the design process steps involved in CAD of VLSI Design, Section III includes the description of the architecture of Island based Homogeneous FPGAs followed by Hardware Task modeling. Section IV includes the definition of cost metrics of PaR used. Section V involves brief explanation about the optimization algorithms such as PSO, ARPSO, GSA and CSO. Section VI includes the experimentation details and simulation results followed by comparison table and discussion. Conclusion of this paper is discussed in Section VII.

2. CAD of VLSI Design Process Flow

A typical design flow for Digital System circuits design is shown in Figure 1. To implement any application in FPGAs, it involves the various steps as indicated in design flow diagram. Step1 involves the complete description of abstracting the functionality, interface and overall architecture of the application in digital form followed by analysis In terms of functionality and performance that complaints to standards and high level issues. Designer

*Author for correspondence

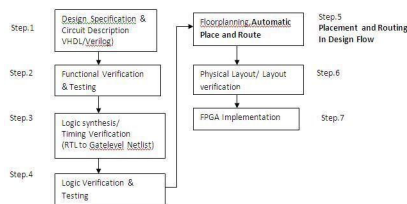


Figure 1. CAD Flow of VLSI Design.

has to describe the dataflow/behavior of desired circuit design RTL description using HDLs such as VHDL/ Verilog. Functional verification and testing is carried out in test bench form by applying simulation vectors in Step 2. Step 3 involves the conversion of RTL description to gate level net list that consists of the information about the logic gates in the circuits and its interconnection and this conversion process is called Logic Synthesis. Logic synthesis should ensure that the gate level net list meets the timing, area and power specification followed by logic verification and testing in Step 4. Gate level net list is the input to a PaR tool, which is the crucial step after floor planning because the overall performance of system gets affected by the PaR performance. VPR tool in FPGAs used in Industries is based on Heuristic called Simulated Annealing which purely depends on the concept of temperature cooling. But still there is requirement in effective PaR optimizer for FPGAs. PaR optimization in VLSI Design can be achieved in various levels like logic level, circuit level and algorithmic level of design. Once the tasks related to application is localized in cells of FPGAs with their interconnections routed (called placement and Routing), the next Step is physical layout formation of the architecture, solving so many issues and verification should be done at this level. Finally in step 7, the layout information is implemented in FPGA. In all the above steps, the Placer and Router play a major role which decides the overall performance of the input application. Factors like wirelength, delay, area and power are the metrics for Placement and Routing Process. PaR process requires most optimized approach to minimize the metrics. PSO and various optimization approaches are applied for FPGA PaR process. In this paper, Attractive & Repulsive PSO (ARPSO) algorithm, Gravitational Search Algorithm (GSA), Cuckoo search Optimization algorithm (CSO) based Hardware tasks placement process are newly introduced with constraints of minimizing the wirelength, area and both.

3. Island based Homogeneous FPGA Architecture

Any kind of digital system or circuit can be implemented by electrically programming the prefabricated silicon device called Field Programmable Gate arrays (FPGAs). Reconfigurability, Fast turn around and low complexity make FPGAs are advantageous than ASIC design. Architecture and CAD tool together distinguish the FPGA in VLSI design Community. FPGA architecture consists of array of Logic cells surrounded by the Switch blocks, Connector matrix (the intersection of horizontal and vertical channels) used for routing the logic cells and I/O Pads. Homogeneous FPGAs consists of logic cells of same type but the heterogeneous FPGAs consists of various special purpose blocks such as multipliers, Memories etc., in addition to logic cells and I/O Pads. Switch block S and connector matrix C together forms routing fabric that programmably interconnects the logic cells for the execution. Once the FPGA is said to be programmed for an application, the function can be directly programmed into the chip after fabrication. Changing the behavior of pre-fabricated chip as per the system design by the user is called programmability. Programming Technologies differ the FPGAs in Market. The architecture of Island based Homogeneous FPGAs is shown in Figure 2.

3.1 Modelling of Hardware Tasks

Placement is defined as the process of assigning the Logic cells called CLBs (Configurable Logic Blocks) for particular application execution inside the FPGAs with proper routing(interconnections) using minimal wirelength and area. Minimal area influence the chip size as small as a result, high density ICs are fabricated in single silicon die. Minimal wirelength reduces the net delay of the circuit. Netlist created by the Logic Synthesis steps in VLSI Design Flow is given as an input to a Placer and router

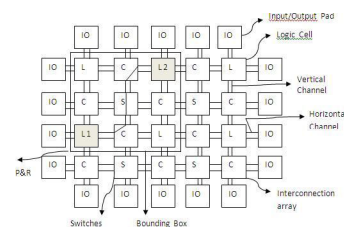


Figure 2. Architecture of Island Based Homogeneous FPGAs.

and the experimentation should be carried out to obtain the optimal wirelength and area. In this paper, netlist is modeled as a directed data flow graph (DDFG), in which each node represents tasks and the edges represent the interconnection between the tasks. An interconnection details is represented as Task Dependency Matrix (TDM). A sample DDFG and its interconnection matrix used in experimentation is shown in Figure 3.

In the Figure 3, DFG consists of five tasks and the dependence between the tasks are represented in task dependence matrix (TDM) of size $N \times N$, where N = number of tasks in an DFG. In TDM, logic '0' represents the presence of Interconnection and logic '1' represents there is no interconnection. For example: Row 1 details indicates the task 1 is connected with task 2 and task 3, task 2 is connected with task 4, task 3 is connected with task 4 and task 5. No interconnection exists from task 4 and task 5.

4. Definition of Hardware tasks PaR and its Fitness Function

By assuming the initial locations of these tasks, the cost function is calculated. Optimize the Cost function using global optimization method such as PSO, ARPSO, GSA, CSO etc., Generally, Routing can be carried out by using the switch block and connector matrix channel inside FPGA. Basically, Switches are used to route the signals between the logic cells through channels and switches. If the number of channel tracks in both vertical and horizontal tracks is W , then $(W/2)-1$ Possibility of connections are there for each track. So totally, there are $4 * (W/2)-1$ times the routing path is available. For example: If $W = 8$, Number of possible interconnections are 3 for each track. Total number of routing paths are 12/switch but based on length of interconnection wirelength. The schematic representation of Switches with their channel tracks are shown as follows: Various routing directions are shown in Figure 4.

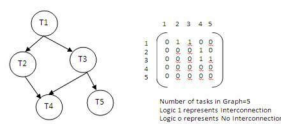


Figure 3. An Sample Directed Data Flow Graph (DDFG) and its Task Dependency Matrix (TDM).

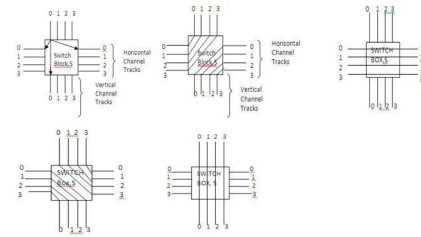


Figure 4. Various Routing Possibilities between Switch Boxes through connectors.

FPGA Architecture after placement and routing is shown for example in Figure 2. L1 is the Logic cell where the task1 is placed and L2 is the logic cell where the task 2 is placed. For routing, it utilizes one switch and two connector matrixes. Bounding box is the area that occupied by the application for execution.

5. Fitness Function

Fitness function of the NP-Complete Placement problem is defines as the minimization of metrics such as wirelength and area. It is denoted by CF and is given by

$$CF = \alpha.CF1 + \beta.CF2, 0 < \alpha, \beta < 1; 0 < \alpha, \beta < 1 \quad (1)$$

Where CF1 is the wirelength fitness function calculation of the circuit calculated by using Euclidean Distance formulae, Perimeter calculation based wirelength estimate and CF2 is the area calculation occupied by the circuit with interconnections using bounding box formulae.

$$CF1 = \min (CF) \quad (2)$$

$$\text{Euclidean Distance} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (3)$$

(x_1, y_1) and (x_2, y_2) are co-ordinates of CLBs position in 2 Dimensional FPGA. For experimentation CLBs are numbered from 1 starts from Bottom left CLB. Upper and lower bound of FPGA will be 1 to $W * H$.

$$CF2 = \text{area fitness Function} = \min(\text{bounding box})$$

$${}^2\text{boundingbox} = \sum q(i) [(bbx(i) + bby(i)) / (cav, x(i) + cav, y(i))], I = 1 \text{ to number of tasks nets} \quad (4)$$

$bbx(i)$ and $bby(i)$ are the horizontal and vertical span in x and y direction of nets. cav is equal in x and y directions which is equal to 50 each.

6. Optimization Algorithms

As the Placement and routing phase of FPGA Based design flow requires an effective optimizer for incoming application execution, the various heuristics are applied to minimize the wirelength, area and combination of both. In this section, we discuss about the ³Particle Swarm optimization (PSO), Attractive Repulsive PSO algorithm (ARPSO), Cuckoo Search Optimization (CSO) algorithm and Gravitational Search Optimization (GSA) algorithm and its details for Optimised Placement and routing.

6.1 Particle Swarm Optimization Algorithm based PaR

As Simulated annealing requires more iterations to converge and ⁴GA involves so many steps, optimization problems solving is focused on Particle Swarm Optimization (PSO) algorithm. PSO⁵⁻¹¹ is a stochastic search, population based global optimization technique developed by Kennedy & Eberhart in 1995, where swarm of particles are considered as population where individual particle in swarm represents candidate solution. It can handle optimizing the functions in high dimensional search space. Concept of PSO based on birds flocking behavior, where the birds are attracted towards the direction of destination in proper way. Flocking behavior arises from the interaction of rules such as collision avoidance, velocity matching and flocking centering. PSO algorithm is a simplified model of determining the nearest neighbors and velocity matching. At each iteration, each individual determines its nearest neighbor and replaces its velocity with its neighbor. Particle position is updated by using velocity equation after each iteration and find the fitness function till the minimal value obtained. Various constraints are there in PSO, which the parameter selection makes more diversible in search space from the best solution. So care must be taken in selecting the parameters. Final solution purely depends on PSO parameters, for experimentation the parameters are swarm size, Particle dimension, Number of iterations, acceleration co-efficient, inertia weight, cognition factors c_1 , c_2 to decide the model. The value of c_1 and c_2 must be equal but greater than 4 is appreciable for smooth optimization. Search process in PSO involves the Position and velocity updates. Let X be the N-Dimensional Vector, in which each vector element represents the solution point in search space. $X_i = \{X_1, X_2, X_3, \dots, X_N\}$. Where I = swarm size. A swarm

of such particle is considered for experimentation. The position of the candidate solution has to be updated by using the equation Where $V_i(t)$ is the velocity factor.

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

$$V_{ij}(t+1) = W * V_{ij}(t) + c_1 * r_1(t)(y_{ij}(t) - x_{ij}(t)) + c_2 * r_2(t)(Y_{ij}(t) - X_{ij}(t)) \quad (6)$$

Where c_1 and c_2 are acceleration co-efficients, $r_1(t)$ and $r_2(t)$ are random values between (0,1), $Y_{ij}(t)$ is the global best parameters, $y_{ij}(t)$ is the local best parameters and W is the weight inertia factor depends on number of iterations.

6.2 Attractive and Repulsive PSO (ARPSO) Algorithm based PaR

ARPSO¹² is an modified version of PSO algorithm in which the velocity updation is carried out based upon the Diversity factor D . A single swarm is used in which the candidate solutions are switched between the two phases depending upon Swarm Diversity.

$$\text{Diversity}, D = 1/ns \sum \sqrt{(\sum (x_{ij}(t) - \text{avg}(X_j(t)))^2)} \quad (7)$$

Where i varies from 1 to swarm size, ns j varies from 1 to dimension of each particles in swarm $X_j(t)$ is average of j th Dimension overall particles i.e.,

$$\text{Avg}(X_j(t)) = \sum X_j(t) / ns \quad (8)$$

The threshold $\Phi_{\min}$ and $\Phi_{\max}$ is fixed, if Diversity, D is $> \Phi_{\min}$ switch the velocity updation to attraction mode. If Diversity D is $< \Phi_{\min}$ then velocity updation switches to repulsion phase. In normal velocity updation equation in PSO, the new constant variable k is introduced for which the value of k is 1 for attraction mode and k is 0 for repulsion mode. Number of Iterations has to be fixed

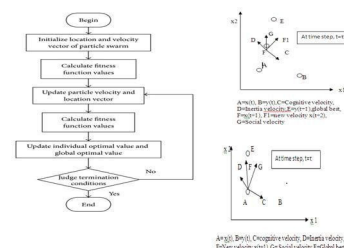


Figure 5. Flowchart and Geometric illustration of PSO position, Velocity Updation.

and the process continues till the iteration over to find the minimal fitness value. ARPSO outperforms than PSO.

Velocity update equation for repulsion Phase:

$$V_{ij}(t+1) = W * V_{ij}(t) - c1 * r1(t)(y_{ij}(t) - x_{ij}(t)) - c2 * r2(t)(Y_{ij}(t) - x_{ij}(t)) \quad (9)$$

Velocity update equation for Attractive Phase:

$$V_{ij}(t+1) = W * V_{ij}(t) + c1 * r1(t)(y_{ij}(t) - x_{ij}(t)) + c2 * r2(t)(Y_{ij}(t) - x_{ij}(t)) \quad (10)$$

6.3 Cuckoo Search Optimization (CSO) Algorithm based PaR

Cuckoo search¹³⁻¹⁶ is correspondingly a recent Meta heuristic algorithm developed by Xin-She Yang and Suash Deb in the year 2009, to be proficient for solving global optimization problem. It deploys on broad parasitism and enhanced by Levy flight, more preferred than by isotropic random walks. Current research works on cuckoo search is prospectively more efficient than PSO. Cuckoo is interesting bird, because of its dynamic reproduction policy in addition to wonderful sound they make. There are three idealized rules to describe the standard cuckoo search algorithm as follows:

- Each one of cuckoo lays one egg at a time and dumps in unmethodically chosen nest.
- The premier nest with superior eggs will be carried over to the subsequent generation.
- The number of available host nest is fixed, the egg laid by the cuckoo is observed with the probability P_a = 0.25, by the host bird.

While in this case, the host birds, neither gets rid of the egg, nor plainly discards the nest and build utterly new nest. In the implementation point of view, each egg in the nest is being considered as the solution. Each egg

that is being laid by cuckoo is considered as one solution. The main aim of the CSO is to use new and better solution (cuckoo) to replace a worst solution in nest. This algorithm uses a balanced mixture of local random walk and the global explorative random walk controlled by P_a . The equation for the local random walk can be written as

$X_i(t+1) = X_i(t) + K \cdot \text{stepsize}$, where K represents fraction of p_a worst nests to be abandoned. Stepsize will be the random walk by random permutation of solutions according to the similarity/difference to the host solutions.

$$K \cdot \text{stepsize} = H(P_a - \epsilon) * (X_i^t - X_i^t). \quad (11)$$

6.4 Gravitational Search Algorithm (GSA) based PaR

Gravitational Search algorithm¹⁷ proposed by Rashedi et al. in 2009, based on the physical law of gravity and the law of motion. The gravitational force is directly proportional to product of their masses and inversely proportional to the square of the distances between them. All possible solutions called agents named masses has been proposed to find the optimum solution. Masses or agents are called possible solutions are represented in the form of vector $x_{im} = \{x_{i1}, x_{i2}, x_{i3}, \dots\}$.

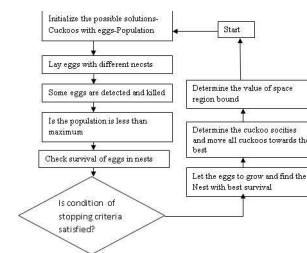


Figure 7. Flow chart of CSO Algorithm

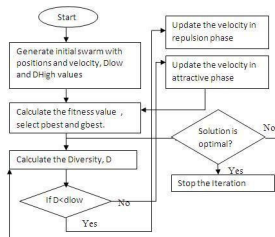


Figure 6. Flowchart of ARPSO Algorithm.

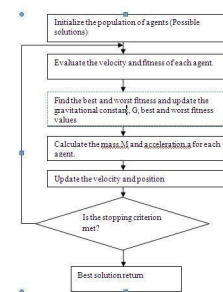


Figure 8. Flow chart of GSA.

Table 1. Multi-objective wirelength and area optimization using PSO, ARPSO, GSA and CSO

S.No	Exper.al DDFGs	FPGA Size	Total CLBs	Nodes/Popsize/Maxiter	Wirelength and Area minimization			
					PSO	ARPSO	GSA	CSO
Wirelength and Area optimization in FPGA Placement								
1.	DDFG1	8x8	64	10/10/1000	13.74	13.32	13.55	10.62
2	DDFG2	8x8	64	10/10/1000	12.24	12.30	12.30	11.60
3.	DDFG3	10x10	100	20/10/1000	30.90	32.40	29.40	10.90
4.	DDFG4	8x8	64	10/10/1000	13.40	13.20	11.12	12.50
5.	DDFG5	10x10	100	40/10/1500	52.27	52.12	45.90	10.70
6.	DDFG6	10x10	100	20/10/1000	30.60	29.00	27.10	11.55
7.	DDFG7	8x8	64	20/10/1000	35.00	33.10	31.90	11.24
8.	DDFG8	8x8	64	30/10/1000	42.00	42.00	36.00	10.40

Betterment : CSO > GSA>ARPSO>PSO

Table 2. Single objective Wirelength /Area Minimization using PSO, ARPSO, GSA and CSO

S.no	Exper.al DDFGs	FPGA Size	Total CLB	Nodes/Popsize/Maxiter	Wirelength minimization (ED)			
					PSO	ARPSO	GSA	CSO
Case 1:Wirelength minimization based on Euclidean distance metric for FPGA Placement								
1.	DDFG1	8x8	64	10/10/1000	36.40	34.40	36.40	27.10
2	DDFG2	8x8	64	10/10/1000	39.40	35.80	32.80	25.90
3.	DDFG3	10x10	100	20/10/1000	144.2	122.9	134.2	81.50
4.	DDFG4	10x10	100	40/10/1000	336.0	334.0	299.0	233.0
5.	DDFG5	10x10	100	40/10/1500	235.4	229.6	207.1	141.3
6.	DDFG6	10x10	100	20/10/1000	121.7	118.5	97.30	73.10
7.	DDFG7	8x8	64	20/10/1000	133.4	137.2	115.9	96.76
8.	DDFG8	8x8	64	30/10/1000	183.8	175.5	163.9	126.7
Case 2:Area minimization based on bounding box metric for FPGA Placement								
1.	DDFG1	8x8	64	10/10/1000	0.867	0.765	0.707	0.568
2.	DDFG2	8x8	64	10/10/1000	0.615	0.504	0.590	0.392
3.	DDFG3	10x10	100	20/10/1000	2.452	2.527	2.040	1.728
4.	DDFG4	10x10	100	40/10/1000	5.960	5.880	4.630	4.330
5.	DDFG5	8x8	64	30/10/1000	2.652	2.850	2.409	2.113
6.	DDFG6	10x10	100	40/10/1500	3.410	3.440	3.123	2.680
7.	DDFG7	10x10	100	20/10/1000	2.070	1.896	1.816	1.467
8.	DDFG8	8x8	64	20/10/1000	2.292	2.000	2.609	1.750

Betterment : CSO > GSA>ARPSO>PSO

--- x_{im} }, $I = 1, 2, 3 \dots N$. where N is the population size for optimization. Search space dimension should be more than I and two-dimensional. FPGA area of size $W \times H$ is the search space and the number of tasks represents the dimension. Including the information with

tasks interconnections, the fitness calculation, updating velocity is carried out to find the optimum solution. Equations involved in GSA Algorithm are listed below. The flow chart of Gravitational search algorithm is shown in Figure.8

Force equation: $F_{ij} = G(t) [M_{pi}(t) * M_{aj}(t)] / [R_{ij}(t) + \epsilon] * [x_j(t) - x_i(t)]$

Gravitational constant equation: $G(t) = G_0 * \exp(-at/T)$

Acceleration equation: $a_i(t) = F_i(t) / M_i(t)$, Velocity equation: $V_i(t+1) = \text{rand}() * V_i(t) + a_i(t)$, Updation equation: $X_i(t+1) = X_i(t) + V_i(t+1)$, Best Fitness: $\text{best}(t) = \min(\text{fitness})$; Worst Fitness: $\text{worst}(t) = \max(\text{fitness})$ (12)

7. Experimentation and Simulation Results

Experimentation is carried out on randomly generated DDFG for different FPGA sizes. For each DDFG, Initial locations are fixed and iteration is performed by using optimization algorithms discussed in this paper. Simulation is done in Mat lab software and the results are discussed. The performance of CSO algorithm outperforms PSO, ARPSO and GSA algorithms. Both single and double constrained optimization of FPGA PaR is considered for experimentation. In all cases, the CSO performs better than the other algorithms. Table (1.) and Table (2.) shows the comparison results for cost function minimization.

Simulation results for the above experimentation is shown in Figure 9 to Figure 14.

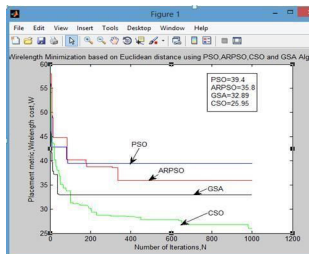


Figure 9. Comparison of wavelength minimization using PSO, ARPSO, GSA and CSO algorithms.

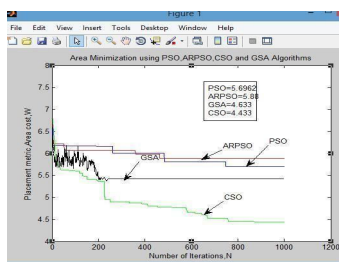


Figure 10. Comparison of area minimization using PSO, ARPSO, GSA and CSO algorithms.

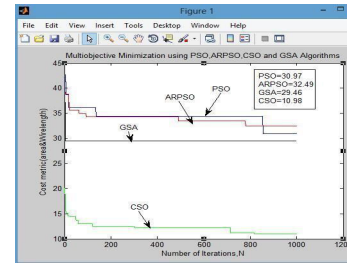


Figure 11. Comparison of wavelength and area minimization using PSO, ARPSO, GSA and CSO algorithms.

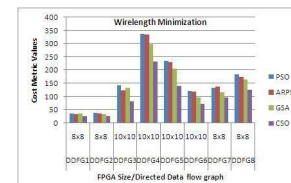


Figure 12. FPGA Size/Directed Data Flow Graph Vs Cost metric Values (Single Objective-wavelength).

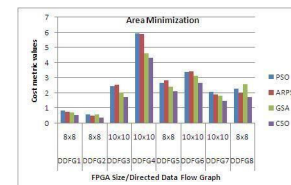


Figure 13. FPGA Size/Directed Data Flow Graph Vs Cost metric Values (Single Objective-area).

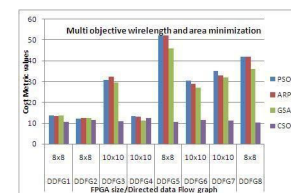


Figure 14. FPGA Size/Directed Data Flow Graph Vs Cost metric Values (Multi-objective -wavelength and area).

Thus the proposed CSO algorithm based Wavelength and area minimization of FPGA Placement and routing Process in CAD flow proves better than the other. CSO can be used as a best optimizer in CAD based Design flow in industries for compact placement in turn results in better resource utilization.

8. Conclusion

Field Programmable Gate arrays are used to implement the digital design of high speed, high performance applications. The advanced features of FPGAs can be utilized effectively only when successful CAD tools are developed. CAD tool for placement and routing is an important phase in design Flow which can be optimized by using an effective optimization algorithm. In this paper, CSO based Placement and routing proves better optimization in terms of area and wirelength. Simulation results shows that CSO outperforms PSO, ARPSO and GSA.

9. References

1. Gerez SH. Algorithms for VLSI Design Automation. Tata McGraw Hill. 1999.
2. Caralconti C, Cabral LAF. A New approach to VPR Tool's FPGA Placement. Proceedings of WCECS. 2007; 24–6.
3. Iztok Fister Jr, Yang XS, Fister I, Brest J, Fister D. A brief review of nature –Inspired algorithms for optimization. Elektrotehnisk Vestnik English edition. 2013; 80(3):1–7.
4. Nousias I, Khawam S, Milward M, Muir M, Arslan T. A Multiobjective GA Based Physical Placement Algorithm for Heterogeneous Dynamically Reconfigurable Arrays. 2007; 497–500. 1-4222-1060-6/07
5. Venayagamoorthy GK, Gudise VG. Swarm Intelligence for Digital Circuits Implementation on Field Programmable Gate arrays Platforms. Proceedings of the 2004 NASA/DoD Conference on Evolution Hardware (EH'04). 2004Jun. p. 83–6.
6. Bali T, Khosla M, Anjum N. Placement in FPGA using Hybrid PSO-SA Technique. International Journal of Engineering Research and Technology. 2013; 2(9). ISSN: 2278-0181.
7. Xu W, Xu K, Xu X. A Novel Placement Algorithm for Symmetrical FPGA. IEEE Guilin, 2007 Oct; 1281–4.
8. Gudise VG, Venayagamoorthy GK. FPGA Placement and Routing Using Particle Swarm Optimization. Proceedings of the IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design. 2004; 307–8.
9. Raj BSA, Jancy AY, Senthilkumar KM, Sangeetha M. Optimisation of placement and routing process in Xilinx XC4000 FPGA with 14x14 CLBs using PSO. International Journal of Emerging Technology and Advanced Engineering. 2012; 2(7). ISSN 2250-2459.
10. Premalatha B, Divya D, Abinaiya N, Monisha S. Particle Swarm Optimization Based Placement and Routing of Hardware Tasks in 2D Homogeneous FPGAs. International Journal of Scientific and Engineering Research. 2013; 4(3):1–6. ISSN 2229-5518.
11. ELAbd M, Hassan H, Anis M, Kamel MS. Discrete cooperative particle swarm optimization for FPGA placement. Applied Soft Computing. Elsevier. 2010; (10):284–95.
12. Lenin K, Reddy BR, Kalavathi MS. Attractive and repulsive particle swarm optimization, hybrid artificial bee algorithm for solving reactive power optimization problem. International Journal of Engineering Sciences and Emerging Technologies. 2013; 5(1):41–52.
13. Roy S, Chaudri SS. Cuckoo Search Algorithm using Levy Flight: Review. IJ Modern Education of Computer and Information Technology (IJCIT). 2013; 12:10–5.
14. Rajabioun R. Cuckoo Optimization Algorithm, Elsevier. 2011; 11(8):5508–18.
15. Yang XS, Deb S. Multiobjective cuckoo search for design optimization. Computers and Operation research (Elsevier). 2013; 40(6):1616–24.
16. Valian E, Mohanna S, Tavakoli S. Improved Cuckoo Search Algorithm for Global Optimization. International Journal of Communication and Information Technology. 2011; (1):1–44.
17. Sabri NM, Puteh M, Mahmood MR. A Review of Gravitational Search Algorithm. Int J Advance Soft Comput Applications. 2013; 5(3):31–9. ISSN 2074-8523.