Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 167 – 175

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

# Natural Computation for Optimal Scheduling with ILP Modeling in High Level Synthesis

Shilpa K. C[a], LakshmiNarayana.C[b,*]

[a,b] *BMSCE, Department of Electrical Engineering Science, Visvesvaraya Technological University, Bangalore, 560019, India.*

**Abstract**

The concept of the natural computation for optimal scheduling in high level synthesis, for resource constraint and time constraint scheduling problem in automated integrated circuit synthesis using Integer Linear Programming (ILP) modeling is presented in this paper. This paper compares three natural computations paradigms: (i) evolution optimizer technique genetic algorithm, (ii) evolutionary programming, and (iii) swarm intelligence based particle swarm optimization. Experimental results indicate that evolution based Genetic Algorithm search is more powerful search compared to Evolutionary Programming and Particle Swam Optimization.



Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

*Keywords:* High Level Synthesis ; Data Flow Graph; Evolutionary Programming ; Genetic Algorithm ; Particle Swarm Optimization ; Very Large Scale Integration ; Integer Linear Programming

## 1. Introduction

Very Large Scale Integration (VLSI) circuits built with hundreds and thousands of transistors on a single chip, the design complexity of the chip increases in terms of number of gates, transistors and functionality.

* Corresponding author. Tel.: + 91-974-3300352.
*E-mail address:* shilpa.kc2@gmail.com



Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.008

Indeed it would be extremely difficult to design system starting at transistor level or at the logic level. Computer Aided Design (CAD) algorithms are the driving forces behind the optimization of circuit quality and design.

The High-Level Synthesis (HLS) process is transformation from algorithm synthesis to its equivalent register transfer level description of VLSI circuits [1]. In HLS, the algorithm level of description is transformed into unique graphical representation which is usually termed as Data Flow Graph (DFG). The most important step in HLS is Scheduling and Allocation process, where Scheduling process provides the performance and cost trade off, and Allocation process which provides hardware minimization and register or memory tradeoff [2]. Scheduling generates required control time slots for computation design process. Allocation determines the required number and type of functional components to perform computation operations.

## 2. Related Work

Many scheduling techniques are reported for scheduling problem, which is (nondeterministic polynomial time) NP-complete. As Soon As Possible (ASAP) and As Late As Possible (ALAP) [1], the two most basic scheduling method, where no priority of operation is assigned in this technique. The List based Scheduling algorithm [2] which is based on priorities of nodes, and increases the time of computation based on priority. The global scheduling algorithm, Integer Linear Programming (ILP) [3] approach is based on mathematical formulation, the computation complexity increases with number of control steps. In Force Directed scheduling (FDS) algorithm [4, 5] and Path Based Scheduling [6] does not guarantee global optimization in design space, and struck at local minima. Simulated annealing scheduling [7] suffers in scheduling speed.

Evolutionary based search [8] provides optimization approaches to handle problems in an efficient manner. Genetic Algorithm (GA) [9] an optimization technique based on the principle of biological evolution to solve complex problem. In comparison with Genetic Algorithm, Evolutionary Programming [10] is simple and more flexible. Particle Swarm Optimization (PS0) [11] is an optimization technique, based on social swarm behavior. PSO appears to be a simpler algorithm than GA and EP.

Experimental result shows that evolutionary optimizer technique yields optimization approaches to handle problem in efficient manner.

## 3. Problem formulation

### *3.1. Time Constraint Scheduling*

The objective of Time Constraint Scheduling minimizes the functional units for fixed number of control step. To illustrate the scheduling problem as constraints optimization formulation [1], the variables used in the formulation are as follows:

- $Nt_k$ be integer variables which represent the required number of functional units ($FUt_k$) of type $t_k$, $k = 1,2, \ldots . m$, for each operation $o_i$, $i = 1,2, \ldots . n$. $Ct_k$ be the cost of the functional units of type $t_k$, $a_j$, $j = 1,2, \ldots . r$, be the control time slot assigned for scheduling tasks.
- $x_{i,j}$ be the zero to one integer variable associated for each operation $o_i$, $S_i$ and L*i* be the earliest and latest time bounds for each operation $o_i$ using ASAP and ALAP algorithms. The scheduling problem is formulated by the equation (1)

$$\text{Minimize} \ \sum_{k=1}^{m} [Ct_k * Nt_k] \tag{1}$$

Such that

$$[\sum_{S_i \le j \le L_i} x_{i,j} = 1] \ , \ \forall i, \ 1 \le i \le n \tag{2}$$

$$[\sum_{oi \in FUt_k} x_{i,j} \le Nt_k] \ , \ \forall j, \ 1 \le j \le r \ , \forall k, 1 \le k \ \le \ m \tag{3}$$

$$[\sum_{S_i \le j \le L_i} [j * x_{i,j}] - [\sum_{S_j \le k \le L_j} [k * x_{j,k}]] \le -1] \ , \ \forall o_i \to o_k \tag{4}$$

Constraints (2) confirms that each operation $o_i$ be arranged into one and exactly one time slot based on the time bound of $S_i$ and $L_i$. Resource constraints (3) confirms that no time slot holds more than $Nt_k$ operations of type $t_k$. Precedence and Successors constraints (4) ensure that there must be minimum difference of one cycle between the execution of successor and predecessors module.

### *3.2. Resource Constraint Scheduling*

In Resource Constraint Scheduling, the total computation time is minimized for fixed number of functional units [2]. The following constraints to minimize the schedule time while satisfying the resource constraints are as follows

- The schedule is progressively built, one operation at time, so that resource restrictions and data reliance are not disturbed.
- The overall number of time slot are minimized by the approach that the number of operation scheduled in any control steps does not exceed the number of functional units available.
- The dependence restrictions for which all the predecessor node operation $o_i$ are scheduled before the successor node operation $o_i$

## 4. Optimization Algorithms (Genetic Algorithm (GA), Evolutionary Programming (EP), Particle Swarm Optimization (PSO))

### *4.1. Genetic Algorithm (GA)*

Genetic Algorithm [9], the principle of evolutionary computation exploration procedure based on Darwin Principle of Natural Selection. The Genetic Algorithm typically requires three major evolutionary operators such as crossover, mutation and selection. The simple Genetic Algorithm evolution search starts as follows (i) Initializing population randomly. (ii) Evaluating the population based on objective function to find the fitness function. (iii) Select individual for reproduction operation, mutation, and selection process. (iv) Perform the process until the fitness level reaches satisfactorily. GA supports multi-objective optimization and applied for solving complex problems.

### *4.2. Evolutionary Programming (EP)*

Evolutionary Programming (EP) [10] solves complex tasks in similar ways as real-coded genetic algorithms. An important difference between evolutionary programming from real-coded genetic algorithms is (i) Evolutionary Programming do not use crossover or any other kind of exchange of genetic material between individuals. (ii) Offspring are generated by mutation only.

### *4.3. Particle Swarm Optimization (PS0)*

Particle Swarm Optimization [11, 12], an evolutionary computation search optimization approach based on social swarm behavior. An important difference between PSO from GA and EP is (i) PSO appears to be a simpler algorithm than GA and EP. (ii) PSO does not have genetic operators. (iii) In PSO, topology is constant.

## 5. Methodology

The scheduling problems taken in this paper is as follows

- In Time constraint scheduling problem, functional units are minimized for fixed number of control steps.
- In Resource constraint scheduling problem, time slots are minimized for fixed number of functional units.

To solve the scheduling problem as constraints optimization formulation using Integer Linear Programming, the Hardware Abstraction Layer (HAL) benchmark problem [2] taken in this paper and is shown in Figure 1.

The following are the step for ILP formulation for scheduling. (i) four control time slot is considered in the data flow graph. (ii) The four diverse varieties of functional components taken in DFG contains multiplier component, addition component, subtraction component and comparator component, four types of functional components from library are needed. (iii) The cost of multiplier component = $C_m$, adder component = $C_a$, subtraction component = $C_s$ and comparator component = $C_c$ are considered respectively. (iv) the number of functional components considered in this paper are represented by multiplier component = $N_m$, adder component = $N_a$, subtraction component = $N_s$, and comparator component = $N_c$ respectively. (v) Assuming the cost of the multiplier component= 2, the cost of adder component = cost of subtraction component =cost of comparator component =1. (vi) the cost of the functional components has to be minimized and all the inequalities constraints have to be satisfied.

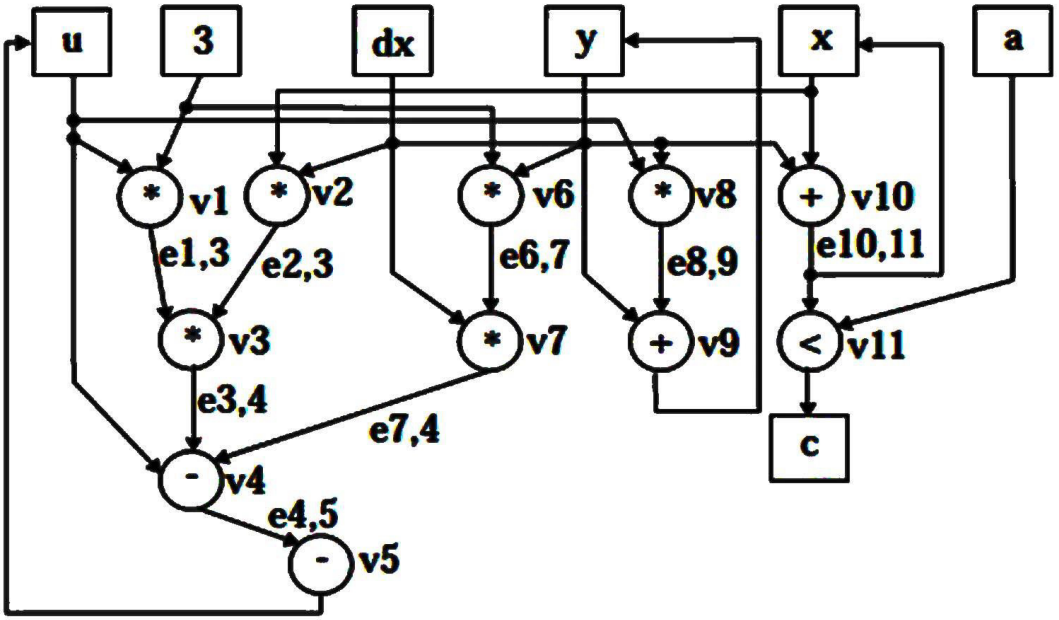



Fig.1. HAL benchmark problem

The Integer Linear Programming can be formulated as follows

$$minimize \quad C_m \times N_m + C_a \times N_a + C_s \times N_s + C_c \times N_c \tag{5}$$

$x_{1,1} = 1$
$x_{2,1} = 1$
$x_{3,1} + x_{3,2} = 1$
$x_{4,1} + x_{4,2} + x_{4,3} = 1$
$x_{5,2} = 1$
$x_{6,2} + x_{6,3} = 1$
$x_{7,3} = 1$
$x_{8,4} = 1$

$$x_{9,2} + x_{9,3} + x_{9,4} = 1$$
$$x_{10,1} + x_{10,2} + x_{10,3} = 1$$
$$x_{11,2} + x_{11,3} + x_{11,4} = 1$$

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} \leq Nm$$
$$x_{3,2} + x_{4,2} + x_{5,2} + x_{6,2} \leq Nm$$
$$x_{4,3} + x_{6,3} \leq Nm$$
$$x_{7,3} \leq Ns$$
$$x_{8,4} \leq Ns$$
$$x_{10,1} \leq Na$$
$$x_{9,2} + x_{10,2} \leq Na$$
$$x_{9,3} + x_{10,3} \leq Na$$
$$x_{9,4} \leq Na$$
$$x_{11,2} \leq Nc$$
$$x_{11,3} \leq Nc$$
$$x_{11,4} \leq Nc$$

$$x_{3,1} + 2x_{3,2} - 2x_{6,2} - 3x_{6,3} \leq -1$$
$$1x_{4,1} + 2x_{4,2} + 3x_{4,3} - 2x_{9,2} - 3x_{9,3} - 4x_{9,4} \leq -1$$
$$1x_{10,1} + 2x_{10,2} + 3x_{10,3} - 2x_{11,2} - 3x_{11,3} - 4x_{11,4} \leq -1$$

## 6. Simulation Results & Analysis

For all the cases (GA, EP, PSO), the following simulation values is taken (i) random number of the uniform range between zero to one for the 200 size population. (ii) In GA two point cross over applied with probability 1 whereas mutation is real valued mutation with probability 0.01. (iii) Constriction constant $\chi$ taken as 0.72, learning factor is taken as 2.5, decreasing value of inertia weight taken for 500 iteration from 1.2 to 0.1. (iv) $\eta$ values in EP for all member is taken as initialization of mutation vector values is 0.00001.Terminating criteria is defined as for 10 continues generation best chromosomes fitness difference should be less than 0.01. (v) Selection process in EP and in GA defined as tournament selection with challenger population size equal to 10% of parent population. MATLAB is used in this paper to solve the scheduling algorithm problem.

A fitness function F is given in (6)

$$F = f + P\left[\sum_{k=1}^{r}(g_k^+(x_i))^2 + \sum_{m=1}^{n}(h(x_i))^2\right] \quad (6)$$

P= penalty factor = 1000, $g_k$ ($\leq 0$) and h (= 0) are constraints violation terms.

Comparison between PSO, EP and GA have given with respect to performance parameters like resources required in time constraint scheduling or control steps required for resourced constraint scheduling along with convergence characteristics. Table 1, Table 2 and Table 3 shows required optimal number of Multipliers components, number of Adder components, number of Subtraction components and number of Comparators components respectively.

### *6.1. Time Constraint Scheduling*

Table 1. Result of GA performance for Time constraint scheduling

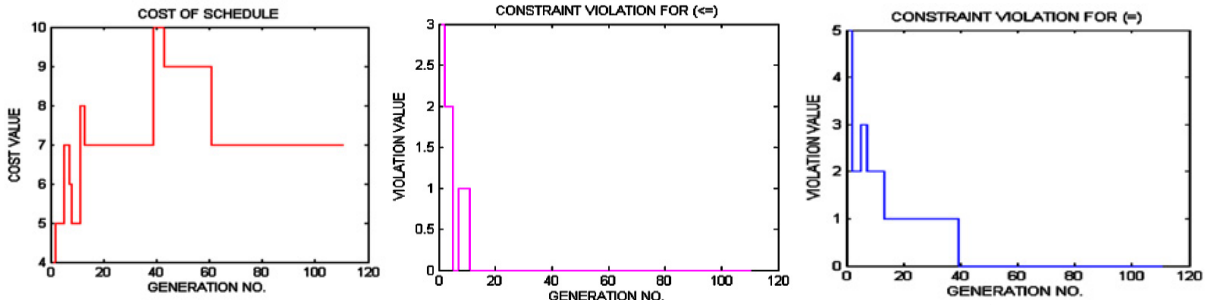
| | Required Resources | | | | |
|---|---|---|---|---|---|
| **Trail No.** | **Multipliers component** | **Adder component** | **Subtraction component** | **Comparators component** | **No. of Generations** |
| 1 | 2 | 1 | 1 | 1 | 128 |
| 2 | 2 | 1 | 1 | 1 | 138 |
| 3 | 2 | 1 | 1 | 1 | 105 |
| 4 | 2 | 1 | 1 | 1 | 112 |
| 5 | 2 | 1 | 1 | 1 | 106 |
| 6 | 2 | 1 | 1 | 1 | 115 |
| 7 | 2 | 1 | 1 | 1 | 119 |
| 8 | 2 | 1 | 1 | 1 | 115 |
| 9 | 2 | 1 | 1 | 1 | 124 |
| 10 | 2 | 1 | 1 | 1 | 128 |
| Mean/S.D | 2 | 1 | 1 | 1 | 119/11 |

Table 2. Result of EP performance for Time constraint scheduling

| | Required Resources | | | | |
|---|---|---|---|---|---|
| **Trail No.** | **Multipliers component** | **Adder component** | **Subtraction component** | **Comparators component** | **No. of Generations** |
| 1 | 2 | 1 | 1 | 1 | 324 |
| 2 | 2 | 1 | 1 | 2 | 410 |
| 3 | 2 | 1 | 1 | 1 | 276 |
| 4 | 2 | 1 | 1 | 1 | 367 |
| 5 | 2 | 1 | 1 | 1 | 484 |
| 6 | 2 | 1 | 1 | 1 | 424 |
| 7 | 2 | 1 | 1 | 1 | 256 |
| 8 | 2 | 1 | 1 | 1 | 378 |
| 9 | 3 | 1 | 1 | 1 | 398 |
| 10 | 2 | 1 | 1 | 1 | 128 |
| Mean/S.D | 2.1/0.32 | 1/0 | 1/0 | 1.1/0.32 | 345/102 |

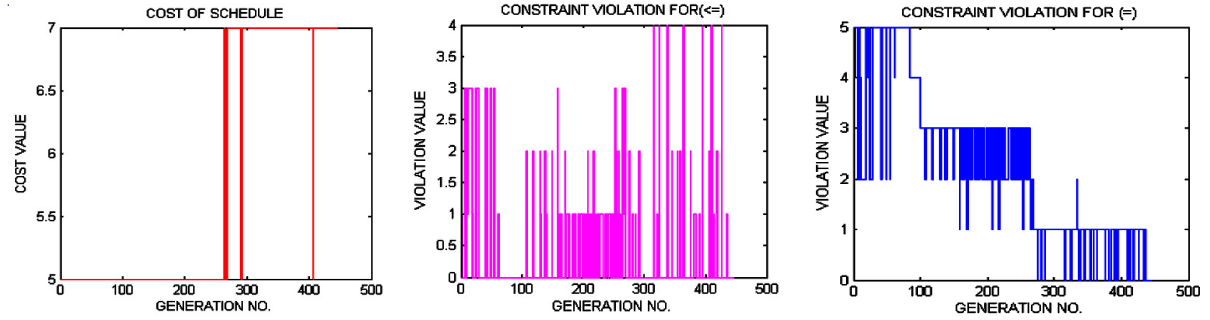
Table 3. Results of PSO performance for Time constraint scheduling

| | Required Resources | | | | |
|---|---|---|---|---|---|
| **Trail No.** | **Multipliers component** | **Adder component** | **Subtraction component** | **Comparators component** | **No. of Generations** |
| 1 | 2 | 1 | 1 | 1 | 56 |
| 2 | 3 | 1 | 1 | 2 | 54 |
| 3 | 3 | 1 | 1 | 1 | 56 |
| 4 | 3 | 1 | 1 | 1 | 56 |
| 5 | 3 | 2 | 1 | 1 | 56 |
| 6 | 3 | 1 | 1 | 1 | 57 |
| 7 | 3 | 1 | 1 | 1 | 58 |
| 8 | 3 | 1 | 1 | 1 | 56 |
| 9 | × | × | × | × | 500 |
| 10 | 2 | 1 | 1 | 1 | 128 |
| Mean/S.D | 2.7/0.44 | 1.1/0.33 | 1 | 1.1/0.33 | 56/1.2 |



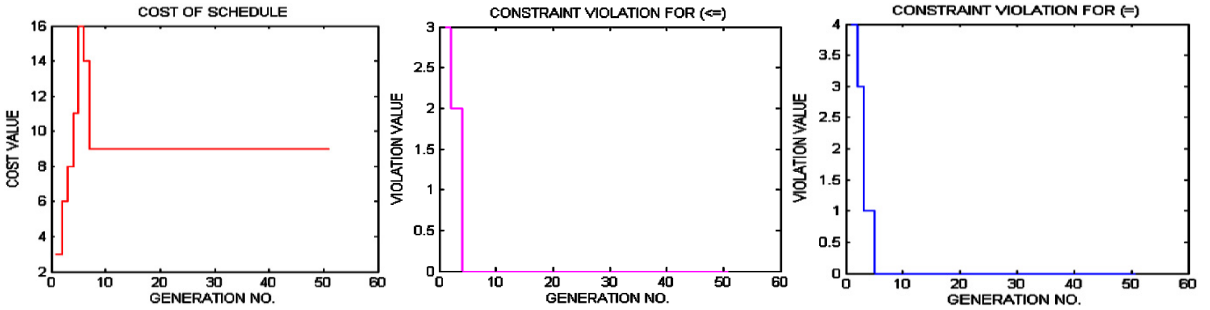

(a) (b) (c)

Fig.2. GA convergence performance (a) cost minimization (b) constraint (<=) violation (c) constraint (=) violation with generation for the best solution.




(a) (b) (c)

Fig.3. EP convergence performance (a) cost minimization (b) constraint (<=) violation (c) constraint (=) violation with generation for the best solution.




(a) (b) (c)

Fig.4. PSO convergence performance (a) cost minimization (b) constraint (<=) violation (c) constraint (=) violation with generation for the best solution.

### *6.1.1. Discussion*

Performance for time constraint schedule using the evolutionary based optimization algorithm Genetic Algorithm, Evolutionary Programming and Particle Swarm Optimization is presented in table 1, table 2, and table 3 for 10 independent trails in all cases. It is clear from the table 1 that GA have delivered all-time 100% optimal scheduling without failure with an average less number of generation compare to EP and there is no violation in constraints in any cases. EP have delivered optimal solution 80% time without any violation in constraint and taken maximum number of generations for convergence, whereas PSO performance is comparatively poor in terms of optimality, but rate of convergence is fastest, which basically indicate the trapping in local minima. In one case PSO

could satisfy the constraint condition as shown in table 3, for trail no 9. Best solution performances in each generation for a single trail with GA, EP and PSO are presented in Figure 2, Figure 3 and Figure 4.

### *6.2. Resource Constraint Scheduling*

Consider the same DFG as shown in Figure 1, in which there are two types of sources multiplier and an ALU unit which perform addition/subtraction and comparison operation is considered. Further assumption has made that each operation required one cycle for execution and constraints with number of resources in each case is equal to 2, with application of heuristic algorithm upper bound of control steps is 4. GA, EP and PSO are applied as applied for time constraint case.

Table 4. Performance for resource constraint scheduling

| Trail No. | GA | | EP | | PSO | |
|---|---|---|---|---|---|---|
| | [Control Step | [Gen.] | [Control Step | [Gen.] | [Control Step | [Gen.] |
| 1 | 4 | 105 | 4 | 635 | × | |
| 2 | 4 | 123 | 4 | 724 | × | |
| 3 | 4 | 130 | × | | × | |
| 4 | 4 | 75 | 4 | 489 | 4 | 52 |
| 5 | 4 | 90 | 4 | 520 | × | |
| 6 | 4 | 127 | 4 | 680 | × | |
| 7 | 4 | 86 | × | | × | |
| 8 | 4 | 94 | 4 | 547 | 4 | 35 |
| 9 | 4 | 120 | 4 | 588 | × | |
| 10 | 4 | 113 | 4 | 634 | × | |

#### *6.2.1. Discussion*

Among all the three natural computation, GA has given optimal result for all 10 different trails as shown in table 4, whereas EP has deliver the optimal results but in two cases it could not converge at all. Number of generation required for convergence in EP is more compare to GA. Performance of PSO is worse in this case. For most of cases it could not converge at all as shown in table.4 with (×).

## 7. Conclusion

Automated synthesis is very important for efficient design and in cost limitation. It is always better to explore the solution domain rather than crafting the solution for optimal solution. Experimental result and analysis have shown that Genetic Algorithm is having better exploration and exploitation capability compare to Evolutionary Programming and Particle Swarm Optimization. Genetic Algorithm gets this advantage because of parent's interaction in terms of offspring development. In both cases of scheduling Genetic Algorithm outperformed the Evolutionary Programming and Particle Swarm Optimization. Particle Swarm Optimization convergence level is earlier than Evolutionary Programming and Genetic Algorithm but having serious drawback in terms of trapping in local solution and curse of dimensionality.

## References


1. De Micheli .G. *Synthesis and Optimization of Digital Circuits*. USA: McGraw-Hill; 1994.
2. Gajski.D, Dutt N.D, Wu.A, Lin.S. *High Level Synthesis: Introduction to chip and System Design*. USA:Kluwer Academic Publisher; 1992.
3. Hwang C.T, Lee J.H, Hsu Y.C, Lin. Y.L. A formal approach to the scheduling problem in high level synthesis. In: *IEEE Trans. on Computer-Aided Design*. Feb 1991. p. 464-475.
4. Paulin P.G, Knight J.P. Force-directed scheduling in automatic data path synthesis. In: Proc. of the 24th ACM/IEEE Conference on Design Automation . 1987.p.195–202.
5. Paulin P.G, Knight J.P. Force-directed scheduling for the behavioral synthesis of ASIC's. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.1989. p.661–679.
6. Camposano.R . Path-based scheduling for synthesis. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 1991. p. 85–93.
7. Ly.T.A, Mowchenko J.T. Applying simulated evolution to high level synthesis. In: *IEEE Trans. On Computer-Aided Design*.1993. p. 389-409.
8. Grewal G, Cleirigh M.O, Wineberg M. An Evolutionary Approach to Behavioral-Level Synthesis. In: *IEEE Trans. VLSI circuits*. 2003. p. 264-272.
9. Goldberg DE. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley Publishing Co; 1989.
10. Eiben A.E, Smith J.E. *Introduction to Evolutionary Computing*. Natural Computing Series Springer; 2003.
11. Eberhart. R.C, Shi. Y. Comparison between Genetic Algorithms and Particle Swarm Optimization. In: *Proceedings of seventh annual conference on Evolutionary Programming*.1998. p.611-616.
12. Kennedy, Eberhart R. C. Particle Swarm Optimization. In: *Proceedings IEEE Int. Conf. on Neural Networks (Perth Australia)*. Piscataway, IEEE service Center; 1995.