ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(31), DOI: 10.17485/ijst/2016/v9i31/96819, August 2016* ISSN (Online) : 0974-5645

Optimal Resource Schedule in Architectural Level Synthesis using Evolutionary Computations

K. C. Shilpa[1*], C. LakshmiNarayana[1], Manoj Kumar Singh[2*]

[1]BMSCE, Department of Electrical Engineering Science, Visvesvaraya Technological University, Bangalore - 560019, Karnataka, India; shilpa.kc2@gmail.com, ln_gp@yahoo.com
[2]Manuro Tech Research Pvt. Ltd, Bangalore - 560097, Karnataka, India; mksingh@manuroresearch.com

Abstract

Objectives: This paper aims to find optimal resource schedule in Architectural level synthesis using Evolutionary Computation. **Methods and Statistical Analysis:** The paper is a comparative study of four Evolutionary Computations Algorithm: Differential Evolution (DE), Genetic Algorithm (GA), Evolutionary Programming (EP) and Particle Swarm Optimization (PSO). The problem area chosen is Hardware Abstraction Layer (HAL) benchmark scheduling problem using Integer Linear Programming method. **Findings:** The nature inspired computation algorithms should satisfy the Latency constrained Schedule, simulation results are implemented using MATLAB software. **Conclusion/Application:** The performance with respect to optimal resource schedule, number of generations, convergence time is compared among the four optimized algorithm are presented. The results prove Differential Evolution is better among the other optimized algorithm.

Keywords: Architectural Level Synthesis, Evolutionary Computation, Hardware Abstraction Layer, Resource Schedule

1. Introduction

Optimization is a goal to explore the excellent solutions in a given discrete space. The challenging optimize problem are NP (Nondeterministic Polynomial time) - hard problem. These challenging optimization problems are solved by trial and error using various optimization techniques. To solve the optimization problem Nature is key source of inspiration for many researchers. Nature inspiration computation comes to the group of meta-heuristics search. The nature inspired optimization meta-heuristic algorithms focused in this paper are Evolutionary Computations techniques: Differential Evolution Genetic Algorithm and Evolutionary Programming, Particle Swarm Optimization. With advance in VLSI (Very Large Scale Integration) fabrication technology, with plentiful of transistor on an integrated circuit, the design complexity gradually increases. CAD tools can be used to provide the designer the effective design in term of quality and time.

The Architecture level synthesis is mapping of Algorithm to hardware module[1]. The major steps in Architecture level synthesis are Resource Scheduling, which assign the behavioral operator to control system and Resource Allocation involves selection and binding of hardware components to behavioral operator and variable.

The motivation for this paper is to formally have an Integer Linear Programming (ILP)[2] approach which guarantees solution quality and guarantee of quickly finding for optimal resource solution problem using Evolutionary Computations.

The different resource scheduling algorithm and its draw back has been reported and shown before paper[3]. The Evolutionary Algorithm inspired by natural evolution, to solve complex function and combinatorial optimization with precision results[4]. Genetic Algorithm[5] and Evolutionary Programming[6] effectively solves nonlinear optimization problem, it main drawback is slow convergence and may struck to local minima. PSO[7] has few operators and better convergence, Differential Evolution[8] has better convergence and few control parameters.

** Author for correspondence*

2. Overview of Nature Inspired Algorithm (DE, GA, EP, PSO)

2.1 Differential Evolution (DE)

Differential Evolution (DE)[9,10] an evolution search initiated by Storn and Price (1997). The method is based on Evolutionary operator crossover, mutation and seclection, with unique feature of DE is its differential weight technique in mutation operation. The DE is simple algorithm, robust with fast convergence to optimal solution.

2.2 Genetic Algorithm (GA)

Genetic Algorithm[11,12] is exploration approach of natural evolutionary theory for combinatorial optimization problem. The general procedure of GA is to evaluate objective function value for random initial population, followed by the diversity operators: Crossover, mutation, selection function. The diversity operator avoids local optima; this process is carried until best solution is achieved. GA[13] is very effective for multi-dimensional optimization.

2.3 Evolutionary Programming (EP)

Evolutionary Programming[14] was coined by Fogel, the flow of the algorithm is to initialize population and calculate fitness values for initial population, secondly mutate the parents and generate new population. Calculate fitness values of new generation and continue from the second step. Mutation is the key point which leads to exploration in the alogrithm. EP convergence speed is very slow.

Table 1. A brief comparison of four nature inspired algorithm

GA (Genetic Algorithm)	EP (Evolutionary Programming)	PSO (Particle swarm optimization)	DE(Differential Evolution)
GA an optimization method of biological evolution.	EP an optimization mode of biological evolution.	PSO an optimization technique inspired by social behavior of bird flocking or fish schooling	DE an optimization method of biological evolution.
John Henry Holland introduced the concept of GA	Lawrence J. Fogel introduced EP	PSO initiated by Eberhart and Kennedy	DE introduced by Storn and Price
Original Binary valued representation	Real valued representation	Real valued representation	Real valued representation
Random population is initialized and searches updating generation	Random population is initialized and searches updating generation	potential solutions are called particles, fly through the problem space by following the current optimum particles	Random population is initialized and searches updating generation
Diversity operator: selection, crossover, and mutation.	EP typically requires only mutation and selection operator	In the PSO, there is one simple operator: velocity calculation.	Diversity operator: selection, crossover, and mutation.
GA crossover has more effect at the beginning of the run	EP does not have crossover operation	PSO does not have crossover	DE crossover has more effect at the end of the run
GA mutation has more effect at the end of the run	Mutation is the only main reproduction operator	PSO does not mutation operator	DE mutation has more effect at the beginning of the run
More operators for computation	Less operators for computation than GA	PSO is easy to perform, few parameters to adjust.	DE is simple to implement.
Convergence rate is less than PSO	Convergence rate is slow compared to GA and PSO	Convergences rate is faster	Convergences rate is faster
Number of Computation is more than PSO	Number of Computation is more than GA	Number of computation is minimum	Number of computation is minimum
Maximum probability to achieve global solution	Delivers the global solution better compared to PSO and equivalent to GA	Probability to struck to local minima.	Maximum probability to achieve global solution

2.4 Particle Swarm Optimization (PSO)

Particle Swarm Optimization[15,16] based on swarm intelligence introduced by Kennedy and Eberhart (1995). The computational operator are very less, the member of swarm have a cognitive behavior (personal best) and social behavior (global best) among them to explore the random search in design space.

The brief comparison of the four different nature inspired algorithm are summarized below in Table 1.

3. Problem Formulation

In Latency constrained Schedule[17], for the fixed the control steps, minimize the required resource. The Resource Schedule problem is np (nondeterministic polynomial time) - hard problem; The Integer Linear Programming (ILP) formulation[18, 19] for the resource schedule is given below:

- Firstly the mobility for each operation is calculated, where E_k= ASAP (AS SOON AS POSSIBLE) andL_k= ALAP(AS LATE AS POSSIBLE) values

$$M = \{0_j | E_k \le j \le L_k\} \quad (1)$$

- Secondly the INTEGER LINEAR PROGRAMMING formulation is given as follows

$$Min \sum_{k=1}^{m} [C_k * R_k] \; while \sum_{E_i \le j \le L_i} x_{i,j} = 1] \quad (2)$$

Where $1 \le k \le m$ indicate the number of resource operation available, R_k term is the computing unit of resource type k, and C_k term is the cost of each resource computing type.

$[x_{ij} = 1]$, ∀i operation = j (3)
else = 0, otherwise

- Thirdly the constraints on resource type,

$$\sum_{k=1}^{n} \left[x_{ij} \le R_i \right] \quad (4)$$

- Finally the constraint on data dependency,

$(s^* x_{j,s})$)-$(t^* x_{j,t})$ ≤ -1, s ≤ t, s and t are control step for each operation. (5)

Latency constrained for the Hardware Abstraction Layer (HAL) benchmark problem is shown in Figure 1,Eleven vertices{v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11} and eight edges $\{e_{1,3}, e_{2,3}, e_{3,4}, e_{4,5}, e_{6,7}, e_{7,4}, e_{8,9}, e_{10,11}\}$. There are six multipliers, two adders, two subs tractors and one comparator, the goal is to minimize these resource units using Integer Linear Programming formulation is depicted in Figure 1.

The mobility M is four. C_m, C_a, C_s, C_c : computing unit cost of the multiplier unit, adder unit, subtraction unit, comparator unit. R_m, R_a, R_s, R_c: Number of computing resource of the multiplier unit, adder unit, subtraction unit, comparator unit. Let the assumption be $C_m=2, C_a=1, C_s=1, C_c=1$. The goal of the problem is to minimize the Resource unit for the scheduling problem, and satisfy the above mentioned constraints.

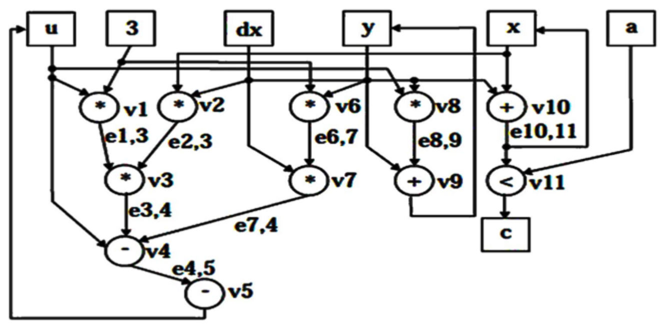

Figure 1. Hardware abstraction layer benchmark problem.

4. Experimental Setup

The comparison of nature inspired computational algorithm taken is DE, GA, EP, and PSO. Each algorithm is tested with uniform random number for population size = 200 population. Matlab simulation is considered for the optimization algorithms to solve for the optimal schedule. The fitness function considered is shown in (6).

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

a = 1000, $g^+_h \le 0$ and h = 0 are constraints violation terms.

The parameters setting for each optimization algorithm is described below:

- DE Setup: f_m scaling factor = 0.7, cr: cross over rate = 0.8, the strategy = DE/rand to best/1 is considered, where rand: randomly chosen population, best: minimum value of objective function, 1: difference vector =1.
- GA Setup: Two point crossover probability =1, mutation probability = 0.01. Tournament Selection process is used
- EP Setup: Mutation probability = 0.00001. Tournament Selection process is used.

- PSO Setup: Constriction constant = 0.72, learning factor = 2.5, inertia weight w = decreasing to 1.2 to 0.1.

5. Results and Discussion

The following Table 2 compares the performance of DE, GA, EP, and PSO. The performances parameters are checked with optimization algorithm are optimal solution obtained for computing unit (R_m multiplier unit, R_a adder unit, R_s subtraction unit, and R_c comparator units). Numbers of generation taken for convergence, (convergence time taken in seconds) are presented. The convergence performance graph obtained to achieve the minimum optimal cost minimized factor is shown in Figure 2.

5.1 Discussion

A comparative study for the performance of latency constrained scheduling using DE, GA, EP and PSO is presented in Table 2 for 5 trails. Among all the trails, DE is the best in finding optimal solution, takes minimum convergence time and minimum number of generation taken to achieve minimum objective function. GA is also best in finding optimal solution similar to DE; but number of generation taken is more compared to DE, convergence time is more when compared to DE, EP, PSO. EP delivers only 90% of optimal solution, but suffers badly in convergence time and number of generation taken for convergence compared to DE, PSO. PSO badly gets struck at local minima, fails to deliver optimal solution, but the convergence time and the number generation taken for convergence is much closed match

Table 1. Comparative results for the performance of DE, GA, EP, PSO

Trail No.	Perfomance Parameters	DE				GA				EP				PSO			
01	Computing Units	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c
	Optimal solution for required resource	2	1	1	1	2	1	1	1	2	1	1	1	3	1	1	1
	Convergence time (second)	12.4060				40.6410				29.4370				12.8120			
	No. of generation	51				101				291				52			
02	Computing Units	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c
	Optimal solution for required resource	2	1	1	1	2	1	1	1	2	1	1	1	4	1	1	1
	Convergence time(second)	12.6560				40.7500				25.2970				12.3910			
	No. of generation	51				101				271				51			
03	Computing Units	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c
	Optimal solution for required resource	2	1	1	1	2	1	1	1	3	1	1	1	2	1	1	1
	Convergence time(second)	12.6400				41.0471				44.3910				14.6870			
	No. of generation	51				101				463				52			
04	Computing Units	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c
	Optimal solution for required resource	2	1	1	1	2	1	1	1	2	1	1	1	3	2	1	1
	Convergence time(second)	12.2650				41.8600				29.9840				12.656			
	No. of generation	51				104				315				51			
05	Computing Units	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c	R_m	R_a	R_s	R_c
	Optimal solution for required resource	2	1	1	1	2	1	1	1	2	1	1	1	3	1	1	1
	Convergence time(second)	12.1560				41.3590				54.8130				13.1720			
	No. of generation	51				101				551				51			

with DE. The minimum convergence to obtain minimum objective value for DE, GA, EP and PSO of the 5th trail is shown in Figure 2 (a), (b), (c), (d). In the optimal resources for scheduling in architectural level synthesis, the optimal value is for multiplier unit = 2, adder unit = 1, substractor unit = 1 and comparator = 1; hence the minimum objective function value obtained is 7 as shown in Figure 3.

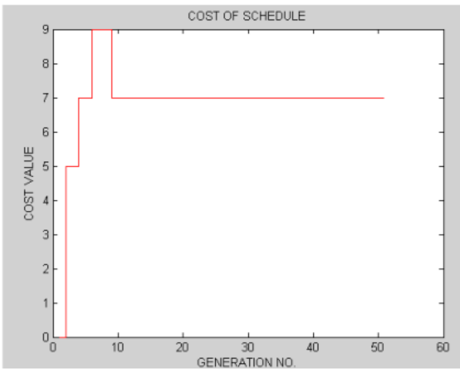

Figure 2a. DE convergence performances.

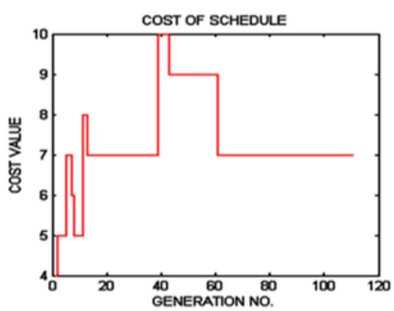

Figure 2b. GA convergence performances.

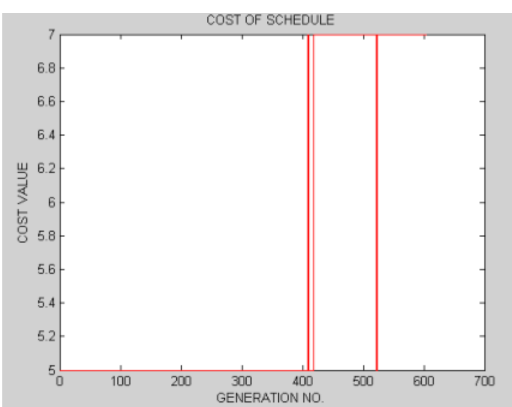

Figure 2c. EP convergence performances.

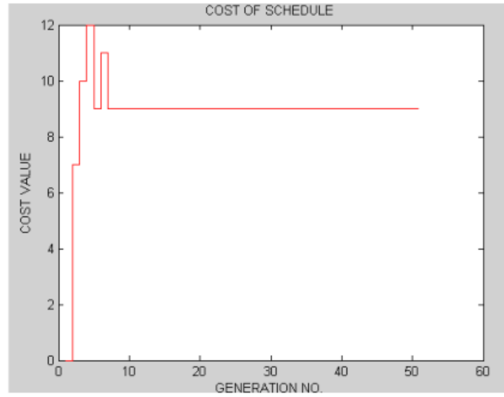

Figure 2d. PSO convergence performances.

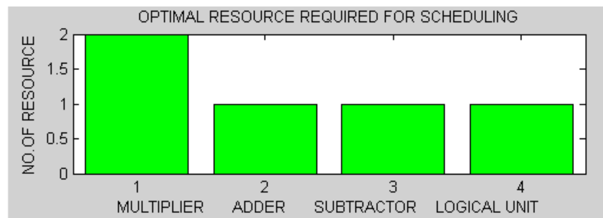

Figure 3. Optimal resource required for scheduling.

6. Conclusion

Comparative study for the performance of Architectural Level Synthesis for Resource Schedule using Nature Inspired Algorithm using DE, GA, EP, PSO are presented. Experimental result indicates DE outperformed PSO, EP and GA in terms of optimal solution achieved, convergence speed, number of generation taken to achieve optimal solution. PSO struck at local minima and EP and GA suffer in having more convergence time. DE proves to be excellent nature inspired algorithm to solve scheduling problem in Architectural Level Synthesis.

7. References

1. Micheli DG. Synthesis and optimization of digital circuits. USA: McGraw-Hill; 1994.
2. Gajski D, Dutt ND, Wu A, Lin S. High level synthesis: Introduction to chip and system design. USA: Kluwer Academic Publisher; 1992. p. 219–24.
3. Shilpa KC, Narayana L. Natural computation for optimal scheduling with ilp modeling in high level synthesis. Science Direct, Procedia Engineering ELSEVIER Journal Publication. 2015; 46:167–75.

4. Back T, Fogel DB, Michalewicz M. Hand book on Evolutionary Computation. Institute of Physics Publication and Oxford University Press; 1997.
5. Goldberg DE. Genetic Algorithms in search, optimization and machine learning. Reading. MA: Addison-Wesley Publishing Co; 1989; 3(2):95–9.
6. Eiben AE, Smith JE. Introduction to evolutionary computing. Natural Computing Series Springer: Berlin Heidelberg: Springer-Verlag; 2003.
7. Eberhart RC, Shi Y. Comparison between Genetic Algorithms and Particle Swarm Optimization. Proceedings of Seventh Annual Conference on Evolutionary Programming; 1998. p. 611–6.
8. Storn R, Price K. Differential Evolution: A simple and efficient adaptive scheme of global optimization over continuous spaces. Journal of Global Optimization; 1997. p. 341–59.
9. Price K. Differential Evolution: A fast and simple numerical optimizer. NAFIPS. Biennial Conference of the North American Berkeley, CA; 1996. p. 842–4.
10. Grewal G, Cleirigh MO, Wineberg M. An evolutionary approach to behavioral-level synthesis. IEEE Trans VLSI circuits; 2003. p. 264–7.
11. Rohini V, Natrarajan AM. Comparison of Genetic Algorithm with Particle Swarm Optimization, Ant Colony Optimization and Tabu Search based on University Course Scheduling System. Indian Journal of Science and Technology. 2016 Jun; 9(21):1–5.
12. Nanda Kumar E, Dhanasekaran R, Mani R. Optimal location and improvement of voltage stability by UPFC using Genetic Algorithm (GA). Indian Journal of Science and Technology. 2015 Jun; 8(11):1–6.
13. Eiben AE, Smith JE. Introduction to Evolutionary Computing. Berlin: Springer; 2003.
14. Kennedy K, Eberhart RC. Particle Swarm Optimization. Proceedings IEEE Int Conf on Neural Networks (Perth Australia). Piscataway, IEEE Service Center: Springer Science+Business Media, LLC; 1995.p. 760–6.
15. Effatnejad R, Rouhi F. Unit commitment in power system by combination of Dynamic Programming (DP), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Indian Journal of Science and Technology. 2015 Jan; 8(2):131–41.
16. Hwang CT, Lee JH, Hsu YC, Lin YL. A formal approach to the scheduling problem in high level synthesis. IEEE Trans on Computer-Aided Design; 1991. p. 464–75.
17. Mohamad NH, Said F. Integer Linear Programming approach to scheduling toll booth collectors problem. Indian Journal of Science and Technology. 2013 May; 6(5):1–6.
18. Lee J, Hsu Y, Lin Y. A new integer linear programming formulation for the scheduling problem in data path synthesis. Proceedings of the International Conference on Computer Aided Design; 1989. p. 20–3.