Indian Journal of Science and Technology, *Vol 9(40), DOI: 10.17485/ijst/2016/v9i40/96821, October 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Comparative Study of Different Strategies using adaptive Differential Evolution for Best Scheduling in Architectural Level Synthesis

K. C. Shilpa[1]*, C. Lakshmi Narayana[1] and Manoj Kumar Singh[2]

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

Abstract

This paper is a comparative study for optimal scheduling in architectural level synthesis using five different strategies in Differential Evolution. In this paper the comparison is performed using Hardware Abstraction Layer (HAL) benchmark scheduling problem using Integer Linear Programming method. The paper implements adaptive scaling factor for mutation operation and variable cross over operation in differential evolution. The experiment results evaluate the performance parameters optimal resource schedule, convergence time among the five strategies are presented.

Keywords: Architectural Level Synthesis, Differential Evolution, Evolutionary Computation, Hardware Abstraction Layer, Integer Linear Programming, Very Large Scale Integration

1. Introduction

Optimization is a procedure of searching optimal solution to satisfy all constraints. The optimization algorithms are proved to be better approach to discover the optimal solution for optimization problem.

Evolutionary Computation has been significant in solving multi objective optimization problem successfully. Evolutionary Computation is successful by its simplicity, robust, achieving global optimization.

The Architecture Level Synthesis (ALS) is mapping of Algorithm to Register Transfer level module[1]. The major task in ALS is Resource Scheduling, which assign the behavioral operator to control time slots.

The motivation for this paper is to formally apply Integer Linear Programming (ILP) approach which guarantees solution quality and guarantee of quickly finding for optimal resource solution problem using five different strategies in Differential Evolution.

The different resource scheduling algorithm and its draw back has shown before[2], Differential Evolution[3,4] has better convergence and few control parameters. The advantages of DE are its simple structure, ease of use, speed and robustness. DE is one of the best genetic type algorithms for solving problems with the real valued variable.

2. Differential Evolution (DE)

Differential Evolution (DE)[5,6] is a Evolutionary Computation search algorithm introduced by Storn and Price (1997). The method is based on evolution of population and operators crossover, mutation and selection with unique feature of DE is differential weight. The DE is simple algorithm, robust with fast convergence to optimal solution.

3. Problem Formulation

Latency constrained for the Hardware Abstraction Layer (HAL) benchmark problem[7] shown in above Figure 1, the number of computing resource of the multi-

*Author for correspondence

plier, adder, subtraction and comparator in the Figure 1 is: $R_m = 5, R_a = 2, R_s = 2, R_c = 1$. Computing unit are cost of the multiplier, adder, subtraction and comparator: C_m, C_a, C_s, C_c. 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.

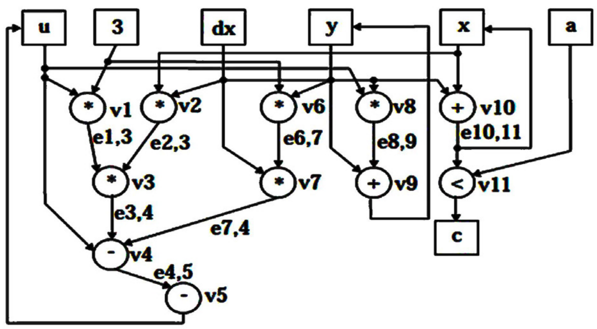

Figure 1. Hardware Abstraction Layer benchmark problem.

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 = ASAP (AS SOON AS POSSIBLE) and = ALAP(AS LATE AS POSSIBLE) values

$$M = \{0 j | E_k \le j \le k \tag{1}$$

- Secondly the INTEGER LINEAR PROGRAMMING formulation is given as follows

$$\text{Min} \sum_{k=1}^{m} [C_k * R_k] \text{ while } \sum_{E_i \le j \le i} X_{i,j} = 1 \tag{2}$$

Where 1 ≤ k ≤ m indicate the number of resource operation available, R_k term is the computing resource type for operation k and C_k term is the cost of each resource computing type.

$$[x_{i,j} = 1], \ \forall i \ operation = \text{j} \tag{3}$$

else = 0, otherwise

- Thirdly the constraints on resource type:

$$\sum_{k=1}^{n} [x_{ij} \le R_i] \tag{4}$$

- Finally the constraint on data dependency:

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

5. Experiments

5.1 Experimental Setup

The fitness function considered is shown in (6):

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

a = 1000, g_k^{+} (≤ 0) and h (= 0) are constraints violation terms.

The parameters setting for algorithm are DE Setup: N = population size = 200, Dimensional vector Xi = (xi1, xi2, xi3,….., xiD), D-dimensional of search space, adaptive Differential Evolution scaling factor is the estimated by mean eucline distance in (7).

$$f_i = \left(d_g - d_{\max}\right) / \left(d_{\max} - d_{\min}\right) \tag{7}$$

d_g = *distance value for best solution*
d_{max} = *maximum value of mean eucline distance*
d_{min} = *minimum value of mean eucline distance*

Mean eucline distance[10,11] is estimated as follows in (8):

$$d_i = \frac{1}{N-1}\sum_{j=1}^{N} j \ne 1 \sqrt{\sum_{k=1}^{D}\left(x_i^k - x_j^k\right)^2} \tag{8}$$

τ1=0.1,*ran, ran* are four different random variable.
Variable factor for binomial crossover cr in (9):

$$cr = \text{rand}1 \quad \text{if } rand2 \ < \tau 1;$$
$$cr = 0.8 \tag{9}$$

The strategies used for Differential Evolution[12] are given below:

- DE1: DE/best/1 = DE: Differential Evolution, best: Minimum value of objective function, 1: Number of difference vector = 1.
- DE2: DE/best/2 = DE: Differential Evolution, best: Minimum value of objective function, 2: Number of difference vector = 2.
- DE3: DE/rand to best/1 = DE: Differential Evolution, rand: Randomly chosen population,

best: Minimum value of objective function, 1: Number of difference vector = 1.

- *DE4:* DE/rand/1 = DE: Differential Evolution, rand: Randomly chosen population, 1: Number of difference vector = 1.
- DE5: DE/rand/2 = DE: Differential Evolution, best: Minimum value of objective function, 2 is number of difference vector = 2.

6. Results and Discussion

The 5 different strategies comparative results for the performance of DE with variable scaling factor and variable cross over factor. The performances parameters are checked with optimization algorithm are optimal solution obtained for computing unit (multiplier, adder, subtraction and comparator). Numbers of generation taken for convergence, Convergence time (taken in seconds) are presented. Figure 2 shows the convergence performance graph obtained to achieve the minimum optimal cost minimized factor.

6.1 Discussion

Comparative study for the performance of latency constrained scheduling using DE is shown in Table 1 for 2 trails. For all the trails DE3 is the best in finding optimal solution, takes minimum convergence time taken to achieve minimum objective function. DE1 is also best in finding optimal solution similar to DE3; but compared to DE3, convergence time is more. DE2 convergence time is less than DE1, but fails in getting optimal solution. DE4, DE5 fails to satisfy the constraints, suffer badly to deliver optimal solution. DE4, DE5 shows the worst performance for the scheduling problem.

The minimum convergence to obtain minimum objective value for DE1, DE2 and DE3 are presented in Figure 2 (a), (b), (c). Figure 3 shows the required optimal resources for scheduling in architectural level synthesis, the optimal value of multiplier unit = 2, adder unit = 1, substractor unit = 1, comparator = 1, hence minimum objective function value obtained is 7.

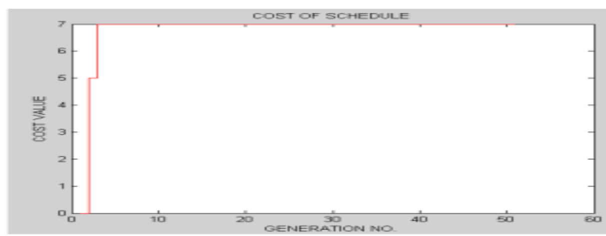

Figure 2a. DE1 convergence performance.

Table 1. Comparative results for the performance of DE

Strategies			PerformanceParameters					
			Computing Units Optimal solution for required resource				Convergence time(second)	No. of generation taken to converge
			R_m	R_a	R_s	R_c		
DE1	DE/best/1	Trail 1	2	1	1	1	13.8750	51
		Trail 2	2	1	1	1	13.4530	51
DE2	DE/best/2	Trail 1	3	1	1	1	13.0160	51
		Trail 2	3	1	1	1	12.9850	51
DE3	DE/rand to best/1	Trail 1	2	1	1	1	12.8430	51
		Trail 2	2	1	1	1	12.7180	51
DE4	DE/rand/1	Trail 1	-	-	-	-	-	-
		Trail 2						
DE5	DE/rand/2	Trail 1	-	-	-	-	-	-
		Trail 2						

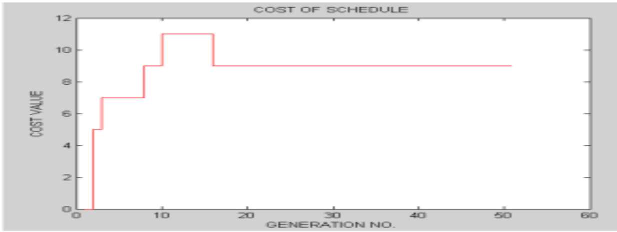

Figure 2b. DE2 convergence performances.

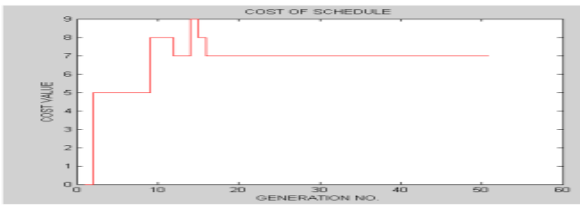

Figure 2c. DE3 convergence performance.

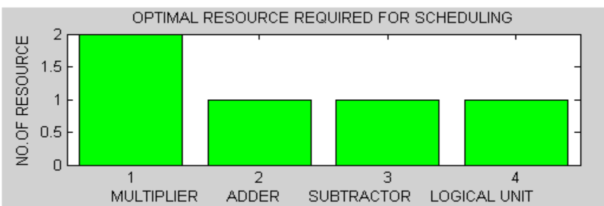

Figure 3. Optimal resource required for scheduling.

7. Conclusion

Comparative study for the performance of Architectural Level Synthesis for resource schedule using Differential Evolution is presented. Experimental result indicates DE3 outperformed DE1, DE2 in terms of optimal solution achieved, convergence speed taken to achieve optimal solution. DE4, DE5 fails to deliver optimal solution.

DE3 proves to be excellent optimization algorithm to solve scheduling problem in architectural level synthesis.

8. References

1. De Micheli G. Synthesis and optimization of digital circuits. USA: McGraw-Hill; 1994.
2. Shilpa KC, Lakshmi Narayana. Natural computation for optimal scheduling with ILP modeling in high level synthesis. Science Direct Procedia Engineering ELSEVIER Journal Publication. 2015; 46:167–75.
3. Storn R, Price K. Differential evolution: A simple and efficient adaptive scheme of global optimization over continuous spaces. Journal of Global Optimization. 1997; 11(4):341–59.
4. Price K. Differential Evolution: A fast and simple numerical optimizer. NAFIPS; 1996. p. 842–4.
5. Price KV. An introduction to Differential Evolution. New Ideas in Optimization. D. Corne, M. Dorigo and F. Glover, editors. UK: McGraw-Hill; 1999. p. 79–108.
6. Price K, Storn RM, Lampinen JA. Differential Evolution: A practical approach to global optimization. Springer-Verlag; 2005. ISBN: 3-540-20950-6.
7. 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.
8. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V. Self-adapting control parameters in Differential Evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation. 2006; 10(6):646–57.
9. Teo J. Exploring dynamic self-adaptive populations in Differential Evolution. Soft Computing - A Fusion of Foundations, Methodologies and Applications. 2006; 10(8):673–86.
10. Brest J, Boskovic B, Greiner VZS, Maucec M. Performance comparison of self-adaptive and adaptive Differential Evolution algorithms. Soft Computing - A Fusion of Foundations, Methodologies and Applications. 2007; 11(7):617–29.
11. Brest J, Maucec M. Self-adaptive Differential Evolution algorithm using population size reduction and three strategies. Soft Computing - A Fusion of Foundations, Methodologies and Applications. 2011; 15:2157–74.
12. Price K, Storn R, Lampinen J. Differential Evolution: A practical approach to global optimization. Springer-Verlag; 2005.