IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 4, Ver. I (Jul. - Aug. 2016), PP 55-59
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Adaptive Differential Evolution For Optimal Schedule In Behavioral Level Synthesis

Shilpa K.C [1], LakshmiNarayan .C [2], Manoj Kumar Singh
[1]*(BMSCE/ Visvesvaraya Technological University, India*
[2]*(BMSCE/ Visvesvaraya Technological University, India*
[3]*(Manuro Research Tech, India)*

Abstract: *This paper presents the adaptive differential evolution for optimal scheduling in Behavioral level synthesis. The benchmark problem for the scheduling problem taken is Hardware Abstraction Layer (HAL) benchmark scheduling problem using Integer Linear Programming method. The adaptive scaling factor for mutation operation in differential evolution is implemented. The experiment results evaluate the performance parameters optimal resource schedule. The exploration and exploitation to global optimal scheduling with minimum convergence time and minimum number of computations are presented.*

Keywords: *optimal schedule, evolution computation, differential evolution, hardware abstraction layer, Integer Linear Programming.*

I. Introduction

The procedure of High Level Synthesis [1] with Scheduling includes the obligation of the Data Flow Graph procedures to numerous time periods. Data Flow Graph tasks to physical arithmetic functional units that is adders, multipliers etc. simultaneously the consequent Allocation of hardware resources that is registers and mux to enable the data transmission in the Data Flow Graph.

The High Level Synthesis gives the rise to automatic or manual process for elimination of source of many errors which are related to design that can be called as design errors and part of development cycle is accelerating for a very long cycle. High Level Synthesis gives the great benefits which also gives the disruptive solution of technology. The process of High Level Synthesis with Scheduling will give the many problems like area consumption, increase in the delay, more power consumption also the soft errors because of the minimizing the designing process which all this give rise on effective increase in cost of the design or methodology.

The Scheduling problem is characterized as (nondeterministic polynomial time) NP-complete algorithms The various scheduling algorithm exist in literature to find the optimal solution for NP problems. The simple Optimal scheduling algorithm are ASAP (As soon as possible) and ALAP (As late as possible) [2], List Scheduling [3], Forced-Directed Scheduling (FDS) [4], Path-Based Scheduling [5], Integer Linear Programming (ILP) [6].

Integer Linear Programming (ILP) approach leads to exact solutions the computation time process associated to ILP usually become too large.

Evolution Algorithm differs from techniques that change a populace of solution or separate points in research, instead of starting by a single point. It is called as offspring by mutating and/or by generating of Evolution Algorithm. Differential Evolution is one of Evolutionary Algorithm [7] which is simple and efficient optimization approach for solving and reducing several benchmark problem also real world applications.

Experimental result proves the proposed algorithm Differential Evolution with ILP formulation was found to generate better scheduling results in the Hardware Abstraction Layer (HAL) differential equation solver high-level synthesis benchmark, with minimum convergence time and minimum of computations.

II. Differential Evolution (DE)

The critical based idea after Differential Evolution is a scheme for producing experimental parameter paths. DE is real valued representation introduced by Storn and Price [8]; this adds the subjective alteration amongst two populace vectors to third direction. This involvement delivers tasks to the best parameter fixed by Evolutionary Algorithm.

The advantages of Differential Evolutions are as follows:

- DE is an optimization method of evolutionary computation.
- The main advantage of DE [9] is the rate of convergences is faster, with minimum number of computations.
- This is simple for implementation.
- The exploration and exploitation operators are mutation, crossover and selection.

- It has been widely used in various science and engineering fields such as electromagnetic, power system optimization, chaotic systems, engineering design problems etc.

2.1. The Steps Involved in Differential Evolution:

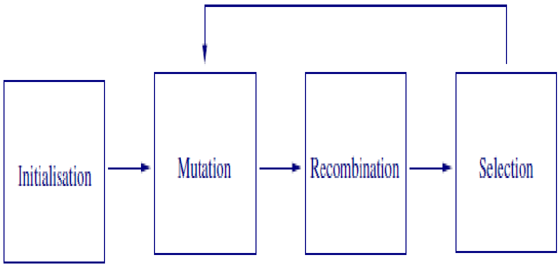

Fig.1. Differential Evolution Representation

Initialization: the process is population based, so initially the population selected is Xi= (xi1, xi2, xi3,….., xiD), D- Dimensional of search space, where i=1,2,3…N where N is population size , d is dimension. Given constraint vector xi, it will take three vectors like xr1, xr2, xr3 that will indicates the indices i, r1, r2, r3 and r4.

- **Initialization:** the process is population based, so initially the population selected is Xi= (xi1, xi2, xi3,….., xiD), D- Dimensional of search space, where i=1,2,3…N where N is population size , d is dimension. Given constraint vector xi, it will take three vectors like xr1, xr2, xr3 that will indicates the indices i, r1, r2, r3 and r4.
- **Mutation:** introduce the indication into the populace by arbitrarily producing differences in the exiting isolated characters. Following are the few various mutation strategies used for literature.
- DE/rand/1 vi=xr0+ fi(xr1-xr2)
- DE/rand-to-best/vi,g=xr1+fi(xbest-xr2)+fi(xr3-xr4)
- DE/best/vi, g=xbest+fi (xr1-xr2).

Where xr1, x2... is the

- **Recombination** or it is also called as **Crossover:** which perform the exchange of data in between the population of individuals.
- **Selection:** This step selects best from the parent path xi, and the trial path vi, as the selection process with respect to the fitness values f.

III. Problem Statement

Optimal Schedule for the Hardware Abstraction Layer (HAL) [10] benchmark problem shown in above Figure 2, the number of computing resource of the multiplier, adder, subtraction, and comparator in the Figure 2 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.

The objective function for Integer Linear Programming [10] formulation is given as follows

$$\text{Min } \sum_{k=1}^{m} [C_k * R_k] \qquad (1)$$

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. The goal of the problem is to minimize the Resource unit for the scheduling problem.

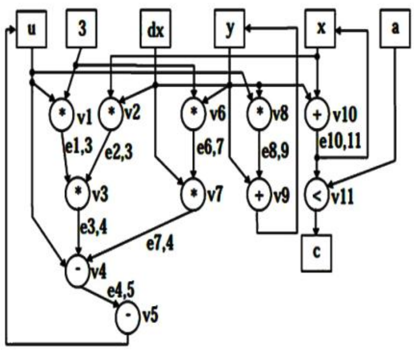

Fig.1. Hardware Abstraction Layer benchmark problem

IV. Experimental Setup

The objective function is to minimize the cost of the computational units. Resource scheduling in high level synthesis with differential evolution algorithm the number of functional units (adders, multipliers, etc.), as well as the subsequent allocation of hardware resources (registers, multiplexors, etc.) are reduced with the result of area , power and delay reduced with effective cost reduction.

- The fitness function considered is shown in (2)

$$F = f + a[\sum_{k=1}^{r}(g_k^{+}(x_i))^2 + \sum_{m=1}^{n}(h(x_i))^2] \qquad (2)$$

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

The pseudo code of DE is as shown in Figure 3:

```
For each particle
  Initialize particle and fitness
End

Do
For each particle
  Create mutant
  Perform crossover
  If an offspring is better than the parent
    Replace parent in the next generation
  Calculate fitness
End
While maximum iterations is not reached
```

Fig.3. The pseudo code of DE

Initialization:

- 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 fi.

Mutation:

- The strategy taken in this paper is:
  DE/rand to best/1 = DE: Differential Evolution, rand: randomly chosen population, best: minimum value of objective function, 1: number of difference vector=1, The mutation vi strategy is given as follows:
  vi = rand1 +fi (xbest-rand2) + fi (rand3-rand4), where, rand1, rand2, rand3, rand4 are four random variable, fi is the adaptive scalar adaptive scalar mutation factor (Evolutionary factor fi)

- The adaptive scalar mutation factor to explore to optimal solution is given by
  fi is the estimated by mean eucline distance in (3)
  fi = (dg-dmax)/ (dmax-dmin) (3)
  dg=distance value for best solution

dmax=maximum value of mean eucline distance.
dmin=minimunm value of mean eucline distance.
Mean eucline distance is estimated as follows in (4)

$$d_i = 1/(N-1)(\sum_{j=1,j\neq i}^{N}\sqrt{\sum_{k=1}^{D}(x_i^k - x_j^k)^2}) \qquad (4)$$

Crossover:

- τ1=0.3, rand1, rand2 are two different random variable. τ1= scaling crossover factor.

Variable factor for binomial crossover cr in (5):
cr= rand1 if rand2< τ1; (5)
else cr=0.7.

Selection

- The selection process select best from the parent path xi, and the trial path vi, as the selection process with respect to the fitness values.

V. Results And Discussion

- 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.
- The performance of ILP formulation using DE for 10 trails is shown in Table 1 for DE/rand to best/1, the optimal solution, and takes minimum convergence time taken to achieve minimum objective function. For all the 10 trails DE achieved the minimum convergence time and minimum generation to achieve the global solution

Table1. Performance of DE achieved for the HAL bench mark problem

Figure 4 shows the best leader achieved in the population, Figure 5 shows the variable adaptive scaling factor convergence, The computing units optimal values obtained are two *multiplier, one adder, one subtraction and one comparator* The minimum convergence to obtain minimum objective value obtained is

DE/rand to best/1	Performance Parameters					
	Computing Units Optimal solution for required resource				Convergence time (second)	No. of generation taken to converge
	R_m	R_a	R_s	R_c		
Trail 1	2	1	1	1	9.2500	25
Trail 2	2	1	1	1	9.9380	25
Trail 3	2	1	1	1	8.7810	25
Trail 4	2	1	1	1	9.2040	25
Trail 5	2	1	1	1	9.1720	25
Trail 6	2	1	1	1	8.7970	25
Trail 7	2	1	1	1	9.5630	25
Trail 8	2	1	1	1	9.3200	25
Trail 9	2	1	1	1	9.5940	25
Trail 10	2	1	1	1	9.8130	25

2*2 + 1*1 +1*1+1*1= 7 and presented in Figure 6.

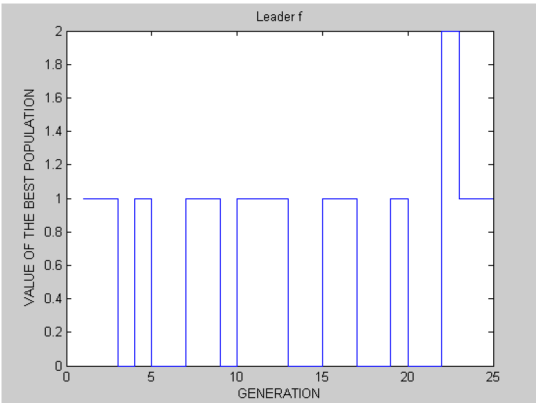

Fig 4. DE convergence performance

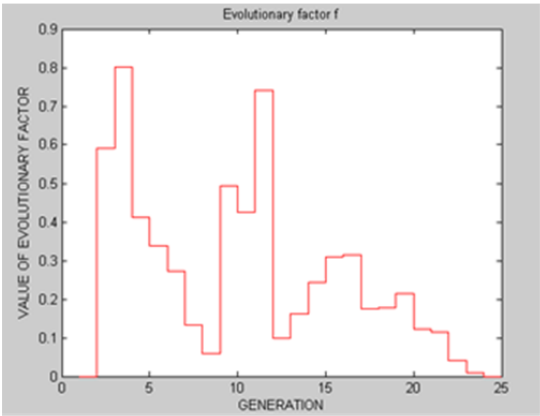

Fig 5. Scaling factor fi convergence performance

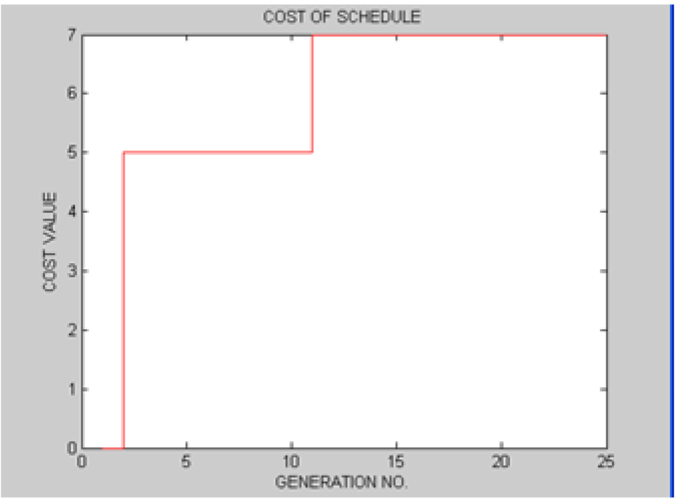

Fig 6. Minimum objective value obtained

VI. Conclusion

The performance of Behavioral Level Synthesis for optimal Schedule using Differential Evolution with ILP formulation is presented. Experimental result indicates DE/rand to best/1 outperformed in terms of optimal solution achieved with minimum convergence speed and minimum computation taken to explore optimal solution.

References

[1] De Micheli .G. Synthesis and Optimization of Digital Circuits. USA: McGraw-Hill; 1994. J.
[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] Camposano.R . Path-based scheduling for synthesis. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1991. p. 85–93.
[6] Shilpa K.C, LakshmiNarayana, "Natural Computation for Optimal Scheduling with ILP modeling in High Level Synthesis. In: Science Direct, Procedia Engineering ELSEVIER Journal Publication. vol 46.
[7] Grewal G, Cleirigh M.O, Wineberg M. An Evolutionary Approach to Behavioral-Level Synthesis. In: *IEEE Trans. VLSI circuits*. 2003. p. 264-272.
[8] Storn. R, Price.K. Differential evolution: A simple and efficient adaptive scheme of global optimization over continuous spaces. In: Journal of Global Optimization. 1997. p.341-359.
[9] Price.K. Differential Evolution: A Fast and Simple Numerical Optimizer, In: NAFIPS.1996.p.842-844.
[10] Lee.J, HsuY, Lin Y. A new Integer Linear Programming Formulation for the Scheduling Problem in Data Path Synthesis. In: Proceedings of the International conference on Computer Aided Design.1989. p.20-23.