FUZZIFIED ITERATIVE ALGORITHMS FOR PERFORMANCE DRIVEN LOW POWER VLSI PLACEMENT

Sadiq M. Sait *Habib Youssef* *Junaid A. Khan* *Aiman El-Maleh*

King Fahd University of Petroleum and Minerals, KFUPM Box 673, Dhahran-31261, Saudi Arabia
{sadiq,youssef,jakhan,aimane}@ccse.kfupm.edu.sa

Abstract

In this paper, we employ fuzzified simulated evolution and stochastic evolution algorithms for VLSI standard cell placement targeting low power dissipation and high performance. Due to the imprecise nature of design information at the placement stage, the various objectives and constraints are expressed in fuzzy domain. The search is made to evolve towards a vector of fuzzy goals. The proposed algorithms are compared with genetic algorithm.

1. INTRODUCTION

With advancement in VLSI technology, delay and power dissipation due to interconnects have become very significant [1]. In this work, the problem of minimizing interconnect delay and power dissipation is addressed at the placement stage [2]. In CMOS circuits, over 90% power dissipation is due to the switching activity [3], expressed as:

$$P_t \simeq \sum_{i \in M} \frac{1}{2} \cdot C_i \cdot V_{DD}^2 \cdot f \cdot S_i \cdot \beta \qquad (1)$$

where P_t denotes the total power, V_{DD} is the supply voltage, S_i is the switching activity of cell i, and f the clocking frequency. The node total capacitance is denoted by C_i, and β is a technology dependent constant. In case of standard cell design, cell characteristics are fixed for a particular library, therefore we cannot reduce cell capacitances. Furthermore, interconnect capacitances are related to the corresponding interconnect wire-lengths l_i.

The cost function due to timing performance in the placement problem can be expressed as Cost$_{delay} = T_{\pi_c}$, where π_c is the most critical path in the layout. In this work, layout width is considered as a constraint.

2. FUZZY COST MEASURE

In order to select the best solution found so far, a goal directed search approach is adopted, where the best placement is one that satisfies as much as possible a user specified vector of fuzzy goals [4].

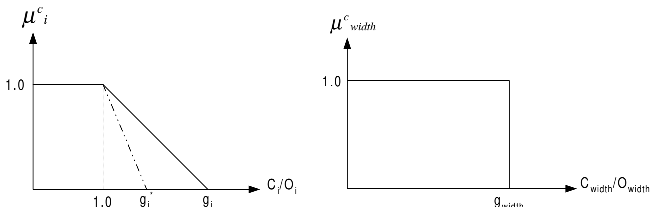

Figure 1: Membership functions within acceptable range.

In order to combine three parameters and one constraint, the following fuzzy rule is suggested.

Rule R1: **IF** a solution is within *acceptable wire-length* AND *acceptable power* AND *acceptable delay* AND *within acceptable layout width* **THEN** it is an acceptable solution.

Using ordered weighted averaging operator (OWA) [5], the above fuzzy rule translates to the following:

$$\mu^c_{pdw}(x) = \beta^c \times \min(\mu^c_p(x), \mu^c_d(x), \mu^c_w(x)) + (1-\beta^c) \times \frac{1}{3} \sum_{j=p,d,w} \mu^c_j(x) \qquad (2)$$

$$\mu^c(x) = \min(\mu^c_{pdw}(x), \mu^c_{width}(x)) \qquad (3)$$

where $\mu^c(x)$ is the membership of solution x in fuzzy set of acceptable solutions, $\mu^c_{pdw}(x)$ is the membership in fuzzy set of "acceptable power AND acceptable delay AND acceptable wire-length", whereas $\mu^c_j(x)$ for $j = p, d, w, width$, are the individual membership values in the fuzzy sets *within acceptable power, delay, wire-length, and layout width,* respectively. In our case we chose $\beta^c = 0.7$, the superscript c represents "cost". The solution that results in maximum value of $\mu^c(x)$ is reported as the best solution found by the search heuristic. Notice that the third AND operator in the above fuzzy rule is implemented as a pure $\min$ because the width constraint has to be always satisfied.

The shape of membership functions for fuzzy sets *within acceptable power, delay and wire-length* are as shown in Figure 1(a), whereas the constraint *within acceptable layout* *width* is given as a crisp set as shown in Figure 1(b). Since layout width is a constraint, its membership value is either 1 or 0 depending on $goal_{width}$ (in our case $goal_{width} = 1.25$). However, for other objectives, by increasing or decreasing the value of $goal_i$ one can vary its preference in the overall membership function. O_is for $i \in \{w, p, d, width\}$ represent the lower bounds for wire-length, power, delay and layout width respectively.

3. EVOLUTIONARY ALGORITHMS

3.1. Stochastic Evolution (StocE)

Stochastic evolution [6], is a relatively recent iterative search heuristic. Unlike simulated annealing, it totally avoids the random search even at the early stages. The general stochastic evolution algorithm can be found in [6]. StocE takes as inputs (1) an initial solution S_0, (2) an initial value of a control parameter p_0, and (3) a stopping criteria parameter R.

There are two StocE parameters that have to be decided, p_0 and R. p_0 is a control parameter to accept uphill moves (bad solutions). In order to select the value of p_0, N random trial moves are made (where N is the number of cells in the circuit). The standard deviation σ_N of the membership in the fuzzy set of acceptable solutions due to these trial moves is computed. Then, p_0 is assigned the value $2\sigma_N$. If the search is stuck at some local optimal solution, then the value of p is increased to $2p_0$, otherwise $p = p_0$.

The second parameter to be specified for StocE is the value of R. It represents the expected number of iterations the StocE algorithm needs until an improvement in the cost with respect to the best solution seen so far takes place. It is suggested in [7] that the best results are obtained with $10 < R < 20$. In this work, $R = 15$ is chosen. Upon termination, the algorithm returns the best solution found so far by the heuristic.

3.2. Fuzzy Simulated Evolution (FSE)

The general Simulated Evolution (SE) algorithm was proposed in [8]. In order to apply simulated evolution, one has to design a suitable goodness measure, a cost function, and an appropriate allocation operator. These three together have the most impact on the behavior of the SE algorithm. Due to the multi-objective nature of the placement problem, the goodness measure, cost function, and the allocation operator should take into consideration all objectives.

Fuzzy Goodness Evaluation: A designated location of a cell is considered good if it results in short wire-length for its nets, reduced delay, and reduced power. These conflicting requirements can be expressed by the following fuzzy logic rule.

Rule R2: **IF** cell i is *near its optimal wire-length* AND *near its optimal power* AND (*near its optimal net delay* OR $T_{\max}(i)$ *is much smaller than* $T_{\max}$) **THEN** it has a high goodness.

where $T_{\max}$ is the delay of the most critical path in the current iteration and $T_{\max}(i)$ is the delay of the longest path traversing cell i in the current iteration.

With the AND and OR fuzzy operators implemented as OWA operators, rule **R2** evaluates to the expression below:

$$\text{goodness}_i = \mu_i^e(x) = \beta^e \times \min(\mu_{iw}^e(x), \mu_{ip}^e(x), \mu_{id}^e(x)) + (1 - \beta^e) \times \frac{1}{3} \sum_{j=w,p,d} \mu_{ij}^e(x) \qquad (4)$$

where

$$\mu_{id}^e(x) = \beta_d^e \times \max(\mu_{inet}^e(x), \mu_{ipath}^e(x)) + (1 - \beta_d^e) \times \frac{1}{2}(\mu_{inet}^e(x) + \mu_{ipath}^e(x)) \qquad (5)$$

β^e and β_d^e are constants between 0 and 1 to control OWA operators, $\mu_{id}^e(x)$ represents the membership in fuzzy set of *good timing performance*.

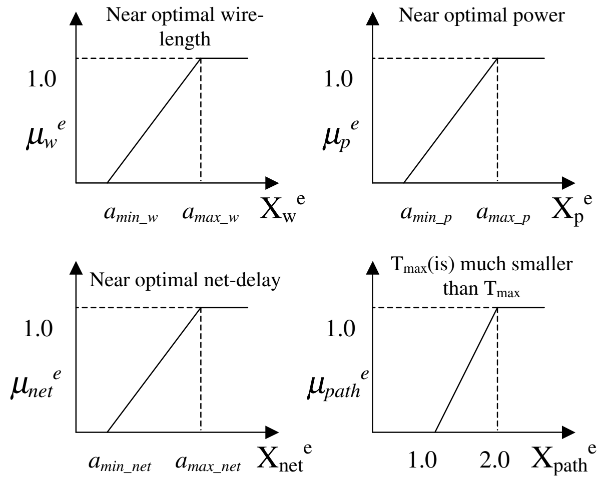

Figure 2: Membership functions used in fuzzy evaluation.

The base values for fuzzy sets near optimal wire-length, power, net delay, and for the fuzzy set "$T_{\max}(i)$ much smaller than $T_{\max}$", for each cell, are represented by $X_{iw}(x)$, $X_{ip}(x)$, $X_{inet}(x)$ and $X_{ipath}(x)$, respectively. Membership functions of these base values are shown in Figure 2.

Selection: In this stage of the algorithm, some cells are selected probabilistically depending on their goodness values. For this purpose we have used an adaptive bias scheme [9].

Allocation: In the allocation stage, the selected cells are to be placed in the best available locations. We have considered selected cells as movable modules and remaining cells as fixed modules. Selected cells are sorted in descending order of their goodnesses with respect to their partial connectivity with unselected cells. One cell from the sorted list is selected at a time and its location is swapped with other movable cells in the current solution. The swap that results in the maximum gain is accepted and the cell is removed from the selection set.

The goodness of the new location is characterized by the following fuzzy rule:

Rule R3: **IF** a swap results in *reduced overall wire-length* AND *reduced overall power* AND *reduced delay* AND *within acceptable layout width* **THEN** it gives good location.

The above rule is interpreted as follows:

$$\mu^a_{i\_wpd}(l) = \beta^a \times \min(\mu^a_{iw}(l),\ \mu^a_{ip}(l),\ \mu^a_{id}(l)) + (1-\beta^a) \times \frac{1}{3} \sum_{j=p,w,d} \mu^a_{ij}(l) \qquad (6)$$

$$\mu^a_i(l) = \min(\ \mu^a_{i\_width}(l),\ \mu^a_{i\_wpd}(l)\ ) \qquad (7)$$

the superscript a is used here to represent allocation. $\mu^a_i(l)$ is the membership of cell i at location l in the fuzzy set of good location. $\mu^a_{i\_wpd}(l)$ is the membership in the fuzzy set of "reduced wire-length and reduced power and reduced delay". $\mu^a_{iw}(l)$, $\mu^a_{ip}(l)$, $\mu^a_{id}(l)$, and $\mu^a_{iwidth}(l)$ are the membership in the fuzzy sets of reduced wire-length, reduced power, reduced delay and within acceptable width, respectively.

The base values of membership functions in allocation are represented as $X^a_{iw}(l)$, $X^a_{ip}(l)$ $X^a_{id}(l)$, and $X^a_{i\_width}(l)$. Membership functions for these base values are shown in Figure 3. The values of a_w, a_p, a_d and a_{width} depend upon priority on the optimization level of the respective objective. In our case, we have set $a_w = 0.75$, $a_p = 0.75$, $a_d = 0.85$ and $a_{width} = 0.25$. The algorithm terminates when no further improvement is observed in the best solution found.

3.3. GA Based Optimization

For comparison purposes, we have also implemented genetic algorithm (GA) [7] with $\mu^c(x)$ as fitness value. Roulette wheel selection scheme [7], is used. Partially mapped crossover (PMX) is used to generate new offsprings. For the selection of next generation, Extended Elitism Random Selection scheme is used, where half of the chromosomes in the next population are the best among offspring and current population and half are selected randomly. A variable mutation is used that depends upon the standard deviation of fitness in the current population. Stopping criteria is the maximum number of generations.

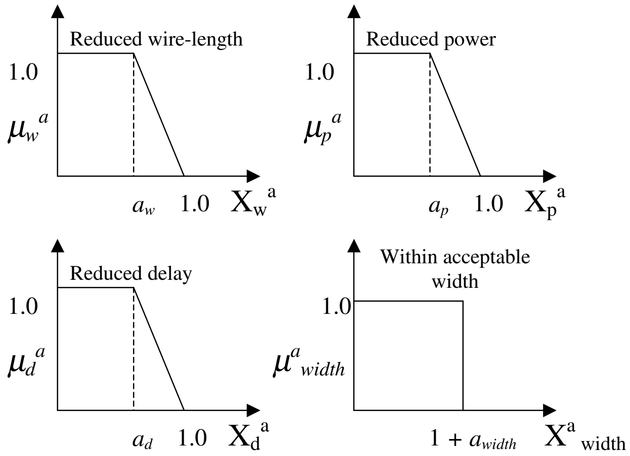

Figure 3: Membership functions used in allocation.

4. RESULTS & DISCUSSION

We applied FSE, StocE and GA on eleven different ISCAS-85 and ISCAS-89 benchmark circuits. In case of FSE and StocE, execution is aborted when no improvement is observed in the last 500 iterations. In case of GA, the stopping criteria was 10,000 generations.

Table 1 compares the quality of the final solution generated by FSE, StocE and GA. The circuits are listed in order of their size (122-1753 modules). From the results, it is clear that StocE performs better than FSE and GA for smaller circuits, whereas for circuits with large number of cells FSE outperforms StocE and GA. In all circuits, it is observed that GA takes considerably large amount of execution time as compared to FSE and StocE.

In order to compare improvement in the quality of solution versus time, we plot the current membership values of the solution obtained by FSE and StocE (Figure 4(a) and (b)). The average fitness (membership) values in a current population obtained by GA versus execution time are plotted in Figure 4(c). These plots are for test case S1196. It can be observed that the quality of solution improves rapidly in FSE- and StocE-based search as compared to GA. This behavior was observed for all test cases

Figures 4(d), (e) and (f) track with time the total number of solutions found by FSE, StocE and GA for various membership ranges. Note however that FSE and StocE exhibited much faster evolutionary rate than GA. For example, after about 100 seconds, almost all new solutions discovered by StocE have a membership in the interval 0.5-0.6 in the fuzzy subset of good solutions with respect to all objectives, and almost none were found with lower membership values. In contrast, for GA, it is only after 10,000 seconds that the first solution with membership in the interval 0.5-0.6 was found (see Figure 4). This behavior was observed for all test cases.

Circuit	FSE				StocE				GA			
	W (μm)	P (μm)	D (ps)	T (s)	W (μm)	P (μm)	D (ps)	T(s)	W (μm)	P (μm)	D (ps)	T (s)
S2081	2693	462	112	43	2139	384	112	414	2426	388	113	2341
S298	4989	1013	133	104	3490	660	125	452	4062	838	130	2922
S386	7088	1640	197	152	5896	1368	191	679	6824	1665	181	3945
S832	24705	5827	258	1643	19644	4517	372	950	21015	4787	232	7206
S641	13906	3321	702	618	15841	3888	689	1426	18320	4365	736	21982
S953	32340	5242	245	1278	29894	4634	225	833	32031	5156	230	11221
S1238	39629	12377	371	1168	49148	14373	393	1104	52679	15473	410	16208
S1196	42426	12745	364	1521	47323	14464	371	844	51804	15259	370	23070
S1494	56961	14071	719	3378	77793	18876	712	477	71021	17497	803	26032
S1488	57091	13887	710	3529	79406	19214	735	143	69792	17346	784	21434
C3540	164897	58055	734	18318	495822	175098	991	2041	310996	109850	924	43232

Table 1: Layout found by FSE, StocE and GA, "W", "P" and "D" represent the wire-length, power, and delay costs and "T" represents execution time in seconds.

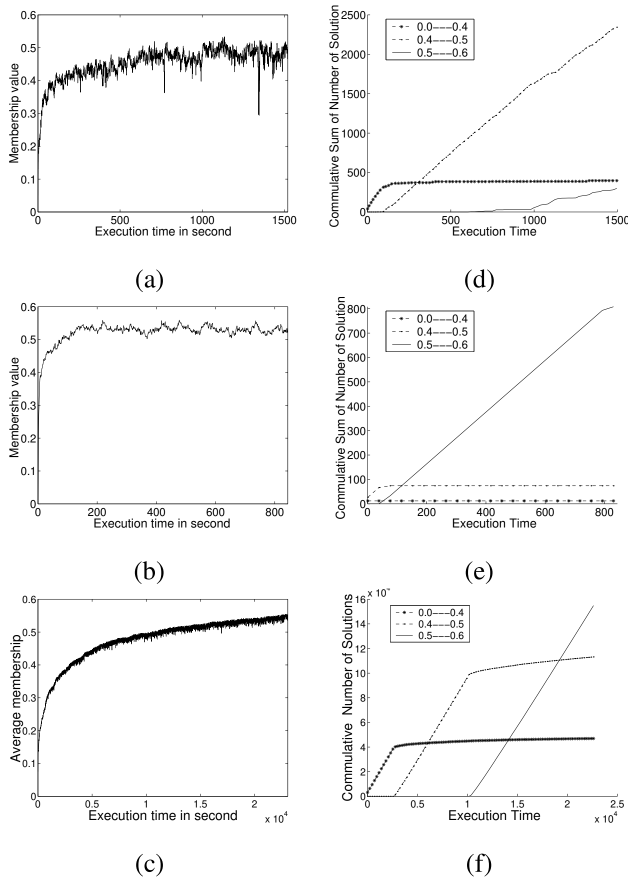

Figure 4: (a), (b) and (c) show Membership values versus execution time for FSE, StocE and GA respectively. (d), (e) and (f) show cumulative number of solutions visited in a specific membership range versus execution time for FSE, StocE and GA respectively.

5. REFERENCES

[1] J. Cong, L. He, C. Koh, and P. Madden. Performance Optimization of VLSI Interconnect Layout. *Integration, the VLSI Journal*, 21:1–94, 1996.

[2] Sadiq M. Sait and Habib Youssef. VLSI Physical Design Automation: Theory and Practice. *McGraw-Hill Book Company, Europe*, 1995.

[3] Srinivas Devadas and Sharad Malik. A Survey of Optimization Techniques Targeting Low Power VLSI Circuits. *32nd ACM/IEEE DAC*, 1995.

[4] Sadiq M. Sait, Habib Youssef, and Ali Hussain. Fuzzy Simulated Evolution Algorithm for Multiobjective Optimization of VLSI Placement. *IEEE Congress on Evolutionary Computation*, pages 91–97, July 1999.

[5] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transaction on Systems, MAN, and Cybernetics*, 18(1), January 1988.

[6] Y. Saab and V. Rao. Stochastic Evolution: A Fast Effective Heuristic for some Generic Layout Problems. *In 27th ACM/IEEE DAC*, pages 26–31, 1990.

[7] Sadiq M. Sait and Habib Youssef. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*. IEEE Computer Society Press, California, December 1999.

[8] R. M. Kling and P. Banerjee. ESP: Placement by Simulated Evolution. *IEEE Transaction on Computer-Aided Design*, 3(8):245–255, March 1989.

[9] Habib Youssef, Sadiq M. Sait, and Ali Hussain. Adaptive Bias Simulated Evolution Algorithm for Placement. *IEEE International Symposium on Circuits and Systems*, pages 355–358, May 2001.

Acknowledgment: Authors thank King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, for support. This work is carried out under university funded project number COE/ITERATE/221.