FUZZY BIASLESS SIMULATED EVOLUTION FOR MULTIOBJECTIVE VLSI PLACEMENT

Junaid A. Khan [1] Sadiq M. Sait [2] Mahmood R. Minhas [2]

[1] Department of Electrical and Computer Engineering,
University of British Columbia, Canada
junaidk@ece.ubc.ca

[2] Department of Computer Engineering,
King Fahd University of Petroleum and Minerals Dhahran 31261, Saudi Arabia
{sadiq, minhas}@ccse.kfupm.edu.sa

Abstract

In each iteration of Simulated Evolution (SE) algorithm for placement poorly placed cells are selected probabilistically based on a measure known as 'goodness'. To compensate for the errors in goodness calculation (and to maintain the number of selected cells within some limit), a parameter known as Bias is used which has major impact on the algorithm run-time and on the quality of solution subspace searched. However, it is difficult to select the appropriate value of this selection bias because, it varies for each problem instance. In this work, a biasless selection scheme for simulated evolution algorithm is proposed. This scheme eliminates the human interaction needed in the selection of bias value for each problem instance. Due to the imprecise nature of design information at placement stage, fuzzy logic is used in all stages of SE algorithm. The proposed scheme was compared with an adaptive bias scheme and was always able to achieve better solutions.

1. Introduction

The placement problem can be stated as follows: Given a set of modules (cells) $M = \{m_1, m_2, \cdots, m_n\}$, and a set of signals $V = \{v_1, v_2, \cdots, v_k\}$, each module $m_i \in M$ is associated with a set of signals V_{m_i}, where $V_{m_i} \subseteq V$. Also each signal $v_i \in V$ is associated with a set of modules M_{v_i}, where $M_{v_i} = \{m_j | v_i \in V_{m_j}\}$. M_{v_i} is called a signal net. Placement consists of assigning each module $m_i \in M$ to a unique location such that a given cost function is optimized and constraints are satisfied [?]. Simulated Evolution (SE) is used in this work to traverse the search space to find an optimal solution [?].

SE is a general, iterative meta-heuristic to solve combinatorial optimization problems [?, ?, ?]. The general SE algorithm is illustrated in Figure ?? and comprises three main steps namely **Evaluation**, **Selection**, and **Allocation**. In **Evaluation** step the **goodness** of each cell, in the range $[0, 1]$, is measured at its current location. The goodness is an approximate indicator of how near a cell is to its optimum location. In **Selection** step unfit cells are selected probabilistically for relocation and this is based on their goodness value. Higher the goodness value, lower is the chance of it being selected. The selected cells are removed from the current solution and reassigned one at a time to new locations in a constructive **Allocation** step, thereby increasing the overall goodness.

In SE, it is not possible to find accurate goodness value for a cell, because accurate optimum location of a cell is unknown. Further, goodness only gives the local view of cell, and in order to compensate the error in goodness calculation and to limit the size of selection set, authors in [?] proposed a bias parameter where a cell i is selected if $goodness_i + B < Random$, where $Random$ is a uniformly distributed random number in the range $[0, 1]$, B is the selection bias with its typical values in the range $[-0.2, 0.2]$, and $goodness_i$ is the goodness value of cell i. However, it is not easy to select the value of B, because it varies for each problem instance and requires the undesireable human interaction. Addressing this problem an adaptive bias scheme was proposed in [?], where in k^{th} iteration bias B_k is computed as follows

$$B_k = 1 - G_{k-1} \tag{1}$$

where G_{k-1} is the average goodness of all cells at the end of $k - 1^{th}$ iteration. This has the following advantages.

- No trial runs are required to find the fixed bias

```
ALGORITHM Simulated_Evolution(B, Φ_initial, StoppingCondition)
NOTATION
B= Bias Value.   Φ= Complete solution.
m_i= Module i.   g_i= Goodness of m_i.
ALLOCATE(m_i, Φ_i)=Function to allocate m_i in partial solution Φ_i
Begin
Repeat
  EVALUATION:
        ForEach m_i ∈ Φ evaluate g_i;
        /* Only elements that were affected by moves of previous */
        /* iteration get their goodnesses recalculated*/
  SELECTION:
        ForEach m_i ∈ Φ DO
            begin
                IF Random > Min(g_i + B, 1)
                THEN
                    begin
                        S = S ∪ m_i; Remove m_i from Φ
                    end
            end
        Sort the elements of S
  ALLOCATION:
        ForEach m_i ∈ S DO
            begin
                ALLOCATE(m_i, Φ_i)
            end
Until   Stopping Condition is satisfied
Return Best solution.
End (Simulated_Evolution)
```

Figure 1: Structure of the simulated evolution algorithm.

value and bias is adjusted automatically according to problem state.

- For poor quality solutions G_k is low, resulting in high bias value and low cardinality of selection set, avoiding large unnecessary perturbations.

- In any iteration only cells having goodness $< G_{k-1}$ have non-zero probability of selection, thus search is always focused on relocating poorly placed cells.

However, along with these benefits the adaptive bias scheme has following drawbacks,

- All the cells with $goodness_i > G_{k-1}$ have zero probability of selection. This fact may lead the search to some local optimal solution, because statistically half of the cells have zero probability of selection.

- If G_{k-1} is low then size of selection set is small and when G_{k-1} is high then size of selection set is large, which contradicts the basic idea of Simulated Evolution algorithm, where size of the selection set has to be decreased with increase in average goodness. Also it is against the behavior of any other iterative search algorithm where big perturbations are made when the solution is bad and smaller perturbations are made with improvement in the quality of solution.

To solve the above problems a **Biasless Selection** scheme is proposed in this paper. In the next section the **Biasless Selection** scheme is explained, followed by the explanation that how the proposed scheme is applied to multiobjective placement problem using fuzzy logic. In the last section the results of proposed scheme are compared with adaptive bias scheme.

2. Biasless Selection

In this scheme the selection bias B is totally eliminated and a cell is selected if $Random > goodness_i$. This can be done as follows.

When number of cells in the problem is large, as in the case of VLSI placement, the goodness distribution among the cells is Gaussian, with mean G_m and standard deviation G_σ. In the proposed selection scheme, instead of using uniformly distributed random number, a Gaussian random number is used. By using Gaussian random number the problem of having cells with zero selection probability can be avoided. The mean R_m and standard deviation R_σ of the random number are calculated as follows

$$R_m = G_m - G_\sigma \qquad (2)$$
$$R_\sigma = G_\sigma \qquad (3)$$

If we use G_m as the mean of random number then it is most likely that around 50% cells will be selected which is not desirable in case of large number of cells. To avoid larger selection set we change the mean of random number to $G_m - G_\sigma$, so that only $12-13\%$ cells will be selected in the initial iterations. These values are determined only in the first iteration. However, with increased number of iterations average goodness will increase and this may cause very small number of cells to be selected. To avoid this, the mean of random number is updated when the number of selected cells goes down to 5% of total number of cells in the problem, as follows

$$R_m = R_m + 0.1 \times R_\sigma \qquad (4)$$

By using this scheme, (1) there is no cell in the design having zero selection probability, hence avoiding local optima, (2) size of selection set reduces with increase in the average goodness value, and (3) update procedure avoids the extremely small selection set.

3. Fuzzy Cost function

In every optimization problem it is necessary to define a cost function. Based on this cost function a solution is said to be superior or inferior to other solutions. In case of VLSI cell placement, three objectives are considered in this work i.e., (1) wirelength minimization,

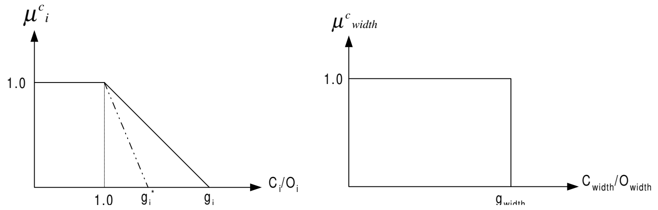

Figure 2: Membership functions within acceptable range.

(2) power dissipation minimization, and (3) circuit delay minimization with satisfying the layout width constraint. It is shown in [?, ?, ?] that cost due to these objectives can be computed as follows,

$$\mathrm{Cost}_{wire} = \sum_{j \in M} l_j \tag{5}$$

$$\mathrm{Cost}_{power} = \sum_{i \in M} S_i l_i \tag{6}$$

$$\mathrm{Cost}_{delay} = T_{\pi_c} \tag{7}$$

where M is the total number of nets in the circuit, l_i is the wirelength estimation of net i, S_i is the switching activity on net i and T_{π_c} is the delay in the current most critical path in the circuit [?, ?].

In order to combine these three objectives 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) [?], the above fuzzy rule translates to the following:

$$\mu^c_{pdw}(x) = \beta^c \times \min(\mu^c_p(x), \mu^c_d(x), \mu^c_w(x)) + (1-\beta^c) \times \frac{1}{3} \sum_{j=p,d,w} \mu^c_j(x) \tag{8}$$

$$\mu^c(x) = \min(\mu^c_{pdw}(x), \mu^c_{width}(x)) \tag{9}$$

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 ??(a), whereas the constraint *within acceptable layout width* is given as a crisp set as shown in Figure ??(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. $O_i s$ for $i \in \{w, p, d, width\}$ represent the lower bounds for wire-length, power, delay and layout width respectively.

4. Biasless Fuzzy Simulated Evolution for VLSI Placement(BLFSE)

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:

$$\mathrm{goodness}_i = \mu^e_i(x) = \beta^e \times \min(\mu^e_{iw}(x), \mu^e_{ip}(x), \mu^e_{id}(x)) + (1-\beta^e) \times \frac{1}{3} \sum_{j=w,p,d} \mu^e_{ij}(x) \tag{10}$$

where

$$\mu^e_{id}(x) = \beta^e_d \times \max(\mu^e_{inet}(x), \mu^e_{ipath}(x)) + (1-\beta^e_d) \times \frac{1}{2}(\mu^e_{inet}(x) + \mu^e_{ipath}(x)) \tag{11}$$

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

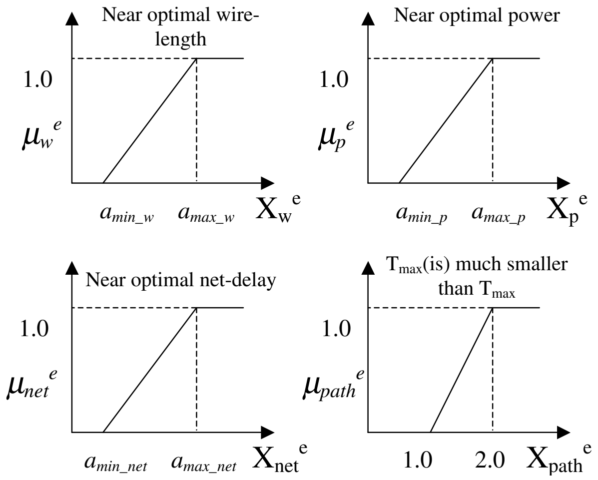

Figure 3: Membership functions used in fuzzy evaluation.

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 ??.

Selection: In this stage of the algorithm, some cells are selected probabilistically depending on their goodness values. For this purpose we have used Biasless selection scheme proposed in Section ??.

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.

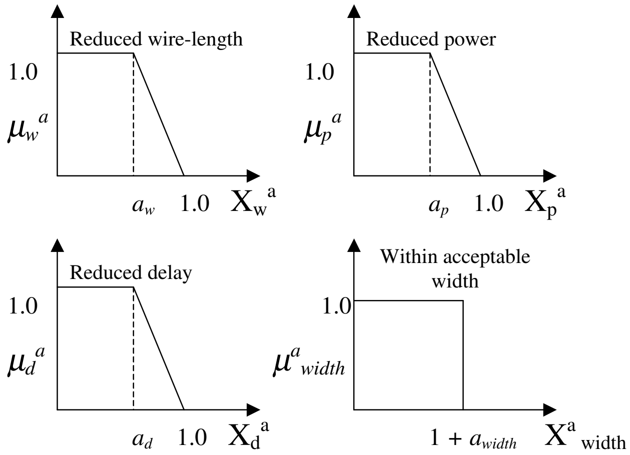

Figure 4: Membership functions used in allocation.

The above rule is interpreted as follows:

$$\mu^a_{i\_wpd}(l) = \beta^a \times \min(\mu^a_{iw}(l),\ \mu^a_{ip}(l),\ \mu^a_{id}(l)) + (1-\beta^a) \times \frac{1}{3} \sum_{j=p,w,d} \mu^a_{ij}(l) \qquad (12)$$

$$\mu^a_i(l) = \min(\ \mu^a_{i\_width}(l),\ \mu^a_{i\_wpd}(l)\ ) \qquad (13)$$

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 ??. 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.

5. Experiments and Results

Biasless Fuzzy Simulated Evolution (BLFSE) and Adaptive Bias Fuzzy Simulated Evolution (ABFSE) is applied on eleven different ISCAS benchmark circuits. In both cases execution is aborted when no improvement is observed in the last 500 iterations.

Table ?? compares the quality of final solution generated by BLFSE, and ABFSE. The circuits are listed in order of their size (136- 2993 modules). From the results, it is clear that BLFSE has outperformed ABFSE

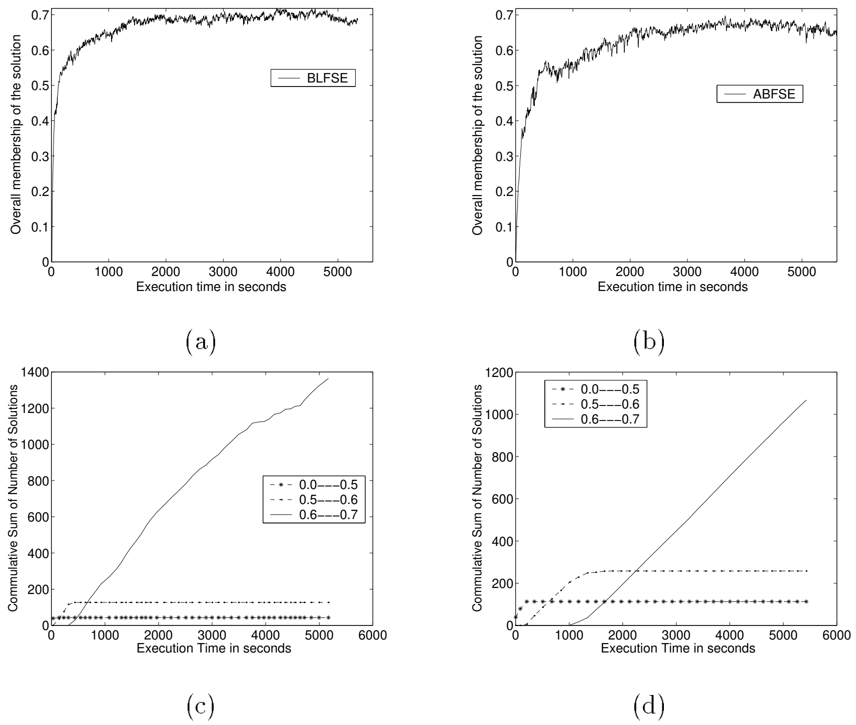

Figure 5: (a), and (b) show Membership values versus execution time for BLFSE, and ABFSE respectively. (c), and (d) show cumulative number of solutions visited in a specific membership range versus execution time for BLFSE, and ABFSE.

for all circuits in terms of final quality of solution. In most of the cases ABFSE has taken smaller execution time, it is because of premature convergence of the search to some local optimal solution. It is due to the fact, that in every iteration half of the cells have zero probability of selection. For larger circuits S3330 and S5378, where execution time is approximately comparable, it can be seen that BLFSE has outperformed ABFSE. In general, BLFSE performs better than ABFSE in terms of quality of final solution.

In order to compare improvement in the quality of solution versus time, we plot the current membership values of the solution obtained by BLFSE and ABFSE (Figure ??-(a) and (b)). These plots are for test case S3330. It can be observed that the quality of solution improves rapidly in BLFSE based search as compared to ABFSE. This behavior was observed for all test cases

Figures ??(c), and (d) track with time the total number of solutions found by BLFSE and ABFSE, for various membership ranges. Note however that BLFSE exhibited much faster evolutionary rate than ABFSE. For example, after about 300 seconds, almost all new solutions discovered by BLFSE have a membership more than 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 ABFSE, it is only after 1,100 seconds that the first solution with membership greater than 0.6 was found (see Figure ??). This behavior was observed for all test cases.

6. Conclusion

In this paper, we have proposed Biasless Fuzzy Simulated Evolution Algorithm for multiobjective VLSI standard cell placement. The use of Bias, which is difficult to find is eliminated from the SE algorithm.

Fuzzy logic is used to overcome the multi-objective nature of the problem. Fuzzy logic is employed at evaluation and allocation stages and in choice of the best solution from the set of generated solutions. The proposed scheme is compared with adaptive bias scheme.

It is observed that BLFSE perform much better than ABFSE in terms of quality of final solution. Furthermore, quality of solution improved more rapidly in BLFSE based search as compared to ABFSE. Unlike ABFSE, non of the cell in BLFSE has zero probability of selection and hence avoid local optimal solutions.

Acknowledgment:
The authors thank King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, for support, under project # : COE/ITERATE/221

Table 1: Layout found by BLFSE, and ABFSE. "L", "P" and "D" represent the wire-length, power, and delay costs and "T" represents execution time in seconds.

Circuit	BLFSE				ABFSE			
	L (μm)	P (μm)	D (ps)	T (s)	L (μm)	P (μm)	D (ps)	T(s)
S298	4548	915	139	46	7130	1395	152	21
S386	8357	2036	203	117	11167	2544	221	33
S832	23140	5251	416	192	28537	6577	485	114
S641	12811	3072	687	175	13773	3107	687	264
S953	29576	5025	223	351	33484	5523	250	130
S1238	41318	12303	363	699	45140	13870	397	295
S1196	35810	11276	360	613	41861	12918	357	433
S1494	54523	12986	768	762	67944	16091	809	279
S1488	57730	13810	700	374	73696	17511	891	216
S3330	183288	24797	459	5351	193731	25373	558	5610
S5378	326840	48360	435	11823	365204	56001	441	11369

7. References

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

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

[3] Sadiq M. Sait, Habib Youssef, and Juaid A. Khan. Fuzzy Evolutionary Algorithm for VLSI Placement. *Genetic and Evolutionary Computer Conference 2001, GECCO-2001*, July 2001.

[4] Junaid A. Khan, Sadiq M. Sait, and A. Almulhem. Algorithms for Fixed Channel Assignment Problem in Wireless Networks. *5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001)*, July 2001.

[5] 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.

[6] Junaid A. Khan. Performance Driven, Low Power, Standard VLSI Cell Placement Using Simulated Evolution. Master's thesis, Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Saudi Arabia, January 2001.

[7] Sadiq M. Sait, Habib Youssef, Juaid A. Khan, and Aiman Al-Maleh. Fuzzy Simulated Evolution for Power and Performance Optimization of VLSI Placement. *INNS-IEEE International Joint Conference on Neural Networks, IJCNN2001*, July 2001.

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