Fuzzy Simulated Evolution Algorithm for Multi-objective Optimization of VLSI Placement

Sadiq M. Sait   Habib Youssef   Hussain Ali

Department of Computer Engineering
King Fahd University of Petroleum and Minerals
Dhahran-31261, Saudi Arabia
e-mail: {sadiq,youssef,hussain}@ccse.kfupm.edu.sa

Abstract: A Fuzzy Simulated Evolution algorithm is presented for multi-objective minimization of VLSI cell placement problem. We propose fuzzy goal-based search strategy combined with a fuzzy allocation scheme. The allocation scheme tries to minimize multiple objectives and adds controlled randomness as opposed to original deterministic allocation schemes. Experiments with benchmark tests demonstrate a noticeable improvement in solution quality.

1 Introduction

Very Large Scale Integration (VLSI) Placement is a combinatorial optimization problem. It consists of arranging circuit blocks on a layout surface such that cost is optimized. This is an NP-Hard problem [1]. Intelligent methods known as "heuristics" are used to get near optimal solutions. Simulated Evolution (SE) [2], simulated annealing [3] and genetic algorithm [4] are iterative stochastic heuristics. SE falls in the category of algorithms which emphasize the behavioral link between parents and offspring, or between reproductive populations, rather than the genetic link [5]. SE combines iterative improvement and constructive perturbation. The advantage of this heuristic is that it requires less execution time compared to simulated annealing and genetic algorithm [2]. In this paper, we report our study of multi-objective optimization of VLSI placement by using fuzzy logic in Simulated Evolution algorithm. During placement process, all desirable objectives can only be imprecisely estimated. Fuzzy logic provides a rigorous algebra for dealing with imprecise information. Furthermore, it is a convenient method of combining conflicting objectives and expert human knowledge. It has been used in many studies related to VLSI placement [6-10]. Most of these studies have used fuzzy logic in constructive heuristics except in [10] where fuzzy cost measure is used in genetic algorithm for the floorplanning problem.

Our proposed Fuzzy Simulated Evolution Algorithm carries out multi-objective optimization of VLSI standard cell placement. In standard cell design, circuit blocks have fixed height and variable widths. Circuit blocks are placed in rows on a two-dimensional layout alternated by routing channels [1]. In our scheme, we minimize three cost parameters of the layout: interconnection wire length, circuit delay, and layout width. The SE algorithm consists of three distinct steps: **evaluation, selection** and **allocation**. Allocation is the most important step of the algorithm [11]. Therefore we propose fuzzification of this step. We propose a "fuzzy controlled stochastic" allocation instead of the previously purely constructive sorted individual best fit allocation strategy. Experiments show that our proposed allocation scheme results in an overall improved solution quality compared to the weighted average allocation scheme. In order to identify the best solution generated by SE algorithm, we use a *fuzzy goal-based cost computation*, which constitutes another major novelty of this work.

The rest of this paper is organized as follows. In Section 2 we briefly review the simulated evolution algorithm and fuzzy logic. In Section 3 we describe the proposed fuzzy allocation scheme and the fuzzy goal-based cost computation. Results of our experiments are given in Section 4. The paper ends with conclusion in Section 5.

2 Preliminaries

2.1 Simulated Evolution Algorithm

Simulated Evolution (SE) is a general iterative heuristic proposed in [2]. This scheme combines iterative improvement and constructive perturbation and saves itself from getting trapped in local minima by using stochastic search approach. This algorithm iteratively operates a sequence of **evaluation, selection** and **allocation** steps on one solution. Other than these three steps, some input parameters like stopping condition, selection bias ($BIAS$) and valid starting solution are initialized in an earlier step known as **initialization.** The flowchart of the algorithm is given in Figure 1.

In the **evaluation** step, **goodness** for each element of the current solution is computed. The goodness of an element is a ratio of its optimum cost to current cost estimate. The goodness is used to probabilistically select elements in the **selection** step. Elements with low

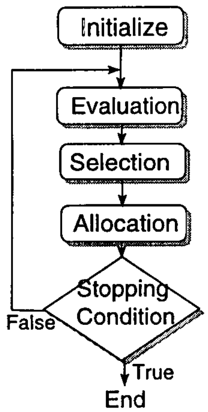


Figure 1: Flow chart of SE algorithm.

goodness have a higher probability of getting selected for reposition. Selection $BIAS$ is used to compensate errors made in estimation of optimum cost. Its objective is to inflate or deflate the goodness of elements. A high positive value of $BIAS$ decreases the probability of selection and vice versa. A carefully tuned $BIAS$ value results in good solution quality and reduced execution time [11]. The selection step results in a partial solution of only unselected elements, while selected elements are saved in a queue for allocation.

The purpose of **allocation** is to perturb the current solution in such a way that the selected elements are assigned to better positions. Different constructive allocation schemes are proposed in [2]. One such scheme is **sorted individual best fit**, where all the selected elements are sorted in descending order in a queue with respect to their connectivity with the partial solution. The sorted elements are removed one at a time and *trial* moves are carried out for all the available empty positions at that time. The element is *finally* placed in a position where maximum reduction in cost for the partial solution is achieved. This process is continued until the selected queue is empty. The overall complexity of this algorithm is $O(s^2)$ where s is the number of selected elements. Other more elaborate allocation schemes are **weighted bipartite matching allocation** and **branch-and-bound search allocation** [2]. However, these schemes are more complex allocation strategies than "sorted individual best fit", while the quality of solution is comparable [2]. In this work we have used fuzzy sorted individual best fit allocation scheme.

2.2 Fuzzy Logic and Fuzzy Set Theory (FST)

A crisp set is normally defined as a collection of elements or objects $x \in X$, where each element can either belong to a set or not. However, in real life situations, objects do not have crisp [1 or 0] membership criteria. fuzzy set theory (FST) aims to represent vague information, like "very hot" and "quite cold", which are difficult to represent in classical (crisp) set theory. In fuzzy sets, an element may partially belong to a set. Formally, a fuzzy set is characterized by a membership function which provides a measure of the degree of presence for every element in the set [12].

Like crisp sets, set operations such as union, intersection, and complementation etc., are also defined on fuzzy sets. There are many operators for fuzzy union and fuzzy intersection. For fuzzy union, the operators are known as **s-norm** operators while fuzzy intersection operators are known as **t-norm**. Generally, **s-norm** is implemented using max and **t-norm** as min function. However, formulation of multi criteria decision functions do not desire pure "anding" of **t-norm** nor the pure "oring" of **s-norm**. The reason for this is the complete lack of compensation of **t-norm** for any partial fulfillment and complete submission of **s-norm** to fulfillment of any criteria. Also the indifference to the individual criteria of each of these two forms of operators led to the development of Ordered Weighted Averaging (OWA) operators [13]. This operator allows easy adjustment of the degree of "anding" and "oring" embedded in the aggregation. According to [13], "orlike" and "andlike" OWA for two fuzzy sets A and B are implemented as given in Equations 1 and 2 respectively.

$$\mu_{A \bigcup B}(x) = \beta \times \max(\mu_A, \mu_B) + (1-\beta) \times \frac{1}{2}(\mu_A + \mu_B) \quad (1)$$

$$\mu_{A \bigcap B}(x) = \beta \times \min(\mu_A, \mu_B) + (1-\beta) \times \frac{1}{2}(\mu_A + \mu_B) \quad (2)$$

β is a constant parameter in the range [0,1]. It represents the degree to which OWA operator resembles the pure "or" or pure "and" respectively.

Fuzzy reasoning: Fuzzy reasoning is a mathematical discipline invented to express human reasoning in vigorous mathematical notation. Unlike classical reasoning in which propositions are either true or false, fuzzy logic establishes approximate truth value of propositions based on linguistic variables and inference rules. In order to represent imprecise ideas, Zadeh [14] introduced the concept of linguistic variable. A **linguistic variable** is a variable whose values are words or sentences in natural or artificial language [6]. The set of values a linguistic variable can take is called a **term set**. This set is constructed by means of primary terms and by placing modifiers known as **hedges** such as "more","many","few" etc., before primary terms. The term set represents a precise syntax in order to form a vast range of values the linguistic variable can take. The linguistic variables can be composed to form propositions using **connectors** like AND, OR and NOT.

3 Fuzzy Simulated Evolution Algorithm

In this section we describe our proposed Fuzzy Simulated Evolution Algorithm. Several stages of SE algorithm can be fuzzified, for instance **allocation, evaluation** and **selection**. In this paper, our emphasis is on fuzzification of the allocation stage. After describing the implementation details of evaluation and selection stages of the SE algorithm in Sections 3.1-3.2, we describe the proposed fuzzy allocation scheme in Section 3.3. In order to evaluate the quality of each layout generated by SE algorithm, we have used a "fuzzy goal-based cost computation" measure. This measure is described in Section 3.4.

3.1 Evaluation

In this stage of the algorithm, individual cell **goodnesses** are computed, where a **cell** is a circuit block. The goodness measure proposed in [2] computes the cell goodness on the basis of wire length. For such a measure, goodness of cell c_i which is a part of $\{v_1, v_2, ..., v_k\}$ nets, where a **net** is an equipotential interconnect of pins on different cells, is computed as follows.

$$g_{c_i} = \frac{1}{k}\sum_{j=1}^{k} \min\left(\frac{L^*_{v_j}}{L_{v_j}}, 1.0\right) \quad (3)$$

where $L^*_{v_j}$ and L_{v_j} are respectively optimum and actual wire length of net v_j. The $L^*_{v_i}$ is computed by placing the cells of a net next to each other on the layout surface and then estimating the wire length.

3.2 Selection

In this stage of the algorithm, for each cell $c_i = \{c_1, c_2, ..., c_n\}$ a random number in the range [0,1] is generated and compared with $g_{c_i} + BIAS$. If the generated random number is greater than $g_{c_i} + BIAS$ then cell c_i is selected for allocation and removed from the layout. The location of cell c_i is marked as *empty*.

3.3 Allocation

During **allocation** stage of the algorithm, the selected cells are repositioned on empty locations in such a way that they result in reduced cost. As described in Section 2.1, "sorted individual best fit" is an allocation scheme for SE algorithm. In this scheme, we identify the best location for the first cell known as *head of line cell* in selected queue which results in maximum reduction in cost. The best location is removed from the list of empty locations for remaining cells in the selected queue. For single objective minimization, it is straightforward to identify such a location. The original SE (OSE) proposal [2] places the cell in a location which results in maximum reduction in wire length cost. However, for multiple and conflicting objectives the decision to identify the best location for selected cell will require many tradeoffs. One way of converting a multi-objective task into a single-objective task is to get a **weighted average** of multiple objectives. For each trial placement, reduction in cost due to individual objective is estimated, normalized and summed together using weights assigned for each objective. The selected cell is finally placed in a position which results in the highest value for the overall reduction in cost.

Assuming a cell c_i is temporarily placed in a location l during the m^{th} iteration of the SE algorithm. This cell is part of $\{v_1, ..., v_k\}$ nets. Let r be the row number of the cell location l. The quality of this trial placement can be measured as follows.

$$\text{gain}^m_{c_i,l} = w^a_1 \times \Delta L_{c_i,l} + w^a_2 \times \Delta D_{c_i,l} + w^a_3 \times \Delta W_{c_i,l} \;; \quad \sum_{h=1}^{3} w^a_h = 1.0 \quad (4)$$

where $\Delta L_{c_i,l}, \Delta D_{c_i,l}, \text{and} \Delta W_{c_i,l}$ are respectively measure of the gains in wire length, delay and width of the layout for the trial placement of cell c_i on location l. Similarly w^a_1, w^a_2 and w^a_3 are respectively averaging weights for these gains in wire length, delay and width of the layout. These gains are computed as follows:

$$\Delta L_{c_i,l} = \frac{\sum_{j=1}^{k}(L^{m-1}_{v_j} - L^m_{v_j})}{\sum_{j=1}^{k} L^{m-1}_{v_j}} \quad (5)$$

$$\Delta D_{c_i,l} = \frac{\sum_{j=1}^{k}(D^{m-1}_{v_j} - D^m_{v_j})}{\sum_{j=1}^{k} D^{m-1}_{v_j}} \quad (6)$$

$$\Delta W_{c_i,l} = \frac{W_{opt} - W^m_r}{W_{opt}} \quad (7)$$

In Equation 5, $L^{m-1}_{v_j}$ and $L^m_{v_j}$ are respectively wire length estimates for net v_j in $m-1^{st}$ and m^{th} iterations of SE algorithms. Similarly $D^{m-1}_{v_j}$ and $D^m_{v_j}$ in Equation 6 are propagation delays for net v_j in $m-1^{st}$ and m^{th} iterations respectively (computation of propagation delays is given in [15]). The W_{opt} in Equation 7 represents the optimum row length (lower bound on maximum row length) for the layout. It is computed by adding the widths of all the cells and dividing it by the number of cell rows. While W^m_r is the row length of row r during the trial placement of cell c_i.

The above weight based sorted individual allocation scheme has two problems.

1. It is difficult to come up with appropriate weights for Equation 4.
2. It is possible that the leading cell in the selected queue will block optimum positions for remaining selected cells.

The use of proper fuzzy rules and membership functions can overcome the first problem. One solution to the second problem is to allow creation of empty space by shifting other cells. However, this will disturb many well placed cells resulting in reduction in overall quality of solution. Therefore we propose a **fuzzy allocation** scheme which is characterized by following two properties.

1. It uses fuzzy rules and membership functions to combine multiple objectives.
2. It adds a *controlled randomness* in placing a cell on an empty location. In this scheme it is possible that a cell is placed any where on a set of empty locations within a *fuzzy window*. The fuzzy window contains locations which result in near identical reductions in cost.

The logic behind the "controlled randomness" is that it decreases the chances of blocking an optimal location for the rest of the selected cells by head of line cell. In the following text we describe the proposed **fuzzy allocation** scheme.

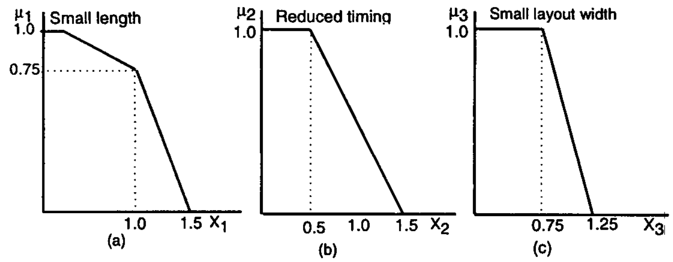


Figure 2: Membership functions for three fuzzy variables used in "Fuzzy Allocation Scheme" of the SE algorithm

Fuzzy Allocation Scheme

Following the sorting of selected cells in descending order with respect to their connectivity with partial placement, the head of line cell is trial placed on all the available empty locations at that time and the membership of these locations in a fuzzy set of *good location* is computed. The fuzzy subset of *good location* falling within a *fuzzy window* is identified and it is called *favorable locations.* The cell will be randomly placed on any location within this subset. Following is the description of how these sets are formed and what rules govern them.

Good Location

Assume E is the set of empty locations and S is the set of selected cells. A location $l \in E$ will be a member in the fuzzy set *good location* for cell $c_i \in S$ with membership function $\mu^a_{c_i}(l)$. The determination of this membership function is carried out through the following rule and is evaluated using Equation 8 below:

Rule 1: **IF** a location results in
small length AND
reduced timing AND
small layout width
THEN it is a *good location.*

$$\mu^a_{c_i}(l) = \beta^a \times \min(\mu^a_1(l), \mu^a_2(l), \mu^a_3(l)) + (1-\beta^a) \times \frac{1}{3}\sum_{i=1}^{3}\mu^a_i(l) \quad (8)$$

where $\mu^a_{c_i}(l)$ is the fuzzy set of good locations and β^a is a constant parameter in the range [0,1]. The values $\mu^a_1(l)$, $\mu^a_2(l)$ and $\mu^a_3(l)$ represent the membership values of location l in fuzzy sets *small length, reduced timing* and *small layout width* respectively.

The base values $X_1(l)$, $X_2(l)$ and $X_3(l)$ for corresponding membership functions $\mu^a_1(l)$, $\mu^a_2(l)$ and $\mu^a_3(l)$, are computed below using the notation of Equations 5-7.

$$X_1(l) = \frac{\sum_{j=1}^{k} L^m_{v_j}}{\sum_{j=1}^{k} L^{m-1}_{v_j}} \quad (9)$$

$$X_2(l) = \frac{\sum_{j=1}^{k} D^m_{v_j}}{\sum_{j=1}^{k} D^{m-1}_{v_j}} \quad (10)$$

$$X_3(l) = \frac{W^m_r}{W_{opt}} \quad (11)$$

Following the computation of base values, memberships in respective fuzzy sets are determined using the functions given in Figure 2.

Favorable Locations

The subset of fuzzy set *good locations* is named as *favorable locations.* The upper and lower boundaries for *favorable locations* are determined by a *fuzzy window.* For a cell $c_i \in S$, location $l \in E$ will fall within *fuzzy window* if it satisfies the following inequality:

$$\max_{\forall e \in E}(\mu^a_{c_i}(e)) \times \left\{1.0 - w \times \frac{\#\text{ of unplaced cells}}{\#\text{ of selected cells}}\right\} \leq \mu^a_{c_i}(l) \leq \max_{\forall e \in E}(\mu^a_{c_i}(e)) \quad (12)$$

where E is the set of empty locations, S is the set of selected cells and w is a small positive value which determines the lower limit of $\mu^a_{c_i}(l)$ for presence in *fuzzy window.* It controls the randomness in the fuzzy allocation scheme. All locations falling in the *fuzzy window* will be identified as *favorable locations.* The cell will be placed randomly in any of the locations within this set. The presence of the ratio of number of unplaced cells to selected cells will make sure that the size of *favorable location* set will decrease as allocation progresses. This is because the head of line cell is strongly connected with the partial layout therefore there are many locations with near identical gains. We can place the cell on any of these locations without adversely affecting the quality of solution. Due to this, a bigger subset is identified for leading cells. However, the cells at the end of the selected queue are sparsely connected. Therefore a narrow *favorable location* set is used for them.

We experimented with different values for w in our proposed fuzzy allocation scheme. It is observed that for small values of w in the range $(0.05, 0.1)$ the quality of solutions improves. However, as the value of w is increased the quality decreases due to increased randomness.

3.4 Fuzzy Goal Based Cost Computation

VLSI placement is a multi-objective combinatorial optimization problem. A placement is evaluated against several objective criteria such as, wire length, delay and layout width. The best placement is the one which scores

	Fuzzy (Fa_SE)			Classical Sum (CSE)			% Gain		
Circuit	L (mic)	D (ns)	W (mic)	L (mic)	D (ns)	W (mic)	L	D	W
highway	7735	5.56	520	9919	6.15	520	22.0	9.5	0.0
fract	31528	13.62	784	37285	14.58	800	15.4	6.5	2.0
c499	56506	14.13	1200	59278	14.51	1200	4.6	2.6	0.0
c532	80779	37.96	1160	72789	38.55	1184	-10.9	1.5	2.0
c880	137309	29.46	1872	135509	30.92	1848	-1.32	4.7	-1.2
c1355	290221	27.05	2320	335589	28.41	2344	13.5	4.7	1.0
struct	667850	28.8	3336	685328	26.65	3312	2.5	-8.0	-0.7
c3540	750153	46.01	3152	844069	54.03	3152	11.1	14.8	0.0

Table 1: Best layout found by Fa_SE and CSE. "L", "D" and "W" respectively stand for wire length, delay and width costs of layouts.

lowest with respect to all objectives. A notion of optimality that respects the integrity of each of the separate criteria is the concept of Pareto optimality [16]. However, the Pareto optimality concept does not assist in making a *single* choice. One usually has to tradeoff these various objectives. In such a case, the concept of optimum is not clear. Traditional approach consists of combining all objectives in a weighted sum cost function, and the placement with lowest weighted sum is reported as the best solution [15]. This approach is at best controversial. Furthermore, the individual placement objectives are very imprecise. In this work we adopt a goal directed search approach, where the best placement is the one that satisfies as much as possible a user specified vector fuzzy goals.

Let there be Π solutions generated by the SE algorithm. Assume that we are optimizing a p-valued cost vector given by $C(x) = (C_1(x), C_2(x), ..., C_p(x))$ where $x \in \Pi$. Assume that a vector $O = (O_1, O_2, ..., O_p)$ gives lower bound estimates on individual objectives such that $O_i \leq C_i(x)$ $\forall i$, $\forall x \in \Pi$. These are lower bounds on each objective which usually are not achievable in practice. Further, assume that there is a user specified goal vector $G = (g_1, g_2, ..., g_p)$ which indicates the relative *acceptable limits* for each objective. It means that x will be an acceptable solution if $C_i(x) \leq g_i \times O_i$ where $\forall i, g_i \geq 1.0$. For a two dimensional optimization problem, Figure 3 shows the region of acceptable solutions.

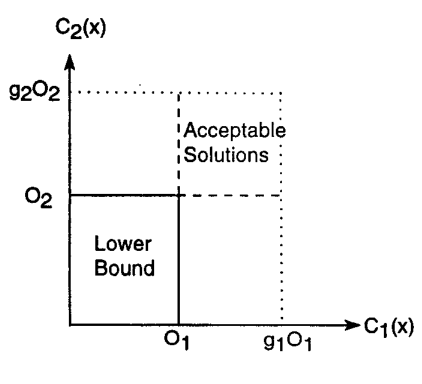


Figure 3: Range of acceptable solution set.

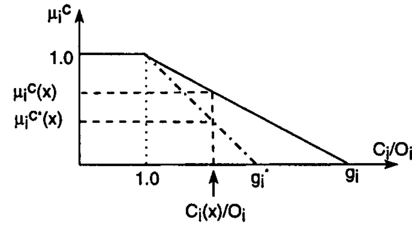


Figure 4: Membership function *within acceptable range.* By lowering the goal g_i to g_i^* the preference for objective "i" has been increased.

In our proposed scheme, the *acceptable solution* set is a fuzzy set. For VLSI cell placement problem of minimizing three parameters, we propose the following rule to determine the membership in fuzzy set *acceptable solution.* This rule is implemented by Equation 13.

Rule 2: **IF** a solution is
within acceptable wire length AND
within acceptable circuit delay AND
within acceptable width
THEN it is an *acceptable solution.*

$$\mu^c(x) = \beta^c \times min(\mu_1^c(x), \mu_2^c(x), \mu_3^c(x)) + (1 - \beta^c) \times \frac{1}{3}\sum_{i=1}^{3} \mu_i^c(x) \quad (13)$$

where $\mu^c(x)$ is the membership value for solution x in fuzzy set *acceptable solution.* While μ_i^c for $i = \{1, 2, 3\}$ represents the membership values of solution x in the fuzzy sets *within acceptable wire length, within acceptable circuit delay* and *within acceptable width* respectively. The solution which results in the maximum value for Equation 13 is reported as the best solution found by the SE algorithm. The membership function for a general objective "i" is shown in Figure 4. User preferences can be easily expressed in goal vector G. For example by decreasing the goal value g_i to g_i^* in Figure 4, the subsequent membership value $\mu_i^{c*}(x)$ for objective i will decrease. This might dictate the acceptance or rejection of solutions. In this work, the lower bounds on objectives are computed at initialization by the placement program.

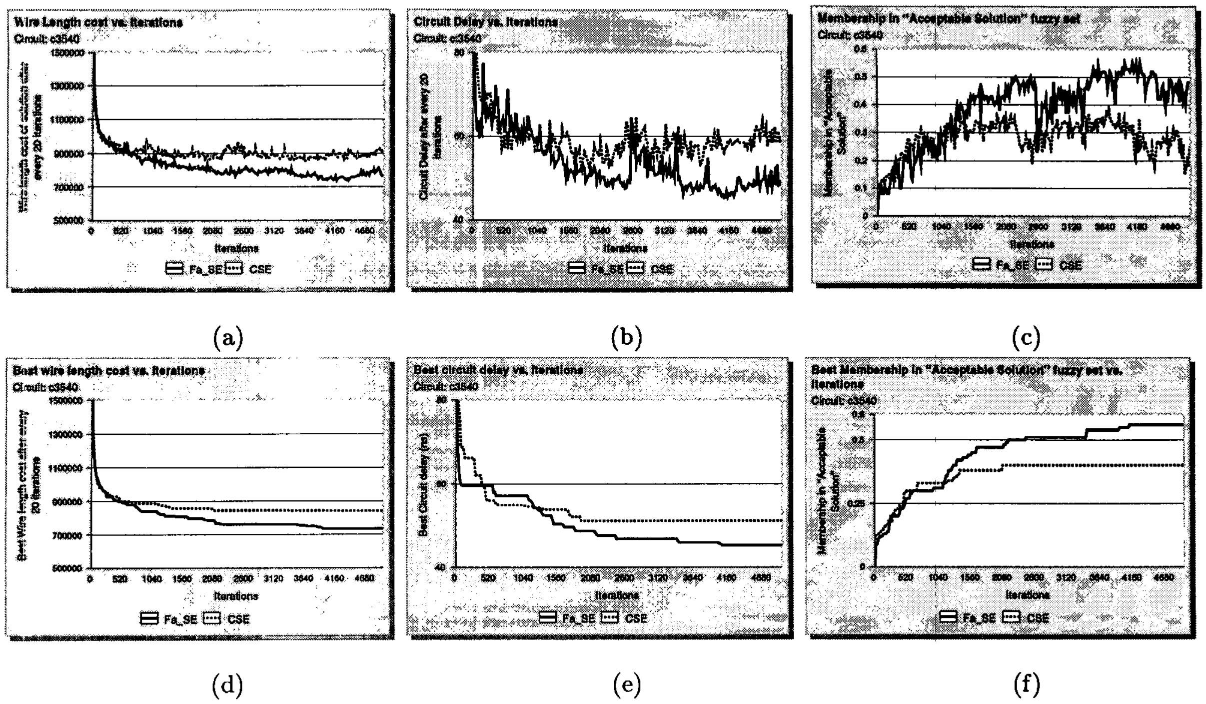


Figure 5: Comparison of Fa_SE algorithm with CSE. (a), (b) & (c) respectively compares the current wire length, circuit delay and membership in *acceptable solution*, (d), (e), & (f) plots the corresponding best values of these parameters.

4 Experiments and Results

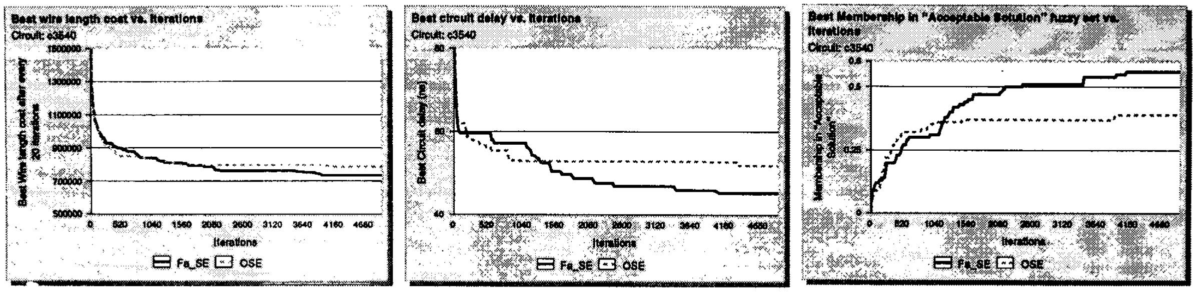
In order to compare the effects of fuzzification of the **allocation** stage of SE algorithm, we implemented two versions of the algorithm. One implementation, identified as Fa_SE, uses the proposed fuzzy allocation scheme. The second implementation is called classical SE (CSE) and it uses the weighted average allocation strategy. For weighted allocation, several experiments were carried out to find appropriate weight values for Equation 4. The weight combination ($w_1^a = 0.6, w_2^a = 0.1, w_3^a = 0.3$) resulted in the best solution. Both these algorithms use a uniform "fuzzy goal-based cost" measure to identify the best solution. These algorithms were tested on eight ISCAS-89 benchmark circuits. Initial solutions are randomly generated and algorithms are executed for a fixed number of iterations.

Table 1 compares the quality of final solution generated by Fa_SE and CSE. The circuits are listed in order of their complexity. From the results, it is clear that except for few cases (like **c532** and **c880**) the proposed fuzzy allocation scheme is able to generate better quality solution with respect to wire length. The wire length increase for **c532** and **c880** can be attributed to the fact that the weight combination in CSE allocation strategy is suitable for medium range circuits only. For smaller circuits (like **highway**) or bigger circuits (like **c3540**) these weight combinations do not perform well. Moreover for **c532** and **c880**, the reduction in circuit delay compensates the loss in wire length.

With respect to the circuit delay objective, the proposed scheme generates better quality solutions for all circuits. The **struct** circuit is an exception because it is a structured multiplier which contains all the paths of similar complexity, therefore reduction in the net delays of all nets forming those paths is required to achieve overall reduction in the circuit delay. The layout width of both schemes is comparable for all circuits. This is due to the fact that the widths of generated layouts are very close to the respective lower bound on maximum width.

In order to make a complete search space comparison between Fa_SE and CSE, we have drawn different cost values versus iteration count of the algorithm for circuit c3540, which has above 2000 cells. Figure 5(a) & (b) shows the current wire length and circuit delay cost versus iteration count for both schemes. Respective best values are shown in Figure 5(d) & (e) of the same figure. From these plots it is clear that Fa_SE is able to find solutions with less interconnect length and delay than CSE. Figure 5(c) & (f) compares the current and best overall quality of solution found by drawing the membership in fuzzy set *acceptable solution* respectively. From these plots it is clear that fuzzy allocation based SE algorithm is able to generate overall better quality solutions than weight based SE algorithm.

We also compared the original SE (OSE) algorithm which uses only wire length based sorted individual best fit allocation scheme [2] with our proposed Fuzzy SE algorithm (Fa_SE). For largest test circuit **c3540** results are summarized in Figure 6. As it is clear from Figure 6, Fa_SE produced noticeably better results than OSE. Similar results were observed with other circuits but are not included here due to lack of space.



(a) (b) (c)

Figure 6: Comparison of Fa_SE algorithm with OSE. (a),(b) & (c) respectively compares the best values for wire length, circuit delay and membership in *acceptable solution.*

5 Conclusion

In this paper, we have proposed a Fuzzy Simulated Evolution Algorithm. In the proposed scheme, we have fuzzified the **allocation** stage of the SE algorithm. This scheme combines controlled random move in a purely constructive sorted individual best fit allocation technique. The identification of favorable locations for a cell is carried out through the use of fuzzy rule and membership functions. Being the most important stage of SE algorithm, fuzzy allocation results in noticeable improvement in the quality of final solution.

VLSI placement is a hard and ill-defined problem with many conflicting objectives. In order to identify the best solution generated by the placement algorithm, we proposed a novel approach of *fuzzy goal-based cost measure.* This approach avoids the problems associated with the controversial weighted sum approach. It also allows easy incorporation of user preferences for different objectives.

Acknowledgments

Authors acknowledge King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, for all support.

References

[1] Sadiq M. Sait and Habib Youssef. *VLSI Physical Design Automation: Theory and Practice.* McGraw-Hill Book Company, Europe (also co-published by IEEE Press), 1995.

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

[3] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):498–516, May 1983.

[4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, INC., 1989.

[5] Sadiq M. Sait and Habib Youssef. *Iterative Computer Algorithms and their Application to Engineering.* IEEE Computer Society Press (to appear in 1999).

[6] Q. E. Kang, R. Lin, and E. Shragowitz. Fuzzy Logic Approach to VLSI Placement. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):489–501, Dec. 1994.

[7] M. Razaz. A Fuzzy C-Means Clustering Placement Algorithm. *IEEE Int. Symposium on Circuits and Systems, ISCAS*, 3:2051–2054, May 1993.

[8] C. F. Ball, P. V. Kraus, and D. A. Mlynski. Fuzzy partitioning applied to VLSI-floorplanning and placement. *IEEE Int. Symposium on Circuits and Systems, ISCAS*, 1:177–180, May-June 1994.

[9] T. Yukimatsu, T. Furuhashi, and Y. Uchikawa. A fuzzy expert system for hierarchical placement of parts on printed circuit board. In *Proc. of Second NZ Int. Conf. on Artificial Neural Networks and Expert System*, pages 342–345, Nov. 1995.

[10] Eugene Shragowitz, Habib Youssef, Sadiq M. Sait, and Hakim Adiche. Fuzzy Genetic Algorithm for Floorplanning. In *Proc. of SPIE*, pages 36–47, July 1997.

[11] Ralph M. Kling and Prithviraj Banerjee. Empirical and Theoretical Studies of the Simulated EvolutionMethod Applied to Standard Cell Placement. *IEEE Transactions on Computer-Aided Design*, 10(10):1303–1315, October 1991.

[12] L. A. Zadeh. Fuzzy Sets. *Information Contr.*, 8:338–353, 1965.

[13] Ronald R. Yager. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transaction on Systems, MAN, and Cybernetics*, 18(1):183–190, January 1988.

[14] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transaction Systems Man. Cybern.*, SMC-3(1):28–44, 1973.

[15] Sadiq M. Sait and Habib Youssef. Timing-influenced general-cell genetic floorplanner. *Microelectronics Journal*, 28(2):151–166, 1997.

[16] J. Horn, N. Nafpliotis, and D. E. Goldberg. Niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of 1st International Conf. on Evolutionary Computation*, pages 82–87, 1994.