Fuzzy Aggregating Functions for Multiobjective VLSI Placement

Junaid A. Khan **Sadiq M. Sait**
Computer Engineering Department
King Fahd University of Petroleum and Minerals
KFUPM Box # 673, Dhahran 31261, Saudi Arabia
e-mail: {jakhan,sadiq}@ccse.kfupm.edu.sa
Telephone: +966-3-860-6665, Fax: +966-3-860-3059.

Abstract

When fuzzy logic is used with multiobjective optimization, min/max operators may not be desirable. This is primarily due to lack of compensation/submission of min/max. To overcome this, Ordered Weighted Averaging(OWA) operators were proposed by Yager. OWA requires the selection of a control parameter β, which is different for different problem instances. In this work we propose new fuzzy aggregating functions that simulate the fuzzy AND/OR logic, and have the advantages of OWA without a need of any control parameter. Comparison with OWA for the VLSI cell placement using Simulated Evolution produced encouraging results.

1 Introduction

In a multiobjective optimization problem (MOP), balancing different objectives by weight functions is difficult, or at best controversial. Fuzzy logic is a convenient vehicle for solving this problem. It allows to map values of different criteria into linguistic values, which characterize the level of satisfaction of the designer with the numerical value of the objectives. All these numerical values operate over the interval [0,1] defined by the membership functions for each objective [1].

A fuzzy logic rule is an **If-Then** rule. The **If** part (*antecedent*) is a fuzzy predicate defined in terms of linguistic values and fuzzy operators (**Intersection** and **Union**). The **Then** part is called the *consequent*. There are many implementations of fuzzy union and fuzzy intersection operators. Fuzzy union 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, i.e., $\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$, and $\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))$. This is known as the $\min - \max$ logic initially introduced by Zadeh [2].

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 [3]. This operator falls in the category of compensatory fuzzy operators and allows easy adjustment of the degree of "anding" and "oring" embedded in the aggregation. According to [3], "orlike" and "andlike" OWA for two fuzzy sets A and B are implemented as given in Equations 1-2 respectively.

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

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

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. However, it is difficult to select a suitable value of β without any trial runs of an optimization algorithm for each problem instance, because a suitable value of β is different for each problem instance. In order to solve this problem, a set of aggregating functions (AND like and OR like) is presented in this paper. These aggregating functions do not need any user specified parameter and also provide the compensation/submission in a controlled manner.

2 Proposed Fuzzy Aggregating Functions

Two fuzzy aggregating functions, AND like fuzzy aggregation (AFA) and OR like fuzzy aggregation (OFA) are presented in this work details of these functions are given as follows.

2.1 And Like Fuzzy Aggregation (AFA)

This function does not receive, directly the membership values of fuzzy sets as parameters, it receives the membership values of complementary fuzzy sets.

Let μ, μ_1 and μ_2 be the membership values in fuzzy sets S, S_1 and S_2. The membership $\bar{\mu}$ in $\bar{S}$ (the complementary fuzzy set of S) is obtained by using fuzzy complementary operator.

Now the *And Like Fuzzy Aggregation* (AFA) is defined as follows,

$$\bar{\mu} = \bar{w}_1\ \bar{\mu}_1 + \bar{w}_2\ \bar{\mu}_2 \tag{3}$$

$$\mu = 1 - \bar{\mu} \tag{4}$$

where

$$\bar{w}_n = \frac{\bar{\mu}_n}{\bar{\mu}_1 + \bar{\mu}_2} \tag{5}$$

If the membership value μ_1 in one fuzzy set S_1 is lower than other, then corresponding membership $\bar{\mu}_1$ in complementary fuzzy set $\bar{S}_1$ is higher than the other, resulting in higher weight $\bar{w}_1$, leading to higher membership $\bar{\mu}$ in resulting complementary fuzzy set $\bar{S}$. It results in the lower membership μ in the resulting fuzzy set S. This behavior is analogous to t-norm where, if one membership is low, then the resulting membership is also low. If the membership values in all complementary fuzzy sets are equal then equal weights are assigned and the resulting membership is high. In short the AFA has following advantages.

1. It simulates the behavior of fuzzy AND logic (especially at the boundaries).
2. There is no need to adjust any parameter like β in OWA.
3. All the weights are controlled automatically.
4. It provides the compensation for any partial fulfillment.
5. It rejects the solutions having diverse membership values in different fuzzy sets, that can be accepted in the case of "pure anding" and "andlike OWA".

After combining Equations 3, 4 and 5 and generalizing the function to n fuzzy membership values to be ANDed, we can define the AFA function as follows,

$$\mu = 1 - \frac{\sum_{i=1}^{n} \bar{\mu}_i^2}{\sum_{i=1}^{n} \bar{\mu}_i} \tag{6}$$

2.2 Or Like Fuzzy Aggregation (OFA)

Or like fuzzy aggregation (OFA) is analogous to s-norm in behavior. Unlike AFO it receives directly the membership values. The function is defined as follows,

$$\mu = w_1\ \mu_1 + w_2\ \mu_2 \tag{7}$$

where

$$w_n = \frac{\mu_n}{\mu_1 + \mu_2} \tag{8}$$

If the membership in one fuzzy set is higher than the membership values in the other fuzzy sets then it will be given higher weight, hence the membership value μ in resulting fuzzy set S will be higher, that is analogous to s-norm. Unlike "pure oring" it also provides interaction from other membership functions having lower values.

After combining Equations 7 and 8 and generalizing the function to n fuzzy membership values to be ORed, we can define the OFA as follows,

$$\mu = \frac{\sum_{i=1}^{n} \mu_i^2}{\sum_{i=1}^{n} \mu_i} \tag{9}$$

Figure 1 shows the behavior of proposed fuzzy aggregating functions. Figure 1(a) shows the behavior of AFA, it can be seen that the functions operates as a min operator on the extremes and acts like a compensatory operator in the middle. Due to this fact, it is not possible to unintentionally optimize only a single objective (possible in OWA and not desirable), due to the compensation. It provides compensation in a controlled manner, when the membership values to be aggregated are near each other it behaves as a compensatory function, however if these are diverse, indicating optimization of a single objective, then it behaves as a pure min and force the optimization algorithm to optimize other objectives as well.

Figure 1(b) shows the behavior of OFA, it shows that the functions behaves as pure max in boundaries and also exhibits the effect due to submission of other membership values. However, it does not waste time in differentiating the degree of submission of a particular objective, because in OR logic if one objective is fulfilled then it is sufficient.

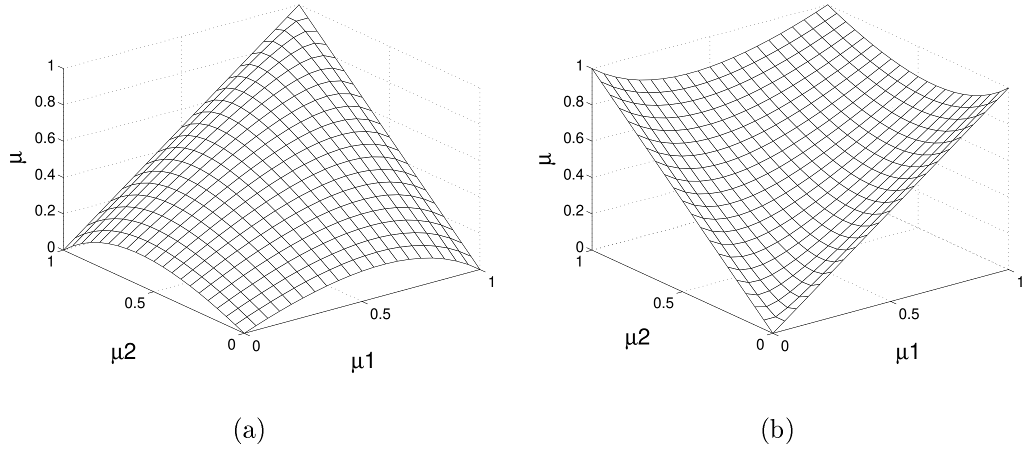

Figure 1: (a) And Like Fuzzy Aggregation , (b) Or Like Fuzzy Aggregation.

3 VLSI Cell Placement

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 [4].

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, (2) power dissipation minimization, and (3) circuit delay minimization with satisfying the layout width constraint. It is shown in [1, 5, 6] that cost due to these objectives can be computed as follows,

$$\text{Cost}_{wire} = \sum_{j \in M} l_j \tag{10}$$

$$\text{Cost}_{power} = \sum_{i \in M} S_i l_i \tag{11}$$

$$\text{Cost}_{delay} = T_{\pi_c} \tag{12}$$

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 [1, 5].

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

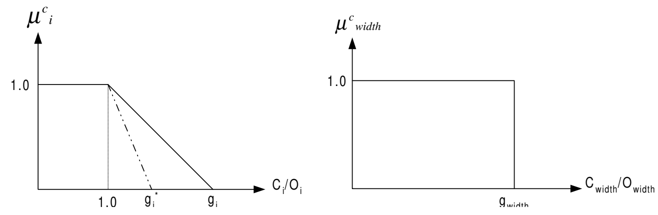

Figure 2: Membership functions within acceptable range.

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

Using And like fuzzy aggregation (AFA) the above fuzzy rule translates to the following:

$$\begin{aligned} \mu^c_{pdw}(x) &= 1 - \frac{\sum_{j=p,d,w} \bar{\mu^c}^2_j(x)}{\sum_{j=p,d,w} \bar{\mu^c}_j(x)} \\ \mu^c(x) &= \min(\mu^c_{pdw}(x), \mu^c_{width}(x)) \end{aligned} \tag{13}$$

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. 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 2(a), whereas the constraint *within acceptable layout width* is given as a crisp set as shown in Figure 2(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 Fuzzy Simulated Evolution for Placement

Simulated Evolution (SE) is a general, iterative metaheuristic to solve combinatorial optimization problems [1, 7, 8]. The general SE algorithm is illustrated in Figure 3 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 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 logic implemented as AFA, and OFA, rule **R2** evaluates to the expression below:

$$\text{goodness}_i = \mu_i^e(x) = 1 - \frac{\sum_{j=w,p,d} \bar{\mu^e}^2_{ij}(x)}{\sum_{j=w,p,d} \bar{\mu^e}_{ij}(x)} \qquad (14)$$

where

$$\mu_{id}^e(x) = \frac{\mu^{e2}_{inet}(x) + \mu^{e2}_{ipath}(x)}{\mu^e_{inet}(x) + \mu^e_{ipath}(x)} \qquad (15)$$

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 [1]. Membership functions of these base values are shown in Figure 4.

Selection: In this stage of the algorithm, some cells are selected probabilistically depending on their goodness values. A cell i is selected if $Random > goodness_i$. Where $Random$ is a Gaussian random number with $mean = G_m - G_\sigma$ and $standard\ deviation = G_\sigma$. G_m, and G_σ are the mean and standard deviation of goodness values of cells in the initial solution.

Allocation: In the allocation stage, the selected cells

```
ALGORITHM Simulated_Evolution(B, Φ_initial, StoppingCondition)
NOTATION
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 > g_i
                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 3: Structure of the simulated evolution algorithm.

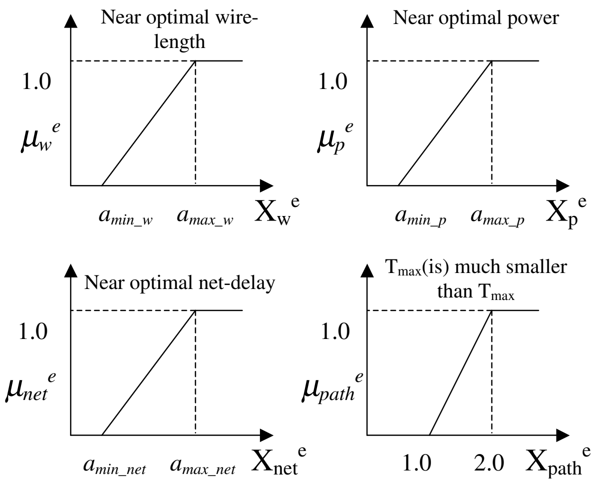

Figure 4: Membership functions used in fuzzy evaluation.

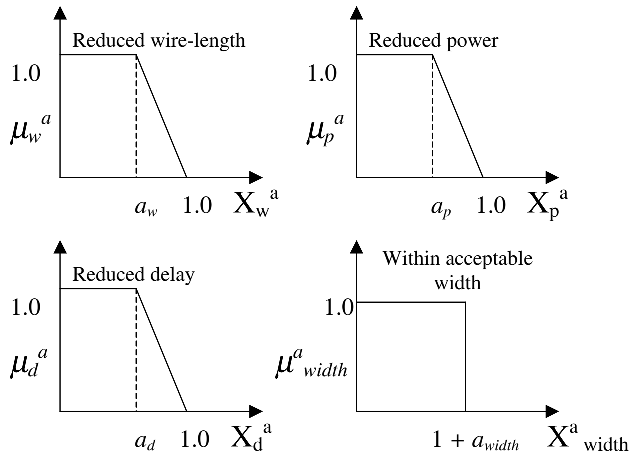

Figure 5: Membership functions used in allocation.

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:

$$\mu^a_{i\_wpd}(l) = 1 - \frac{\sum_{j=p,w,d} \bar{\mu^a}^2_{ij}(l)}{\sum_{j=p,w,d} \bar{\mu^a}_{ij}(l)} \qquad (16)$$

$$\mu^a_i(l) = \min(\ \mu^a_{i\_width}(l),\ \mu^a_{i\_wpd}(l)\ ) \qquad (17)$$

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 5. 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.

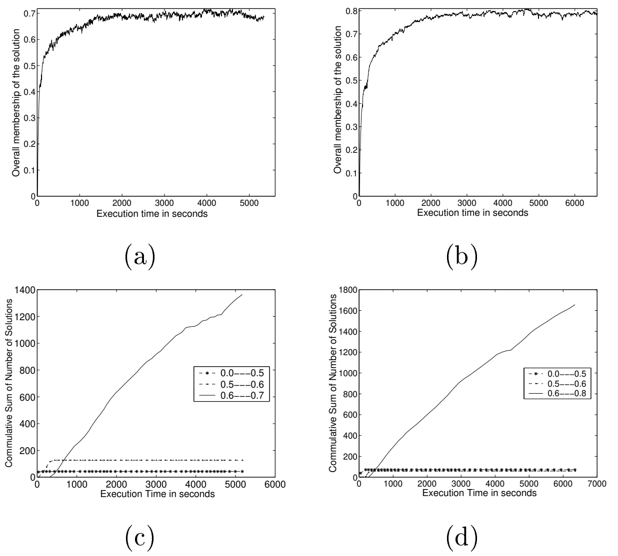

Figure 6: (a), and (b) show Membership values versus execution time for OFSE, and AFSE respectively. (c), and (d) show cumulative number of solutions visited in a specific membership range versus execution time for OFSE, and AFSE.

Circuit	AFSE				OFSE			
	L (μm)	P (μm)	D (ps)	T (s)	L (μm)	P (μm)	D (ps)	T(s)
S2081	2932	452	116	24	2740	422	114	126
S298	4853	925	139	82	4548	915	139	46
S386	7140	1653	202	153	8357	2036	203	117
S641	9445	2092	650	836	12811	3072	687	175
S832	19191	4359	356	293	23140	5251	416	192
S953	28290	4394	236	344	29526	5025	223	351
S1196	34331	10523	340	565	35810	11276	359	613
S1238	36333	11329	382	566	41318	12303	362	699
S1488	51793	12397	712	674	57730	13810	700	374
S1494	52711	12824	763	575	54523	12986	768	762
S3330	135650	17378	437	6619	183288	24797	460	5351
S5378	207252	29432	341	19159	326840	48360	435	11823
S9234	641670	101362	919	49479	857174	137712	923	42692

Table 1: Comparison between proposed aggregating functions and OWA. L is wirelength in μm, P is power cost in μm, D is delay in pico seconds, and T is the execution time in seconds.

5 Experiments and Results

Fuzzy Simulated Evolution using OWA (OFSE) and Fuzzy Simulated Evolution using proposed Fuzzy Aggregating Functions (AFSE) are applied on thirteen different ISCAS benchmark circuits.

Table 1 compares the quality of final solution generated by OFSE, and AFSE. The circuits are listed in order of their size (136- 5844 modules). It is clear that proposed aggregating functions (AFSE) has performed better than OWA operators (OFSE), except for two smaller circuits. In most of the cases AFSE is better in terms of all objectives, because of its better directed search capabilities in the solution space. However, in some cases, slight increase in the cost of one objective has resulted larger decrease in cost of other objectives (see S953 and S1488). In general, AFSE performs better than OFSE 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 OFSE and AFSE (Figure 6-(a) and (b)). These plots are for test case S3330. It can be observed that the quality of solution improves rapidly in AFSE based search as compared to OFSE. This behavior was observed for all test cases.

Figures 6(c), and (d) track with time the total number of solutions found by OFSE and AFSE, for various membership ranges. Note however that AFSE exhibited slightly faster evolutionary rate than OFSE. For example, after about 200 seconds, almost all new solutions discovered by AFSE 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 OFSE, it is after 300 seconds that the first solution with membership greater than 0.6 was found (see Figure 6). This behavior was observed for all test cases.

6 Conclusion

In this paper, we have proposed two fuzzy aggregating functions for multiobjective optimization problem. In this work these functions are applied on VLSI cell placement problem, however, these can be applied to any multiobjective optimization problem. These functions exhibits the good qualities of OWA operators and are more directed towards optimal solution than OWA without the need of any control parameter. Simulation results has shown the superiority of proposed aggregating functions over OWA operators.

References

[1] 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.

[2] 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.

[3] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision-

making. *IEEE Transaction on Systems, MAN, and Cybernetics*, 18(1), January 1988.

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

[5] 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.

[6] Sadiq M. Sait, Habib Youssef, Juaid A. Khan, and Aiman Al-Maleh. Fuzzified Iterative Algorithms for Performance Driven Low Power VLSI Placement. *International Conference on Computer Design, ICCD2001*, September 2001.

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

[8] 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.