2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Real-Time Power Aware Scheduling for Tasks with Type-2 Fuzzy Timing Constraints

Rahul Nath **Amit K. Shukla** **Pranab K. Muhuri**
Department of Computer Science
South Asian University, Akbar Bhavan, Chanakyapuri, New Delhi-110021, India
rahul.nath@outlook.com, amitkshukla@outlook.com, pranabmuhuri@cs.sau.ac.in

Abstract-- The timing constraint of tasks in the mobile real-time computing systems plays the central role in deciding the task schedule as timely completion of the task is very important in such systems. These timing constraints are however completely unquantifiable during the time of system modeling and designing. Thus we consider type-2 fuzzy sets for modeling the timing constraints in mobile and time-critical computing systems and propose a new algorithm FT2EDF (Fuzzy Type-2 Earliest Deadline First) for task scheduling. On the other hand, because of the limitation of the storage power, power efficiency is another foremost design objective for designing mobile real-time computing systems. However, reduction of processor power pulls down the system performance. Timely task completion and power efficiency are therefore two mutually conflicting criteria. In this paper, we propose a heuristic based solution approach that with a modified version of the non-dominated sorting genetic algorithm-II (NSGA-II). Our approach allows that a processor dynamically switches between different voltage levels to ensure optimum reduction in the power requirements without compromising the timeliness of the task completion. The efficacy of our approach is demonstrated with two numerical examples. Comparison with the previous results show that our solution ensures approximately 44% of energy saving as compared to the around 25% of the earlier results.

Keywords: *Task scheduling, Timing constraints, Type-2 Fuzzy Numbers, Deadlines, Processing Times, Power Efficiency, NSGA-II*

I. INTRODUCTION

Real-time computing systems are those where timely production of results is an essential necessity in addition to the logical correctness of the computations [1]. These kinds of systems are very often mobile and operated by the battery, which makes their uses controlled because of the limited power capacity of the storage cell. Task scheduling mechanism for these systems targets the feasible schedules that ensure the minimum consumption of the power as well as the timely task execution. Therefore, timing constraint of the tasks in the mobile and time-critical computing systems plays the central role in deciding the task schedule as timely completion of the task is very important in such systems. These timing constraints are however completely unquantifiable during the time of system modeling and designing. At this point of time, available information on these timing constraints are imprecise, incomplete and somehow non-obtainable [2]. As a result, those are all guessed by the designers when the systems are at the designing phase. This shows that these estimated timing parameters suffer from uncertainties. To model these uncertain timing parameters, type-I fuzzy numbers has been considered by a large section of researchers [3-9] over the last few decades. However, type-I fuzzy set based modeling has drawn severe criticisms because of the interpretability restrictions. This is because different designers may very naturally come up with different estimations as well as different membership grades for the same timing constraint. In [35], P. Melin et. al. applied type-2 fuzzy logic to the pattern recognition and classification problem. An optimization method has been proposed by D. Hidalgo et.al. in [36] for designing a fuzzy type-2 inference model based on the footprint of uncertainty. P. Melin et.al. [37] have proposed modifications to the original equations of Sugeno integrals using interval type-2 fuzzy logic and implemented the same in face recognition problem. The optimal granularity allocation approach is used to design the Interval type-2 fuzzy logic by O. Castillo in [38].

A very good survey on the researchers of the energy efficient algorithms may be found in [11]. Energy efficient task scheduling mechanisms has been extensively studied by a number of researchers [12-22]. K. Li. [23] addressed the problem of scheduling parallel tasks on multiprocessors system using the dynamically variable voltage and speed. W. Y. Lee [24] proposed few methods for energy optimization for the scheduling of periodic real-time tasks. A new approach for energy efficiency with optimum performance in heterogeneous clusters was proposed by X. Zhu et.al. [25]. S. Ehsan et. al. [26] provided a detailed survey on energy efficient routing techniques for wireless sensor networks. The energy-efficient data redistribution problem in data-intensive sensor networks has been addressed by B. Tang et. al. [27]. So far we know, no research has been reported till date that allows dynamic voltage switching of the processors for the power efficient execution of the tasks with type-2 fuzzy timing constraints. We have compared the results with the existing results, which show that our solution ensures approximately 44% of energy saving as compared to the around 25% of the earlier results.

Recently, type-2 fuzzy numbers were considered for modeling the timing constraints in mobile and time-critical computing systems [10]. The authors introduced a NSGA-II based solution technique for addressing the issue of energy-efficient scheduling in real-time embedded systems when deadlines and processing times are modeled with type-2 fuzzy uncertainty [34]. However the model is not so attractive as it does not allow switching of the processor voltage dynamically. In this paper, we have addressed this issue and improved the NSGA-II [32-33] based solution technique that allows a processor dynamic switching etween different voltage levels. Our new model ensures optimum reduction in the power requirements without compromising the timeliness of the task completion. We further propose a new algorithm, FT2EDF (Fuzzy Type-2 Earliest Deadline First), based on the traditional EDF scheduling policy. FT2EDF incorporates the type-2 fuzzy uncertainty in the timing constraints of real-time tasks and takes all interval trapezoidal type-2 fuzzy timing parameters, applies the rules and inference to get the type reduced fuzzy number which is considered as the earliness. The efficacy of our approach is demonstrated with two numerical examples and results are compared with the existing results. The rest of the paper is organized as follows. The mathematical background and the problem formulation is given in the Section II. Our modified NSGA-II based solving mechanism is elaborated in the Section III. Numerical examples are given the Section IV. We conclude the paper with a discussion and comparison of the results in the Section V.

II. Mathematical Background and Problem Formulation

We have considered Interval type-2 Fuzzy numbers to model the timing constraints of the tasks in mobile real-time systems. The notations used for our problem formulations are given in the Table 1. The mathematical details of the IT2FS may be found in [28-30]. However we are providing here a brief summary for ready reference.

TABLE 1. Notations for the problem formulations

Notations	Parameters
p_i	Processing Time of task T_i
d_i	Deadline of task T_i
C_i	Completion time of task T_i
E_i	Earliness of Task T_i
L_i	Lateness of Task T_i
l_i	Length of task T_i
v_j	Voltage levels (j=1, 2, 3,...)
X	It is a primary variable and its measurement domain is denoted by X
J_x	It is a primary membership degree of x
$\underline{\mu}_{\tilde{A}}(x)$	Lower membership function
$\overline{\mu}_{\tilde{A}}(x)$	Upper membership function
$\underline{f}^i$	Lower limit of firing-strength
$\overline{f}^i$	Upper limit of firing-strength
δ	Height of LMF in range [0.70,1]
S_{Ea}	Satisfaction Function for Earliness
S_{En}	Satisfaction Energy Efficiency

Type-2 fuzzy set: A type-2 fuzzy set $\tilde{A}$ is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, with $x \in X$ and $u \in J_x \subseteq [0,1]$, when the primary variable x is measured in the domain X, the secondary variable $u \subseteq J_x$ for each $x \in X$ and J_x is the primary membership degree of x. Mathematically,

$\tilde{A}=\{((x,u),\mu_{\tilde{A}}(x,u)) \mid \forall x \in X\ \forall u \in J_x \subseteq [0,1]\}, 0 \le \mu_{\tilde{A}}(x,u) \le 1.$

Interval Type-2 Fuzzy Set (IT2FS): When the secondary membership functions ($\mu_{\tilde{A}}(x,u)$) is 1 all over the primary domain, then the type-2 fuzzy set is known as an interval type-2 fuzzy set, $\tilde{A}_I$. Mathematically,

$$\tilde{A}_I = \{((x,u),1) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\}$$

Fig.1 shows a typical Trapezoidal Interval type-2 fuzzy membership function is shown represented in the Fig. 1 below:

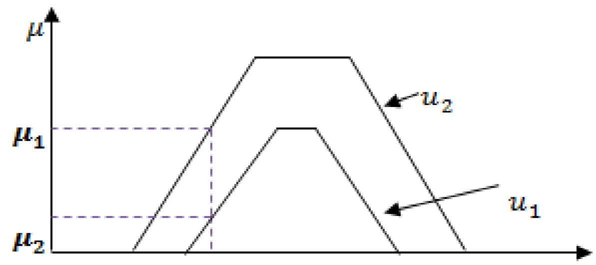

Fig. 1: Typical Trapezoidal Interval Type-2 Membership Function

As shown in the Fig. 1, the pictorial representation of Trapezoidal type-2 membership function, footprint of uncertainty of an interval type-2 Fuzzy set (IT2FS) is the union of all the embedded type-1 fuzzy sets. We have two terms here lower membership function (LMF) and the Upper membership function (UMF) which bounds FOU denoted by u_1 and u_2. We have used trapezoidal fuzzy set for both the LMF and UMF. If the representation of LMF(x) and UMF(x) are $(\underline{a}_L, \underline{a}, \underline{a}', \underline{a}_R)$ and $(\overline{a}_L, \overline{a}, \overline{a}', \overline{a}^R)$ respectively, then the membership function of IT2FLS can be represented as:

$$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} \frac{x-\underline{a}_L}{\underline{a}-\underline{a}_L} & \underline{a}_L \le x \le \underline{a} \\ \delta & \underline{a} \le x \le \underline{a}' \\ \frac{\underline{a}_R - x}{\underline{a}_R - \underline{a}'} & \underline{a}' \le x \le \underline{a}_R \end{cases} \quad (1)$$

$$\overline{\mu}_{\tilde{A}}(x) = \begin{cases} \frac{x-\overline{a}_L}{\overline{a}-\overline{a}_L} & \overline{a}_L \le x \le \overline{a} \\ 1 & \overline{a} \le x \le \overline{a}' \\ \frac{\overline{a}^R - x}{\overline{a}^R - \overline{a}'} & \overline{a}' \le x \le \overline{a}^R \end{cases} \quad (2)$$

In our problem we have considered the task timing constraints e.g. task deadlines (d_i) and processing times (p_i) for modeling as Trapezoidal type-2 fuzzy numbers. Therefore, the completion times (c_i), which are the deterministic sum of the processing times, shall also be trapezoidal. The structure of Interval type-2 fuzzy logic systems we have considered is shown in the Fig. 2. For any point on x-axis we will be having two corresponding membership value in the range $0 \le \mu_{\tilde{A}}(x) \le 1$, which can be represented as $\mu_{\tilde{A}}(x) = [\mu_2, \mu_1]$ as shown in the Fig. 1. A Mamdani IT2FLS with z inputs, $x_1 \in X_1$,, $x_2 \in X_2$,…, $x_n \in X_n$ and one output, $y \in Y$, the rule-base is composed of M rules, when the ith rule R^i will be like following:

R^i: IF x_1 is $\tilde{A}_1^i$ and x_2 is $\tilde{A}_2^i$… and x_n is $\tilde{A}_n^i$. THEN y is B^i

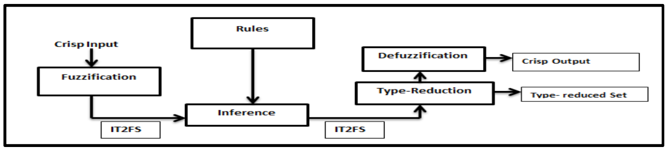

Fig.2: Interval Type-2 Fuzzy Logic System (IT2FLS)

The antecedent part of the inference rule is the linguistic represented by $\tilde{A}_k^i$ for k=1…n and the consequent part is represented by $B^i = [\underline{b}^n, \overline{b}^n]$ for i=1,2,…,M. The firing strength F^n(x) are defined as:

$$\underline{f}^i = \underline{\mu}_{\tilde{A}_1^i}(x_1) \sqcap \underline{\mu}_{\tilde{A}_2^i}(x_2) \sqcap \ldots \sqcap \underline{\mu}_{\tilde{A}_p^i}(x_p) = \mathcal{T}_{k=1}^{p} \underline{\mu}_{\tilde{A}_k^i}(x_k)$$

$$\overline{f}^i = \overline{\mu}_{\tilde{A}_1^i}(x_1) \sqcap \overline{\mu}_{\tilde{A}_2^i}(x_2) \sqcap \ldots \sqcap \overline{\mu}_{\tilde{A}_p^i}(x_p) = \mathcal{T}_{k=1}^{p} \overline{\mu}_{\tilde{A}_k^i}(x_k)$$

where $\mathcal{T}$ is the min or product t-norm. The fired rule output sets in an IT2 FLS are then combined by means of the Join operation to give an aggregated IT2 FS $\tilde{F}$. For our problem encoding we have obtained the type-reduced fuzzy sets using the very efficient method of KM algorithms [28], [29], [30] that calculates the centroid of the IT2FS, which is union of the centroids of all the embedded type-1 Fuzzy sets. In the K-M approach, the left point (y_l) and the right point (y_r) of the centroid of the IT2FS are calculated using the following Algorithms 1 and 2 [30] .

Algorithm 1: *Algorithm for Computing* y_l

a) Sort $\underline{b}^n$ (n=1, 2, 3,…, M) in increasing order and call the sorted $\underline{y}^n$ by the same name. Match the weights $F^n(x')$ with their respective $\underline{b}^n$ and renumber them so that their index corresponds to the renumbered $\underline{b}^n$.

b) Initialize f^n by setting $f^n = \frac{\underline{f}^n + \overline{f}^n}{2}$ n=1, 2, 3,…, M

c) and then compute $y = \frac{\sum_{n=1}^{N} \underline{b}^n f^n}{\sum_{n=1}^{N} f^n}$

d) Find switch point k (1 ≤ k ≤ M - 1) when $\underline{b}^k \leq y \leq \underline{b}^{k+1}$

e) Set $f^{n'} = \begin{cases} \overline{f}^n, & n \leq k \\ \underline{f}^n, & n > k \end{cases}$ and compute $y' = \frac{\sum_{n=1}^{N} \underline{b}^n f^{n'}}{\sum_{n=1}^{N} f^{n'}}$

f) Check if $y' = y$. If yes, stop and set $y_l = y$ and L = k. If no go to step f.

g) Set $y = y'$ and go to step c.

Algorithm 2: *Algorithm for Computing* y_r

a) Sort $\overline{b}^n$ (n=1, 2, 3,…, M) in increasing order and call the sorted $\underline{y}^n$ by the same name. Match the weights $F^n(x')$ with their respective $\overline{b}^n$ and renumber them so that their index corresponds to the renumbered $\underline{y}^n$.

b) Initialize f^n by setting $f^n = \frac{\underline{f}^n + \overline{f}^n}{2}$ n=1, 2, 3,…, N

and then compute $y = \frac{\sum_{n=1}^{N} \overline{b}^n f^n}{\sum_{n=1}^{N} f^n}$

c) Find switch point k (1 ≤ k ≤M - 1) such that $\overline{b}^k \leq y \leq \overline{b}^{k+1}$

d) Set $f^{n'} = \begin{cases} \underline{f}^n, & n \leq k \\ \overline{f}^n, & n > k \end{cases}$ and compute $y' = \frac{\sum_{n=1}^{N} \overline{b}^n f^{n'}}{\sum_{n=1}^{N} f^{n'}}$

e) Check if $y' = y$. If yes, stop and set $y_r = y$ and R = k. Otherwise, go to step f.

f) Set $y = y'$ and go to step c.

Once we get the type reduced fuzzy timing constraints viz. deadlines and completion times, we can calculate the fuzzy earliness, E_i . using the following Eq.

$$E_i = -L_i = C_i - d_i$$

where, L_i is the fuzzy lateness of the task T_i. The type reduced trapezoidal fuzzy earliness is expressed by $E_{ij} = E_{ij}(g_{ij}, g_{ij}^{/}, h_{ij}^{/}, h_{ij})$. With triangular membership function, $g_{ij}^{/} = h_{ij}^{/}$ and the fuzzy earliness shall be of the form as shown Fig.3. The function, satisfaction of schedulability, S_{Ea} , is introduced now to show how satisfied we are with the obtained task schedule and the calculated fuzzy earliness:

$$S_{Ea_i}(C_{ij}, d_{ij}) = \begin{cases} 1 & \text{if } g_{ij} \geq 0 \\ 1 - \dfrac{\int_{g_{ij}}^{0} \mu(E_{ij}) d(E_{ij})}{\int_{g_{ij}}^{h_{ij}} \mu(E_{ij}) d(E_{ij})} & \text{if } g_{ij} \leq 0 \leq h_{ij} \\ 0 & \text{if } h_{ij} \leq 0 \end{cases}$$

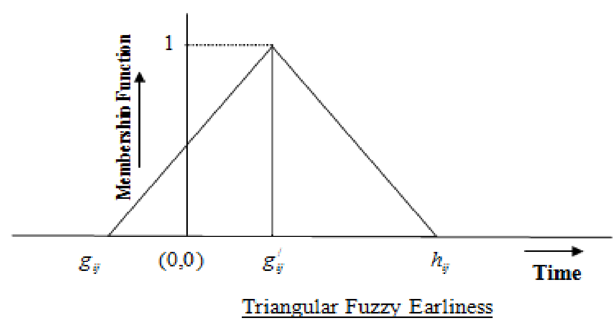

Fig.3: Typical Fuzzy Earliness

Another function, S_{En}, known as the satisfaction of energy efficiency is introduced now for ensuring that the generated task schedule that meets the deadlines at the expense of minimum amount of energy.

$$S_{En} = \frac{E_C - E_m}{E_M - E_m} \quad \text{for } E_m \leq E_C \leq E_M$$

The total energy required for the whole task set is $E_C = \sum_{i=1}^{n} E_{C_i}$ with $E_{C_i} = l_i * v_i^2$. If E_{m_i} and E_{M_i} are the energy needed for running the task T_i at low voltage and at high voltage mode respectively, then for the whole task set energy requirements are $E_m = \sum_{i=1}^{n} E_{m_i}$ and $E_M = \sum_{i=1}^{n} E_{M_i}$

Thus, we formulate problem as the following multi-objective optimization problem, with i =1, 2,…n, j = 1, 2,….P/P$_i$:

Find the optimal schedule subject to

(i) Min. S_{En}

(ii) Max. $S_{Ea} = \min_{i,j} S_{Ea_i}\left(C_{ij}, d_{ij}\right)$

III. FT2EDF Scheduling Algorithm in NSGA-II

We have solved this multi-objective scheduling problem using a modified NSGA-II (non-dominated sorting algorithm-II) algorithm [32-33]. The detailed working of the modified NSGA-II algorithm is given below and the flow chart is shown in Fig. 4

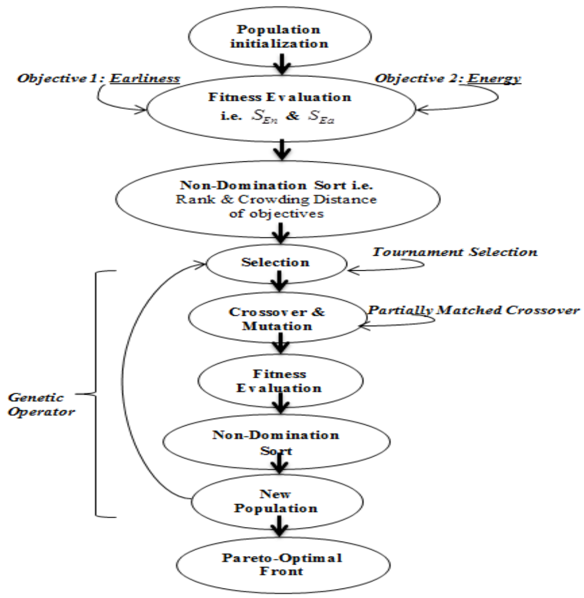

Fig. 4: Flow-Chart of Modified NSGA-II

Step 1: Population Initialization

We generate the large random schedule from which a solution space is chosen as an initial population.

Step 2: Fitness Evaluation: Assigning the fitness to initial population from objective function defined.

We calculate the fitness value to each individual in the solution space. The function S_{Ea} (as Earliness) S_{En} **(**as Energy**)** defined above are the evaluation criteria for this step. Each individual now have two fitness values.

Step 3: Non-domination sort (NSGA).

The outcomes of this step leave each individual with the rank and the crowding distance.

Step 4: Start the Evaluation process: Genetic Operator

a) Selection of the individuals from the initial population: Tournament Selection is used.
b) PMX Crossover and Mutation
c) Calculation of fitness value
d) Non-dominated Sorting.
e) Based on the rank and the crowded distance new population is selected.
f) Repeat Step: 4 for all the iterations

Step 5: Generated Pareto-optimal fronts are obtained

In step-2 for the calculation of the fitness functions we need to calculate the earliness and the energy consumption. The energy consumption for a task may be calculated by Eq. (17). The calculation of the earliness is not so much straightforward as the parameters are represented by trapezoidal type-2 fuzzy numbers. The objective to use the type-2 fuzzy numbers is to effectively model real-time timing constraints (by considering many embedded interval type-1 numbers) and improve the EDF. Therefore, the newly proposed algorithm, FT2EDF, is introduced; whose design is based on the traditional EDF scheduling policy so that it can incorporate type-2 fuzzy uncertainty in the timing constraints. FT2EDF takes all Interval Trapezoidal type-2 fuzzy timing parameters, applies the rules and inference to get the type reduced fuzzy number which is considered as the earliness. The new algorithm is given below:

Algorithm 3: FT2EDF (Fuzzy Type-2 Earliest Deadline First)
Input: Crisp Deadline's <$d_1, d_2, d_3 \dots, d_n$> of the task T_i's and procesings ion timeC_i's
Output: Defuzzified Crisp Deadline's <$d_1', d_2', d_3', \dots, d_n'$>
Method FT2EDF ()
{
Step 1: ∀ x∈X apply Fuzzifier (x) => Fuzzy Type-2 Input Sets (Ã)
Where $\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \square [0,1]\}$ and $\mu_A(x) \le \mu_{d_n}(x)$
Step 2: Apply Inference Rule (Ã) => Fuzzy Type-2 Output Sets ($\grave{\tilde{A}}$)
Step 3: Apply Type-Reduction ($\grave{\tilde{A}}$) => Type-Reduced Set (A), A is Type-1 Fuzzy Set
Where A= (a,c,b)
Step 4: Use Defuzzifier (A) => Crisp Output ($x' \in d_n'$)
Step 5: Sort x' to get Schedule S and compute the completion times
Step 6: Calculate Total Earliness $E_i = d_i - c_i$
Step 7: Rank (E_i) to check the optimality of the Earliness
}

IV. Simulation Experiments

A number simulation experiments are performed using licensed MATLAB software. Here we have included two numerical examples with 5 tasks [10] and 16 tasks [13]. Marvell PXA270 processor [31] with 5 voltage levels is taken as our model variable voltage processor. The characteristics processor details are mentioned in the Table 2.

Table 2. Voltage Levels and Frequency Levels

Voltage	Frequency Levels
0.90	104 MHz
1.15	208 MHz
1.25	312 MHz
1.35	416 MHz
1.45	520 MHz

Numerical Example 1:

First we consider a numerical example of 5 tasks with characteristics given in the Table 3. As shown, the task characteristics consists of deadlines and processing times (at different voltages) with trapezoidal membership function.

TABLE 3: TASKS CHARACTERISTICS (DEADLINES AND PROCESSING TIMES AT DIFFERENT VOLTAGE LEVELS)

Task	Parameter	Type Reduced Trapezoidal Fuzzy Number			
T_1	Deadline	4	7	7	9
	p_1 t 0.9V/104Mhz	6.5	7.69	7.69	8.2
	p_1 at 1.15V/208Mhz	2.5	3.84	3.84	4.2
	p_1 at 1.25V/312Mhz	2	2.56	2.56	2.90
	p_1 at 1.35V/416Mhz	1.4	1.92	1.92	2.5
	p_1 at 1.45V/520Mhz	0.9	1.53	1.53	1.9
T_2	Deadline	18	21	21	23
	p_2 at 0.9V/104Mhz	6.66	7.21	7.21	8.01
	p_2 at 1.15V/208Mhz	2.8	3.6	3.6	4.1
	p_2 at 1.25V/312Mhz	2.1	2.4	2.4	2.9
	p_2 at 1.35V/416Mhz	1.2	1.8	1.8	2.3
	p_2 at 1.45V/520Mhz	1	1.44	1.44	2
T_3	Deadline	3	5	5	7
	p_3 at 0.9V/104Mhz	14.5	15.38	15.38	16.42
	p_3 at 1.15V/208Mhz	6.8	7.69	7.69	8.20
	p_3 at 1.25V/312Mhz	4.6	5.1	5.1	5.6
	p_3 at 1.35V/416Mhz	3	3.84	3.84	4.5
	p_3 at 1.45V/520Mhz	2.8	3.07	3.07	3.5
T_4	Deadline	22	25	25	28
	p_4 at 0.9V/104Mhz	8.5	9.61	9.61	10.21
	p_4 at 1.15V/208Mhz	3.5	4.8	4.8	5.1
	p_4 at 1.25V/312Mhz	2.7	3.2	3.2	3.8
	p_4 at 1.35V/416Mhz	2.1	2.4	2.4	2.9
	p_4 at 1.45V/520Mhz	1.4	1.92	1.92	2.5
T_5	Deadline	14	16	16	19
	p_5 at 0.9V/104Mhz	5.01	5.76	5.76	6.5
	p_5 at 1.15V/208Mhz	2.1	2.88	2.88	3.5
	p_5 at 1.25V/312Mhz	1.4	1.92	1.92	2.5
	p_5 at 1.35V/416Mhz	1	1.44	1.44	2
	p_5 at 1.45V/520Mhz	0.8	1.15	1.15	1.5

TABLE 4: FITNESS EVALUATION VALUES (ENERGY & EARLINESS), PARETO FRONT AND CROWDING DISTANCE

Earliness	Energy	Pareto Front	Crowding
0	0.0885	1	Inf
0	0.0885	1	0
0	0.0885	1	0
0	0.0885	1	0
0.773	0.3361	1	0.791045
0.744	0.2782	1	1.360993
0	0.0885	1	0
0	0.0885	1	0
0	0.0885	1	0
0	0.0885	1	Inf

The two objectives viz. fuzzy earliness and energy efficiency with the resulted crowding distance and the Pareto-front is shown in the Table 4. The evaluation of the Pareto-fronts gives us two schedules with optimal results. Both the schedules and the voltages are shown in Fig. 6 and Fig. 7.

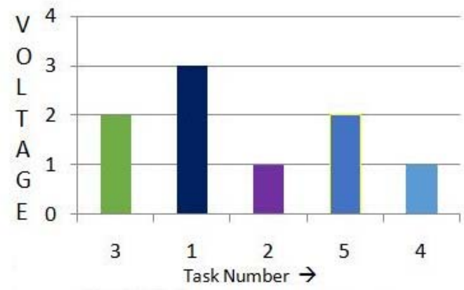

Fig. 6: Schedule 1 run on variable voltages

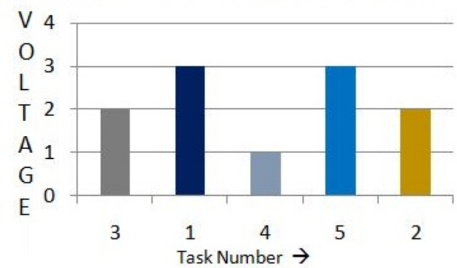

Fig. 7: Schedule 2 run on variable voltages.

Numerical Example 2

We now consider a popular real-life example with 16 tasks for demonstrating our type-2 fuzzy uncertainty and NSGA-II based energy optimization technique. Table 5 shows the task lengths and the deadlines after the type-reduction process. Table 6 to Table 10 give the type-reduced processing times when run at 104MHz, 208 MHz, 312MHz, 416Mhz and 520MHz corresponding to the supply voltages of 0.9V, 1.15V, 1.25V, 1.35V and 1.45V respectively.

TABLE 5: TYPE REDUCED TASK DEADLINES FOR 16 TASKS WITH TASK LENGTHS

T_i	Length	Deadline			
T_1	200	180	200	200	220
T_2	2560	29.7	33	33	36.3
T_3	1350	36	40	40	44
T_4	770	180	200	200	220
T_5	800	180	200	200	220
T_6	200	180	200	200	220
T_7	1750	4500	5000	5000	5500
T_8	5520	180	200	200	220
T_9	549	56.25	62.5	62.5	68.75
T_{10}	3975	112.5	125	125	137.5
T_{11}	1900	90	100	100	110
T_{12}	155	225	250	250	275
T_{13}	360	225	250	250	275
T_{14}	2792	180	200	200	220
T_{15}	2320	1800	2000	2000	2200
T_{16}	2320	1800	2000	2000	2200

TABLE 6: TYPE REDUCED TRAPEZOIDAL PROCESSING TIMES FOR THE 16-TASK SYSTEM WHEN RUN ON 104MHZ AT 0.9V.

T_i	li/104 at 0.9V			
T_1	1.730769	1.923077	1.923077	2.115385
T_2	22.15385	24.61538	24.61538	27.07692
T_3	11.68269	12.98077	12.98077	14.27885
T_4	6.663462	7.403846	7.403846	8.144231
T_5	6.923077	7.692308	7.692308	8.461538
T_6	1.730769	1.923077	1.923077	2.115385
T_7	15.14423	16.82692	16.82692	18.50962
T_8	47.76923	53.07692	53.07692	58.38462
T_9	4.750962	5.278846	5.278846	5.806731
T_{10}	34.39904	38.22115	38.22115	42.04327
T_{11}	16.44231	18.26923	18.26923	20.09615
T_{12}	1.341346	1.490385	1.490385	1.639423
T_{13}	3.115385	3.461538	3.461538	3.807692
T_{14}	24.16154	26.84615	26.84615	29.53077
T_{15}	20.07692	22.30769	22.30769	24.53846
T_{16}	20.07692	22.30769	22.30769	24.53846

TABLE-7: TYPE REDUCED TRAPEZOIDAL PROCESSING TIMES FOR THE 16-TASK SYSTEM WHEN RUN ON 208MHZ AT 1.15V

T_i	li/208 at 1.15V			
T_1	0.865385	0.961538	0.961538	1.057692
T_2	11.07692	12.30769	12.30769	13.53846
T_3	5.841346	6.490385	6.490385	7.139423
T_4	3.331731	3.701923	3.701923	4.072115
T_5	3.461538	3.846154	3.846154	4.230769
T_6	0.865385	0.961538	0.961538	1.057692
T_7	7.572115	8.413462	8.413462	9.254808
T_8	23.88462	26.53846	26.53846	29.19231
T_9	2.375481	2.639423	2.639423	2.903365
T_{10}	17.19952	19.11058	19.11058	21.02163
T_{11}	8.221154	9.134615	9.134615	10.04808
T_{12}	0.670673	0.745192	0.745192	0.819712
T_{13}	1.557692	1.730769	1.730769	1.903846
T_{14}	12.08077	13.42308	13.42308	14.76538
T_{15}	10.03846	11.15385	11.15385	12.26923
T_{16}	10.03846	11.15385	11.15385	12.26923

TABLE-8: TYPE REDUCED TRAPEZOIDAL PROCESSING TIMES FOR THE 16-TASK SYSTEM WHEN RUN ON 312MHZ AT 1.25V

T_i	li/312 at 1.25V			
T_1	0.576923	0.641026	0.641026	0.705128
T_2	7.384615	8.205128	8.205128	9.025641
T_3	3.894231	4.326923	4.326923	4.759615
T_4	2.221154	2.467949	2.467949	2.714744
T_5	2.307692	2.564103	2.564103	2.820513
T_6	0.576923	0.641026	0.641026	0.705128
T_7	5.048077	5.608974	5.608974	6.169872
T_8	15.92308	17.69231	17.69231	19.46154
T_9	1.583654	1.759615	1.759615	1.935577
T_{10}	11.46635	12.74038	12.74038	14.01442
T_{11}	5.480769	6.089744	6.089744	6.698718
T_{12}	0.447115	0.496795	0.496795	0.546474
T_{13}	1.038462	1.153846	1.153846	1.269231
T_{14}	8.053846	8.948718	8.948718	9.84359
T_{15}	6.692308	7.435897	7.435897	8.179487
T_{16}	6.692308	7.435897	7.435897	8.179487

TABLE-9: TYPE REDUCED TRAPEZOIDAL PROCESSING TIMES FOR THE 16-TASK SYSTEM WHEN RUN ON 416MHZ AT 1.35V

T_i	li/416 at 1.35V			
T_1	0.432692	0.480769	0.480769	0.528846
T_2	5.538462	6.153846	6.153846	6.769231
T_3	2.920673	3.245192	3.245192	3.569712
T_4	1.665865	1.850962	1.850962	2.036058
T_5	1.730769	1.923077	1.923077	2.115385
T_6	0.432692	0.480769	0.480769	0.528846
T_7	3.786058	4.206731	4.206731	4.627404
T_8	11.94231	13.26923	13.26923	14.59615
T_9	1.18774	1.319712	1.319712	1.451683
T_{10}	8.59976	9.555288	9.555288	10.51082
T_{11}	4.110577	4.567308	4.567308	5.024038
T_{12}	0.335337	0.372596	0.372596	0.409856
T_{13}	0.778846	0.865385	0.865385	0.951923
T_{14}	6.040385	6.711538	6.711538	7.382692
T_{15}	5.019231	5.576923	5.576923	6.134615
T_{16}	5.019231	5.576923	5.576923	6.134615

TABLE-10: TYPE REDUCED TRAPEZOIDAL PROCESSING TIMES FOR THE 16-TASK SYSTEM WHEN RUN ON 520MHZ AT 1.45V

T_i	li/520 at 1.45V			
T_1	0.346154	0.384615	0.384615	0.423077
T_2	4.430769	4.923077	4.923077	5.415385
T_3	2.336538	2.596154	2.596154	2.855769
T_4	1.332692	1.480769	1.480769	1.628846
T_5	1.384615	1.538462	1.538462	1.692308
T_6	0.346154	0.384615	0.384615	0.423077
T_7	3.028846	3.365385	3.365385	3.701923
T_8	9.553846	10.61538	10.61538	11.67692
T_9	0.950192	1.055769	1.055769	1.161346
T_{10}	6.879808	7.644231	7.644231	8.408654
T_{11}	3.288462	3.653846	3.653846	4.019231
T_{12}	0.268269	0.298077	0.298077	0.327885
T_{13}	0.623077	0.692308	0.692308	0.761538
T_{14}	4.832308	5.369231	5.369231	5.906154
T_{15}	4.015385	4.461538	4.461538	4.907692
T_{16}	4.015385	4.461538	4.461538	4.907692

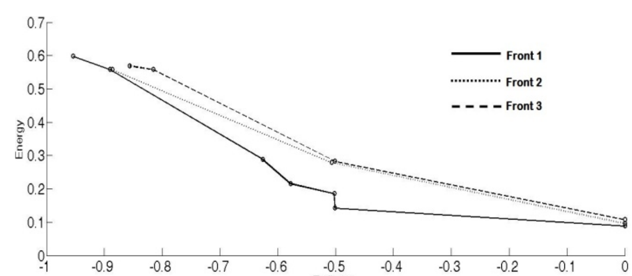

Fig. 7: Pareto-fronts obtained from the modified NSGA-II

Table 11 shows the two objective function values, earliness and energy, with the resulted crowding distance and the Pareto-fronts. Fig. 7 shows the respective Pareto-optimal fronts, which are drawn using the results shown in the Table 11. As it is visible from the Fig. 7, a number of schedules, all which give us the optimal earliness at optimized energy consumption. We have shown three such schedules with tasks running at variable voltages in the Fig. 9 to Fig. 11. As seen from these figures, all the three schedules are quite different from each other. The priority of a particular task in one schedule is quite different from its priority in another schedule. Further, a specific task is assigned different voltage levels in different schedules requiring different execution times. However, one have to choose any one of these schedule based on the favorable values for the conflicting objectives of energy and earliness.

TABLE 11: FITNESS VALUES (ENERGY & EARLINESS), PARETO FRONT AND CROWDING DISTANCE VALUES

Earliness	Energy	Pareto Front	Crowding
0	0.0885	1	Inf
-0.501	0.1421	1	0.717849
0	0.0885	1	0
0	0.0885	1	0
-0.954	0.5983	1	Inf
-0.502	0.1862	1	0.225475
-0.626	0.2888	1	0.999072
-0.89	0.5585	1	0.596117
-0.89	0.5585	1	0
-0.89	0.5585	1	0.354799
-0.578	0.2159	1	0.331234
0	0.0885	1	0
0	0.0885	1	Inf
0	0.0966	2	Inf
-0.507	0.2795	2	2
-0.886	0.5585	2	Inf
-0.815	0.5585	3	1.034572
-0.856	0.5693	3	Inf
-0.501	0.2833	3	1.928696
0	0.1079	3	Inf

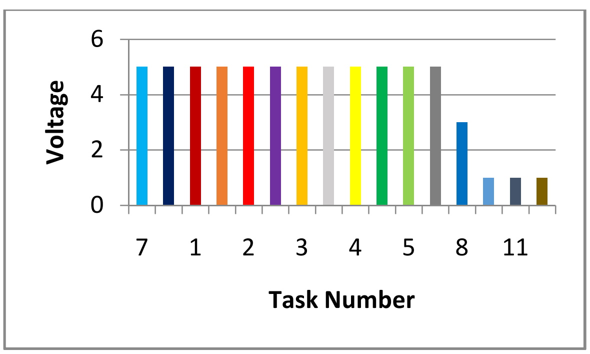

Fig. 8: Schedule 1 run on variable voltages

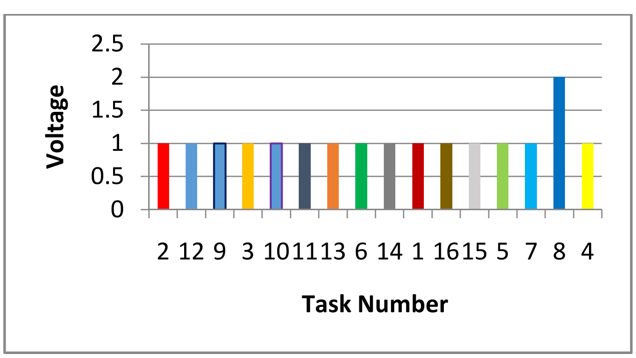

Fig. 9: Schedule 2 run on variable voltages

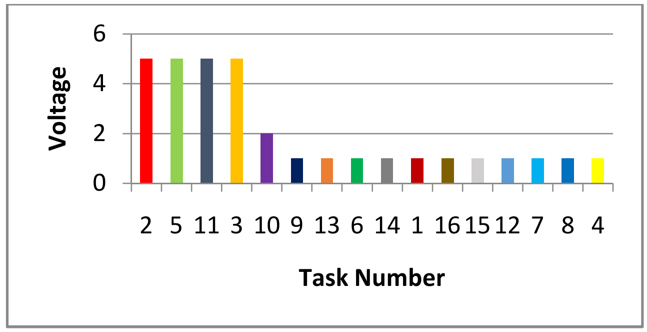

Fig. 10: Schedule 3 run on variable voltages

V. CONCLUSION

We have considered type-2 fuzzy sets for modeling the timing constraints in mobile and time-critical computing systems proposed a new algorithm, called FT2EDF (Fuzzy Type-2 Earliest Deadline First) scheduling algorithm. FT2EDF is developed with the modification of the traditional EDF scheduling policy by incorporating type-2 fuzzy uncertainty in the timing constraints. FT2EDF takes all Interval Trapezoidal type-2 fuzzy timing parameters, applies the rules and inference to get the type reduced fuzzy number which is considered as the earliness. On the other hand, uses of mobile and time-critical are highly restrictive because of the limited source of power storage. Due to which, power efficiency is another foremost design objective. Unfortunately, reduction of processor power pulls down the operating frequency of the processor i.e., power efficiency is achieved only at the cost of system performance. Timely task completion and power efficiency are therefore two mutually conflicting criteria.

TABLE 12 COMPARISON OF RESULTS WITH NUMERICAL EXAMPLE 1

Parameters	Dynamic voltage switching not allowed		Dynamic voltage switching allowed	
	Optimal Solution 1	Optimal Solution2	Optimal Solution1	Optimal Solution2
S_{Ea}	0.8756	0.7112	0.773	0.744
S_{En}	0.4805	0.4805	0.3361	0.2782
Energy Consumed	7423	7423	6105	5577
Voltage Level	{3 3 3 3 3}	{3 3 3 3 3}	{2 3 1 3 2}	{2 3 1 2 1}
Schedule	$[T_1T_3T_5T_2T_4]$	$[T_2T_3T_1T_4T_5]$	$[T_3T_1T_4T_5T_2]$	$[T_3T_1T_2T_5T_4]$
Energy Saving (%)	25.67	25.67	38.87	44.16

In this paper, we have proposed a heuristic based solution approach using a well-known evolutionary algorithm, NSGA-II (non-dominated sorting genetic algorithm-II). Our approach allows a processor dynamically switch between different voltage levels to ensure optimum reduction in the processor power requirements without compromising the timeliness of the task completion. The efficacy of our approach is demonstrated with numerical examples. In the Table 12, we have compared the results obtained from the numerical example 1 with the existing results [10]. The comparison shows that the proposed solution technique ensures approximately 44% of energy saving as compared to the around 25% of the earlier results.

REFERENCES

[1] H. Kopetz, “Real-time systems: design principles for distributed embedded applications”. Springer, 2011.

[2] I. N. Durbach and T. J. Stewart. "Modeling uncertainty in multi-criteria decision analysis" European Journal of Operational Research, Vol. 223, No. 1 (2012), pp. 1-14.

[3] F. Terrier and Z. Chen, “Fuzzy Calculus applied to Real-Time Scheduling”, In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on, (1994),* pp. 1905-1910. IEEE

[4] J. Lee, A. Tiao and J. Yen, “A Fuzzy Rule-Based Approach to Real-Time Scheduling”, In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on* (pp. 1394-1399). IEEE

[5] T. Murata, H. Ishibuchi and M. Gen, “Multi-Objective Fuzzy Scheduling with the OWA Operator for Handling Different Scheduling Criteria and Different Job Importance”, Proceedings of the IEEE International fuzzy Systems Conference (1999) (II) 773-778.

[6] M. Litoiu, R. Tadei, “Real-Time Task Scheduling with Fuzzy Deadlines And Processing Times”, Fuzzy Sets and Systems, Vol. 117, No. 1, (2001), pp. 35-45.

[7] Pranab K. Muhuri and K. K. Shukla, “Real-Time Task Scheduling with Fuzzy Uncertainty in Processing Times and Deadlines”,Applied Soft Computing, Vol. 8. No.1 (2008), 1-13.

[8] Pranab K. Muhuri and K. K. Shukla, “Real-Time Scheduling of Periodic Tasks with Processing Times and Deadlines as Parametric Fuzzy Numbers”, Journal of Applied Soft Computing, Vol. 9, No. 3 (2009), pp. 936-946.

[9] K. K. Shukla and Pranab K. Muhuri. Fuzzy Uncertainty Models and Algorithms For Real-Time Task Scheduling, Lambert Academic Publishing , 2011, ISBN978-3-8443-1647-6

[10] R. Nath, A. K. Shukla, P. K. Muhuri and Q M Danish LOhani, “NSGA-II based energy efficient scheduling in real-time embedded systems for tasks with deadlines and execution times as type-2 fuzzy numbers", Fuzzy Systems (FUZZ), 2013 IEEE International Conference on. IEEE, 2013.

[11] S. Albers, “Energy-Efficient Algorithms” Communications of the ACM, Vol. 53, No. 5, (2010), 86-96

[12] A. Sinha, A. P. Chandrasekasan, “ Energy Efficient Real-Time Scheduling”, Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design,(2001), pp. 458-462

[13] V. Swaminathon, K. Chakraborty, “Real-Time Task Scheduling for Energy-Aware Embedded Systems” Journal of the Franklin Institute, Vol. 338 (2001), pp. 729-750

[14] C. A. Rusu, R. Melhem, D. Mosse, “Maximizing Rewards for Real-Time Applications with Energy Constraints”, ACM Transactions on Embedded Computing Systems (TECS), Vol. 2, No. 4 (2003), pp. 537 - 559

[15] C. A. Rusu, R. Melhem, D. Mosse, “Maximizing the System Value while Satisfying Time and Energy Constraints”, IBM Journal of Research and Development, Vol 47 No. 5/6 (2003) pp. 689-702

[16] P. M. Alvarez, E. Levner, D Mosse, “Adaptive Scheduling Sever for Power-Aware Real-Time Tasks” ACM Tran. on Embedded Computing Systems, Vol. 3, No. 2, (2004), 284-306

[17] H. Wu, B Ravindran, E. D. Jensen, P Li, “Energy-Efficient, Utility Accrual Scheduling under Resource Constraints for Mobile Embedded Systems”, ACM Transactions on Embedded Computing Systems, Vol. 5. No. 3, (2006), pp. 513-542

[18] G. S. A. Kumar, G, Manimaran, Z. Wang, Ënergy-aware Scheduling of Real-Time Tasks in Wireless Networked Embedded Systems”, Proceedings of the 28th IEEE RTSS, (2007), pp. 15-24

[19] R. M. Santos, J. Santos and J. D. Orozco, “Power Saving and Fault-Tolerance in Real-Time Critical Embedded Systems”, Journal of Systems Architecture Vol. 55 (2009), pp. 90-101

[20] A Aminzadeh, A Ejlali, “A Comparative Study of Study of System-Level Energy Management Methods for Fault Tolerant Hard Real-Time Systems”, IEEE Transactions on Computers, Vol. 60, No. 9, (2011), 1288-1299

[21] J. Li, L. C. Shu , “Energy Efficient Scheduling in Non-Preemptive Systems with Real-Time Constraints” , IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43, No. 2, (2013), pp. 332-344.

[22] V. Swaminathan , K. Chakarbarty, “ Real-Time Task Scheduling for Energy-Aware Embedded Systems”, Journal of franklin institute 338 (2001) 729-750.

[23] K Li, “Energy efficient Sch. of parallel tasks on multiprocessor computers”, Journal *of Supercomputing*, Vol. 6*0, No.* 2, (2012), pp. 223-247.

[24] W. Y. Lee, “Energy-efficient scheduling of periodic real-time tasks on lightly loaded multi-core processors”. IEEE Tran. On Parallel and Distributed Systems, Vol. *23, No.* 3, (2012), pp. 530-537.

[25] X. Zhu, C. He, K. Li & X. Qin, “Adaptive energy-efficient scheduling for real-time tasks on DVS-enabled heterogeneous clusters” *Journal of parallel and distributed computing*,Vol. *72, No.* 6, (2012), pp. 751-763.

[26] S. Ehsan and B. Hamdaoui. "A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks", *IEEE Communications Surveys & Tutorials, Vol.* 14, No. 2, (2012): 265-278.

[27] B. Tang, N. Jaggi, H. Wu and R. Kurkal, "Energy-efficient data redistribution in sensor networks" ACM Transactions on Sensor Networks (TOSN), Vol. 9, No. 2, (2013): pp.11:1-28

[28] J. M. Mendel, R. John, “ Type-2 Fuzzy Sets Made Simple”, IEEE transactions on Fuzzy systems, Vol 10, No. 2, 2002.

[29] R. John, “Type-2 Fuzzy Sets: An Appraisal of theory and applications”, International Journal of Uncertainty, Fuzziness and Knowledge-based systems Vol. 6, No. 6 (1998) 563-576

[30] N. N. Karnik, J. M. Mendel, "Introduction to Type-2 Fuzzy logic Systems”, In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence,* Vol. 2, pp. 915-920.

[31] “Marvell PXA270 Processor Electrical, mechanical and Thermal Specification data Sheet”, 2009 http://www.marvell.com/application-processors/pxa-family/assets/pxa_27x_emts.pdf

[32] K.Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, IEEE Transactions on Evolutionary Computation, Vol. 6. No. 2 (2002) pp. 182-197

[33] K. Deb, “Multi-objective Optimization Using Evolutionary Algorithms”, Willey, 2001

[34] S. Wang and M. Mahfouf, "Multi-Objective Optimisation for Fuzzy Modelling Using Interval Type-2 Fuzzy Sets”, 2012 *IEEE World Congress on Computational Intelligence*, 2012

[35] Melin, Patricia, and Oscar Castillo. "A review on the applications of type-2 fuzzy logic in classification and pattern recognition." *Expert Systems with Applications, Vol.* 40, No. 13, (2013), pp. 5413-5423.

[36] Hidalgo, Denisse, Patricia Melin, and Oscar Castillo. "An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms." *Expert Systems with Applications, Vol.* 39, No. 4, (2012), pp. 4590-4598.

[37] Hidalgo, Denisse, Patricia Melin, and Oscar Castillo. "An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms." *Expert Systems with Applications, Vol.* 39, No. 4, (2012), pp. 4590-4598.

[38] Melin, Patricia, Olivia Mendoza, and Oscar Castillo. "Face recognition with an improved interval type-2 fuzzy logic sugeno integral and modular neural networks." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, Vol. 41, No. .5 (2011), pp. 1001-1012.

[39] Castillo, Oscar, Patricia Melin, and Witold Pedrycz. "Design of interval type-2 fuzzy models through optimal granularity allocation." *Applied Soft Computing*11.8 (2011): 5590-5601.