International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 492-501 June 2018-Dec 2018 www.csjournals.com

Time Optimization model with defuzzification for the task allocation in distributed computing system

Abhilasha Sharma
Mathematics Department, DIT, University,Dehradun, India
abhilasha.crown@gmail.com

Abstract: Distributed computing systems (DCS) are of current interest due to the advancement of microprocessors technology and computers networks. It consists of multiple computing nodes that communicate with each other by message passing mechanism. The advancement of the new technologies in communication and information lead to the development of the Distributed System. The task assignment is an essential phase in the Distributed computing systems. In this paper we have the concept of defuzzification by using Robust's ranking method to convert the fuzzy value into the crisp indices of time. A mathematical model has been developed to determine the optimal response time of the system by the optimal assignment of the tasks based on the speed of the processors with Triangular and Trapezoidal fuzzy execution time and Triangular and Trapezoidal fuzzy inter task communication time.
Keywords: Distributed Network, Task Allocation, defuzzification Fuzzy execution times, Fuzzy inter tasks, communication times, Crisp value.

I. Introduction

The fuzzy logic is developed by Professor L. A. Zadeh of the University of California at Berkeley in 1965 [1]. This development was not in use until Dr. E. H. Mamdani, who is a professor at London University, used the fuzzy logic in the implementation practically to control an automatic steam engine in 1974 [2], which was the ten years gap approximately the fuzzy theory was developed. After that, in 1976, Blue Circle Cement and SIRA in Denmark formed an industrial applications to control cement kilns [3].Fuzzy logic is the key for the powerful system to control the various problems; Kosko [4] has been developed some procedures to solve the problem which is difficult to model mathematically. Its technique provides the logical inference of rules in processing the linguistic variable data information to form crisp or numeric values. Fuzzy logic has the various techniques in the formation of controllers, operations research, expert systems and help in various decision making processes. Parallel and distributed computing has its applications in a wide area of real- time engineering problems. One particular instance is scheduling the robot inverse dynamics computation through mapping on a multiprocessor system, Lee & Chen [5]. Distributed ComputingSystem (DCS) is a system in which various processors joined together through a communication connection. This connection plays the role of bridge to access data and programs on remote processors. A definition of distributed computing is presented by [6,7] that " The Multiple Computers utilized cooperatively to solve problems i.e. to process and maintained the large scale database of the programs which are to be executed on these type of computing environment". The allocation of tasks to the processors is an importantphase to make the better system of a DCS and may be done in a variety of ways (i) Static Allocation and (ii) Dynamic Allocation. In static allocation, the allocations of the tasks to the processors have on change while the characteristic of the computation change and a new allocations must be computed. These problems may be categorized in static [8 -15].Nagarajan et al. [16] in 2010, has formed a mathematical model using Robust's ranking method. The cost has been taken trapezoidal and triangular values.The paper [17] deals an optimal solution to allocate the tasks in heuristic model. A membership function of fuzzy is formed for making the bundles of tasks with maximize the throughput and minimize execution time of the system. A tasks allocation model is constructed by [18] for the reliability and cost in DCS. In [19, 20] introduced a fuzzy approach for the multiprocessor real-time scheduling variables are treated as fuzzyvariables.The present model has the mathematical approach for assigning a s "m'' tasks to the "n" processors applying fuzzy execution times and fuzzy inter tasks communication times with the purpose of minimizing the total response time of the system.

II. Definitions

a) Fuzzy Execution Time

The fuzzy execution time $et_{i,j}$is the time to represent the performanceof the execution of the task $t_i$on the processor $p_j$from the set of tasks allocation N, the fuzzy execution time and processor's fuzzy execution time for each processors are obtained by equations (1) and (2) as follows:

$$FUET(f) = \sum_{1 \le i \le n} et_{i,f(i)} \quad (1)$$

$$PFUET(f) = \sum_{1 \le i \le n} et_{i,f(i)} \ , i \in TAS_j \ (2)$$

Where $TAS_j = \{i : f(i) = j, j = 1,2.....m\}$.

b) Fuzzy Inter Task Communication Time

The communication between the tasks t_i and $t_j$is represented by The fuzzy inter task communication time $ct_{i,j}$and these tasks on the various processors at the time of execution for an allocation,the fuzzy inter task communication time and processor's fuzzy inter task communication time for each processors are obtained by equations (3) and (4) as follows:

$$FUITCT(f) = \sum_{\substack{1 \le i \le n \\ i+1 \le j \le n \\ f(i) \ne f(j)}} ct_{f(i),f(j)} \quad (3)$$

$$PFUITCT(f)_j = \sum_{\substack{1 \le i \le n \\ i+1 \le j \le n \\ f(i) \ne f(j)}} ct_{f(i),f(j)} \quad (4)$$

c) Response Time of the System

The response time is defined as the maximum computation by the processor and the communication weight of the processor and it is performed by each processor's activity in the execution process .The fuzzy response time of the system is as follows:

$$FURT(f) = \max_{1 \le j \le m} \{PFUET(f)_j + PFUITCT(f)_j\} \quad (5)$$

III. Mathematical Model

The mapping between tasks and processors is defined by $f : N \to M$ here N represents set of numbers of tasks and M represents set of numbers of processors. $N(i) = j$ The task t_i is assigned to processor $p_j \ 1 \le i \le n, 1 \le j \le m$ each processor catch the data from the local memory only not global memory. The fuzzy execution times (FUET) et_{ij} in the form of matrix called as fuzzy execution time matrix (FUETM), $FUETM = [et_{ij}]$ of order $m \times n$. The fuzzy inter-tasks communication times (FUITCT), ec_{ij} in the form of a symmetric matrix called as fuzzy inter task communication time matrix (FUITCTM), $FUITCTM = [ct_{ij}]$ of order m. In this article et_{ij} and ct_{ij} time is taken as the triangular and trapezoidal numbers. The purpose of the problem is to allocate the n tasks for a set $N = \{t_1, t_2.....t_n\}$ to a set $M = \{p_1, p_2.....p_m\}$ of m processors for optimize the system.

- **Assumptions**

To design the algorithm the following assumptions is as follows

(a) The system has the assumption of "m" tasks execution on the "n" different speeds processors. A task may be a code or a data file.

(b)The allocation of the tasks is more than the processors as real in DCS.The program assumed that the fuzzy execution time of each task on each processor is known.

(c)After execution of the task the storage of the output data in the local memory of the processor before execution of the next task on the same processor it read from the local memory.

(d)The inter task communication time between the same tasks is zero. it is assumed because the allocation of the tasks on the processors should be according the communication of the tasks.

- **Average Load**

Compute the average load which is assigned to each processor p_j by the equation 6 and total load to be assigned on the system by equation 7 using ECM (,).

$$L_{avg}(p_j) = \frac{W_j}{m}, j = 1,2.....m, W_j = \sum_{\substack{1 \le i \le n \\ 1 \le j \le m}} et_{ij} \quad (6)$$

$$\text{Total Load [TL]} = \sum_{1}^{m} L_{avg}(p_j) \quad (7)$$

- **Defuzzification**

Defuzzification or Defuzzified is the conversion of the input times et_{ij} and ct_{ij} into crisp values. In this article we are applying Robust's ranking procedure for the defuzzification of the fuzzy time. If (g_α^L, g_α^U) is α - cut for triangular and trapezoidal fuzzy times (either et_{ij} or ct_{ij}) then its corresponding defuzzified Crisps value is obtained as follows.

$$et_{i,j} = RR(et_{i,j}) = \frac{1}{2}\int_0^1 (\, g_\alpha^L, g_\alpha^U) d\alpha \qquad (8)$$

$$ct_{i,j} = RR(ct_{i,j}) = \frac{1}{2}\int_0^1 (\, g_\alpha^L, g_\alpha^U) d\alpha \qquad (9)$$

Defuzzified fuzzy execution times et_{ij} and fuzzy inter tasks communication times ec_{ij} are stored in the form of matrices $[et_{ij}]$ and $[ec_{ij}]$ respectively.

The Robust's ranking methodtransforms the $FUETM$ into a crisp value problem.

The membership function of the triangular fuzzy number et_{ij} (a,b,c) is:

$$\mu(x) = \begin{cases} \dfrac{x-a}{b-a} & , a \le x \le b \\ 1 & , x = b \\ \dfrac{c-x}{c-b} & , c \le x \le d \end{cases}$$

The membership function of the trapezoidal fuzzy number et_{ij}, (a,b,c,d) is:

$$\mu(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \end{cases}$$

The α-cut of the fuzzy triangular number (a,b,c) is obtained by the following expression

$$(g_\alpha^L, g_\alpha^U) = [a + (b-a)\alpha, c - (c-b)\alpha]$$

$$\text{Crisp value} = \frac{1}{2}\int_0^1 [a + (b-a)\alpha + c - (c-b)\alpha] d\alpha .$$

The α-cut of the fuzzy Trapezoidal number (a,b,c,d) is obtained by the following expression

$$(g_\alpha^L, g_\alpha^U) = [a + (b-a)\alpha, d - (d-c)\alpha]$$

$$\text{Crisp value} = \frac{1}{2}\int_0^1 [a + (b-a)\alpha + d - (d-c)\alpha] d\alpha$$

- **Processors Bundling**

To optimize the response time of the distributed computing system we are applying the processors bundling on the basis of the membership function by taking the average speed of the processors. The membership value of the average speed of the processor obtained by defuzzified execution time et_{ij} of the tasks on each processor, The membership function is as follows

$$\mu_P = \frac{et_{ij}}{et} \qquad (10)$$

Where $et = \max\{et_{ij}; i = 1,2.....n, j = 1,2.....m\}$.

We are choosing four types of linguistic variables {Very Slow Average Speed Processor [VSASP], Slow AverageSpeed Processor[SASP], HighAverage Speed Processor [HASP], and Very HighAverage Speed Processor [VHASP], to grade the high and low average speed processors are graded into four categories on the basis of their membership values as shown in the following table

Table 1. Processor ranking

Membership Values	Linguistic Variables
[0.9, 1]	Very Slow Average Speed Processor [VSASP]
[0.7, 0.9)	Slow AverageSpeed Processor[SASP]
[0.5, 0.7)	HighAverage Speed Processor [HASP]
[0.0, 0.5)	Very HighAverage Speed Processor [VHASP]

- **Tasks Bundling**

Tasks are bundled based on their communication time. The number of tasks are heavily communicating are bundled together to reduce communication time and applied the same fuzzy membership function to categories the task in five categories. The membership values are obtained by using the defuzzified crisp indices ct_{ij} as follows

$$\mu_T = \frac{ct_{ij}}{ct} \quad (11)$$

Where $ct = \max\{ct_{ij}; i = 1,2.....n, j = 1,2.....m\}$

We are selecting five types of linguistic variables {Very Low Communicating Task [VLCT], Low Communicating Task [LCT], Average Communicating Task [ACT], Highly Communicating Task [HCT], Highly Communicating Task [HCT], Very Highly Communicating Task [VHCT]} to grade the high and low communicating pairs of tasks. Tasks pairs are graded into five categories on the basis of their membership values as shown in

Table 2. Tasks Ranking

Membership Values	Linguistic Variables
[0, 0.2)	Very Low Communicating Tasks [VLCT]
[0.2, 0.4)	Low Communicating Tasks [LCT]
[0.4, .6)	Average Communicating Tasks [ACT]
[0.6, 0.8)	Highly Communicating Tasks [HCT]
[0.8, 1]	Very Highly Communicating Tasks[VHCT]

To make proper correspondence between tasks bundles and processors the number of tasks bundles should be equal to the processors. These bundles will be permanently bounded throughout their execution.. If the new bundle, producing from combining two bundles becomes too large, it would not be possible to obtain a load-balancing scheduling properly. We are making control check on the maximum number of tasks in a bundle by $t_c = \frac{n}{m}$ for defining the size of the producing bundles.

- **The allocation algorithm**

The mapping between the tasks bundles to the processors bundles takes place according to the following steps of the algorithm

❖ AVERAGE_LOAD()://*to be allocated to the processors and System*
Calculate the average load on the processors p_j and the total load by using the equation (6) and (7) with tolerance factor.

❖ PROCESSOR_BUNDLE ():// *form FUET the bundles of processors.*

(a) Determine the defuzzified crisps value from equation (8).
(b)Estimate thedefuzzified membership value for all the processors in FUETM from the equation (10).
(c) Determine the average speed of the processor from defuzzified membership value execution time matrix.
(d) Categories the processor on the basis of the given order of the average speed membership values of the processor.

VHASP→ HASP→ SASP → VSASP.

❖ TASK_BUNDLE ()://*form FUITCT the bundles of tasks.*

(a) Determine the defuzzified crisps value from equation (9).
(b)Estimate thedefuzzified membership value for all the tasks in FUITCTM from equation (11).
(c) Categories bundle of tasks on the basis of the given order of the communication time membership values of the tasks.

$$VHCT \rightarrow HCT \rightarrow ACT \rightarrow LCT \rightarrow VLCT.$$

TASK_MAPPING():// *Map the task bundles to the processors bundles*

(a) Consider each individual task t_i as a individual bundle as

Bundle T (i) $\leftarrow \{ t_i \}, \ i = 1,2,.....n$.

(b) Ifnumber of tasks bundles are not equal to number of processors then

(i) Select the pair of tasks bundles (say BundleT (a) and BundleT (b)) with higher priorities starting from VHCT and assignto the categorized order of the processors starting from HASP.

Store the assigned tasks in a linear array $T_{as}(i)$ *and non-assigned tasks in a linear array* $T_{nas}(i)$.

(ii) Calculate NT(number of tasks){a , b} = $no.of tasksin$*BundleT (a) +* $no.of tasksin$*BundleT (b).*

(iii) If NT {a, b} $\leq t_c$ *then*

(iv) Fuse tasks bundles Bundle T (a) with Bundle T (b)
i.e. BundleT (a) ← BundleT (a) U Bundle T (b).
(v) Delete the tasks bundle T (b).
(c)Select the next bundle of tasks pair with next higher priorities for their fusion and assign the tasks to the next higher priorities of the processors (HASP).

❖ $Calculate\, FUET(f), PFUET(f)_j, FUITCT(f), FUITCT(f)_j \ and\, FURT(f).$

End.

IV . ILLUSTRATED EXAMPLE

- ***Triangular Fuzzy Time***

Let a set N= {t_1, t_2, t_3, t_4, t_5} of "n=5" tasks and a set M= {p_1, p_2, p_3} of "m=3" processors. The execution time matrix $FUETM = [et_{ij}]$ *of order m x n with fuzzy triangular numbers as given in Table 1. Inter tasks communication time matrix* $FUITCTM = [ec_{ij}]$ *of order m with fuzzy triangular numbers as given in Table 2.*

Table 1: Fuzzy Execution Time Matrix

	p_1	p_2	p_3
t_1	(5,10,20)	(5,10,15)	(10,15,20)
t_2	(10,15,20)	(10,20,30)	(10,15,25)
t_3	(10,20,30)	(10,15,25)	(10,15,20)
t_4	(5,10,20)	(10,15,20)	(5,10,15)
t_5	(5,10,15)	(5,10,20)	(5,15,20)

Table 2: Fuzzy Inter Task Communication Time Matrix

	t_1	t_2	t_3	t_4	t_5
t_1	(0,0,0)	(20,30,40)	(10,20,30)	(40,45.50)	(5,10,20)
t_2	(20,30,40)	(0,0,0)	(40,50,60)	(10,20,30)	(30,40,50)
t_3	(10,20,30)	(40,50,60)	(0,0,0)	(10,15,25)	(10,20,30)
t_4	(40,45,50)	(10,20,30)	(10,15,25)	(0,0,0)	(15,25,30)
t_5	(5,10,20)	(30,40,50)	(10,20,30)	(15,25,30)	(0,0, 0)

The crisp indices for the fuzzy execution times et_{ij} and the fuzzy inter tasks communication times ct_{ij} are obtained by (8) and (9)shown in the table 3 and 4.

Table3: Crisp indices et_{ij} for the fuzzy execution times et_{ij}

	p_1	p_2	p_3
t_1	11.25	10.00	15.00
t_2	15.00	20.00	16.25
t_3	20.00	16.25	15.00
t_4	11.25	15.00	10.00
t_5	10.00	11.25	13.75

Table 4: Crisp indices ct_{ij} for the fuzzy inter tasks communication times ct_{ij}

	t_1	t_2	t_3	t_4	t_5
t_1	0.00	30.00	20.00	45.00	11.25
t_2	30.00	0.00	50.00	20.00	40.00
t_3	20.00	50.00	0.00	16.25	20.00
t_4	45.00	20.00	16.25	0.00	23.75
t_5	11.25	40.00	20.00	23.75	0.00

Table 5:average load assigned to the processor

Load	p_1	p_2	p_3	Total Load
Average Load	23	24	23	70
Tolerance 10%	25	26	25	76
Total	48	50	48	146

Table 6:*Defuzzified membership value* μ_P *of FUET*

	p_1	p_2	p_3
t_1	0.56	0.50	0.75
t_2	0.75	1.00	0.81
t_3	1.00	0.81	0.75
t_4	0.56	0.75	0.50
t_5	0.50	0.56	0.69
Average Speed of Processor	.67 HASP	.72 SASP	.70 SASP

Table 7:*Defuzzified membership value* μ_T *of FUITCT*

	t_1	t_2	t_3	t_4	t_5
t_1	0.00	0.60	0.40	0.90	0.23
t_2	0.60	0.00	1.00	0.40	0.80
t_3	0.40	1.00	0.00	0.33	0.40
t_4	0.90	0.40	0.33	0.00	0.48
t_5	0.23	0.80	0.40	0.48	0.00

Table8: Inter task communication ranking

	t_1	t_2	t_3	t_4	t_5
t_1	—	HCT	ACT	**VHCT**	LCT
t_2	HCT	—	**VHCT**	ACT	**VHCT**
t_3	ACT	VLCT	—	LCT	ACT
t_4	VHCT	ACT	LCT	—	ACT
t_5	LCT	VHCT	LCT	LCT	—

$T_{as}(i) = \{t_2, t_3, t_5\} \rightarrow \{P_1(\text{HASP})\}$

$T_{nas}(i) = \{t_1, t_4\} \rightarrow \{P_2, P_3\}.$

Table 9: Optimal Result of the Example

Processor	assigned tasks	PFUET (1)	PFUITCT (2)	FURT(f) =(1)+(2)	FURT(f)
p_1	t_2 , t_5,t_3	(25,45,65)	(70,120,175)	(95,165,240)	(95,165,240)
p_2	t_1	(5,10,15)	(75,105,140)	(80,115,155)	
p_3	t_4	(5,10,15)	(75,105,135)	(80,115,150)	

- **Trapezoidal Fuzzy Time**

Let a set N= $\{t_1, t_2, t_3, t_4, t_5, t_6\}$ of "n=6" tasks and a set M= $\{p_1, p_2, p_3, p_4\}$ of "m=4" processors. The execution time matrix $FUETM = [et_{ij}]$ of order m x n whose elements are fuzzy trapezoidal numbers as given in Table 10. Inter tasks communication time matrix $FUITCTM = [ct_{ij}]$ of order m with fuzzy trapezoidal numbers as given in Table11.

Table 10: Fuzzy Execution Time Matrix

	p_1	p_2	p_3	p_4
t_1	(2,4,6,10)	(8,10,12,14)	(5,8,10,12)	(10,15,17,20)
t_2	(6,9,11,14)	(2,4,6,8)	(8,10,12,15)	(6,8,10,12)
t_3	(15,20,23,25)	(8,11,14,16)	(4,7,9,13)	(15,17,19,21)
t_4	(2,3,5,9)	(4,6,9,12)	(3,4,6,9)	(8,10,12,16)
t_5	(7,10,13,15)	(6,10,12,16)	(5,7,10,12)	(1,3,5,8)
t_6	(8,10,12,16)	(10,12,13,15)	(6,9,11,15)	(6,8,11,13)

Table 11: Fuzzy InterTask Execution Time Matrix

	t_1	t_2	t_3
t_1	(0,0,0,0)	(4,6,10,12)	(0,2,4,7)
t_2	(4,6,10,12)	(0,0,0,0)	(2,4,7,10)
t_3	(0,2,4,7)	(2,4,7,10)	(0,0,0,0)
t_4	(10,12,15,17)	(5,7,12,15)	(10,12,14,16)
t_5	(12,14,15,17)	(4,6,9,11)	(16,17,18,20)
t_6	(0,0,0,0)	(4,5,7,9)	(2,5,9,11)
	t_4	t_5	t_6
t_1	(10,12,15,17)	(12,14,15,17)	(0,0,0,0)
t_2	(5,7,12,15)	(4,6,9,11)	(4,5,7,9)
t_3	(10,12,14,16)	(16,17,18,20)	(2,5,9,11)
t_4	(0,0,0,0)	(2,4,6,10)	(0,0,0,0)
t_5	(2,4,6,10)	(0,0,0,0)	(10,12,15,17)
t_6	(0,0,0,0)	(10,12,15,17)	(0,0,0,0)

The crisp indices for the fuzzy execution times et_{ij} and inter task communication time areobtained by (8) and (9) shown in the table 12 and 13.

Table12: Crisp indices et_{ij} j for the fuzzy execution times et_{ij}

	p_1	p_2	p_3	p_4
t_1	5.50	11.00	8.75	15.5
t_2	10.00	5.00	11.25	9.00
t_3	20.75	12.25	8.25	18.00
t_4	4.75	7.75	5.50	11.5
t_5	11.25	9.00	8.50	4.25
t_6	11.50	12.5	10.25	9.50

Table13: Crispindices ct_{ij} for the fuzzy inter tasks communication times ct_{ij} ,

	t_1	t_2	t_3
t_1	0.00	8.00	2.75
t_2	8.00	0.00	5.75
t_3	2.75	5.75	0.00
t_4	13.5	9.75	13.00
t_5	14.5	7.50	17.75
t_6	0.00	6.25	6.75
	t_4	t_5	t_6
t_1	13.5	14.5	0.00
t_2	9.75	7.5	6.25
t_3	13	17.75	6.75
t_4	0.00	5.50	0.00
t_5	5.50	0.00	13.5
t_6	0.00	13.50	0.00

Table 14:average load assigned to the processor

Load	p_1	p_2	p_3	p_4	Total Load
Average Load	16	14	13	17	60
Tolerance 10%	18	15	14	19	66
Total	34	29	27	36	126

Table 15:*Defuzzified membership value* μ_P *of FUET*

	p_1	p_2	p_3	p_4
t_1	0.27	0.53	0.42	0.75
t_2	0.48	0.24	0.54	0.43
t_3	1.00	0.59	0.42	0.87
t_4	0.23	0.37	0.27	0.55
t_5	0.54	0.43	0.41	0.20
t_6	0.54	0.59	0.49	0.46
Average Speed of Processor	.51 HASP	.46 VHASP	.43 VHASP	.54 HASP

Table 16:*Defuzzified membership value* μ_T *of FUITCT*

	t_1	t_2	t_3	t_4	t_5	t_6
t_1	0.00	0.45	0.15	0.76	0.82	0.00
t_2	0.45	0.00	0.32	0.55	0.42	0.35
t_3	0.15	0.32	0.00	0.73	1.00	0.38
t_4	0.76	0.55	0.73	0.00	0.31	0.00
t_5	0.82	0.42	1.00	0.31	0.00	0.76
t_6	0.00	0.35	0.38	0.00	0.76	0.00

Table17: Grading of the communicating tasks

	t_1	t_2	t_3	t_4	t_5	t_6
t_1	-	**ACT**	VLCT	HCT	**VHCT**	VLCT
t_2	ACT	-	LCT	ACT	ACT	LCT
t_3	VLCT	LCT	-	**HCT**	VHCT	LCT
t_4	HCT	ACT	HCT	-	LCT	VLCT
t_5	VHCT	ACT	VHCT	LCT	-	HCT
t_6	VLCT	LCT	LCT	VLCT	HCT	-

$$T_{as}(i) = \{t_2, t_3, t_5\} \rightarrow \{P_1(\text{HASP})\}$$
$$T_{nas}(i) = \{t_1, t_4\} \rightarrow \{P_2, P_3\}.$$

Table 18: Optimal Result of the Example

Processor	assigned tasks	PFUET (1)	PFUITCT (2)	FURT(f) =(1)+(2)	FURT(f)
p_1	t_2	(6,9,11,14)	(19,28,45,57)	(25,37,56,71)	(56,74,97,118)
p_2	t_3, t_4	(12,17,23,28)	(37,51,71,90)	(49,68,94,118)	
p_3	t_5, t_1	(10,15,20,24)	(46,59,77,94)	(56,74,97,118)	
p_4	t_6	(6,8,11,13)	(16,22,31,37)	(22,30,42,50)	

V. Conclusion

In this article we developed a mathematical model by Fuzzy execution times et_{ij} and fuzzy inter tasks communication times ct_{ij} which provide the new direction to optimize the system. This model has the efficient bundles of tasks for the optimal allocation to minimize the response time of the system. This technique also provides the allocation problem of the tasks with fuzzy execution time. In this article the proposed methodology is efficient but do not provide the completeness for the all technique which would exist. Many new ways we can also design in future work. In this work we have focused only on the response time of the system, but we can also consider the load balancing problem in future with fuzzy technique. This model is useful in telephone networks, cellular network, image processing etc.

VI. References

[1] Zadeh L. A. (1965) Fuzzy Sets. Intl J. Information Control 8:338-353.

[2] Mamdani E. H., Assilion S (1974) An Experiment in Linguistic Synthesis With a Fuzzy Logic Controller, Intl J. Man-Machine Stud 7:1-13.

[3] Holmblad L. P., Ostergaard J. J. (1982) Control of Cement Kiln by Fuzzy Logic, Gupta M. M, Sarchez E, Fuzzy Information and Decision Processes, North Holland, pp. 389-399.

[4] B. Kosko, Neural networks and fuzzy systems: a dynamical system approach to machine intelligence, Prentice-Hall, New York, 1992.

[5] C. S. G. Lee & C. L. Chen, Efficient mapping algorithms for scheduling robot int1erse dynamics computation on a multiprocessor system, vol. 20 no. 3, IEEE Trans. Systems, Man and Cybernetics, 1990.

[6] K.K. Bhutani, Distributed Computing, The Indian Journal of Telecommunication, 1994, pp. 41-44.

[7] B.R. Sitaram, Distributed Computing – A User"s View Point, CSI Communication, vol.-18 No. 10, 1965, pp. 26-28.

[8] A. A. Elsadek and B. E. Wells, A Heuristic model for task allocation in heterogeneous distributed Computing systems, The International Journal of Computers and Their Applications, Vol. 6, No. 1, 1999, pp 0-35.

[9] D.W. Coit, and A.E. Smith, Reliability Optimization of Series Parallel Systems using a Genetic Algorithm. IEEE Transactions on Reliabilit,. vol. R-45, 1996, pp. 254-260.

[10] P. K. Yadav and Nadeem Ahmad, Performance Analysis of Heterogeneous Distributed Processing System through Systematic Allocation of Task, International Journal of Intelligent Information Processing, Vol. 5(1), 2011. ,pp. 19– 24.

[11] V. Kumar, M. P. Singh, and P.K. Yadav, An Efficient Algorithm for Allocating Tasks to Processors in a Distributed System. Proc. of the 19th National system conference, SSI, Coimbatore, India, 1995, pp 82- 87.

[12] V. Kumar, P.K. Yadav and K. Bhatia, Optimal Task Allocation in Distributed Systems owing to Inter Tasks Communication Effects. Proc. of the 33rd Annual convention of system society of India, New Delhi, India, 1998, pp. 369-378.

[13] M.P Singh,., V. Kumar and A. Kumar, An Efficient Algorithm for Optimizing Reliability Index in Tasks-Allocation. Acta Ciencia Indica. 1999, pp. 437-444.

[14] Srinivasan, Santhanam and K. Niraj Jha, Safety and Reliability Driven Task Allocation in Distributed System, IEEE Transactions on Parallel and Distributed Systems. vol. 10, 1999, pp. 238-250.

[15] P. K.Yadav, M. P Singh and Harendra Kumar, Scheduling Algorithm: Tasks Scheduling Algorithm for Multiple Processors with dynamic Reassignment. International Journal of Computer System, Network and Communication,2008, pp. 1-9.

[16] R Nagarajan, A. Solairaju, A.,Computing Improved Fuzzy Optimal Hungarian Assignment Problems with Fuzzy Costs under Robust Ranking Techniques, International Journal of Computer Applications , Vol. 6, No.4, 2010, pp.6-13.

[17] P.K..Yadav, P. Pradhan, P.P. Singh, A Fuzzy Bundleing Method to Minimize the Inter Task Communication Effect for Optimal Utilization of Processor's Capacity in Distributed Real Time Systems, Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) (Advances in Intelligent and Soft Computing: Published by Springer) Vol.130, 2012, pp 159-168.

[18] P.K. Yadav, M.P. Singh, K. Sharma., Tasks Allocation Model for Reliability and Cost Optimization in Distributed Computing System, International Journal Of Modeling, Simulation, and Scientific Computing, Vol.2, No.2, 2011, pp.131-149.

[19] M. Sabeghi, H. Deldari, V. Salmani, M. Bahekmat, A Fuzzy Algorithm for Real-Time Scheduling of Soft Periodic Tasks on Multiprocessor System, Procedding of IADIS International Conference Applied Computing, 2006, pp.467-471.

[20] Harendra Kumar,M. P. Singh, Pradeep Kumar Yadav A Tasks Allocation Model with Fuzzy Execution and Fuzzy Inter-Tasks Communication Times in a Distributed Computing System International Journal of Computer Applications (0975 – 8887) Volume 72– No.12, June 2013

[21] Gillett, Introduction to Operations Research: A computer Oriented Algorithmic Approach, McGraw-Hill, New York,1984.