A TRICKY TASK SCHEDULING TECHNIQUE TO OPTIMIZE TIME COST AND RELIABILITY IN MOBILE COMPUTING ENVIRONMENT

Faizul Navi Khan[1], Kapil Govil[2]

[1]*Department of Computer Application, Teerthanker Mahaveer University, Moradabad - 244 001, UP, INDIA*
[2]*Department of Computer Application, Teerthanker Mahaveer University, Moradabad - 244 001, UP, INDIA*

Abstract

Mobile Computing Environment (MCE) consisting portable computing devices interconnected by wireless medium, are increasingly being used in many area of science, business and education etc nowadays. It is the computation made over physical mobility. MCE facilitate the user to perform a task from a distant place from the device. This provides flexible communication between user and continuous access to networked services. Mobile computing shares many features with distributed computing as the application runs in MCE executes in distributed manner i.e. an application is combination of multiple tasks and these task schedule on the available resources to be execute. The performance of an application is depending on task scheduling technique in MCE. Task scheduling technique consist in finding an allocation of the tasks to the processors such that the total execution cost and time are minimized or processing reliability is maximized. So the ultimate objective of finding task scheduling technique is to improve the performance in terms to minimize cost and time and maximize reliability of MCE. In MCE an application can be seen as one or multiple task modules and each of the task modules allocate on available processor for its execution. Therefore, in this paper, the problem of scheduling the tasks in mobile computing environment is explore with objectives to minimize processing cost, processing time or maximize the processing reliability through a new task scheduling technique to optimize performance in MCE. The nature of the assignment will be static.

Keywords – *Cost, Mobile Computing Environment, Performance, Reliability, Task scheduling*

---***---

1. INTRODUCTION

Advances in communication and computing technologies have introduced a new way of computing environment, called ‘Mobile Computing Environment (MCE)’, which allows mobility of users thereby providing “anytime anywhere” computing environment. The mobility of user (or device) results in a highly dynamic environments with respect to the availability of resources/infrastructures for any application running on the mobile devices. Also, in a mobile computing environment, location has a profound effect. A Mobile Computing Environment (MCE) consists of a set of multiple processors interconnected by wireless communication links and equipped with software systems to produce an integrated and consistent processing environment. The task partitioning and scheduling strategies also play an important role for achieving high performance in MCE and an efficient task scheduling technique can utilize the resources in the MCE to complete the task execution at the earliest or in optimize manner. Task scheduling is a very common and challenging problem in MCE and it deals with finding an optimal scheduling of tasks to the processors so that the processing cost and processing time are be minimized and the processing reliability is maximized without violating any of the system constraints.

Processing time is referred as the time taken by the task for its execution, expenses occurs during the task execution is describe as the processing cost and reliability can be consider as the reliability of its processors as well as the reliability of its communication links. Distribute data across processors unevenly so that each processor performs the volume of computation proportional to its speed is a common approach to solve task scheduling problem in MCE. This problem deals with finding an optimal task scheduling technique to the processors so that the processing time and cost can be minimized and processing reliability can be maximized for task scheduling in MCE. In MCE, the objective is to make processors busy executing tasks all the time by ensuring that it does not get idle and this serves the purpose. Task optimization is highly dependent on the tasks allocation method onto the available processor. In order to improve the performance of MCE, application workloads are divided into small independents units called tasks and these tasks need to be execute on available processor through a task scheduling technique.

Task scheduling technique should be capable enough in terms of optimize processing capabilities in MCE i.e. minimize processing cost, minimize processing time and maximize processing reliability. This research paper explore task scheduling problem with n number of processors and m number of tasks where m>n in MCE and these tasks are need to schedule on the available processors in optimize manner with satisfying the processing constraints i.e. time, cost and reliability. In task scheduling problem once available processors are scheduled with a single task while the remaining tasks have to wait (if the numbers of task are greater than numbers of processor) until the present scheduled task will execute. To overcome such problem in a MCE, a task scheduling technique would be needed where more than one task to a single processor can be schedule in order to achieve minimum processing time, processing cost and maximum processing reliability in MCE. Some of the task allocation schemes have been reported in the literature, such as Task modeling [1], Scheduling of tasks [2, 5, 7, 8, 11], Task allocation scheme [3, 15, 20], Resource allocation [4, 13, 14, 16, 19], Load balancing [10], Static task assignment [6], Resource scheduling [9], Job scheduling algorithm [12], Resource management and scheduling [17], Application scheduling in Mobile Cloud Computing [18]. This research paper propose a new task scheduling technique to task allocation in MCE by using the proper utilization of processors of the MCE so the problem of load balancing can be avoid.

2. NOTATIONS

p	Processor
t	Task
n	Number of Processors
m	Number of Tasks
TCR	Task Cost Reliability
MTCR	Modified Task Cost Reliability

3. OBJECTIVE

The objective of this research paper is to solve task scheduling problem in an efficient manner so that maximum level of optimization is achieved in order to minimize processing cost and time and maximize reliability by the proper utilization of resource in Mobile Computing Environment (MCE). The applied technique would also ensure that processing of all the tasks and its sub tasks as task modules are more than the numbers of processors in the MCE. The type of assignment of tasks to the processor is static. In this research paper performance is measured in term of processing time, cost and reliability of the task that have to be get processed on the processors of the environment and it have to be optimally processed i.e., time, cost to be minimized and reliability maximized.

4. TECHNIQUE

In order to obtain the overall optimal processing cost or processing time or processing reliability of a Mobile Computing Environment (MCE), this research paper consider a problem of task scheduling where a set $P = \{p_1, p_2, p_3,\ldots\ldots\ldots p_n\}$ of 'n' processors with different processing capabilities and a set $T = \{t_1, t_2, t_3,\ldots\ldots\ldots t_m\}$ of 'm' tasks, where m>n, every task has also contain some number of sub tasks module. Processing time, cost and reliability are known for each tasks module to the processor and arrange in TCTR. Initially processing cost, processing time will be initializing as zero (0) and processing reliability as one (1). Task scheduling algorithm will find for the minimum value by row (without repeating the column in the matrix) for processing time, cost or maximum for processing reliability, in result would get the tasks equal to number of processors available in the MCE and those task will get assigned. The process will repeat until number of tasks will remain lesser than the processor available in the environment. Once that situation will occur where the numbers of processors are greater than the tasks waiting for the processing then will make slide change in the technique. Instead of searching element row wise, the search will be column wise that will make enable the final allocation of remaining unscheduled task in MCE.

The function to calculate overall time [Etime], cost [Ecost] and reliability [Ereilability] is given here:

$$\mathrm{Etime} = \left[\sum_{i=1}^{n}\left\{\sum_{i=1}^{n} \mathrm{ET_{ij}}\ \mathrm{X_{ij}}\right\}\right] \qquad \text{(i)}$$

$$\mathrm{Ecost} = \left[\sum_{i=1}^{n}\left\{\sum_{i=1}^{n} \mathrm{EC_{ij}}\ \mathrm{X_{ij}}\right\}\right] \qquad \text{(ii)}$$

$$\mathrm{Ereliablity} = \left[\prod_{i=1}^{n}\left\{\sum_{i=0}^{n} \mathrm{ER_{ij}}\ \mathrm{X_{ij}}\right\}\right] \qquad \text{(iii)}$$

5. ALGORITHM

1. Start Algorithm
2. Read the number of task in m
3. Read the number of processor in n
4. Store task and Processing Cost, time and reliability into Matrix PCTR (,) n x m of order
5. While (All task! = Assigned)

{

i. Check if the matrix containing numbers of tasks are greater than or equal to numbers of processors (m>=n) then go to step (ii) else step (iv)

ii. Search minimum value row wise in the matrix for time, cost or maximum value for reliability.

iii. Check if the column is previously selected for minimum/maximum value then GO TO step – (ii) to find next minimum value for the row else Goto step (vi) to assign eligible task.

iv. Search the minimum value for cost, time or maximum value for reliability column wise in the matrix

v. Check if the row is previously selected for minimum or maximum value then GO TO step – (iv) to find next minimum or maximum value for the column else Goto step (vi) to assign eligible task.

vi. Assign the eligible tasks to available processors

}

6. State the results
7. End of algorithm

6. IMPLEMENTATION

This research paper consider Mobile Computing Environment (MCE) which consist a set P of 3 processors $\{p_1, p_2, p_3\}$, and a set T of 3 tasks $\{t_1, t_2, t_3\}$. Each tasks contained some number of modules and these modules belongs to different tasks would be process on available processors in MCE as mentioned in mentioned in Figure 1.

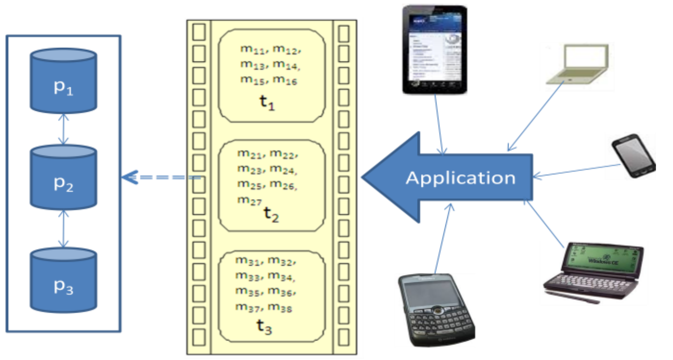

Fig 1: Task modules originating from different mobile hosts are waiting in queue.

Table 1: Tasks and their modules

t_1	$\{m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}\}$
t_2	$\{m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}, m_{27}\}$
t_3	$\{m_{31}, m_{32}, m_{33}, m_{34}, m_{35}, m_{36}, m_{37}, m_{38}\}$

Each task contained different individual task components which are known as modules. The processing time (t), processing cost (c) and processing reliability (r) of each task modules to each processor are known and mentioned in Processing Cost Time Reliability (PCTR) matrix as mentioned in Table 2

Table 2: Processing Time Cost Reliability Matrix

	Processors	p_1	p_2	p_3
Task	Modules	t-c-r	t-c-r	t-c-r
t_1	m_{11}	15-100-0.999111	18-135-0.999125	17-110-0.999450
	m_{12}	16-110-0.999429	20-125-0.999159	18-115-0.999429
	m_{13}	21-115-0.999423	16-105-0.999328	15-120-0.999418
	m_{14}	18-120-0.999218	22-130-0.999419	23-100-0.999458
	m_{15}	19-105-0.999451	30-150-0.999480	26-125-0.999450
	m_{16}	33-125-0.999329	25-145-0.999219	28-155-0.998219
t_2	m_{21}	16-110-0.999112	18-125-0.941124	17-115-0.999418
	m_{22}	17-100-0.999149	20-135-0.982152	18-110-0.999429
	m_{23}	22-125-0.999533	16-130-0.979317	15-120-0.999418
	m_{24}	19-110-0.973218	22-105-0.999419	23-100-0.999413
	m_{25}	29-105-0.929431	30-150-0.989411	26-125-0.999414
	m_{26}	33-125-0.979328	25-145-0.969019	28-150-0.998226
	m_{27}	31-115-0.924322	22-140-0.924219	25-155-0.998214
t_3	m_{31}	15-125-0.999669	19-110-0. 997854	18-150-0. 998967
	m_{32}	18-100-0.999433	21-135-0. 998780	19-110-0. 999232
	m_{33}	26-130-0.998798	14-125-0. 998955	16-115-0. 987432
	m_{34}	21-110-0.999754	21-135-0. 987432	22-100-0. 999876
	m_{35}	29-105-0. 998766	32-155-0. 999578	24-185-0. 999866
	m_{36}	37-115-0. 998654	22-145-0. 998643	29-140-0. 998456
	m_{37}	35-125-0. 999478	25-160-0. 998903	24-105-0. 999754
	m_{38}	30-155-0. 999268	21-140-0. 998458	25-165-0. 996371

To proceed further with task scheduling problem in PCTR matrix, considering that t_1 is based on processing time (t) (it may be processing cost or reliability), task t_2 is based on processing cost (c) (it may be processing time and reliability) and task t_3 is based on processing reliability (r) (it may be processing time and processing cost). Hence a new matrix named MPCTR can be derived by using PCTR, In MPCTR task t_1 will represent processing time (t), t_2 task processing cost (c) and t_3 task processing reliability (r). New matrix MPCTR represent as Table 3:

Table 3: Modified Processor Cost Time Reliability Matrix

	Processors	p_1	p_2	p_3
Task	Modules	t-c-r	t-c-r	t-c-r
t_1	m_{11}	15-…-…	18-…-…	17-…-…
	m_{12}	16-…-…	20-…-…	18-…-…
	m_{13}	21-…-…	16-…-…	15-…-…
	m_{14}	18-…-…	22-…-…	23-…-…
	m_{15}	19-…-…	30-…-…	26-…-…

	m_{16}	33-…-…	25-…-…	28-…-…
t_2	m_{21}	...-110-…	...-125-…	...-115-…
	m_{22}	...-100-…	...-135-…	...-110-…
	m_{23}	...-125-…	...-130-…	...-120-…
	m_{24}	...-110-…	...-105-…	...-100-…
	m_{25}	...-105-…	...-150-…	...-125-…
	m_{26}	...-125-…	...-145-…	...-150-…
	m_{27}	...-115-…	...-140-…	...-155--…
t_3	m_{31}	…-…-0. 999669	…-…-0. 997854	…-…-0. 998967
	m_{32}	…-…-0. 999433	…-…-0. 998780	…-…-0. 999232
	m_{33}	…-…-0.998798	…-…-0. 998955	…-…-0. 987432
	m_{34}	…-…-0. 999754	…-…-0. 987432	…-…-0. 999876
	m_{35}	…-…-0. 998766	…-…-0. 999578	…-…-0. 999866
	m_{36}	…-…-0. 998654	…-…-0. 998643	…-…-0. 998456
	m_{37}	…-…-0. 999478	…-…-0. 998903	…-…-0. 999754
	m_{38}	…-…-0. 999268	…-…-0. 998458	…-…-0. 996371

MPCTR can be break into three different tables for each constraint i.e. Table 4 for processing time, Table 5 for processing cost and Table 6 for processing reliability in order to demonstrate all the three constraints of processing i.e. time cost and reliability in Mobile Computing Environment (MCE).

Table 4: Processing Time

	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
p_1	15	16	21	18	19
p_2	18	20	16	22	30
p_3	17	18	15	23	26

Table 5: Processing Cost

	m_{21}	m_{22}	m_{23}	m_{24}	m_{25}	m_{26}	m_{27}
p_1	110	100	125	110	105	125	115
p_2	125	135	130	105	150	145	140
p_3	115	110	120	100	125	150	155

Table 6: Processing Reliability

	m_{31}	m_{32}	m_{33}	m_{34}	m_{35}	m_{36}	m_{37}	m_{38}
p_1	0. 999669	0. 999433	0.998798	0. 999754	0. 998766	0. 998654	0. 999478	0. 999268
p_2	0. 997854	0. 998780	0. 998955	0. 987432	0. 999578	0. 998643	0. 998903	0. 998458
p_3	0. 998967	0. 999232	0. 987432	0. 999876	0. 999866	0. 998456	0. 999754	0. 996371

To demonstrate new scheduling technique, this research paper is initially considering Table 4 for task scheduling in MCE. Table 4 represent number of five modules which are originally belongs to task t_1 need to get schedule on number of 3 processors. In Table 4 numbers of task modules are 5 and the numbers of processors are 3 (m>n), scheduling technique will determine minimum value for each row and will get the resulting output as mentioned in Table 7:

Table 7: Matrix representing minimum value

	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
p_1	15				
p_2			16		
p_3		18			

Hence there are only three processors in the environment, employed technique would schedule only three task module in MCE as mentioned in Table 8.

Table 8: Scheduling Table

Processor	Task	Processing Cost
p_1	m_{11}	15
p_2	m_{12}	16
p_3	m_{13}	18

After the first task scheduling step execution, still two task in the queue and gets unscheduled, here numbers of processors are greater than number of task (two) (m<n), now the element will be searched by column wise and that approach will ensure that none of the task module will get remain

unexecuted. And the final task scheduling is mentioned in Table 9.

Table 9: Overall processing time

Processor	Task	Processing time	Etime
p_1	$m_{11} * m_{14}$	33	93
p_2	m_{12}	16	
p_3	$m_{13} * m_{15}$	44	

Graphical representation of overall processing time is mentioned in Fig 2:

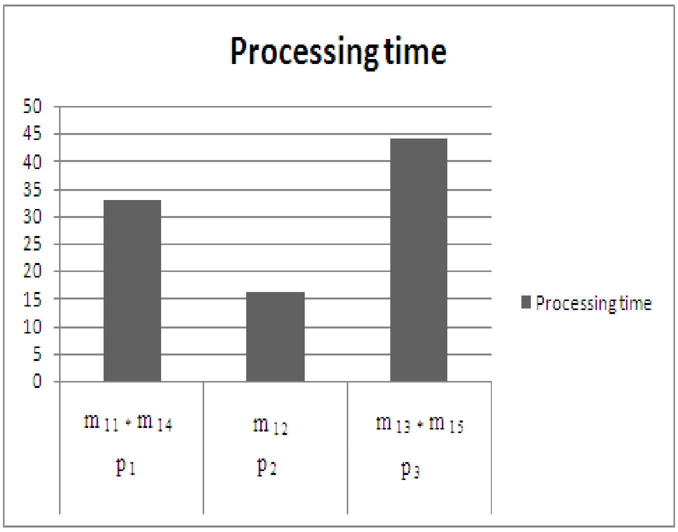

Fig 2: Overall processing time taken by the processor in MCE

By applying the same task scheduling technique, this research paper would also calculate processing cost and processing reliability for the given example in this research paper as mentioned in Table 10 and Table 11 respectively.

Table 10: Overall processing cost

Processor	Task	Processing Cost	Ecost
p_1	$m_{22} * m_{25} * m_{27}$	320	820
p_2	$m_{23} * m_{24}$	235	
p_3	$m_{21} * m_{26}$	265	

Graphical representation of overall processing cost is mentioned in Fig 3:

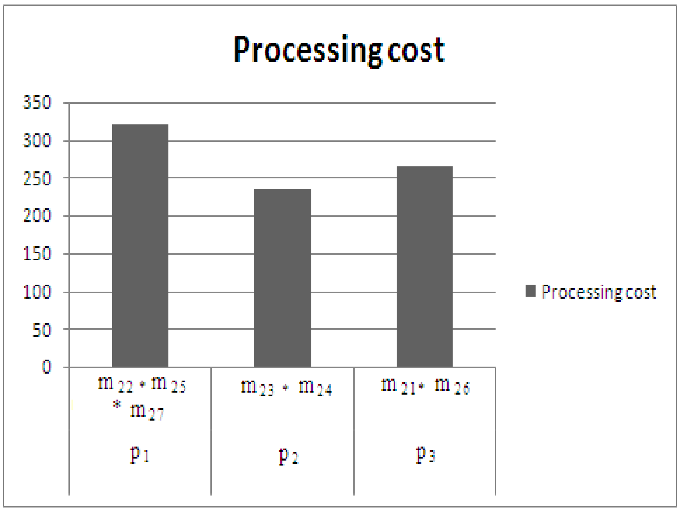

Fig 3: Overall processing cost occurred during the execution in MCE

Table 11: Overall processing reliability

Processor	Task	Processing Cost	Ereliability
p_1	$m_{31} * m_{34} * m_{36}$	0.998077	0.993508
p_2	$m_{33} * m_{37} * m_{38}$	0.996320	
p_3	$m_{32} * m_{35}$	0.999098	

Figure 4 showing graphical representation of overall processing reliability of given example:

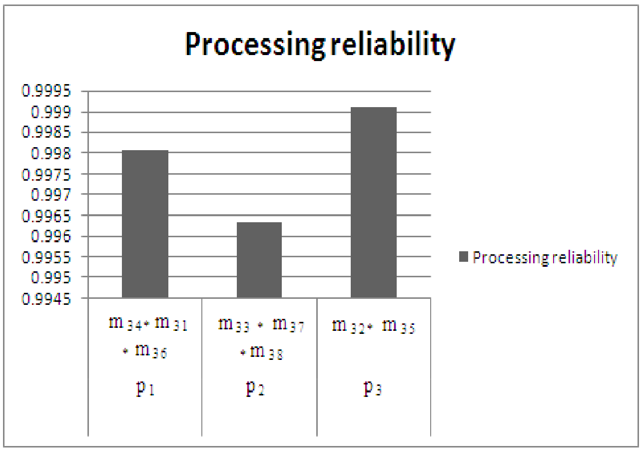

Fig 4: Overall processing reliability of the task in MCE

7. CONCLUSIONS

This research paper provide the solution to the problem of task scheduling through a smart task scheduling technique, in which the number of the tasks is more than the number of processors in Mobile Computing Environment (MCE). Opted scheduling technique will ensure to satisfy various constraints i.e. time, cost and reliability in MCE. The task scheduling technique is presented in pseudo code and applied on the several sets of input data to test the performance and effectiveness of the pseudo code. Optimization of task is the common objective for any task scheduling problem that the task needs to be processed with optimal time, cost and reliability. This paper consider three tasks i.e. t_1, t_2 and t_3 with different tasks module and process t_1 with minimum time, t_2 process with minimum cost and t_3 process with maximum reliability in MCE. The optimal output of the given example that is consider in research paper to test the task scheduling technique is mentioned in the implementation section of the paper is mentioned in Table 12:

Table 12: Consolidated results for Time, Cost and Reliability in MCE

Task	p_1	p_2	p_3	Optimal ETime	Optimal ECost	Optimal Ereliablity
t1	m_{11} * m_{14}	m_{12}	m_{13} * m_{15}	93	---	---
t2	m_{22} * m_{25} * m_{27}	m_{23} * m_{24}	m_{21} * m_{26}	---	820	---
t3	m_{34} * m_{31} * m_{36}	m_{33} * m_{37} * m_{38}	m_{32}* m_{35}	---	---	0.993508

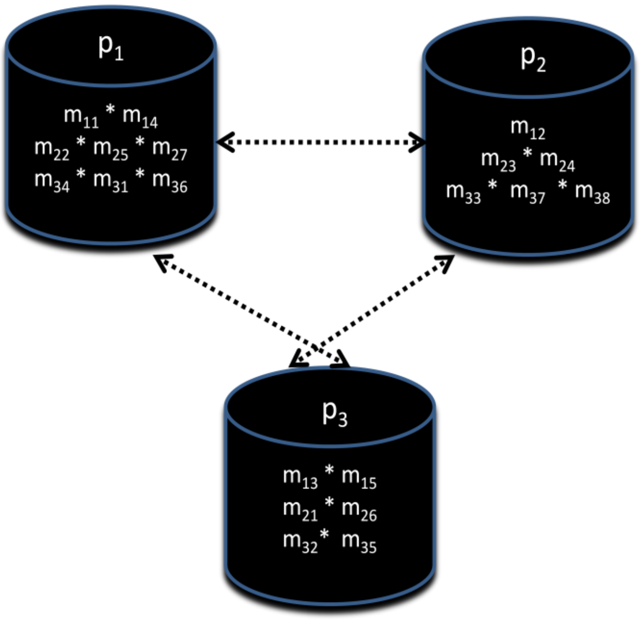

Fig 5: Final Task Scheduling in Mobile Computing Environment (MCE)

The technique stated in pseudo code applied on several sets of input data and that verified the objective of get maximum processing reliability for given tasks for their execution. The analysis of an algorithm mainly focuses on time complexity. The time complexity of above mentioned algorithm is O(m+n). By taking several input examples, the above algorithm returns results as mentioned in Table 13.

Table 13 Time Complexity

Number of Processors (n)	Number of tasks (m)	Complexity of algorithm [5] $O(mn^2)$	Complexity of present algorithm O(m+n)
3	5	45	8
3	6	54	9
3	7	63	10
3	8	72	11
3	9	81	12
4	5	80	9
4	6	96	10
4	7	112	11
4	8	128	12
4	9	144	13
5	6	125	11
5	7	150	12
5	8	175	13
5	9	200	14
5	10	225	15

Time complexity comparisons (as mentioned in Table 13) are shown in Figure 6, 7 and 8.

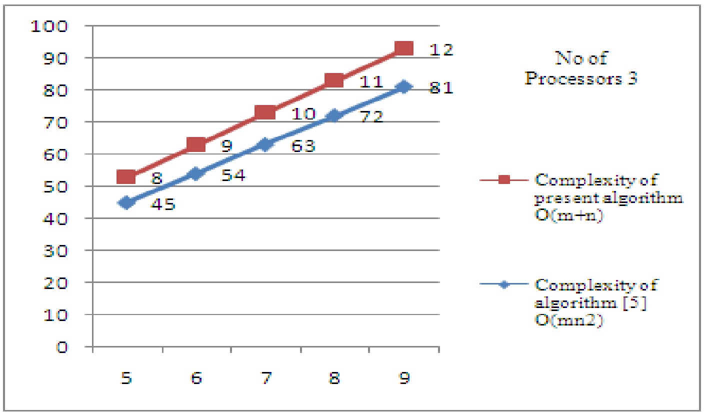

Fig 6: Time complexity for No of Processors 3

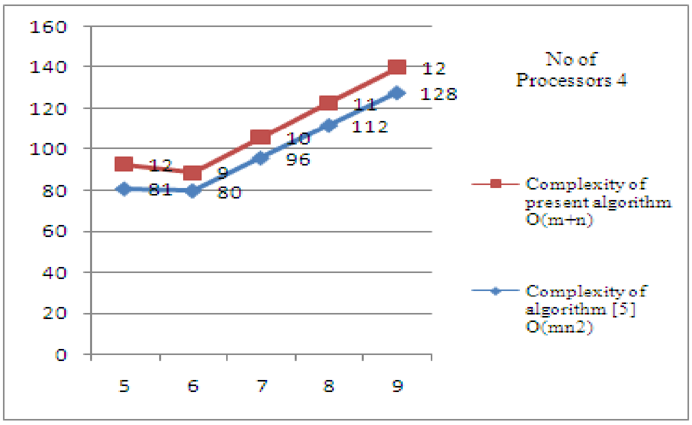

Fig 7: Time complexity for No of Processors 4

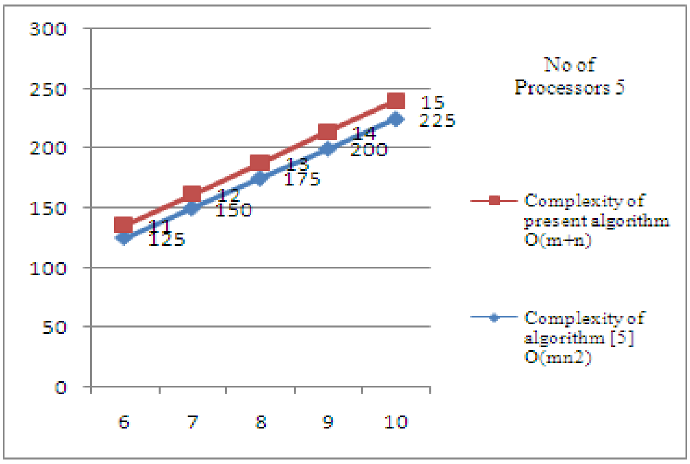

Fig 8: Time complexity for No of Processors 5

REFERENCES

[1] Ana Isabel Molina, Miguel Ángel Redondo, Manuel Ortega, 2007, Applying Task Modeling and Pattern-based techniques in Reengineering Processes for Mobile Learning User Interfaces: A case study, JOURNAL OF COMPUTERS, VOL. 2, Issue 4, 23-30

[2] Benazir Fateh, G. Manimaran, 2013, Joint Scheduling of Tasks and Messages for Energy Minimization in Interference-aware Real-time Sensor Networks, IEEE Transactions on Mobile Computing, 25 June 2013. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TMC.2013.81>, ISSN: 1536-1233

[3] Faizul Navi Khan, Kapil Govil, 2013, Distributed Task Allocation Scheme for Performance Improvement in Mobile Computing Network, International Journal of Trends in Computer Science, vol: 2 issue: 3, pp: 809-817

[4] Hongbin Liang, Dijiang Huang, Lin X. Cai, Xuemin (Sherman) Shen, Daiyuan Peng, 2011, Resource Allocation for Security Services in Mobile Cloud Computing, IEEE INFOCOM 2011 Workshop on M2MCN-2011, ISBN: 978-1-4244-9920-5, 191-195

[5] Ilavarasan E, Manoharan R, 2010, High Performance and Energy Efficient Task Scheduling Algorithm for Heterogeneous Mobile Computing System, International Journal of Computer Science & Information Technology, Vol. 2, Issue 2, 10-27

[6] Kapil Govil and Dr. Avanish Kumar. 2011. A modified and efficient algorithm for Static task assignment in Distributed Processing Environment. International Journal of Computer Applications, Vol. 23, Number 8, Article 1, 1–5, ISBN: 978-93-80752-82-3, ISSN: 0975 – 8887

[7] L. Yu, G. Zhou, Y. Pu, 2011, An Improved Task Scheduling Algorithm in Grid Computing Environment, Int'l J. of Communications, Network and System Sciences, Vol. 4 No. 4, 227-231. doi: 10.4236/ijcns.2011.44027.

[8] Marjan Kuchaki Rafsanjani, Amid Khatibi Bardsiri, 2012, A New Heuristic Approach for Scheduling Independent Tasks on Heterogeneous Computing Systems, International Journal of Machine Learning and Computing, Vol: 2, No 4, 371-376

[9] S. Nagendram, J. Vijaya Lakshmi, D. Venkata Narasimha Rao, 2011, Efficient resource scheduling in data centers using MRIS, Indian Journal of Computer Science and Engineering, vol. 2, no. 5, pp. 764–769.

[10] S. Suryadevera, J. Chourasia, S. Rathore, A. Jhummarwala, 2012, Load balancing in computational grids using ant colony optimization algorithm," International Journal of Computer & Communication Technology, vol. 3, Issue 3, 20-23

[11] S. Thenmozhi , A. Tamilarasi, K. Thangavel, 2012, Fuzzy Based Task Scheduling for Hierarchical Resource Allocation in Mobile Grid Environment, International Journal of Soft Computing, Vol: 7, Issue: 3, 97-103

[12] Sandeep Kaur, Sukhpreet Kaur, 2013, Survey of Resource and Grouping Based Job Scheduling Algorithm in Grid Computing, International Journal of Computer Science and Mobile Computing, Vol: 2, Issue. 5, 214-218

[13] Sayed Chhattan Shah, Myong-Soon Park, 2010, Resource Allocation Scheme to Minimize Communication Cost in Mobile Ad Hoc Computational Grids, International Conference on Intelligent Networking and Collaborative Systems, ISBN: 978-0-7695-4278-2, 169-176

[14] Sayed Chhattan Shah, Qurat-Ul-Ain Nizamanib, Sajjad Hussain Chauhdaryc, Myong-Soon Park, 2012, An effective and robust two-phase resource allocation scheme for interdependent tasks in mobile ad hoc computational Grids, Journal of Parallel and Distributed Computing, Vol: 72, Issue 12, 1664–1679

[15] T. Xie and X. Qin, 2008, An Energy-Delay Tunable Task Allocation Strategy for Collaborative Applications in Networked Embedded Systems, IEEE Trans. Computers, Vol: 57, Issue 3, 329-343.

[16] Thenmozhi S., A. Tamilarasi and P.T. Vanathi, 2012, A Fault Tolerant Resource Allocation Architecture for Mobile Grid, Journal of Computer Science, Vol: 8, Issue 6, 978-982

[17] Vignesh V, Sendhil Kumar KS, Jaisankar N, 2013, Resource Management and Scheduling in Cloud Environment, International Journal of Scientific and Research Publications, Volume 3, Issue 6, 1-6

[18] Xianglin Wei,Jianhua Fan, Ziyi Lu,Ke Ding, 2013, Application Scheduling in Mobile Cloud Computing with Load Balancing, Journal of Applied Mathematics, Vol: 2013, 1-13

[19] Yashpal Rohilla, Preeti Gulia, 2012, Resource Allocation in cellular Network : A cross layer approach, International Journal of Engineering Research and Applications, Vol. 2, Issue 4, 912-915

[20] Yichao Jin, Jiong Jin, Alexander Gluhak, Klaus Moessner and Marimuthu Palaniswami, 2012, An Intelligent Task Allocation Scheme for Multihop Wireless Networks, IEEE Transactions on Parallel and Distributed Systems, Vol: 23, Issue 3, 444-451