

Heuristic Algorithm for Independent Task Scheduling In Grid Computing

B.Anitha, G.K.Kamalam

Abstract- Grid enables the integration of large number of geographically distributed heterogeneous resources owned by different organizations for resource sharing and collaboration in solving advanced sciences and engineering applications. In a distributed heterogeneous computing grid environment, scheduling independent tasks on the grid resources is more complicated and is an NP-Complete problem. Scheduling is the process of mapping the tasks to the available resources. In order to utilize the essence of grid efficiently, this paper presents a heuristic technique for scheduling/mapping the tasks to the resources. The efficiency of the proposed algorithm (WSSLVA) in terms of reduced makespan as well as better resource utilization is achieved. The experimental results indicate that the proposed WSSLVA algorithm is a promising algorithm than the Min-min heuristic scheduling algorithm in terms of makespan and resource utilization.

Keywords: Independent task, Task Scheduling, grid resources, makespan

I. INTRODUCTION

Grid computing and scheduling is an emerging field of research of how efficiently a task is assigned to a grid resources and achieving high performance [1]. The resources are heterogeneous, distributed and dynamic. Integration of geographically distributed resources through the Internet creates a virtual organization to solve complex problems in less time. The grid infrastructure is aggregated by different computing resources. Many research issues are addressed in Grid computing like resource management, job scheduling and security. One of the most challenging issues in grid computing environment is the task scheduling problem [2,10]. Why scheduling? To reduce the completion time of tasks and to make a system balanced by efficient utilization. Scheduling of tasks can be done by simply assigning the incoming tasks to the available resources. But the Grid scheduler, while allocating resources for tasks, would expected to consider various characteristics like load of the resource, length of the task, when new resources join or drop from the system and cost. Efficient grid scheduler is required to allocate the resources to the tasks in dynamic environment and to satisfy the user requirements. To achieve an optimum distribution of tasks in a heterogeneous network is an unresolved question in Grid environment. The proposed computational scheduler is heuristic and meta-heuristic scheduler. Heuristic task scheduling algorithms provides better solution compared to the

traditional techniques [3]. Metaheuristics use a random search based strategies to identify optimal solution. The proposed method has been proved to be efficient in solving many hard computational problems and achieve an optimal performance in terms of makespan and cost.

II. RELATED WORK

In Min-min scheduling algorithm, for each unassigned task, expected completion time of the task is found out over all allocated resources. The resource with minimum completion time for each task is selected and then a task with minimum completion time is assigned to the resource. The assigned task is removed from the task pool and the procedure is repeated for rest of the unassigned tasks [1,2].

In Max-min scheduling algorithm, analogous to Min-Min, task with maximum completion time is found out and assigned to the resource [2].

In On-line mode heuristic scheduling algorithm, tasks are scheduled to the resources as soon as it arrives. Grid environment includes many heterogeneous resources, online mode heuristic algorithms are more appropriate [1].

In Opportunistic Load Balancing (OLB) algorithm, each job is assigned in random order to the next available machine that is expected to be free, without considering the jobs execution time on that machine. This algorithm improves resource utilization. OLB results in poor makespan and flow time [4].

In the Minimum Execution Time (MET) algorithm, each job is assigned to the resource that has the minimum expected execution time. It does not consider the availability and the current load of the resource. Thus MET allocates each job to its best resource and leads to load imbalance between the resources [4].

The benefits of OLB and MET scheduling approach are combined to design new algorithm called MCT. In Minimum Completion Time (MCT) scheduling algorithm, the expected finishing time of a job is calculated by adding the resource availability time and the expected task execution time on that resource. The algorithm assigns each task to the resource that has minimum finishing time for that task. It avoids the limitations of poor execution of OLB and MET algorithms [2].

Min-mean heuristic scheduling algorithm works in two phases. In phase I, the algorithm takes a set of all unassigned tasks. It calculates the execution time for each task on each machine and a minimum execution time of each task is found out. The task with the overall minimum completion time is selected and allocated to the suitable machine. Then it removes the task from the

Revised Manuscript Received on November 14, 2019

Ms.B.Anitha, Assistant Professor in Kongu Engineering College Erode
Dr.G.K.Kamalam, Assistant Professor(SLG) in Kongu Engineering College Erode

task set. Until all the tasks get assigned, the process repeats. It calculates the total completion time of the entire machine and the mean completion time. In phase II, the algorithm takes the mean completion time of all machines and the machine whose completion time is greater than mean value is taken. The selected machines are arranged in decreasing order of the completion time. Then the assigned tasks on the selected machines are rescheduled whose completion time is less than the mean value. Min-mean achieves better makespan than Min-min algorithm [2].

Grid environment is a cluster computing environment with multiple WN in a cluster. The Best Cluster Decentralized Job Scheduling Algorithm divides the tasks into subtasks to optimize the execution time. The subtasks are assigned to the WN of different clusters in a grid environment. The algorithm efficiently schedules both the computing-intensive jobs and data-intensive jobs based on the best cluster value [3].

Task Unique Credit System Scheduling (TUCSA) Algorithm, based on the highest credit value of the task, schedule order is determined [4].

Guaranteed Best Effort QoS (GBEQoS) Algorithm, schedules the task efficiently to the appropriate resource and results in satisfying the user criteria such as minimum time and minimum cost in completing execution of the task [5].

Sort Completion Time Mean Tasks Scheduling (SCTMST) algorithm specifies the task schedule order to the appropriate resources. The first stage of the (SCTMST) algorithm sorts the list of completion time of the task. The mean value of each task is determined in the next stage of the algorithm. The maximum mean value is calculated and maximum mean value task is selected and scheduled to the resource that has the minimum completion time [6].

In Double Min-Min Credit Worth Task Scheduling algorithm, for the task in task set, the task worth value is calculated. Tasks are grouped accordingly in the Worth Set (WS) in the ascending order of the Task worth value. From the Worth Set, each task is assigned to the best resource that gives the minimum completion time. Until all the tasks in worth set are assigned the process repeats. In the next phase, tasks that are allocated to the resources are again reallocated to provide minimum makespan [7].

The Task Group Scheduling Algorithm used in a distributed grid environment. It divides the tasks into different groups based on the priority of the task and task attributes. Each group will have tasks with similar attributes. Groups are ordered based on priority that is given to the task attributes. The first included group will have tasks with high task attributes value or high priority compared to that of the other groups. Task with minimum execution time will be scheduled first in the selected group [9].

The heuristic scheduling algorithm aims to minimize the makespan, i.e., completion time of the metatask as soon as possible [10, 11]. The proposed algorithm (WSSLVA) schedules the task in the order specified in the Worth Set (WS) to the resource until the completion time of the tasks is less than the Scheduling Limit Value of the resources. Otherwise, the remaining tasks are scheduled to the next resource and so on. The mean completion time of all the resources is calculated. The tasks whose completion time greater than the mean completion time are selected. The tasks are rescheduled to the resources whose completion time is minimum compared

to that of the completion time of the previously allocated resource. The proposed algorithm performs the mapping of the tasks to the resources efficiently and achieves reduced makespan and better resource utilization.

III. HEURISTIC DESCRIPTIONS

A. Notations and definitions are listed below:

Notations	Definitions
$Length_i$	Length of the task T_i in MI
$Power_j$	Processing capacity of resource R_j in MIPS
$ETC(T_i, R_j) = Length_i / Power_j$	expected execution time of the task T_i on resource R_j
$T = \{T_1, T_2, \dots, T_n\}$	Task set
$R = \{R_1, \dots, R_m\}$	Resource set
CTT_i	expected completion time of task T_i
RT_j	ready time of resource R_j
CT_j	expected completion time of all task T_i scheduled on resource R_j
$makespan = \max(CTT_i)$	overall completion time of all tasks $1 \leq i \leq n$

B. Proposed Algorithm: Worth Set Scheduling Limit Value Task Scheduling (WSSLVA) Algorithm

This section explains (WSSLVA) algorithm. The algorithm enhances system performance in terms of reduced makespan and improved resource utilization. The proposed algorithm (WSSLVA) identifies the order in which the task to be scheduled based on the previous work proposed in Double Min-Min Credit Worth Task Scheduling Algorithm (DMMCWTS) and the tasks specified in the Worth Set (WS) is scheduled based on the Scheduling Limit Value (SLV). The large number of tasks, that is either a more number of tasks that take less execution time or tasks that takes more time for execution are allocated to resources that has the highest processing capacity. To further reduce makespan, and better resource utilization, in stage 2, the proposed algorithm (WSSLVA) calculates the mean completion time of the resources and reschedules the tasks whose completion time greater than the mean completion time (MeanCT) to the suitable resources by considering the Scheduling Limit value of the resources with reduced makespan.

The WSSLVA algorithm consists of two stages: a ordering stage and an allocating stage.

1. Ordering stage: Firstly, the information about resource availability and task requirements is considered. Secondly, the credit value and the worth value of the task is calculated as proposed in the previous work (DMMCWTS) and the task schedule order is maintained in WS (Worth Set).

2. Scheduling stage: In this stage, the proposed algorithm (WSSLVA) calculates the Scheduling Limit Value (SLV) of the scheduling model.

Equation (1) calculates the total execution time of the task

$$TET = \sum_{i=1, j=1}^{i=n, j=m} ETC_{ij}$$



Equation (2) calculates the Scheduling Limit Value (SLV) of the task

$$SLV = \frac{TET}{\text{Number of resources} * \text{Number of tasks}} \quad (2)$$

The tasks listed in the Worth Set (WS) determine the order of the tasks to be assigned to the resource until the completion time of the tasks is less than the Scheduling Limit Value of the resources. Otherwise, the remaining tasks are scheduled to the next resource and so on. The completion time of each resource is calculated. Sum of the completion time of all the available resources is calculated. The average of the resources completion time is computed. The tasks whose completion time greater than the mean completion time are identified and are rescheduled to the resources whose completion time is minimum compared to that of the completion time of the previously allocated resource.

Equation (3) finds the completion time of each resource is

$$CT_j = \sum_{i=1}^k ETC_{ij} \quad (3)$$

where k – represents the number of tasks

The pseudocode for finding Task Worth (TW) for each task t_i is given below:

Algorithm Task Worth

```

Fix a unique random value (URVi) for each task  $t_i$  in the range 1 to n
for all submitted task in the metatask Mt
    Compute  $UVC_i = URV_i / d_i$ 
endfor
for each task  $t_i$  in the metatask Mt
    Compute  $TW_i = \text{Credit Value}_i * UVC_i$ 
endfor
Order the task in the Worth Set WS in the ascending order of  $TW_i$ 
    
```

scheduled on the resource_j

Equation (4) computes the sum of the completion time of the resources

$$TotalCT = \sum_{j=1}^m CT_j \quad (4)$$

Equation (5) calculates the mean completion time

$$MeanCT = TotalCT / Total Resources$$

C. Worth Set Scheduling Limit Value Algorithm (WSSLVA): Ordering stage:

I. HELPFUL HINTS

your figure has two parts, include the labels “(a)”

Scheduling Stage:

```

Worth Set Scheduling Limit Value Algorithm (WSSLVA):
for all tasks  $T_i$ 
    for all resources  $R_j$ 
        compute  $TET += ETC_{ij}$ 
     $SLV = TET / (\text{Number of tasks} * \text{Number of resources})$ 
do until all tasks in the Worth Set WS are mapped
for all tasks  $T_i$  in the Worth Set WS
for all resources  $R_j$ 
    compute  $T = ETC_{ij} + RT_i$ 
    if ( $T < SLV$ )
        update  $RT_i = RT_i + ETC_{ij}$ 
        update  $CT_i = CT_i + ETC_{ij}$ 
        assign  $CTI_i = T$ 
    end if
end for
end for
assign  $k = 0$ 
for all resources  $R_j$ 
    if  $CT_j > 0$ 
        compute  $X = X + CT_j$ 
    end if
end for
 $k++$ 
compute  $MeanCT = X / k$ 
for all tasks  $T_i$ 
    
```

```

    if  $CTI_i > MeanCT$ 
        assign task  $T_i$  to the resource  $R_j$  that gives the minimum completion time compared to that of the previously scheduled resources
        update ready time and completion time of newly assigned resource
        update ready time and completion time of previously assigned resource
        update completion time of task  $T_i$ 
    end if
end for
Compute  $makespan = \max(CTI_i)$  for all  $i$ 
    
```

Table – I: ETC Matrix-8X8 matrix for consistent, high task, high resource heterogeneity

Tasks	R1	R2	R3	R4	R5	R6	R7	R8
t1	25,137.5	52,468.0	150,206.8	289,992.5	392,348.2	399,562.1	441,485.5	518,283.1
t2	30,802.6	42,744.5	49,578.3	50,575.6	58,268.1	58,987.9	85,213.2	87,893.0
t3	242,727.1	661,498.5	796,048.1	817,745.8	915,235.9	925,875.6	978,057.6	1,017,448.1
t4	68,050.1	303,515.9	324,093.1	643,133.7	841,877.3	856,312.9	861,314.8	978,066.3
t5	6,480.2	42,396.7	98,105.4	166,346.8	240,319.5	782,658.5	871,532.6	1,203,339.8
t6	175,953.8	210,341.9	261,825.0	306,034.2	393,292.2	412,085.4	483,691.9	515,645.9
t7	116,821.4	240,577.6	241,127.9	406,791.4	1,108,758.0	1,246,430.8	1,393,067.0	1,587,743.1
t8	36,760.6	111,631.5	150,926.0	221,390.0	259,491.1	383,709.7	442,605.7	520,276.8

I. EXPERIMENTAL RESULTS

A. Illustration

To illustrate how the proposed algorithm Worth Set Scheduling Limit Value Algorithm (WSSLVA) assigns the tasks to the available resources, the sample data consisting of eight tasks of different task length and eight resources of different processing capacity is considered. Consistent Expected Time to Compute (ETC) Matrix Model for high task and resource heterogeneity is shown in Table I.

If the number of tasks greater than 9 and less than 100, $dv=100$, if the number of tasks greater than 99 and less than 1000, $dv=1000$ and so on.

From Table 1, among the execution time of eight tasks on eight different resources, the maximum value is represented as MAXET.

$$MAXET = 1587,743.1$$

$$W1 = 793,871.55$$

$$W2 = 529,247.7$$

$$W3 = 1323,119.25$$

$$W4 = 1852,366.95$$

Depending on the task execution time, the task credit value is determined and is shown in Table II.

Table – II: Tasks Credit Value

Task	Credit Value
T1	4
T2	4
T3	3
T4	3
T5	3
T6	4
T7	2
T8	4

A unique random value in the range 1 to the total number of tasks to be scheduled is assigned for each task and is given in Table 3. The task worth value for every task is determined using $TW_i = CreditValue_i * UVC_i$ and is given in Table III.

Table – III: Tasks Worth Value

Tasks	Credit Value	URV	UVC	TW
T1	4	7	0.7	2.8
T2	4	6	0.6	2.4
T3	3	1	0.1	0.3
T4	3	4	0.4	1.2
T5	3	8	0.8	2.4
T6	4	2	0.2	0.8
T7	2	3	0.3	0.6
T8	4	5	0.5	2.0

To achieve efficient mapping, the proposed algorithm reschedules the tasks which are scheduled on the resource whose $CT > MeanCT$.

The tasks selected for rescheduling and the resource to which the tasks are rescheduled are shown in the Table IV.

Table – IV: Tasks Selected for Rescheduling

Tasks	Scheduled on Resource	Rescheduled on resource
T1	R2	R2
T2	R2	R2
T4	R3	R3
T5	R1	R5
T7	R1	R1
T8	R1	R4

The makespan, task-resource scheduled pair produced by Min-min algorithm, proposed algorithm (WSSLVA) is shown in Table V. From the result it is evident that the (WSSLVA) algorithm achieves reduced makespan and better utilization of resource than the existing Min-min algorithm.

Table-V: Comparisons between algorithms in makespan

Algorithm	R1	R2	R3	R4	R5	makespan
Proposed Algorithm (WSSLVA)	T3,T7	T6,T2,T1	T4	T8	T5	359,548.5
Min-min Algorithm	T1,T5,T3,T8,T4	T2,T6	T7			379,155.5

B. Benchmark ETC Matrix Model

Benchmark ETC matrix model by Braun et al[2] consists of 512 *16 matrix model. 512*16 represents the number of tasks is equal to 512 and the number of resources is equal to 16.

The three parameters for algorithm evaluation is task, resource heterogeneity, and consistency Benchmark ETC matrix model.

Instances denoted by u-x-yyzz.k,

u – denotes uniform distribution in ETC matrix generation

x – denotes consistency like c-consistent, i-inconsistent,

s/pc-semi-consistent or partially consistent ETC matrix model.

An ETC matrix is consistent, whenever a resource r_i executes any tasks t_i faster than resource r_j , then resource r_i executes all tasks faster than r_j . An ETC matrix is inconsistent, resource r_i may be faster than resource r_j for executing some tasks and slower for others. An ETC matrix is partially-consistent if it includes a consistent sub-matrix. yy – task heterogeneity, which means the execution time for a task t_i on all resources is not the same, hi -high, lo -low zz – resource heterogeneity, which means the execution time of all the tasks on a resource r_j is different, hi -high, lo -low

The twelve instances are classified into three groups of four instances and shown in Table VI.

Table – VI: Expected Time to Compute (ETC) Matrix Model

Consistency	Heterogeneity			
	Task (High)		Task (Low)	
	Resource (High)	Resource (Low)	Resource (High)	Resource (Low)
Consistent	u-c-hihi-0	u-c-hilo-0	u-c-lohi-0	u-c-lolo-0
Inconsistent	u-i-hihi-0	u-i-hilo-0	u-i-lohi-0	u-i-lolo-0
Partially-consistent	u-pc-hihi-0	u-pc-hilo-0	u-pc-lohi-0	u-pc-lolo-0

C. Scheduling Parameters

The scheduling parameter, makespan is termed as $makespan = \max(CTT_i), 1 \leq i \leq n$.

Fig. 1 shows that the Worth Set Scheduling List Value Algorithm (WSSLVA) provides better makespan than Min-min Heuristic Scheduling Algorithm. Fig. 2 shows the makespan values for High Task Machine Heterogeneity. Fig. 3 shows the makespan values for High Task Low Machine Heterogeneity. Fig. 4 shows the makespan values for Low Task High Machine Heterogeneity. Fig. 5 shows the makespan values for Low Task Machine Heterogeneity. The figure shows the various instances represented for consistent ETC matrix, inconsistent ETC matrix, semi-consistent or partially consistent ETC matrix model. It is evident from the Fig. 1,2,3,4 and 5, the algorithm (WSSLVA) results in reduced makespan value than the existing Min-min.

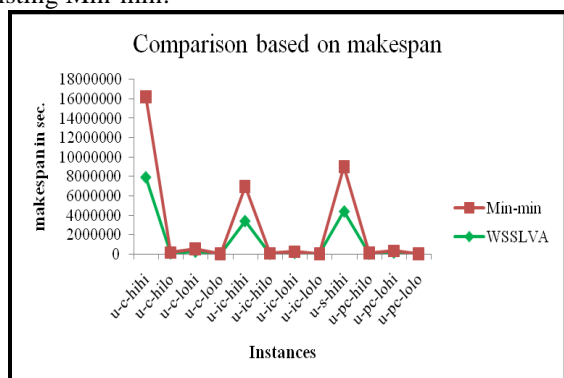


Fig.1. makespan values

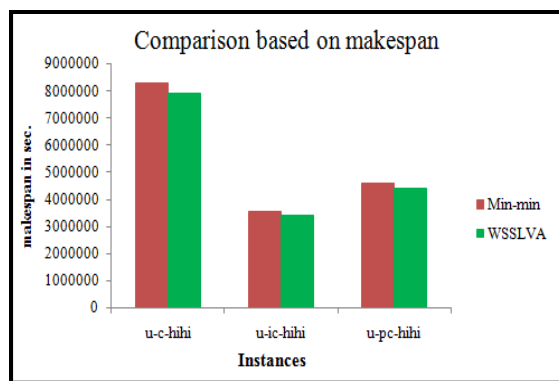


Fig. 2. makespan values for High Task Machine Heterogeneity

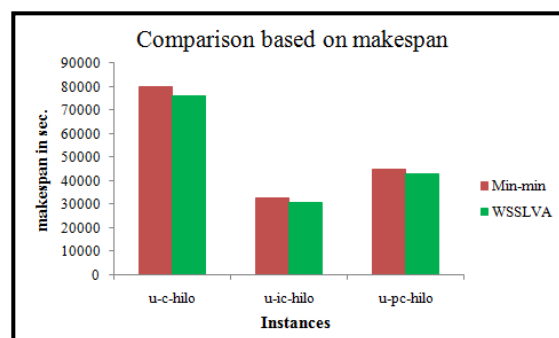


Fig. 3. makespan values for High Task Low Machine Heterogeneity

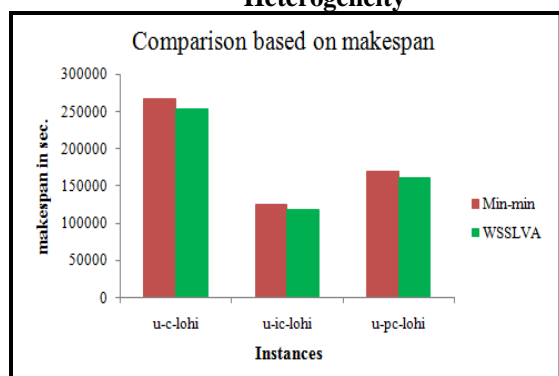


Fig. 4. makespan values for Low Task High Machine Heterogeneity

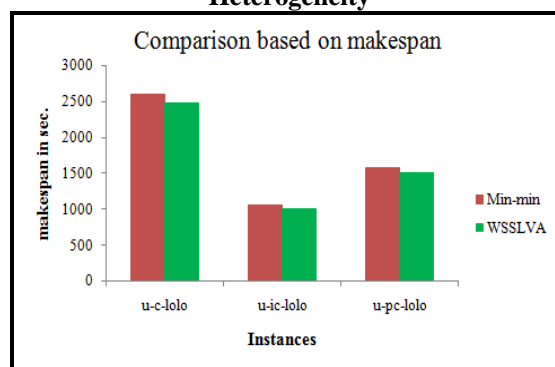


Fig. 5. makespan values for Low Task Machine Heterogeneity

IV. CONCLUSION AND FUTURE DIRECTIONS

A Worth Set Scheduling List Value (WSSLVA) Algorithm is proposed in this paper. The proposed (WSSLVA) algorithm determines the task schedule order from the Double Min-Min Credit Worth Task Scheduling Algorithm (DMMCWTTSA) as specified in the previous work. The tasks schedule list is maintained in the Worth Set (WS) and scheduled depending on the Scheduling Limit Value (SLV). The large number of tasks, that is either a more number of tasks that take less execution time or tasks that takes more time for execution are allocated to resources that has the highest processing capacity. To further reduce makespan, and improve resource utilization, the proposed algorithm (WSSLVA) further calculates the mean completion time of the resources and reschedules the tasks whose completion time greater than the mean completion time (MeanCT) to the suitable resources within the Scheduling Limit value of the resources. WSSLVA algorithm and existing Min-min algorithm examined through Braun et. al.[1] benchmark ETC matrix simulation model. From the results, it is evident that the proposed algorithm (WSSLVA) achieves better makespan and resource utilization than the existing Min-min algorithm. The future direction of the algorithm is to design in such a way to schedule data-intensive and dependent jobs.

REFERENCES

1. T.D.Braun, H.J.Siegel, and N.Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", *Journal of Parallel and Distributed Computing* 61, 2001, pp.810-837.
2. G.K.Kamalam, and Dr.V.Murali Bhaskaran,"A New Heuristic Approach:Min-mean Algorithm For Scheduling Meta-Tasks On Heterogeneous Computing Systems", *IJCSNS International Journal of Computer Science and Network Security*, vol.10(1), 2010, pp. 24-31.
3. G.K.Kamalam,"Best Cluster Decentralized Job Scheduling Algorithm for Scheduling Tasks on Heterogeneous Grid Environment", *Australian Journal of Basic and Applied Sciences*, vol.8(18), 2014,pp. 171-177.
4. Dr.G.K.Kamalam, B.Anitha, and V.Manju Barkavi, "Task unique credit system based scheduling algorithm for mapping meta-tasks on heterogeneous grid environment", *International Journal of innovative research in Computer Science and Engineering (IJIRCSE)*, vol.1(13), 2015, pp.13-18.
5. Dr.G.K.Kamalam, "Guaranteed Best Effort QoS Algorithm for Task Scheduling on Heterogeneous Grid Resources", *Asian Journal of Research in Social Sciences and Humanities*, vol.6(7), 2016, pp.400-411.
6. Dr.G.K.Kamalam, "Sort completion time mean task scheduling Algorithm in Decentralized grid environment", *Asian journal of information technology* vol.15 (12), 2016, pp.1957-1963.
7. Dr.G.K.Kamalam, and B.Anitha,"Double Min-min Credit Worth Tasks Scheduling Algorithm for Mapping Meta-tasks on Heterogeneous Grid Environment", *International journal for Modern Trends in Science and Technology*, vol.3(12), 2017, pp.18-24.
8. Dr.G.K.Kamalam, and B.Anitha," Task Credit Score Min-Mean Scheduling Algorithm for Mapping Meta-Tasks on Heterogeneous Grid Environment", *Journal of Computer Aided Parallel Programming*, vol.2(1), 2017, pp.1-19.
9. Dr.G.K.Kamalam, and B.Anitha,"Task Group Scheduling Algorithm for Mapping a Set of Independent Tasks in Each Group Based on QoS onto Heterogeneous Resources in a Distributed Grid Environment", *International Journal of Applied Engineering Research*, vol.14(1),2019, pp. 92-96.
10. Muhanad Tahrir Younis, and Shengxiang Yang, "Hybrid meta-heuristic algorithms for independent job scheduling in grid computing", *Applied Soft Computing*, vol. 72, ,2018, pp.498-517.

11. Paolo Bellavista, et. al, "GAMESH: A grid architecture for scalable monitoring and enhanced dependable job scheduling", *Future Generation Computer Systems*, vol. 71, 2017, pp.192-201.

AUTHORS PROFILE



B.Anitha received BE degree in Computer Science and Engineering from Madras University and M.Tech degree from Anna University. She is currently working as an Assistant Professor in the Department of Information Technology, Kongu Engineering College, Tamil Nadu, India. Her research area is Grid Computing and Data Analysis. She has published papers in national and international journal. She is a life member of Computer Society of India (CSI). Her research interest includes Machine Learning and Deep Learning.



Dr.G.K.Kamalam is working as an Assistant Professor (SLG) in the Department of Information Technology, Kongu Engineering College, Tamil Nadu, and India. Her research area is Grid and Cloud Computing. She has presented 35 papers in national and international conferences and published 25 papers in international journals in her research and other technical areas. Her Area of Interest includes Algorithm Analysis and Optimization Techniques.