An Inherent Approach based on ACO and Tabu Search for Resource Allocation in Grid Environment

Sonal Nagariya
Department of IT,
LNCT, RGPV, BHOPAL

Mahendra Mishra
Department of IT,
LNCT, RGPV, BHOPAL

Manish Shrivastava
Department of IT,
LNCT, RGPV, BHOPAL

ABSTRACT

The persistence of grid environment is to yield a virtual mainframe by using different types of assets available in the large area using broad networks. Grid computing is expanding promptly in the existing era. Grid Computing for controlling and allotment large-scale resources to solve complex technical obstacles. Arrangement is the greatest contemporary theme used to achieve high performance in grid environments. The rotation of the assets floods dishonest announcement infrastructure and availability of the resources carriage a major challenge for proficient resource carriage. Due to fundamental scattering of the assets and their observation, grid provision is a NP-hard exertion. It has been exposed that heuristic algorithms such as tabu search offer optimal performance for resource provision in grid atmosphere. This paper presented a unusual merging algorithm inspired by the tabu search with ant colony optimization .The proposed fusion approach ambitions to breed energetically, an peak skeleton to widespread responsibilities within tiniest time period and also to use assets skillfully.

Keywords

Ant Colony Optimization, Tabu Search, Backfilling, Makespan, Tardiness, Grid Computing, Hybrid algorithm

1. INTRODUCTION

The term Grid Computing initiated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid in Ian Foster 's and Carl Kesselman's formative effort, "The Grid: Blueprint for a new computing infrastructure".[1] Grid Computing represent a topographical circulation of the resources in the distributed computing environment. Grid computing has been acknowledged as an acute technology by industry for initiative computing and commercial computing. Thus, grid computing has concerned the devotion of industrial managers and investors in companies that have become elaborate in grid computing, such as IBM, HP, Intel, and Sun [2].In Grid Computing organization there are numerous kind of challenges such as job development, resources circulation, load balancing and security.

To motivate the resources allocation, resource management system provides the transparency.

Resource allocation systems basically have two type static allocation and dynamic provision scheme. There are many preparation policies for resource allocation such as FCFS, SJF are stationary and backfilling, random search etc. are active.

The physical appearance of the grid for resource allocation should be

1. Distribution, Resource allotment, among dissimilar organizations
2. Heterogeneity of the resources.
3. Multiple administrations
4. Resource coordination
5. Scheduling is highly problematical by the scattered ownership of the grid resources

The Key points under such resource allocation algorithms are studied they should be-

- An estimation of the make span.
- Reduce the various parameters values as compared to the existing algorithm like make span, machine usage and tardiness.
- An approximation of the computational capacity of every job
- The figuring capability of each resource.
- An estimation of the prior load of each one of the resources is required.

Concluded the complete analysis we realize that resource allocation is a NP hard problematic. So there are no one any algorithm that deliver a best solution in all the field .So grouping of the two or more than two algorithm provide a better explanation as compared to the other and heuristic algorithm ACO is offer the optimal resolution with Tabu Search that try to convert NP hard to NP Complete .

1.1 Benefits of Grid Computing

- **Exploiting underutilized resources**: In some administrations, even the server technologies can often be comparatively idle. Grid computing provides an outline for exploiting these underutilized resources and thus has the opportunity of substantially cumulative the efficiency of resource practice. The treating resources are not the solitary ones that may be underutilized. Regularly, machines may have vast unused disk drive ability.

- **Parallel CPU capacity**: A CPU-intensive grid application can be supposed of as many smaller sub jobs, each finishing on a different machine in the grid. To the magnitude that these sub jobs do not need to interconnect with each other, the more mountable the claim suits.

- **Virtual resources and virtual Organizations for Collaboration:** Distribution starts with data in the form of files or databanks. A data grid can magnify data competences in several ways. Data can be duplicated throughout the grid to serve as a backup and can be presented on or near the machines most likely to need the data, in aggregation with advanced arrangement techniques.

- **Access to additional resources**: The grid can support more decorative access, potentially to remote medical diagnostic and robotic surgery tools with two-way interaction from a distance
- **Resource balancing**: A grid delivers brilliant infrastructure for brokering resources. Individual resources can be summarized to determine their obtainability and their ability, and this can be factored into forecast on the grid.
- **Reliability**: Thus, if there is a power or other kind of disappointment at one location, the additional fragments of the grid are not likely to be affected. Grid management software can mechanically resubmit jobs to other machines on the grid when a disappointment is sensed.
- **Management**: The objective to virtualize the assets on the grid and more consistently handle heterogeneous systems will create new opportunities to better manage a larger, more circulated IT organization.

This paper is organized as follows: Section II introduces the problem of resource allocation in different exiting scheduling algorithm in grid networks and survey of related work of our era. Section III and IV present proposed work and algorithm. Section V shows the experimental results and Section VI conclude the overall work .Section VII describe the future enhancement of the proposed algorithm.

2. LITERATURE SURVEY

In Grid Groundwork have variety of unrelated resources associated with each other. The accessible resources are frequently mutable with respect to every existing node. The insistently changing distinguishing of the heterogeneous resources is known as dynamic resources. To make it more critical, the jobs that need to be process by these resources are arriving from different length of time and not knowing by the system until the particular jobs arrive to the system. [3]

Aram Galstyan, et. al., [4] suggested that resource allocation algorithm like Random Selection intermediaries is select randomly with constant opportunity between the resources. So the presentation of this algorithm is very restricted in the case of widely disparate resource aptitudes that we are interested in.

Shengwei YI et. al., [5] advised that in resource allocation algorithm conditions of figuring resource consumption, backfilling scheduling can find the free resource province. Based on the resource necessities of the approaching jobs in the queue, backfilling planning governs which short-and-narrow job can be transmitted to the indolent resources. It can successfully increase the consumption rate of resources at some extent and increase the throughput of the system. There are two traditional backfilling scheduling methods concluding conservative backfilling and an aggressive backfilling called EASY (the Extensible Argonne Scheduling sYstem) backfilling. Conservative backfilling can backfill a trivial job in the back of the line provided that it does not suspension the start time of all the jobs in the front of it. EASY backfilling chooses a lesser job to backfill if it does not rearrangement the start time of the head job in the queue. The resource consumption is improved. The requirement of user appraised run-time of jobs is inferior.

The small jobs will be talented to get more prospects for backfilling. It is more flexible to backfill. Though, the large job may be delayed to route more simply. Shortest Job Backfilled First (SJBF) which used the system-predicted execution time, rather than user assessed runtime of jobs. [5]

A comparison of features between conservative backfilling strategy and aggressive backfilling strategy for corresponding job scheduling is made in with the imprecision of user-estimated runtime the classic backfilling scheduling approaches including conservative backfilling, aggressive backfilling are all enhancements based on FCFS. These approaches surge the resource consumption and decrease the middling waiting time of jobs to some quantity. Combinational backfilling algorithm extended from is able to select a combination of multiple small jobs to development on suitable indolent resources. It can backfill numerous jobs at a era to feat the exploitation of assets in clusters system.

Fred Glover, et. al.,[7] recommended that the term tabu comes from Tongan, a voiced of Polynesia, where it was used by the aborigines of Tonga Island to indicate things that cannot be touched because they are consecrated. According to Webster's Dictionary, the word now also means “a prohibition imposed by social custom as a protective measure" or of something "banned as constituting a risk . The meta-heuristic tactic called tabu search (TS) is histrionically changing our ability to solve problems of practical significance. TS amount the realms of source planning, financial analysis, scheduling, space planning, energy distribution, molecular engineering, logistics, and pattern classification, flexible manufacturing, waste management, mineral survey, biomedical analysis and scores of others.

Fatos Xhafa, et. al., [6] proposed that the Outline for Tabu Search Algorithm

1. Compute an initial solution;
2. Reset the tabu and aspiration conditions;
3. While not termination-condition do
4. Generate a subset of solutions that do not violate the tabu conditions or hold the aspiration criteria;
5. Choose the best solution with respect to the cost function;
6. If improvement in the solution then
7. Update the recency and frequency;
8. If (intensification condition) then Perform intensification procedure;
9. If (diversification condition) then Perform diversification procedures;
10. End while
11. Return solution;

It is conclude that the total tardiness strongly be determined by on the number of available machines. Total tardiness develops higher when the quantity of machine small for scheduling technique. [8]

Vittorio Maniezzo, et. al., [9] advised that Ant Colony Optimization (ACO) is an example for designing Meta heuristic procedures for combinatorial optimization problems. The first algorithm which can be categorized within this agenda was obtainable in 1991. The important characteristic of ACO algorithms is the permutation of a priori statistics about the structure of a promising solution with a posteriori notifies action about the structure of earlier attained good solutions.

A general overview of the proposed methodology for solving the problem is firstly initializing the value and now starts the making of assembly. firstly we discovery the fitness function simulation value for local pheromone update ,and set the best local update as a global update .These steps perform repeatedly till the process not stop.

Przemysław Korytkowski, et. al.,[10] recommended that proper organization of the distribution of orders in the system can improve the effectiveness by several percent. Optimization of the job processing order does not involve the need for additional speculation in machinery or equipment. For all presentation criteria, the ACO procedure converged and gave, in an acceptable time, good results for the scheduling problem.

Due to the appearance of Heuristic in Grid Resource Scheduling provide the following features:
1. Relaxed job control and expand allocation efficiency
2. Expand resource consumption
3. Decrease makespan
4. Reduce response time
5. Improved workload balancing:
6. Service reliability
7. Decrease fairness abnormality
8. Great throughput

Discovery of optimum plans for such an atmosphere is an NP-hard problem, and so heuristic tactics must be used. Throughout all the studies of paper we pragmatic that an ant colony optimization (ACO) algorithm, when pooled with tabu search, can treasure shorter schedules on benchmark problems than other techniques found in the literature in different area such as job scheduling, power distribution .ACO is used for global and tabu search is used for local search and combination of both ACO and tabu search provide optimal solution.

3. PROPOSED WORK

In this paper we recommend a new hybrid resources allocation algorithm which associations of Tabu search with ACO. **"The advised merging technique assists to find the optimal makespan time comprising the solution for blind alley problem.** "Assured Trials the convenience of the recommended algorithm will be paralleled with conventional resources allocation algorithm such as Backfilling, Tabu search etc. algorithms.

4. PROPOSED METHOD

In our dissertation work we combine Tabu search algorithm with Ant Colony optimization algorithm to find the optimal solution in terms of various parameters and also remove closed looping problem. For this dissertation work here we represent an algorithm:

```
1) Initialization: fixed the algorithm constraints.
2) Generation of novel nest (assessment):
  // use the tabu-search method to generate a new nest.
        2.1 Generate preliminary situation
        2.2 WHILE the stopping is not met DO
            2.2.1 Identify Locality set
            2.2.2 Identify Tabu set
            2.2.3 Identify Applicant set
        2.3 Choose the best
        2.4 Memories s' if it progresses the previous best known solution
            Go to step 2.2
          END WHILE
3) Manipulation:
      3.1 Fitness function appraisal (simulation)
      3.2 Local pheromone apprise
 (If current ant quantity ≤ total quantity of ants, go to step 2)
      3.3   Global pheromone apprise
      3.4   Stop condition
 (If current recurrence ≤ total number of recurrences, then
            Go to step 2)
```

During our algorithm we compute some values, so the correlated principles of these values are-

- For fitness function

$$f(s) = \frac{1}{ms(s)}$$

ms(s) = makespan of the solution

- For updating the pheromone trail

$$\tau(i,j) = \begin{cases} \rho.\tau(i,j) + \dfrac{f(sbest\ )}{f(sgb)} & if\ job\ i\ is\ alloted\ to\ processor\ j\ in\ sbest \\ \rho.\tau(i,j) & otherwise \end{cases}$$

$\tau(i,j)$= pheromone value of the job i to the processor j,
ρ = relative parameter

- The heuristic

$$\eta\,(j) = \frac{1}{ct(j,P^{j}_{best})}$$

ct(j, P^{j}_{best})= completion time of job j for best processor p

- For building a solution

$$prob(j) = \frac{\left[\tau\left(j,P^{j}_{best}\right)\right]^{\alpha}[\eta(j)]^{\beta}}{\sum_{i=1}^{n}\left[\tau\left(i,P^{i}_{best}\right)\right]^{\alpha}[\eta(i)]^{\beta}}$$

α & β= relative parameters

5. EXPERIMENTAL SETUP AND RESULT

The experimental simulation is designed and implemented based on grid simulation architecture Alea 3.2 beta version which is operate on net beans environment. Grid features is provided by GridSim 5.0 version .Experimental parameters are shown in (See Table 1, Table 2) and the different results related to the various parameter is shown in (see Table 3).This algorithm has many parameters as mentioned in algorithm in section V which will be used in the implementation of this Hybrid algorithm. These values are defined on the basis of our literature survey and they are given below.

Table1. Parameter Values

Symbol	Value	Symbol	Value
α	10	*numAnts*	10
β	10	τ_{min}	0.01
Γ	0-1	ρ	0.75

Table 2. Simulation Parameter

Parameter Name	Value
Total Number of Jobs	3000
Number of Cluster	07
Number of CPU per cluster	80
RAM	517018.0 MB
Baud Rate	10000
CPU Speed	1GHz
Resource allocation algorithm	ACO with Tabu Search
Starting Pheromone value	0.1

Table 3. Simulation Result

Parameter Name	Value
submit	3000
completed	3000
killed	0.0
Response time	70712.87
Run time	0.51
makespan	5749264
Weigh usage	54.49
Tardiness	5141.66

In our Dissertation work, simulation is implemented on Alea simulator, which provide the graphs related with the various parameters obtained during the execution phase.

Average Machine Usage

See Figure 5.1 and 5.2 shows the Number of hours machine and see figure 5.3 and 5.4 cluster usage in a day in proposed hybrid optimization algorithm and Tabu search respectively.

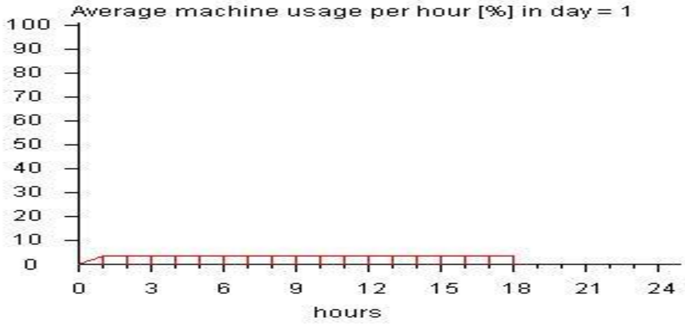

Figure: 5.1 Number of hours machine for proposed algorithm

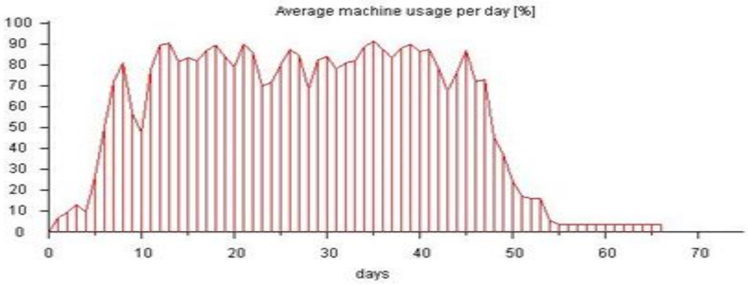

Figure: 5.2 Number of hours machine for tabu search algorithm

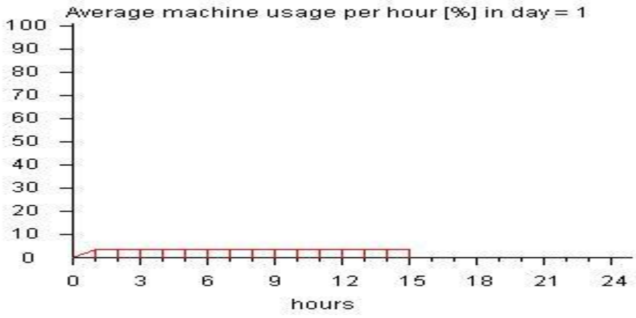

Figure 5.3 cluster usage in a day for proposed algorithm

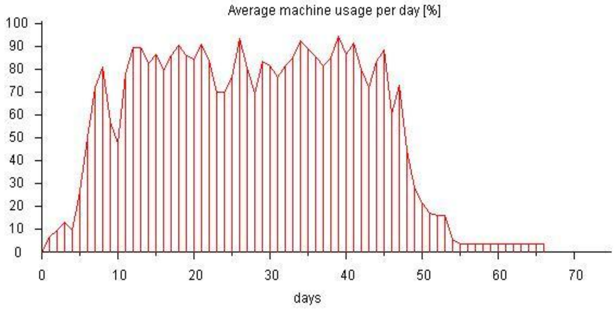

Figure 5.4 cluster usage in a day for tabu search algorithm

Number of Waiting and Running Jobs

See Figure 5.5 and 5.6 shows the number of waiting and running jobs over the days in proposed hybrid optimization algorithm and Tabu search respectively

Number of Requested and Used CPU's

See Figure 5.7 and 5.8 shows the number of Requested, used and available CPU's in all seven clusters over the days in proposed hybrid optimization algorithm and Tabu search respectively.

Simulation results are show in table 3 and comparison are shown in table 4, it is clear that proposed method performs well compare to the some existing resource allocation algorithms. Simulations are performed using Alea 3.1 beta version.

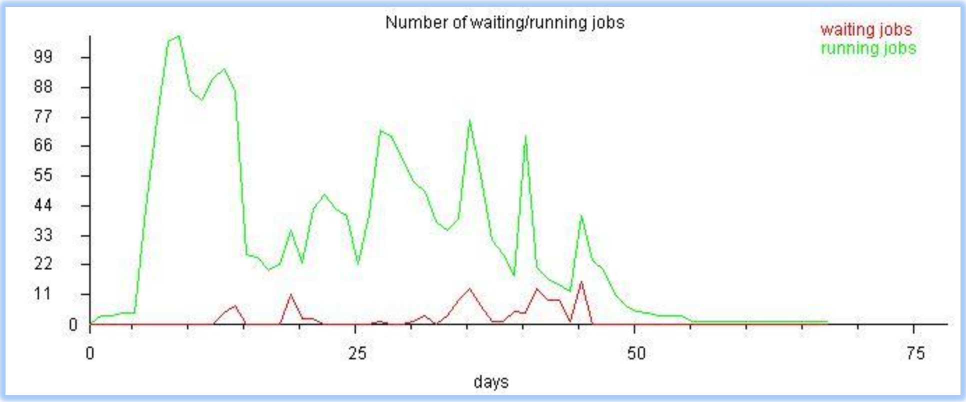

Figure 5.5 Number of waiting and running jobs for proposed algorithm

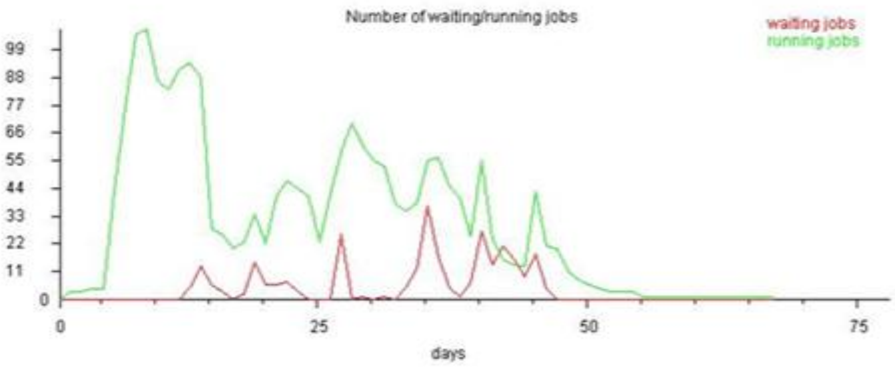

Figure 5.6 Number of waiting and running jobs for tabu search algorithm

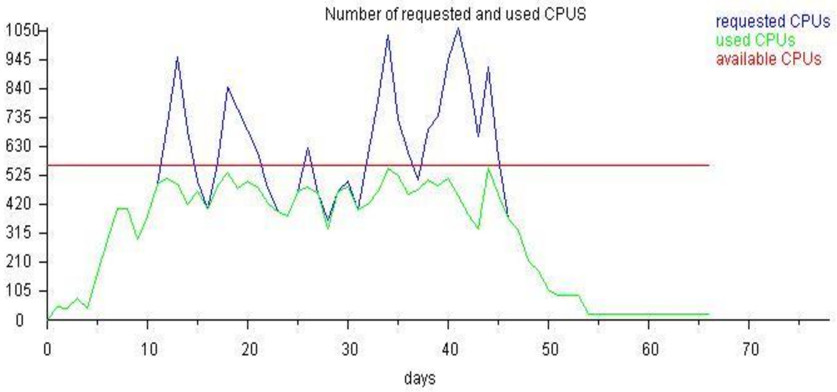

Figure: 5.7 Number of Requested, used and available CPU's for proposed algorithm

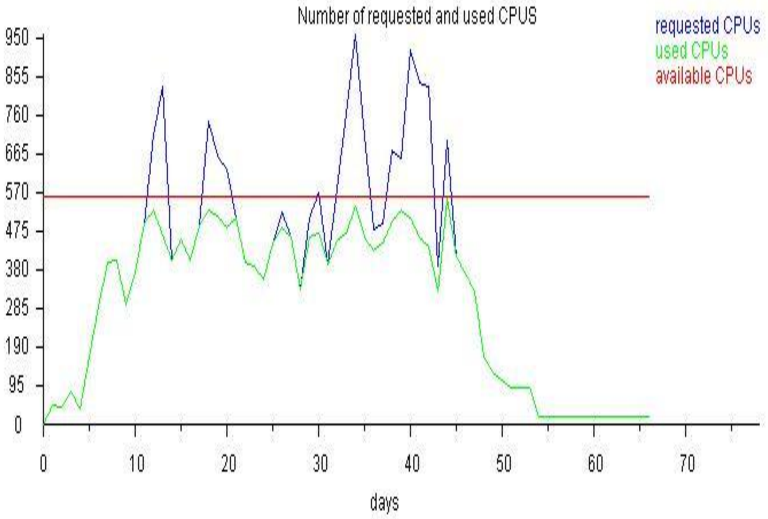

Figure: 5.8 Number of Requested, used and available CPU's for tabu search algorithm

Table 4. Result Summary

Parameters	Algorithm	TABU + ACO	TABU	RANDOM SEARCH	BACK FILLING
MAKESPAN TIME		5749264	5761063	5757129	5749694
NET WAITING TIME		0.43394590363	0.169918116371	0.021924139197	0.10349029374
MACHINE USAGE		15 hours	18 hours	17 hours	15 hours
RESPONSE TIME		70712.87	66020.96	65637.25	65921.92
WEIGHT USAGE		54.38	54.38	54.42	54.49
RUN TIME		0.53	11.15	29.54	0.45

6. CONCLUSION

On the basis of all the study and experimental results proposed method is based on Tabu with ACO resources allocation is an optimal method. The hybrid resource allocation algorithm labeled here reliably find better agendas for several standard problems than other techniques found in the literature, In this policy makespan time is reduced as compared to the some exiting algorithms. Suggested method affords better performance as compared to the other existing algorithms in term of machine usage and makespan time .The hybrid method remove the blind alley problem of ACO and also reduces power consumption and cost.

7. FUTURE WORK

The hybrid resource allocation procedure pronounced at this juncture consistently find better schedules for several target problems than other skills found in the literature, and it seems a brilliant tactic to scheduling in grid atmospheres. There is, how-ever, much scope for advance investigation. More effort could be permitted out with the algorithm described here, for example examining different constraint locations, ant explanation building practices, or dissimilar local search approaches, and also challenging it in a additional genuine surroundings. In broader terms we feel that exploring the use of ACO strategies in different forms of grid scheduling, such as preparation of jobs with precedence restrictions or in dynamic environments might also be productive.

8. REFERENCES

[1] Avijit Bhowmick, Nirmalya Mukhopadhyay, Abhishek Bandyopadhyay, “A Proposed Robust Authentication Approach for Secure Data Transmission in Grid Computing Environment”, International Journal of Computer Applications (0975 – 8887) ,Volume 50– No.22, July 2012.

[2] Zahra Pooranian, Mohammad Shojafar, Jemal H. Abawajy, and Mukesh”, GLOA: A New Job Scheduling Algorithm for Grid Computing”, International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 2, Nº 1.

[3] Zafril Rizal M Azmi, Kamalrulnizam Abu Bakar, Mohd Shahir Shamsir, Wan Nurulsafawati Wan Manan, Abdul Hanan Abdullah, “Scheduling Grid Jobs Using Priority Rule Algorithms and Gap Filling Techniques”,International Journal of Advanced Science and Technology Vol. 37, December, 2011.

[4] Aram Galstyan, Karl Czajkowski and Kristina Lerman ,”Resource Allocation in the Grid with Learning Agents” .Journal of Grid Computing 2005, Volume 3,pp. 91–100 .

[5] Shengwei YI, Zhichao WANG, Shilong MA, Zhanbin CHE, Yonggang HUANG , Xin CHEN ,” An Effective Algorithm of Jobs Scheduling in Clusters “ , Journal of Computational Information Systems, Volume 6:10 ,2010 , pp.3163-3171 .

[6] Fatos Xhafa, Javier Carretero, “A TABU SEARCH ALGORITHM FOR SCHEDULING INDEPENDENT JOBS IN COMPUTATIONAL GRIDS”, Computing and Informatics, Vol. 28, 2009, 1001–1014, 2009-Mar-2

[7] Fred Glover , Manuel Laguna, TABU SEARCH,

[8] Mohd Kamir Yusof ,Kampus Gong Badak ,Muhamad Azahar Stapa ,” Achieving of Tabu Search Algorithm for Scheduling Technique in Grid Computing Using GridSim Simulation Tool: Multiple Jobs on Limited Resource, “ International Journal of Grid and Distributed Computing Grid and Distributed Computing, Vol. 3, No. 4, December No. 4, 2010 .

[9] Vittorio Maniezzo,Luca Maria Gambardella, Fabio de Luigi,” Ant Colony Optimization”.

[10] Przemysław Korytkowski & Szymon Rymaszewski & Tomasz Wiśniewski ,“ Ant colony optimization for job shop scheduling using multi-attribute dispatching rules”, Springerlink.com.