ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Genetic Algorithm Based Artificial Bee Colony Algorithm for Grid Scheduling

Dr.D.Ramyachitra[1], Ms.P.Suganthi[2]

Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore, India[1]

Assistant Professor, Department of Computer Applications, Sudharsan Engineering College, Pudukkottai, India[2]

Abstract: Grid computing is an emerging technology in distributed systems that provide a high performance computing platform to solve larger scale applications by coordinating and sharing computational power, data storage and network resources. All Grid resources are heterogeneous. A grid integrates and coordinates resources and users of different administrative domains inside the same company or in different countries. Task scheduling is one of the key research areas in grid computing. Scheduling is the NP Complete problem which allocates resources for the incoming tasks to the resource broker. Scheduling problem may become tedious when the resources are dynamic and the size of Grid in increased. The goal of scheduling is to achieve highest possible system throughput and to match the application's need with the available computing resources. This paper primarily focuses on scheduling the tasks to the available resources in the Grid environment. The resource allocation process is based on Artificial Bee Colony (ABC) algorithm. The objective of this algorithm is to generate optimal solution dynamically. By this scheduling, the task is completed in minimum time and uses the available resources in efficient manner. In proposed method Single Shift Neighborhood (SSN), Double Shift Neighborhood (DSN), Ejection Chain Neighborhood (ECN) and Genetic Algorithm (GA) techniques are applied for migration of jobs to the resources and the result are compared based on makespan and resource utilization. The result shows that Genetic Algorithm technique will give better results when compared to Single Shift and Double Shift Neighborhoods. This heuristic algorithm provides an optimal task scheduling in all heterogeneous computing environments.

Keywords: Grid Computing, Task Scheduling, Heuristic Algorithms, Makespan, ABC, SSN, DSN, ECN, GA.

I. INTRODUCTION

Grid Computing is a term referring to the combination of computer resources from multiple administrative domains to reach common goal. Grid computing provides a high performance computing platform to solve large scale applications by coordinating and sharing computational power, data storage and network resources across dynamic and geographically dispersed organizations.

Grid Computing is an infrastructure that enables the integrated collaborative use of high-end computers, networks, databases and scientific instruments owned and managed by multiple organizations [1]. Grid applications often involve large amount of data and/or computing and often require secure resources sharing across organizational boundaries. Grid computing links potentially vast numbers of computational resources such as computers, storage devices, and scientific instruments that are distributed over a network to solve complex, massive, computation-intensive problems. The size of a grid may vary from small confined such as a network of computer workstations within a corporation to large such as public collaborations across many companies and networks [2]. The goal of grid computing is high performance, achieved by using spare computer power at little or no cost.

Grid Computing Supports Layered architecture which is a technique used in designing computer software, hardware, and communications in which system or network components are isolated in layers so that changes can be made in one layer without affecting the others. To enable interoperability of heterogeneous distributed resources, a grid computing system is often divided into five layers:
The application layer which is the highest level contains applications that use the lower layers to access distributed resources. The application layer adaptively adjusts user's resource demand based on the current resource conditions [3]. The collective layer is responsible for coordinating distributed resources, such as scheduling a task to analyse data received from a scientific device. The resources layer enables applications to request and share a resource. The connectivity layer carries out reliable and secure network communications between resources. The fabric layer accesses physical resources such as disks. The fabric layer allocates CPU, storage and bandwidth required by the upper layer.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

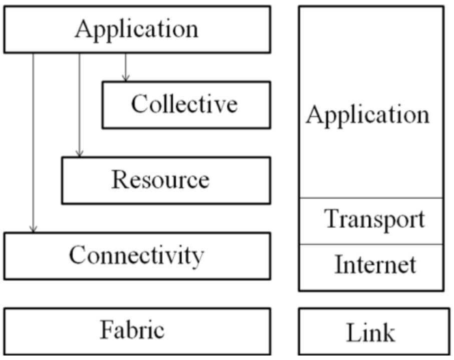

Figure 1: Grid Architecture

Resource scheduling is the process of mapping tasks into specific available physical resources, trying to minimize some cost function specified by the user. Task is a single unit of the application that can be independently assigned to a resource for execution. A task scheduler is a software application that is in charge of unattended background executions, commonly known for historical reasons as batch processing. Effective computation and task scheduling is rapidly becoming one of the main challenges in grid computing and is seen as being vital for its success.

A scheduling algorithm is the method by which processes or data flows are given access to system resources for example processor time, communications bandwidth likewise [4]. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking which executes more than one process at a time and multiplexing which transmits multiple flows simultaneously. This is usually done to load balance a system effectively or achieve a target quality of service [5].

The remaining sections of the papers organized as follows. Section 2 deals with reviews on various literatures on grid computing and describes different algorithms used for Grid Scheduling. In section 3 the detailed methodology of ABC for Grid Scheduling is explained. Section 4 gives the proposed GA based ABC algorithm which is applied for grid scheduling. Section 5 deals with implementation of proposed work which is done in Java. Testing and analyses the result by comparing the results of the proposed method with ABC and section 6 concludes the work.

II. Related Work

Schedulers are types of applications responsible for the management of jobs such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of task, data management, event correlations, and serial-level management capabilities. These schedulers then form a hierarchical structure, with meta-schedulers that form the root and lower level schedulers [15]. These schedulers may be constructed with a local scheduler implementation approach for specific job execution, or another meta-schedulers or a cluster scheduler for parallel executions.

The jobs submitted to Grid Computing schedulers are evaluated based on their service-level requirements, and then allocated to the respective resources for execution. This will involve complex workflow management and data movement activities to occur on a regular basis. There are schedulers that must provide capabilities for areas such as

1) Advanced resource reservation.
2) Serial-level argument validation and enforcement.
3) Job and resource policy management and enforcement for best turnaround times within the allowable budget constraints.
4) Monitoring job execution and status.
5) Rescheduling and corrective actions of partial fail over situations.

Job scheduling is known to be NP-complete [18], therefore the use of heuristics is the de facto approach in order to cope in practice with its difficulty. Thus, the Meta heuristics computing research community has already started to examine this problem. Ritchie and Levine used Single heuristic approaches for the problem in Local Search [19].
QoS parameters such as makespan, resource utilization, and security are found to be crucial for the performance of grids [36], [37], [38]. Besides makespan and cost, some recent researches also take security and reliability into account. Literature [39] presents an iterative scheduling algorithm for a set of independent tasks in computational grids.
Yarkhanan and Dongarra used simulated annealing for grid job scheduling. The Simulated Annealing scheduler is compared to an Ad Hoc Greedy scheduler used in earlier experiments. The Simulated Annealing scheduler exposes some assumptions built into the Ad Hoc scheduler and some problems with the Performance Model being used [20]. Abraham et al. used Tabu Search and Simulated Annealing for scheduling jobs on computational grids [21].

Similarly, GAs for scheduling are addressed in several works by Braun et al. [26], Zomaya and Teh [24], Martino and Mililotti [22], Abraham et al. [21], Page and Naughton [25]. Some hybrid heuristic approaches have also been reported for the problem. Thus, Abraham et al. [21] addressed the hybridization of GA, SA and TS heuristics for dynamic job scheduling on large-scale distributed systems.
In these hybridizations a population based heuristic, such as GAs, is combined with two other Local Search heuristics, such as TS and SA, which deal with only one solution at a time. Ritchie and Levine [27], [28] combined an Ant Colony Optimization algorithm with a TS algorithm for the problem.
Other approaches for the problem include the use of AI techniques used by Cao et al. [29], use of predictive models to schedule jobs at both system and application

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

level by Gao et al. [30], and PSO scheduling by Lei et al. [31]. In [32], scheduling the jobs using ACO method in computational grid system is dealt.

Particle Swarm Optimization (PSO) is another evolutionary optimization technique. It simulates the process of a swarm of insects preying and works well in many global optimal problems [31]. K. Kousalya and P. Balasubramanie used task parcelling based Ant Algorithm for Grid Scheduling [32]. The great performance of ACO algorithms for scheduling problems has been demonstrated in [40] and ant colony system (ACS) algorithm [41], which is one of the ACO algorithms.
D.Jeya Mala, V. Mohan used Artificial Bee Colony Algorithm for Software Testing Approach [34]. Adil Baykasolu, Lale Ozbakir and Pinar Tapkan used Artificial Bee Colony Algorithm and its application to Generalized Assignment Problem for grid scheduling [33]. This paper takes the advantage of the ABC for the grid workflow scheduling problem.
Genetic Algorithm for grid job scheduling is addressed in several works [22], [23]. The benefits of the usage of the Genetic Algorithms to improve the quality of the scheduling are discussed. The result of this paper suggests the usage of local search strategy to improve the convergence when the number of jobs to be considered is big as in real world operation [22]. In this paper, proposed an evolutionary approach to solve Grid job scheduling problem with time uncertainties. The aim of the optimization problem was to find the best CF value at fastest convergent speed without losing stability [23].

III. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) algorithm is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm. In common ABC algorithm is used for optimizing multivariable functions [21]. In ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts. It is assumed that there is only one artificial employed bee for each food source. That is, the number of employed bees in the colony is equal to the number of food sources around the hive. Employed bees go to their food source and come back to hive and dance on this area. The employed bee whose food source has been discarded becomes a scout and starts to search for finding a new food source. Onlookers watch the dances of employed bees and choose food sources depending on dances. The main steps of the algorithm are given below:

• Initialize
• Repeat
1) Place the food sources on the employed bee's memory; Determines neighbour source, then evaluates its nectar amount and dances in the hive;
2) Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source; After choosing a neighbour around that, evaluates its nectar amount; Place the onlooker bees on the food sources in the memory;
3) Send the scouts to the search area for discovering new food sources; The best food source found so far is registered.
• Until (requirements are met)

In ABC which is a population based algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality of the associated solution. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts. Several applications of ABC algorithm includes Biological simulation, Genetic Algorithm Improvement, Continuous Optimization, Travelling Salesman Problem (TSP), Ride-Matching Problem, Dynamic Allocation of Internet Service, Telecommunication Network Routing, Large Scale Precise, Job Shop Scheduling.

IV. GA BASED ABC ALGORITHM

The proposed algorithm is the implementation of ABC algorithm with the key concept of GA technique. This algorithm provides various allocations of tasks to the available resources. The resource matrix is given as input to the algorithm. The probabilities of execution time calculated for each and every machine. Then allocate the task to the resource which executes the task in least time. Likewise all tasks allocated to the resources. After the allocation of tasks to the best available resources the makespan will be calculated. The Single Shift Neighborhood (SSN), Double Shift Neighborhood (DSN) and Ejection Chain Neiborhood (ECN) techniques will be applied on that scheduling and the respective makespans will be calculated. Then the makespans are compared for choosing the best scheduling. On the selected allocation the GA technique will be applied for better performance of the algorithm. The following flow diagram shows the system process:

This algorithm implements various allocation techniques of tasks to the available resources. The resource matrix is given as input to the algorithm. Based on the size of the matrix the random process time assigned to the available resources. The probabilities of execution time calculated for each and every machine.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

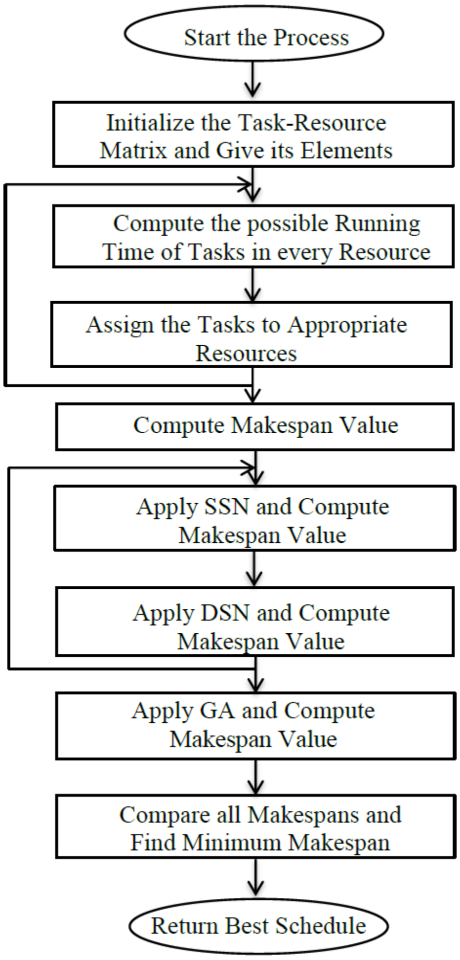

Figure 2: Flow diagram of GA based ABC Algorithm

Then allocate the task to the resource which executes the task in least time. The minimum execution time evaluated by the following formula:

$$\text{Process Time} = \frac{n_i}{\sum_{i=1}^{n} (n_i)}$$

The proposed method has used ABC approaches with the GA technique to best schedule the jobs. Here Single Shift Neighborhood, Double Shift Neighborhood and Ejection Chain Neighborhood are applied to find the minimum makespan. From the selected allocation is applied to GA technique to obtain best optimal solution.

The algorithm can be viewed in the following procedure:

1) Initialization of algorithm: The maximum value of task's process time and resource's capacity are initialized at the beginning of the algorithm.
2) Assignment of Resources: Calculate the process time of each and every job in all available resources. Assign the resource which executes the job at minimum time.
3) Makespan: Calculate the makespan and resource utilization for the above assignment.
4) Single Shift Neighborhood: Apply Single Shift Neighborhood for each task to increase the resource utilization and calculate makespan.
5) Double Shift Neighborhood: Apply Double Shift Neighborhood for each task based on the unused resources.
6) Ejection chain Neighborhood: Apply Ejection Chain Neighborhood and calculate makespan.
7) GA Technique: Among the best schedule get so far applied to GA process to reduce makespan and to increase resource utilization.
8) Terminal test: Find the best optimal solution from the solution construction.

V. PERFORMANCE EVALUATION

The proposed GA based algorithm is developed and the result is compared with ABC. In ABC, we use different techniques like Single Shift Neighborhood (SSN), Double Shift Neighborhood (DSN) and Ejection Chain Neighborhood (ECN) for resource allocation. Then one of the Genetic Algorithm (GA) techniques implemented. Among the above calculated makespans the best schedule is given to the Grid scheduler who has minimum makespan and the maximum resource utilization.

First ABC with minimum makespan value among SSN, DSN, and ECN techniques selected is compared with GA. ABC with SSN shows better result as compared with other techniques. Then the GA produces much better results than SSN. The Comparison is done using various ranges of Task Length, Resource Speed. The range is given below:

Low Task Length and Low Heterogeneity Task Length	: 1000 – 3000 MI;
Low Task Length and High Heterogeneity Task Length	: 1000 – 6000 MI;
High Task Length and Low Heterogeneity Task Length	: 6001 – 9000 MI;
Low Task Length and High Heterogeneity Task Length	: 6001 – 10000 MI;
Low Resource Speed and Low Heterogeneity Resource Speed	: 0 – 3 MIPS;
Low Resource Speed and High Heterogeneity Resource Speed	: 0 – 6 MIPS;
High Resource Speed and Low Heterogeneity Resource Speed	: 6 – 8 MIPS;
High Resource Speed and High Heterogeneity Resource Speed	: 6 – 10 MIPS;

The proposed and existing methods are simulated for various combinations of heterogeneity of task and resources with different ETC Matrix such as 100 x 10, 200 x 20, 300 x 30, 400 x 40 and 500 x 50. Here, 100, 200, 300, 400 and 500 represents number of tasks and 10, 20, 30, 40 and 50 represents number of resources. The result was compared with different QoS such as Makespan, and Resource Utilization. GA based ABC gives less makespan and high resource utilization. GA approximately gives a reduction of 5% for makespan and approximately 5% to 6% increase in resource utilization compared to ABC in most of the cases. The combinations are abbreviated as, for example, l l l h for Low Task Length Low Resource Speed Low Heterogeneity Task Length High Heterogeneity Resource Speed, i.e., Low – l and High – h.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

TABLE I
PERFORMANCE COMPARISON OF PROPOSED ALGORITHM FOR L L L L TYPE OF ETC MATRIX

QoS Constraints	Resource Matrix	ABC	SSN	DSN	ECN	GA
Makespan	100 x 10	200	166	191	150	145
Resource Utilization		94.5	98.5	93	96.66	99.17
Makespan	200 x 10	240	247	274	243	227
Resource Utilization		94.58	98.75	93.33	96.66	99.24
Makespan	300 x 10	410	370	407	345	332
Resource Utilization		94.66	97.5	94.33	97.68	99.56
Makespan	400 x 10	594	513	584	498	483
Resource Utilization		94.86	98.5	96.66	97.87	100
Makespan	500 x 10	532	545	560	530	528
Resource Utilization		94.88	98.57	96.77	97.86	100

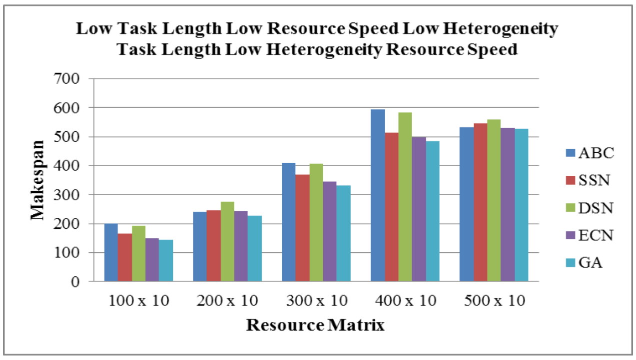

Figure 1. Performance Comparison of ABC techniques with proposed GA for l l l l type of ETC matrix for Makespan

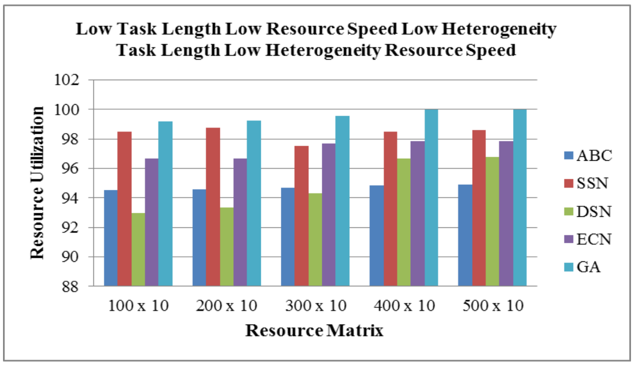

Figure 2. Performance Comparison of ABC techniques with proposed GA for l l l l type of ETC matrix for Resource Utilization

TABLE II
PERFORMANCE COMPARISON OF PROPOSED ALGORITHM FOR L L L H TYPE OF ETC MATRIX

QoS Constraints	Resource Matrix	ABC	SSN	DSN	ECN	GA
Makespan	100 x 20	52	69	108	52	52
Resource Utilization		85.5	91.56	85.11	93.55	95.66
Makespan	200 x 20	80	126	137	78	77
Resource Utilization		85.86	94.53	87.14	93.84	96.62
Makespan	300 x 20	134	128	142	121	117
Resource Utilization		86.5	95.56	86.71	94.58	96.86
Makespan	400 x 20	176	169	174	164	155
Resource Utilization		86.52	96.68	85.28	95.56	96.54
Makespan	500 x 20	216	203	206	198	193
Resource Utilization		87.65	96.97	87.23	96.66	98.34

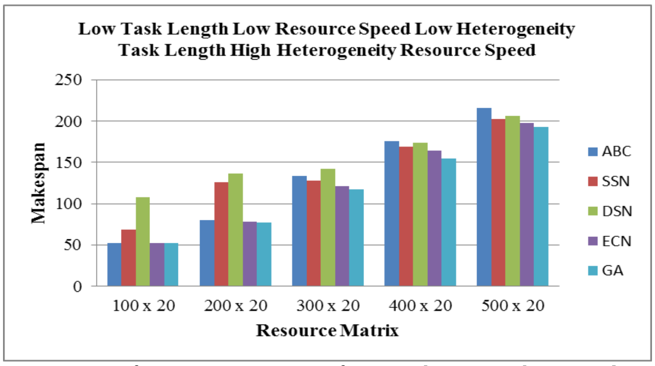

Figure 3. Performance Comparison of ABC techniques with proposed GA for l l l h type of ETC matrix for Makespan

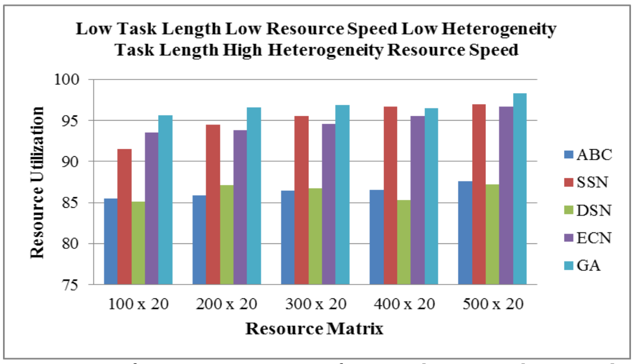

Figure 4. Performance Comparison of ABC techniques with proposed GA for l l l h type of ETC matrix for Resource Utilization

TABLE III
PERFORMANCE COMPARISON OF PROPOSED ALGORITHM FOR H H H L TYPE OF ETC MATRIX

QoS Constraints	Resource Matrix	ABC	SSN	DSN	ECN	GA
Makespan	50 x 20	50	46	49	40	33
Resource Utilization		85	95	85	95	95
Makespan	150 x 20	71	72	68	62	59
Resource Utilization		93.15	98.75	94.28	96.66	100
Makespan	250 x 20	126	114	119	107	101
Resource Utilization		97.52	97.89	97.56	99.35	100
Makespan	250 x 20	129	128	149	138	128
Resource Utilization		97.85	98.24	98.28	100	100
Makespan	350 x 20	145	187	199	160	143
Resource Utilization		98.12	98.46	99.16	100	100

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

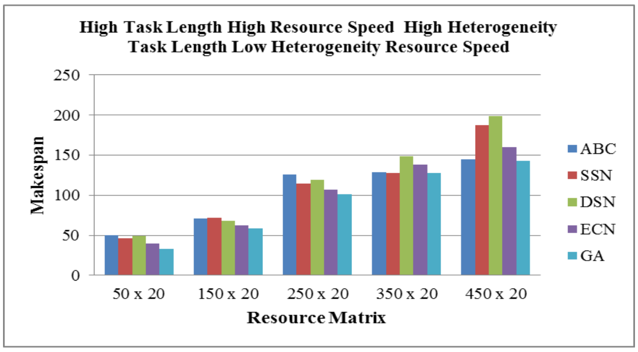

Figure 5. Performance Comparison of ABC techniques with proposed GA for h h h l type of ETC matrix for Makespan

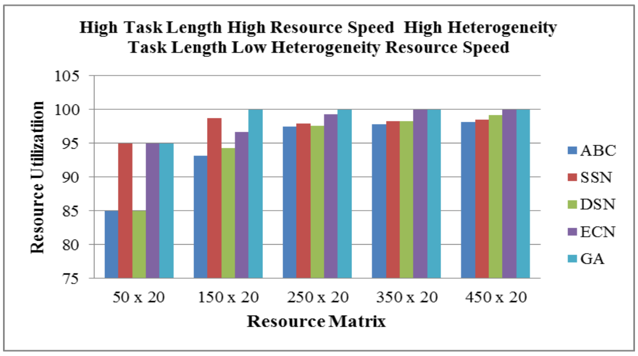

Figure 6. Performance Comparison of ABC techniques with proposed GA for h h h l type of ETC matrix for Resource Utilization

Genetic Algorithm (GA) based ABC gives less makespan and it gives maximum resource utilization. GA approximately shows 5 – 6% less than SSN and DSN for 100 x 10 resource matrix and 1% less for 500 x 10 for makespan. Also, GA shows 5 – 6% higher resource utilization compared to SSN and DSN for 100 x 10 resource matrix and 1% higher resource utilization for 500 x 10 matrixes. In general, the proposed GA based ABC produces better performance in terms of makespan and resource utilization. When the size of resource matrix is increased 100% resources are utilized. For instance, the resource matrix 1024 x 64 always utilizes 100% resources whereas 128 x 8 resource matrix utilizes 90 – 96% of resources. Thus, from the simulation results, it is inferred that GA based ABC shows far better results compared to SSN and DSN for low order matrices compared to high order matrices. The results got from ECN, SSN, DSN and GA for consistent and inconsistent matrices are given below in the following table.

TABLE IV
PERFORMANCE COMPARISON OF SSN, DSN AND ECN WITH GA TECHNIQUE FOR H H H H TYPE OF ETC MATRIX

Scheduling Techniques	Resource Matrix	Makespan	Resource Utilization
SSN	128 x 8	243	97.23
DSN		260	95.67
ECN		270	97.58
GA		239	99.78
SSN	256 x 16	148	98.16
DSN		162	96.58
ECN		163	99.25
GA		137	100
SSN	512 x 32	111	98.26
DSN		94	96.67
ECN		91	99.67
GA		88	100
SSN	1024 x 64	74	99.46
DSN		79	98.23
ECN		74	100
GA		71	100

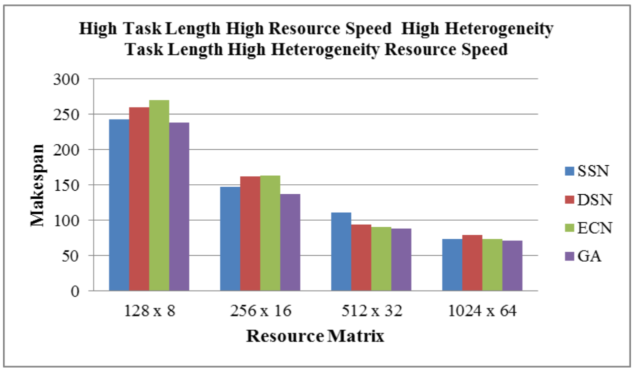

Figure 7. Performance Comparison of SSN, DSN and ECN with GA technique for h h h h type of ETC matrix for Makespan

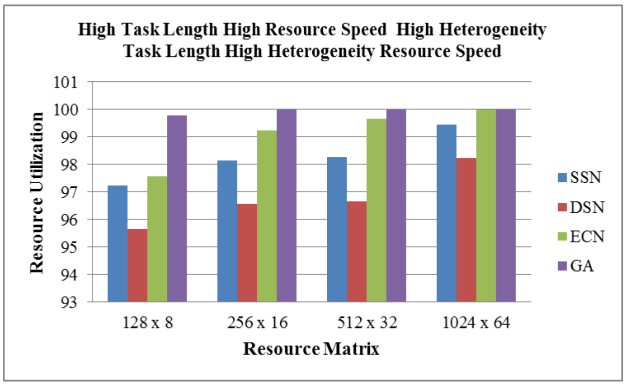

Figure 8. Performance Comparison of SSN, DSN and ECN with GA technique for h h h h type of ETC matrix for Resource Utilization

VI. CONCLUSION

Grid computing presents a new trend in distributed systems and Internet computing for coordinating large-scale heterogeneous resources sharing and problem solving in dynamic organizations. Grid computing allow the challenges such as security, computational economy, uniform access, system management, resource discovery, resource allocation and scheduling, data locality, and network management. Grid scheduling is the process of scheduling applications over Grid resources. Grid Scheduling in the heterogeneous and dynamic nature of grid resources continues to be a tedious task. This paper is proposed with the packages of task scheduling, scheduling algorithms and a hybrid task scheduling algorithms with various factors. In this paper, a heuristic Algorithm is developed based on the QoS makespan and resource utilization. Results show that GA based ABC technique is better than ABC with respect to the QoS constraints such

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

as makespan and resource utilization. The proposed method shows better result for high resource matrix in most of the cases. Future work will focus on inclusion of Deadline and Load balancing the resources and to implement and evaluate different QoS and different Job error ratio.

REFERENCES

[1] Foster, C. Kesselman, and S. Tuecke. "The anatomy of the Grid: Enabling Scalable Virtual Organizations". International Journal of Supercomputing Applications, pp.200-222, Fall.2001.

[2] Foster and C.Kesselman, editors,"The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 1999.

[3] Fatos Xhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", Future Generation Computer Systems 26 (2010), pp.608_621.

[4] T. Casavant and J. Kuhl, "A Taxonomy of Scheduling in General Purpose Distributed Computing Systems", IEEE Trans. on Software Engineering Vol. 14, No. 2, PP 141-154, February 1988.

[5] M. Arora, S. K. Das, R. Biswas, "A Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments", in Proc. of International Conference on Parallel Processing Workshops (ICPPW'02), pp.:499-505, Vancouver, British Columbia Cananda, August 2002.

[6] L. Lee, C. Liang, H. Chang, "An adaptive task scheduling system for Grid Computing", Proceedings of the Sixth IEEE international Conference on Computer and information Technology, CIT'06, 2006, p. 57.

[7] J.Yu, R.Buyya and K.Ramamohanrao, "Workflow Scheduling Algorithms for Grid Computing, Meta. for Scheduling in Distributed Computing Environment", SCI 146, springerlink.com, pp. 173–214, 2008.

[8] Rajkumar Buyya1, Manzur Murshed, David Abramson and Srikumar Venugopal, "Scheduling parameter sweep applications on global Grids: a deadline and budget constrained cost-time optimization algorithm "Journal of Software—Practice & Experience, Volume 35, No. 5 pp:491-512,2005 .

[9] F. Xhafa, J. Carretero, L. Barolli and A. Durresi."Immediate Mode Scheduling in Grid Systems", International Journal of Web and Grid Services, Vol.3 No.2, 219-236, 2007.

[10] Rajeev Wankar"Grid computing with Globus: an overview and research challenges", International Journal of Computer Science and Applications, Techno mathematics Research Foundation Vol. 5, No. 3, pp 56 - 69.

[11] D. Fernandez-Baca(1989) "Allocation Modules to processors in a Distributed System", IEEE Transactions on Software Engineering. Vol.15 (11): Pages 1427-1436.

[12] Buyya R, Abramson D, and Giddy J. "Nimrod/G: An architecture for a resource management and scheduling system in a global computational Grid". Proceedings of the 4th International Conference and Exhibition on High Performance Computing in Asia-Pacific Region (HPC ASIA 2000), Beijing, China, May 2000. IEEE Computer Society Press: Los Alamitos, CA, 2000.

[13] Feitelson DG, Rudolph L (eds.). Proceedings of the 5th IPPS/SPDP'99 Workshop "Job Scheduling Strategies for Parallel Processing" (JSSPP 1999), San Juan, Puerto Rico, April 1999 (Lecture Notes in Computer Science, vol. 1659). Springer: Heidelberg, 1999.

[14] Stefka Fidanova and Mariya Durchova, "Ant Algorithm for Grid Scheduling Problem", Large Scale Computing, Lecture Notes in Computer Science No. 3743, Springer, germany, pp. 405- 412, 2006.

[15] O. H. Ibarra and C. E. Kim, "Heuristic algorithms for scheduling independent tasks on non-identical processors", J. Assoc. Compute. Mach. 24, 2 (Apr. 1977), 280_289.

[16] M. Dorigo, L.M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.

[17] Dr.K.Vivekanandan, D.Ramyachitra, B.Anbu, "Artificial Bee Colony Algorithm For Grid Scheduling", Journal of Convergence Information Technology, Volume6, Number7, July 2011.

[18] Kleinrock, L. MEMO on Grid Computing, University of California, Los Angeles, 1969. http://www.lk.cs.ucla.edu/LK/Bib/REPORT/press.html.

[19] G. Ritchie and J. Levine. ,"A fast, effective local search for scheduling independent jobs in heterogeneous computing environments". Technical report, Centre for Intelligent Systems and their Applications, School of Informatics, University of Edinburgh, 2003.

[20] Yarkhan .A, J. Dongarra, "Experiments with scheduling using simulated annealing in a grid environment", in: 3rd International Workshop on Grid Computing, 2002.

[21] Abraham, R. Buyya, and B. Nath." Nature's heuristics for scheduling jobs on computational Grids". In The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), India, 2000.

[22] V. Di Martino, M. Mililotti, "Sub optimal scheduling in a grid using genetic algorithms", Parallel Computing 30 (2004) 553–565.

[23] D. Liu, Y. Cao, and CGA: "Chaotic Genetic Algorithm for Fuzzy Job Scheduling in Grid Environment", Springer-Verlag Berlin Heidelberg, 2007.

[24] A.Y. Zomaya and Y.H. Teh."Observations on using genetic algorithms for dynamic load-Balancing".IEEE Transactions on Parallel and Distributed Systems, 12(9):899–911, 2001.

[25] J. Page and J. Naughton"Framework for task scheduling in heterogeneous distributed computing using genetic algorithms."Artificial Intelligence Review, 24:415–429, 2005.

[26] H.J. Braun, T. D.and Siegel, N. Beck, L.L. Blni, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D.Theys, and B. Yao."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(6):810–837, 2001.

[27] G. Ritchie. "Static multi-processor scheduling with ant colony optimization & local search".Master's thesis, School of Informatics, University of Edinburgh, 2003.

[28] G. Ritchie and J. Levine." A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments." In 23rd Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2004), 2004.

[29] J. Cao,D.P.Spooner,S.A.Jarvis and G.R.Nudd,"Grid load balancing using intelligent agents," Future Generation Computer Systems, 21(1), 135-149, 2005.

[30] Y. Gao, H. Rong and J.Z. Huang," Adaptive grid job scheduling with genetic algorithms, "Future Generation Computer Systems, 21(1), 151-161, 2005.

[31] Lei Zhang, et al, "A Task Scheduling Algorithm Based on PSO for Grid Computing", International Journal of Computational Intelligence Research, ISSN 0973-1873 Vol.4, 1 (2008), pp. 37–43.

[32] K. Kousalya and P. Balasubramanie," Task Severance and Task Parceling Based Ant Algorithm for Grid Scheduling", International journal of computational cognition (http://www.ijcc.us), vol. 7, no. 4, december 2009.

[33] Adil Baykasolu1, Lale Özbakır2 and Pınar Tapkan," Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem".

[34] D. Jeya Mala , V. Mohan," ABC Tester - Artificial Bee Colony Based Software Test Suite Optimization Approach", International Journal of Software Engineering, IJSE Vol.2 No.2 July 2009.

[35] Xiao-Min Hu, Jun Zhang, and Yun Li," Orthogonal Methods Based Ant Colony Search for Solving Continuous Optimization Problems", journal of computer science and technology 18 jan. 2008.

[36] R. Prodan and T. Fahringer, "Overhead analysis of scientific workflows in grid environments," IEEE Trans. Parallel Distrib. Syst., to be published.

[37] R. Buyya, D. Abramson, and S. Venugopal, "The Grid Economic," Proc.IEEE, vol. 93, no. 3, pp. 698–714, Mar. 2005.

[38] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut, "Quality of service for workflows and web service processes," J. Web Semantics,vol. 1, pp. 281–308, 2004.

[39] L. Chunlin and L. Layuan, "QoS based resource scheduling by computational economy in computational grid," Inf. Process. Lett., vol. 98,pp. 119–126, 2006.

[40] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," IEEE Trans. Evol. Comput.,vol. 6, no. 4, pp. 333–346, Aug. 2002.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

[41] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE Trans. Syst., Man, Cybern. B,Cybern., vol. 26, no. 1, pp. 29–41, Feb. 1996.
[42] Jennifer M. Schopf," A General Architecture for Scheduling on the Grid", Argonne National Laboratory preprint, ANL/MCS-P1000-1002.
[43] Padmanabhan A., "SOG: A Self-Organized Grouping Infrastructure for Grid Resource Discovery," PhD Thesis, University of Iowa,Iowa, 2006.
[44] S. Saeedia , F. Samadzadegan , N. El-Sheimy," Object Extraction From Lidar Data Using An Artificial Swarm Bee Colony Clustering Algorithm", CMRT09. IAPRS, Paris, France, 3-4 September, 2009.
[45] Foster, A. Roy, V. Sander, "A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation", (8th International Workshop on Quality of Service,2000).
[46] Portal Batch System (PBS), http://pbspro.com/.
[47] Silver, http://www.supercluster.org/projects/silver/.
[48] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W.Smith, and S. Tuecke,"Resource Management Architecture for Metacomputing Systems", Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998.
[49] Snell and Clement, "Metascheduling Query and Reservation Interface",Global Grid Forum Schedule working Group WD 2.2, April 2000.
[50] Rajesh Raman, Miron Livny, and Marvin Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing", Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing, July 28-31, 1998.
[51] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, S.Tuecke, "A Directory Service for Configuring High-Performance Distributed Computations", Proc. 6th IEEE Symp on High-Performance Distributed Computing, pg. 365-375, 1997.
[52] Fran Berman, Richard Wolski, Silvia Figueira, Jennifer Schopf, and Gary Shao, "Application-Level Scheduling on Distributed Heterogeneous Networks", Proceedings of Supercomputing 1996.
[53] Ranjan R., Harwood A., and Buyya R., "Peer-to-Peer Based Resource Discovery in Global Grids a Tutorial," Computer Journal of IEEE Communications Surveys and Tutorials, vol. 10,no. 6, pp. 6-33, 2008.
[54] Wei-Neng Chen, Jun Zhang," An Ant Colony Optimization Approach to a Grid workflow Scheduling Problem With Various QoS Requirements", IEEE Transactions on Systems man and cybernetics—Part c: Applications and Reviews, vol. 39, no. 1, January 2009.
[55] Ian Foster, Adriana Iamnitchi," On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing".