

IBCO Based Task Scheduling for E-Learning Environments in Grid

¹A.C. Kaladevi and ²M.V. Srinath

¹Department of Computer Engineering, Sona College of Technology, Salem, Tamil Nadu, India
²Department of Computer Engineering, Mahendra Engineering College, Namakkal, Tamil Nadu, India

Abstract: Grid computing enables synchronization of shared distributed resources and supplement collaboration via virtual learning environment. Server management is essential in this research work with the intention of formulating consistent and harmonized use of ubiquitous and heterogeneous grids. Thus, reducing the server virtualization problems like receding the number of servers for too little research minimizing the high infrastructure requirements and the inadequate flexibility in shared environments. In this study an approach called Inclusive Bee Colony Optimization (IBCO) is developed for scheduling that gathers the learner's requests initially. By estimating the intrinsic attributes of the grid nodes and task intrinsic rate, the observed node cost is acquired. Subsequently the random cost is evaluated following the hypothesis and finally applying the Chi-square distribution for ascertaining the fitness value. Scheduling prolongs according to the fitness function until the stopping criterion is met. Experimental results forecast that server can efficiently schedule the learners request on demand at sustaining moment. Moreover, this research perpetuates the possibility in reducing the cost constraint in time for scheduling the learner's demands.

Key words: Swarm intelligence, scheduling, Chi-square distribution, ACO, BCO, poisson distribution

INTRODUCTION

In present scenario due to the dynamicity, heterogeneity and autonomy of the grid resources, task scheduling and load balancing within the grid systems have become confronting research area. Keeping this concern in mind researchers have derived a mechanism for e-Learning environment with effective task scheduling and load balancing using IBCO in grid. In spite of encroachment in technologies e-Learning is an adept learning mechanism endowed to lessen time, cost, distance factors, etc. The concept of e-Learning is that teaching can be available anywhere any time. This kind of system delivers teaching principles to remote places via synchronous and asynchronous mode without time delay. In addition to teaching, it generates the learner's request, delivers the learner's request, manages the request on demand, etc. An architecture diagram regarding the grid scheduling is portrayed in Fig. 1 which shows that various tasks are given to the distributed nodes in grid environment regarding user requirements in e-Learning process.

To provide pro-active learning environments, demands of server must be congregated. The server must be consistent enough to handle all requests and schedule it providing non trivial service to the learners. Among the

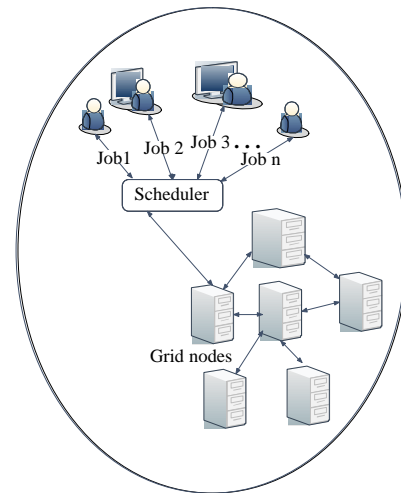


Fig. 1: Architecture for job scheduling in grid environments

assured potential of the large number of distributed resources, effectual scheduling algorithms are enacting an imperative role. Grid computing systems are said to be varying in its characteristics compared with other systems such as Symmetric Multiple Processors (SMPs) and cluster computing. The feasibility of grid environments is

the dynamic sharing of resources at runtime regarding its accessibility of system requirements. Hence, scheduling plays an important role in e-learning environment.

The e-Learning environments server facilitates the sharing and cumulating the jobs with geographically distributed heterogeneous resources that provide an important solution for scheduling constraints. In this study, e-Learning environments are created with the grid framework. The grid scheduler receives the request from the grid users, allocates the request depending on the available resources. The system information has to be acquired from the Grid Information Service (GIS).

The main objective of scheduling includes maximizing the performance by federating computational resources from multiple domains. The performance measures are subjected to the constraints such as resource utilization, cycle time, throughput rate and inventory level. In order to meet the demand terms of current scenario it is better to expose an approach that minimizes the resource utilization by means of affording a minimized makespan.

Scheduling methods are required to address distinct requests from heterogeneous nodes. One of the ancient and lively approaches is the Ant Colony Optimization (ACO) mechanism. These ACO and Bee Colony Optimization (BCO) mechanism has its origin in swarms intelligence. The natural life-cycle of birds and ants is adopted in these approaches. The ACO is called as Meta Heuristic approach since it has a wide range of application areas that have been applied successfully to various combinatorial optimization problems such as scheduling problems, routing or assignment problems.

Colomi *et al.* (1994), Bauer *et al.* (1999), Dorigo and di Caro (1999) and Dorigo *et al.* (1996) have initially proposed this approach in their research work. The ultimate goal of their research is to increase the impact of the influence of various significant parameters for ant colony that exhibits vital characteristics in deriving the optimal solution. The state transition parameters and the pheromone updating parameters are being gained a little insight in this approach.

There were several solutions proposed for ACO mechanism. The Jin and Buyya (2003) have also provided an optimistic solution. Normally in ant colony optimization ants search for the best location of food. On having the communicative behavior of ants the ants in their colony will be co-operating with each other. Using various probabilistic techniques several good solutions were found. In the research ACO mechanism had been proposed for single machine total weighted tardiness problem.

Chong *et al.* (2006) have proposed BCO mechanism. BCO is also an agent based approach as ACO where

low-level agent is the bee. The BCO apparently provides well coordinated interaction, organized team work, work splitting and perfect communication. In this manner, the adduced research aims to explore an evolutionary computation methodology which is dependent on nectar collection in honeybee colonies to job shop scheduling problems. The affirmed methodology is for affording prominent e-Learning environment with flexible task scheduling mechanisms using IBCO in grid network. Depending upon the reliability estimations of distributed grid nodes over software cost, hardware cost and linkage cost the task scheduling is performed. The essence part, critical cost is evaluated for determining the rate of change of the hardware failure intensity and the computational time using poisson distributions. In order to reveal the efficacy of the precise mechanism, some parameters are taken from the benchmark algorithm ACO in the experimental study for comparison. The task scheduling based on these effective manners satisfies the e-Learners by providing appropriate knowledge in an adept way.

Literature review: Recently with the evolution of internet as a global platform for education, the need of efficient task scheduling mechanisms has increased which paves way to produce engrossing research works. The solution for job-shop problem by Applegate and Cook (1991) provided a mechanism to reduce the job completion times of a set of machines by affording heuristic procedures. Addressing the problems on multiple machine scheduling and resource allocation, a genetic algorithm with random keys was demonstrated (Bean, 1994). It has produced attainable offspring for many optimization and scheduling problems. A generic algorithm developed for solving the job-shop scheduling problem with deadlock free manner by Ombuki and Ventresca (2004). The research could be extended with dynamic job-shop scheduling mechanisms in distributed environment. A taboo search algorithm (Nowicki and Smutnicki, 1996) described for minimizing the increasing makespan of job scheduling problem based on specific neighborhood definition, provides base for more researches. The conceits of dynamic programming from (Sutton and Barto, 1998) were referred. A BOINC approach (Berkeley Open Infrastructure for Network Computing) for effective public-resource computing and storage was described by Anderson (2004). While analyzing, optimal disk usage was not effectively prescribed in this research work. Dynamic scheduling algorithm in grid computing environment was given in (Lee *et al.*, 2006) based on DH (Dynamic HEFT) algorithm. Instead of distributing tasks to physical processors directly, the DH mechanism dispatches tasks to multiple

queues. During execution, the DH algorithm persists to dispatch the scheduled tasks to consequent physical processor and determines the transfer rate. There performed rescheduling when the difference between two successive transfer rates is greater than the threshold value. The rescheduling cost could be reduced by efficient methodologies in further research.

A low complexity task scheduling algorithms are presented by Ilavarasan and Thambidurai (2007). It was a three phase approach level sorting consisting of task prioritization and processor-selection. In level-sorting approach, task execution is done in a parallel way. Obviously in task prioritization phase the tasks are being allotted within the prioritization limit. Moreover, insertion based scheduling policy is applied in processor selection phase. In this approach as directed acyclic graphs are intended for processing the linearity may be decreased. There are still more techniques called Heterogeneous Earliest Finish Time (HEFT) and Critical Path on Processor (CPOP) algorithms for grid scheduling which are termed as the benchmark algorithms in scheduling.

A data flow model developed for. Net based grid computing systems due to the demands in acclamation of network computing and distributed computing environments to support transparent fault tolerance and scalable performance (Jin *et al.*, 2007). Since, researchers incorporate the immense concepts of poisson distribution and χ^2 -test functions in the mechanism research are referred from researchers.

Moreover, to meet the demands of scheduling in large grid based environments, Reinforcement Learning (RL) methods have been adopted. The main criteria of RL lie in solving decision making problems (Wu *et al.*, 2011). Following evolved the algorithm of Fair Action Learner (FAL) which applies the policy gradient ascent algorithm (Zhang *et al.*, 2009; Vengerov, 2007).

An Ordinal Sharing Method (OSL) is projected by Jun Wu, Xin Xu, Pengcheng Zhang, Chunming Liu is a Multi Agent based Reinforcement Learning (MARL) Method. In this methodology the decision of job scheduling in MARL framework is changed to decentralize MARL where cooperative control aspect is changed to ordinal sharing. Reliability-driven scheduling algorithm (Tanga *et al.*, 2012) analyzed task reliability. The reliability-driven scheduler and hierarchical scheduler were incorporated to outperform the limitations of related algorithms in avoiding application failures and poor performance. Researchers made an in-depth research in this domain with multidimensional computing resources. A mechanism in (Quezada-Pina *et al.*, 2012) integrated two major tasks of grid, job allocation to sites and local scheduling at each site. Though, the results

revealed robustness and stability by adaptive admissible allocation strategies, the approach had some computational complexities that could be considerably abated from the methodology. However, when compared all the RL based algorithms tend exhibit less convergence rate in learning process. Nevertheless, the proposed approaches have quite strategic approach since the nodes characteristics features are well analyzed and the decision is taken according to the fitness value.

The main contribution of this study is the stack composition of user requests arriving from ports. The bees signifies as the collected jobs. Then, the nodes connected with server are considered flower patches. The foraging bees will be selecting the best site. The experimental result depicts that this algorithm performs excellently well in job scheduling scenarios. Likewise all jobs in the stack are extracted and dispatched to the best sites.

MATERIALS AND METHODS

In this proposal, initially the learners are procured with necessary authentication. The learners will submit their request to the servers. The open grid service architecture is formulated for these e-Learning environments. Let N be the number of task that needs to be scheduled. The master scheduler will be monitoring other systems. On the arrival of service request, the necessary service will be invoked. Every process is divided into subtask for ongoing parallel execution among grid nodes. The primary thing is determining the job allocation. Each individual node will be returning its result to the master scheduler. The master scheduler integrates all the results and passing it to the learners. As researchers have formulated grid based environment, there is a master having monitory access with other nodes. Accordingly, some masters in grid will be selecting more than one node for allocating the same process however it will provide higher cost on allocating the same task to more than one node. Subsequently, an environment consists of grid resources along with a master is originated. Assume each task will be allocated to each node such that only once the task should be executed in a node. Once when the decision is made regarding the nodes the task is allocated and the nodes will be starting its execution with the nodes. Finally, the result is associated with the server. Figure 2 shows the architectural flow of the proposed system.

Node intrinsic aspects: The node intrinsic aspects play an essential function in this scheduling. Usually the node intrinsic aspect is wrapped with three costs, hardware

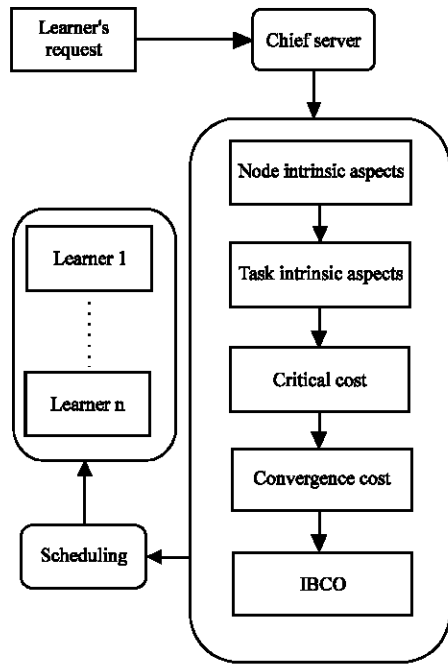


Fig. 2: System architecture diagram

cost (The cost occurred due to the hardware h_k), software cost (cost occurred due to the program vulnerability S_k), link failure (cost occurred in the link l_k) that are encountered for scheduling. Among those failures some would be recoverable that may be caused by users and occurred because of user interrupting normal operation. The hardware failures occurring on grid nodes can be belonging to both the categories of recoverable or unrecoverable declines. Link failures are recoverable because it is highly tough enough to correct the recovering system. Likewise the software failures are said to be recoverable unless the faulty programs are corrected.

There are many node intrinsic aspects available. A foremost criterion for estimation is the computational processing capability of node. Subsequent measurements include the link's failure rate the bandwidth capacity of the link and the velocity rate in hardware failure.

Task intrinsic rate: Subsequent to the node intrinsic aspects the task intrinsic rate is estimated. The task intrinsic rate corresponds to the difficulties encountered in the subtasks. Sometimes the task intrinsic rate increases as it extends to other systems on distributing its requirements to other systems.

On the arrival of all tasks, it is being split into subtask. Each task is allocated to one node. The computational cost for executing one process say (i) in k th node is evaluated:

$$T_k = \frac{T_i}{V_k} \quad (1)$$

The computational time is the time taken to execute the task in a node k. The computational time (T_k) is directly proportional to the task intrinsic rate (T_i) and inversely proportional to the node (k) velocity (speed) which is denoted as V_k .

Critical cost: The peculiarity of poisson distribution is that the probability of capturing number of events occurring in a fixed interval of time. In other words, the events will be occurring in average rate. Exponential distribution is depending on the poisson processes variation.

The scheduling phenomenon comes under the category of poisson distribution with the rate of change of the hardware failure intensity and the computational time. The failure recoverability rate is also a significant factor for assessing the hardware failure. The hardware failure consistency of node k executing subtask (i) is imported as exponential distribution which is given by:

$$P(x) = P_1 + P_2 \quad (2)$$

$$P_1 = e^{-h_k T_{ik}} \quad (3)$$

$$P_2 = e^{-R_k h_k T_{ik}} \quad (4)$$

P_1 = The rate of change when no failures occur

P_2 = Refers to when failures occurred are recoverable

The distribution P_2 is dependent on the number of failure recoverability on a node. The probability distribution is applied along with the Erlang parameters. Finally, the equation for discovering the probability on hardware failure is:

$$P(H)_{ik} = e^{-(1-R_k)h_k T_{ik}} \quad (5)$$

$P(H)_{ik}$ = The hardware failure that occurred on the k th node for task i

R_k = Mean failure rate

In this way, the cost for hardware failure is formulated. It is obvious that the software failure is proportional to the task that is executing on node (k). The software failure intrinsic rate is also measured. Consequently the probability for software failure is predicted apparently by means of exponential function:

$$P(S)_{ik} = e^{-s_i T_{ik}} \quad (6)$$

$$\mu_{ik} = T_i \times C_{ik} + \eta_{ik} \quad (11)$$

Besides hardware and software failures the probability of link failures is also measured stochastically. On behalf of measuring this link cost the data transferred in the link is measured and its link capacity rate is also calculated. The inverse ratio between the link capacity (denoted as c_k) for k th node and data transferred (indicated as d_{ik}) in the link of k th node in other words the data required for executing i th task in node k is deliberated. The Data transferred in the link for executing the subtask i in node k is evaluated by Eq. 7. The predicted parameters are h_k the hardware failure rate and the software failure rate s_k are premeditated per sec. Moreover, the rate of change of average link cost is evaluated as Eq. 8:

$$D_{ik} = \frac{d_{ik}}{c_k} \quad (7)$$

$$P(L)_{ik} = e^{-h_k D_{ik}} \quad (8)$$

The critical cost is further applied in exponential distribution. The critical cost is defined as cost considering hardware failure, software failure and the link failure. It is specified by Eq. 9:

$$C_{ik} = P(H)_{ik} + P(S)_{ik} + P(L)_{ik} \quad (9)$$

Convergence cost: On succeeding the critical cost, there is a possibility of delay which may occur during the process is entered in the system until it gains CPU. The decision is left with the resource management system for allocating the related process with the appropriate nodes. The nodes may be undergoing certain process. Resultantly, the system is assumed as constituted without process conflicts hence, the time elapsed for the new process will be then increased. The elapsed time is denoted as η_{ik} . Moreover, the remaining time to complete a particular process excluding the waiting time is taken into account which is denoted as λ_{ik} . The latency time evaluated is defined as the time taken by i th subtask to entirely complete its execution on node k :

$$LT_{ik} = \eta_{ik} + \lambda_{ik} \quad (10)$$

The task intrinsic rate for each task is evaluated. Similarly, the computing capacity is found for each process (i) on the k th node is represented by C_{ik} . The predicted convergence cost is estimated for each node as:

The predicted values so far evaluated are subjected to replace as the fitness function in Bee's algorithm. The intrinsic nature of sub task is depicted to be (T_i). The inclusive bee colony optimization algorithm is stated as follows:

Inclusive bee colony optimization: The BCO algorithm has its origin from swarm intelligence. Systems will collectively accomplish its task in dynamic environments. Stimulating the regular foraging performance of bees the optimal solution is obtained. Flower patches will be present in the space in a surplus manner. Scout bees are responsive to discover different patches. The bees returning to the hives have to analyze the patches regarding the quality threshold. The other bees in the group will be communicating with each other having the waggle dance.

This approach is practically applied for these scheduling problems. There are some crucial parameters which are still setting. Let us consider the total number of processes (n), number of sites selected (k), the best e sites (e) and finally, the stopping criterion. The salient decisive factor lies within the fitness function. In the approach this is kept according to the χ^2 -test which is used to evaluate the goodness of fit. Based on this the fitness function is symbolized as:

$$fit(n) = \sum_{k=0}^n \frac{(\mu_{ik} - C_{ik})}{C_{ik}} \quad (12)$$

Moreover, the fitness function as per χ^2 -test is slightly modified for the purpose of precisions. The fitness function is therefore obtained by Eq. 12. Let n be the number of nodes. As all the hardware cost, software cost and link cost are estimated depending on the predicted parameters as link failure rate, hardware failure, the critical cost is termed as the expected value. The observed value is depicted as the convergence cost since all those observed values are truly observed as the time related values. Therefore, the best sites (nodes) are collected by applying the fitness function. Subsequently following the fitness values for each node, the job prioritization is done. The similar type of request will be sometimes generated by the user although, the similar type of sub process are collected, the sub process which is having larger cost is being provided with the higher priority. The sub process is being allocated to the node which gives the least cost.

Overview of inclusive BCO approach: The overview of IBCO approach is given as:

Step 1: Collect the total jobs (bees) submitted.

Step 2: Initialize the populations with its intrinsic aspects.

Step 3: While all jobs are dispatched:

- Evaluate fitness of each site
 - Find the best site from each patch
 - Locate the fittest bees into best site
- End while

Step 4: Repeat the steps for fore coming jobs.

According to IBCO the scout bee starts its flight for searching best sites. The forage dance related to the flowers subjected to node intrinsic aspects applying the fitness value the probable best sites are evaluated. The solution obtained are found to be best promising solutions.

RESULTS AND DISCUSSION

In the e-Learning approach under a dedicated grid environment, precision of the evaluation strongly relies on applied workloads. For this concern, researchers use Grid Workload Archive (GWA) which is based on actual production traces to concede results adequately. The probability of scheduling has been increased with the proposed approach. Initially the problem formulation consists of set of jobs $i = \{1, 2, \dots, n\}$. The available resources in the grid environment are represented as $k = \{1, 2, \dots, N\}$. Profitability in proposed system is the resources chosen and the job which is having the largest cost is allocated first.

The forager will return to the hive to perform the waggle dance. The analysis is made regarding the patch cost for each site. The sites are evenly ranked. In order to perform this analysis many parameters such task intrinsic rate and node intrinsic aspects are taken into account. Initially this procedure starts by examining the node intrinsic attributes such as hardware failure, link failure rate, etc. It is observed that processing capability rate for five nodes lies between the range minimum 10 (mega operations per sec). The processing capability of the nodes $\{20, 38, 21, 20, 19\}$ is kept to be arbitrary. The h_k values for five nodes are taken to be $\{0.93 \leq h_k \leq 1.23\}$. Likewise follows the software failures. The link failure rates are observed to be $\{0.02 \leq l_k \leq 0.69\}$. Similarly the data transferred in the k th node are taken to

be $\{0.02 \leq l_k \leq 0.69\}$. The link capacity rates are observed to be $\{2 \leq c_k \leq 9\}$. A sample of performance evaluation is described consisting of seven nodes. The former obsession is calculating the computation cost comprising the task intrinsic rate and computation capability. Figure 3 predicts the computation cost and its corresponding task intrinsic rate. It is obvious that when the complexity rate increases, the computation cost also increases. It states that the computation cost is directly proportional to the complexity rate. The convergence cost is evaluated using Eq. 11.

The convergence cost depends on the processing capability and the elapsed time. Figure 4 shows the critical cost estimated for the successive nodes. The critical cost is proportional to the node intrinsic aspects such as hardware cost, software cost and link cost. In addition, it is also more related to the failure dependency rate. The

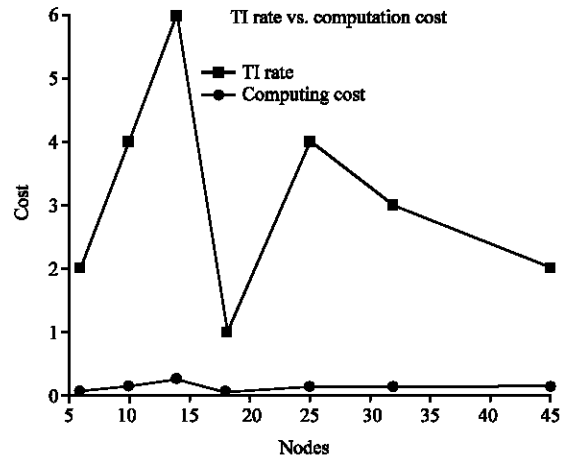


Fig. 3: Depiction of computation cost and its respective task intrinsic rate

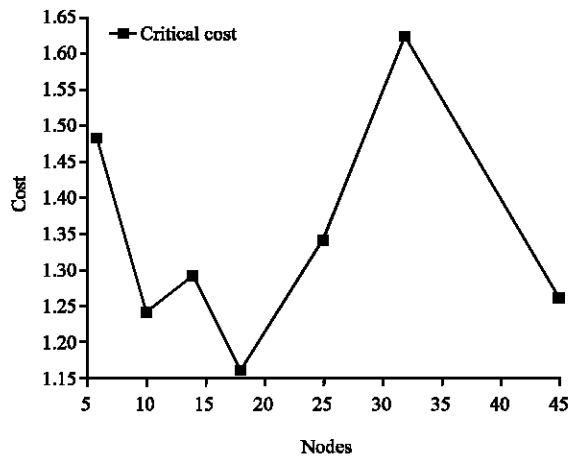


Fig. 4: Portrayal of critical cost among nodes

more the failure the less is the critical cost of the node. In essence for node ID 6 the critical cost is 1.5. For the subject of accuracy if the critical cost is < 2 , the node's scalability is well.

Similarly in Fig. 5, the same proportional rate prevails but the dependency is related to the node's computational capacity and the elapsed time constraints.

Figure 6 symbolizes the fitness value estimated. Chi-square test is partially applied and it is quite modified for the sake of accuracy. The fitness value for node 6 is pragmatic by 65.89. Likewise for node 45 it is observed as 109.32. The node which is provided with higher cost is subjected to undergo degraded performance.

Table 1 is arbitrary similar to the node intrinsic aspects. Moreover, performance is raised with the proposed system. It is experiential from Fig. 7 for 50 jobs the performance time were observed to be 60 sec with the proposed approach and for ACO it is found to be 70 sec. Likewise for IBCO approach the average time is about 146

and 155 sec for ACO. Therefore, the probability of latency rate in execution time is also considerably increased with inclusive IBCO approach.

On these criteria the nodes are organized on the arrival of each batch of jobs. On predicting the nodes behavior in advance, the jobs are prioritized based on the maximum job size. The proposed approach called IBCO is compared with the Benchmark algorithm ACO for inspection of performance in Fig. 7. It is apparent from the figure that the proposed IBCO mechanism improves performance by exploiting optimal route than the traditional ACO. It also outperforms the limitations of ACO by providing direct information sharing between nodes. IBCO approach tremendously reduces the computation time in task scheduling process.

Furthermore, as in Fig. 8 a distinctive scheduling order is presented for the incoming population using the proposed Inclusive BCO approach. The X-axis shows the

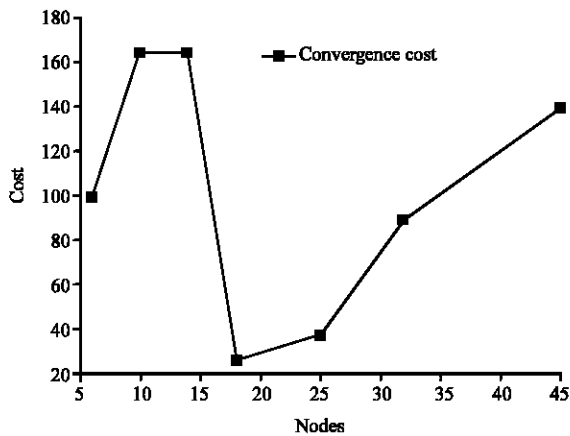


Fig. 5: Interpretation of convergence cost among nodes

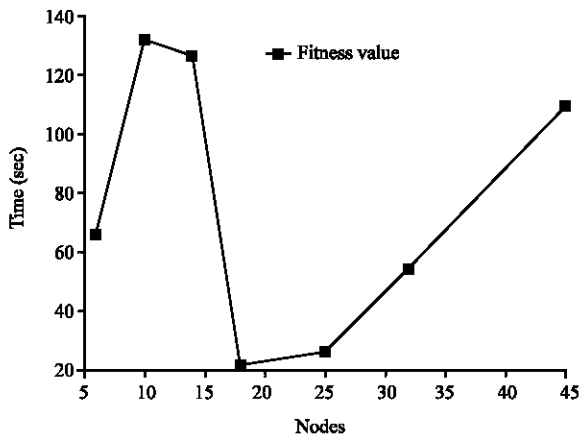


Fig. 6: Representing fitness value among nodes

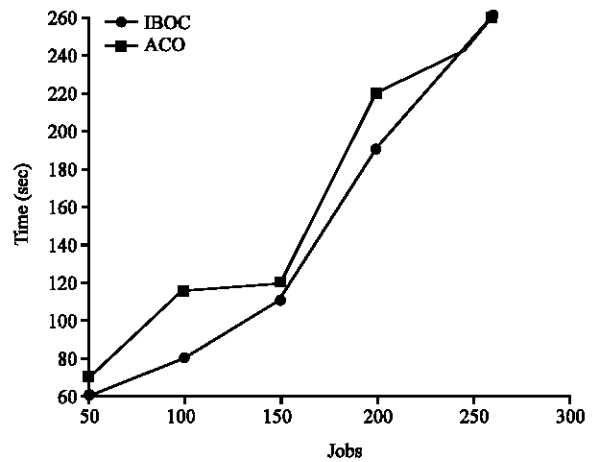


Fig. 7: Comparison between IBCO and ACO approaches

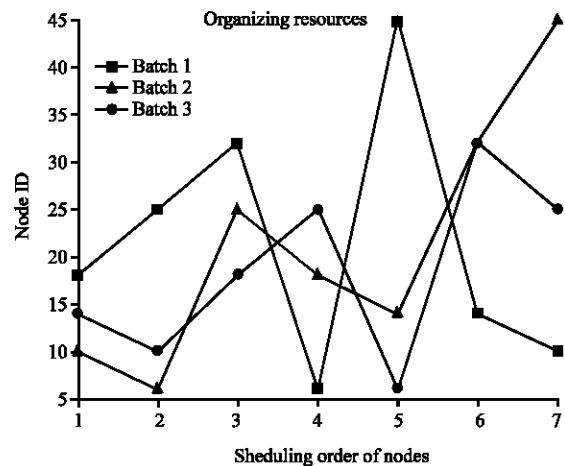


Fig. 8: Illustration of scheduling order

Table 1: Convergence cost's attributes

Node ID	6	10	14	18	25	32	45
Capability (Mega operations sec ⁻¹)	38	34	26	23	23	24	34
Elapsed time (τ_{pa}) (sec)	23	28	8	3	14	17	3
Completion time (λ_{ip}) (sec)	42	6	8	13	29	18	17

scheduling order of nodes and Y-axis its corresponding node Ids. Say {18, 25, 32, 6, 45, 14, 10} are shown as the best sites for patching. The fittest bees are located in these best sites.

CONCLUSION

The primary concern related to organically located grid system is the self capability of scheduling jobs. It is known that swarm intelligence based algorithms are really affording best solutions in decision making scenarios for these scheduling aspects. With these esteems, the proposal provides a novel deliberated approach called Inclusive bee colony optimization that schedules the tasks inclusively in grid environment and supports e-Learning effectively. The key factors in the proposed approach are analyzing the node's distinctiveness with respect to the cost constraints and on measuring several costs including some time constraints and finally predicting the cost of the node for making the right decision.

The Inclusive Bee Colony Optimization algorithm tends to reveal good performance results in making the scheduling decisions. On comparing with the existing approaches the simulation result offers a decreased time cost and moreover, balances with large quantity of loads. The affirmed research can be enhanced with effective application on real grid scenarios, competing to formulate right decisions at quick time. The bee algorithm, essence of this research can be augmented in fusion with some other effective reliability models and QoS measures in future research.

REFERENCES

Anderson, D.P., 2004. BOINC: A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, November 8, 2004, Pittsburgh, USA., pp: 4-10.

Applegate, D. and W. Cook, 1991. A computational study of the job-shop scheduling problem. ORSA J. Comput., 3: 149-156.

Bauer, A., B. Bullnheimer, R.F. Hartl and C. Strauss, 1999. An ant colony optimization, approach for the single machine total tardiness problem. Proceedings of the 1999 Congress on Evolutionary Computation, July -6-9, 1999, Washington, DC., USA., pp: 1445-1450.

Bean, J.C., 1994. Genetics and random keys for sequencing and optimization. ORSA J. Comput., 6: 154-160.

Chong, C.S., M.Y.H. Low, A.I. Sivakumar and K.L. Gay, 2006. A bee colony optimization algorithm to job shop scheduling. Proceedings of the Winter Simulation Conference, December 3-6, 2006, Monterey, CA., USA., pp: 1954-1961.

Colomi, A., M. Dorigo, V. Maniezzo and M. Trubian, 1994. Ant system for job-shop scheduling. Belgian J. Operat. Res. Stat. Comput. Sci., 34: 39-53.

Dorigo, M. and G. di Caro, 1999. The Ant Colony Optimization Metaheuristic. In: New Ideas in Optimization, Corne, D., M. Dorigo and F. Glover (Eds.). McGraw-Hill, New York, USA., pp: 11-32.

Dorigo, M., V. Maniezzo and A. Colomi, 1996. The ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernetics Part B, 26: 29-41.

Ilavarasan, E. and P. Thambidurai, 2007. Low complexity performance effective task scheduling algorithm for heterogeneous computing environments. J. Comput. Sci., 3: 94-103.

Jin, C. and R. Buyya, 2003. A dataflow system with a local optima-based scheduling for enterprise grids. The University of Melbourne, Australia, pp: 1-15.

Jin, C., R. Buyya, L. Stein and Z. Zhang, 2007. A dataflow model for.NET-based grid computing systems. Proceedings of the 3rd International Workshop on Grid Computing and Applications, June 6-7, 2007, World Scientific Publishing Co., USA., pp: 25-37.

Lee, L.T., H.Y. Chang, K.Y. Liu, G.M. Chang, C.C. Lien, 2006. A dynamic scheduling algorithm in heterogeneous computing environments. Proceedings of the International Symposium on Communications and Information Technologies, October 18-September 20, 2006, Bangkok, pp: 313-318.

Nowicki, E. and C. Smutnicki, 1996. A fast taboo search algorithm for the job shop problem. Manage. Sci., 42: 797-813.

Ombuki, B. and M. Ventresca, 2004. Local search genetic algorithms for the job shop scheduling problem. Applied Intell., 21: 99-109.

Quezada-Pina, A., A. Tchernykh, J.L. Gonzalez-Garcia, A. Hiraes-Carbajal and J.M. Ramirez-Alcaraz *et al.*, 2012. Adaptive parallel job scheduling with resource admissible allocation on two-level hierarchical grids. Future Gener. Comput. Syst., 28: 965-976.

- Sutton, R.S. and A.G. Barto, 1998. Reinforcement Learning: An Introduction. MIT Press, Cambridge.
- Tanga, X., K. Li, M. Qiu and E.H.M. Sha, 2012. A hierarchical reliability-driven scheduling algorithm in grid systems. *J. Parallel Distrib. Comput.*, 72: 525-535.
- Vengerov, D., 2007. A reinforcement learning approach of dynamic resource allocation. *Eng. Appl. Artif. Intell.*, 20: 383-390.
- Wu, J., X. Xu, P. Zhang and C. Liu, 2011. A novel multi-agent reinforcement learning approach for job scheduling in grid computing. *Future Gener. Comput. Syst.*, 27: 430-439.
- Zhang, C.J., V. Lesser and P. Shenoy, 2009. A multi-agent learning approach to online distributed resource allocation. *Proceedings of the 21st International Joint Conferences on Artificial Intelligence, Volume 1, July 11-17, 2009, Pasadena, California, USA.*, pp: 361-366.