Energy-Efficient Heterogeneous Multi-Processor Environment in Cloud using Modern Artificial BEE Colony

Sakshi Kapoor, Surya Narayan Panda

Abstract: Cloud Computing is an expansion in distributed, parallel as well as grid computing. The purpose behind cloud computing is the provision of dynamic hiring of server proficiencies as a virtualized and accessible service for customers and end-users. A key issue found in the cloud is the management of resources. Load balancing is a key problem in the management of resources. *The job scheduling issue has charmed abundant courtesy in the field of operation research. There are various algorithms like Ant optimization, genetic algorithms, artificial bee colony which can be used to solve the problem of scheduling. No doubt, Parallelization is proved to be the best method that can be utilized for improving the concert of the above algorithms. In this article, a modified artificial bee colony is utilized in order to crack the problem of scheduling in a heterogeneous multi-processor environment. In this, ABC has various colonies located on dissimilar network hosts as well as the algorithm is accepted in several colonies in parallel fashion. The colonies communicate with each other, which is approved through exchanging immigrants. In order to determine the communication of colonies with neighbors, a dynamic strategy is followed up. The algorithm is useful in making the parallel environment more efficient by reducing energy consumption. The energy consumption is reduced for each job in the DAG. Scheduling with MABC in the heterogeneous environment becomes easy as well as effective.*

***Keywords** : Cloud Computing, Parallelization, Multiprocessor, Energy Consumption, Scheduling, Heterogeneous Environment.*

I. INTRODUCTION

Job scheduling is found to be the most significant issue in parallel computing [1]. It can be defined as n jobs Jm where m= 1,2,3,4,...,m running on X machines where Xk= 1,2,… , m. Each activity comprises a big chain of tasks, each task from the activity should be prepared during a continuous time span of a particular length on a given machine.

In the concept of scheduling, job scheduling alludes to a lot of arrangements that characterize the request for implementation of various processes. As of the majority of the accessible resources of a system that requires scheduling formerly their utilization, the CPU is reproachful of them. Multi-programming is a known fundamental planning strategies [2]. In CPU, scheduling is basically done to keep it as occupied as could reasonably be expected. In parallel computing, there is a need for scheduling different processors as well as the management of resources for different processors. For the management of resources for different processors, it ought to be guaranteed that, overlapping of resources should not be there, also it ought not give any clashing results or outcomes. So, the scheduling in a single-processor is easier and less complicated in comparison with complex scheduling in multiprocessors [3]. There is a great need to ensure that no processor should be in underloaded or overloaded state in case of multiprocessor scheduling. So, the overall framework ought to be balanced and adjusted.

In case of multi-programmed memory systems, the scheduler is the one which decides the order in which processes are to be executed. On considering a parallel system, there are several queues because of several processors, so, there is a big requirement for there is a need for multiple queues to be scheduled concurrently [4,5]. All the scheduling algorithms which are preferable for multiple processors should be proficient for the scheduling of all the queues efficiently.

Task scheduling tackles the issues regarding the allocation of assets to various tasks or jobs with the goal to increase the utilization of resources and reduction in the execution time. For better and efficient performance, scheduling algorithm should be effective and consider both under-loaded as well as overloaded processors, fault tolerance, reduction in total execution time. The role of the client is to present their tasks for completion to the cloud, which in turn, needs to allot to the processor in order to execute [5-7]. Presently, the worry is these tasks or jobs are allocated to various processors with the goal that extreme profit along with minimum execution time is achieved by cloud owner. In this way, Task scheduling settles the issues related to the assignment of tasks to the most optimal processor. It is the most efficient technique that better utilizes resources and accomplishing economic efficiency [6,7]. Distributed scheduling where an assignment of tasks on various assets that are not found topographically on the same processor and on the other hand, centralized scheduling where every one of the assets is on the same processor, however, the level of complexity is very low contrast with distributed planning. Further, the Heuristic strategies are arranged into static and dynamic and hybrid are cost based, vitality based, effectiveness based. On the other hand, Metaheuristic techniques are nature propelled and swarm insight [7,8].

II. LITERATURE SURVEY

In recent years, the popular techniques and algorithms like Genetic algorithms, Tabo search, Neural Networks, Artificial bee colony, Ant optimization, and various such algorithms are widely used in order to solve these types of problems.

Watanabe, Ida, and Gen [9] presented an algorithm having an adjusted crossover operator and search area adjustment technique in order to control the tradeoff stability among worldwide and nearby searches.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

Sakshi Kapoor, Chitkara University Institute of Engineering and Technology Chitkara University, Punjab, India
Surya Narayan Panda, Chitkara University Institute of Engineering and Technology Chitkara University, Punjab, India

Retrieval Number: B2835129219/2019©BEIESP
DOI: 10.35940/ijeat.B2835.129219
Journal Website: www.ijeat.org

Asadzadeh (2015) proposed a genetic algorithm namely agent-based local search

genetic algorithm. In order to implement a local search genetic algorithm, a multi-agent system consists of several agents in which each is having some sort of special behavior is developed. The consequences of analyses demonstrated that the proposed methodology advances effectiveness [10]. Adems [11] proposed a novel swarm-based ABC in order to solve job scheduling issues. Local property of the issue is found, and after that, a tree search algorithm is conceived in order to upgrade the misuse capacity of swarm-based ABC. This methodology is proficient in solving scheduling issues with absolute weighted tardiness criteria. Banharnsakun, Sirinaovakul, and Achalakul [12] proposed a very effective technique dependent on the best-so-far ABC algorithm for a better understanding of job scheduling [12]. Kurdi [13] developed another hybrid island model genetic algorithm (HIMGA) for job scheduling. For progressively successful island model genetic algorithm(IMGA), their methodology has another self-adaption approach where the best entities are selected for performing a local search with the use of taboo search, and the most exceedingly awful ones are enrolled to perform a worldwide search utilizing random mutation operator. Kannan [14] developed a methodology which is the agent-based parallel genetic algorithm. The methodology depends on a coarse-grained model. The initial population is partitioned in various subpopulations and each of which is developed independently.

III. METHODS AND METHODOLOGY

A. Scheduling of Tasks Using DAG Approach

We consider n parallel jobs or tasks to be scheduled using the DAG Model. The task set in the DAG is described by T = {t1, t2,....., tm}. In the DAG Model, each task is represented as Ti where 1 ≤ I ≤ n, where the nodes represent distinctive execution requirements and the edges describe conditions among different nodes. A node in Ti is symbolized as W b_i, 1 ≤ b≤ ni, where ni is a total number of nodes in T_i [15]. The implementation constraint of node W b_i is indicated by Ib_i . There is a directed arc between node A directed edge from node Wb_i to Wh_i, indicated as $Wb_i \rightarrow Wh_i$, means Wh_i executes after the execution of W b_i. So, W b_i is the parent of Wh_i . In DAG, a node can have any number of children and parents but the execution of a child will start only when the execution of the parent gets finished. Figure 1 shows a DAG Model with 10 different nodes to be done [14-16]. The path in the DAG which has the largest execution prerequisite is known as a critical path of a DAG. We can consider a task set as schedulable when all the jobs or tasks encounter their targets or deadlines.

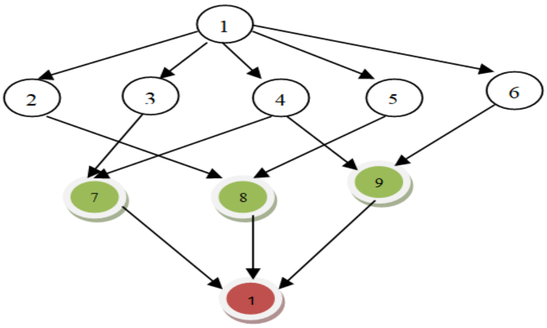

Figure 1. DAG MODEL

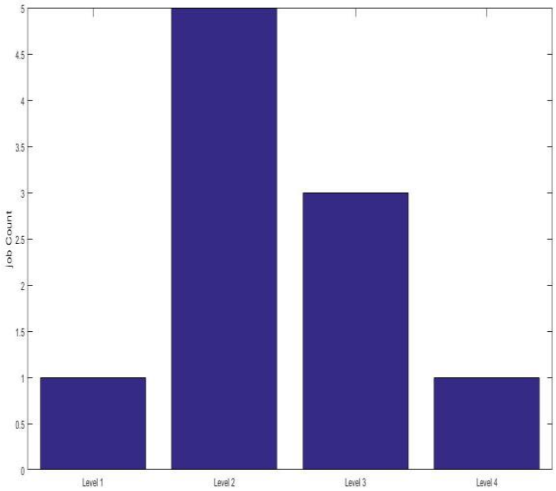

Figure 2. Levels in DAG Model

B. Modified Artificial Bee Colony (Modified ABC)

ABC is an optimization algorithm which is inspired by the intelligence of bees – how the bees collect their food, monitor the food, how the tasks are divided between the three type of bees namely – employed and unemployed bees. The unemployed contains scout and onlooker bees. The bee which makes the decision regarding selecting food is the onlooker bee and the one going towards the food source stayed by it beforehand is titled an employed bee [15]. A bee booming out arbitrary hunt is the scout bee. In ABC, employed artificial bees come in the first half and onlooker in the next half. Each food source corresponds to one employee bee. It means there is equality in total number of food sources and employed bees. The scout bee is the one whose food source gets shattered by both employed and onlooker bees[15,16].

ARTIFICIAL BEE COLONY:

Standard Algorithm: Artificial bee colony (ABC)

```
Initialization of food sources $Z_i$ by random numbers.
Evaluation.
c =1
Repeat
        Update in food sources through Employed Bees: These bees pursue fresh food source $m_i$ and allotted food source $l_i$ using greedy selection.
        Computing possibilities for Onlookers: Selecting food sources considering the probability $p_i$ and looking for fresh food source $m_i$ about the selected one $Z_i$. The update in food sources will be performed by these bees.
        Producing Fresh Food Sources by Scouts: Sending scouts to search areas for finding fresh food sources.
c=c+1
Until
c= MCN.
```

MODIFIED ABC [17]

Population initialization is a very critical job in evolutionary algorithms as it is able to disturb the conjunction speed as well as the excellence of the concluding solution. If there is no information available regarding the solution, then, in that case, random initialization is considered as the most frequently used process for producing candidate solutions. A chaotic map rearranges the population of search space information which is easy to extract in order to increase the diversity of the population. The interchange of random initial population and opposition-based initialization gives efficient initial solutions as well as quicken the convergence rapidity in the speed. There is a proper selection of sinusoidal iterator and the equation for the same is as follows:

$$\mathbf{Cha_{m+1} = \sin(\pi cha_m),\ cha_k\ \varepsilon\ (0,1),\ m=0,1,2....V.}$$

where m is iteration counter and V is the fixed number of chaotic iterations.

A modified differential evolution (DE) is an effective evolutionary algorithm [17]. Many optimization problems exist in the real world. The approach uses a collective evolutionary algorithm which has common actions. There are numerous algorithms having different alteration strategies. The strategy given below is commonly used.

$$\boldsymbol{DE/best/1 : K_i = G_{uest} + M(K_{q1}\text{-}K_{q2})}$$

Where i ranges from 1 to n and q1 and q2 are random numbers taken from i range. M is a positive number, also called amplification number.

In order to develop the convergence routine, best or top solutions in the current population are found to be a useful source. In *DE/best/1*, the best or top solution is utilized in order to uninterrupted the movements or associations of the current population. On the basis of ABC Property, the solution search equation is planned or created in the following manner :

$$\boldsymbol{ABC/best/1: H_{a,b}= J_{best,b}+ \alpha_{a,b}(j_{r1,b}\text{-}j_{r2,b})}$$

Where r1 and r2 are integers which are selected from the range 1 to n. J_{best} is the best fitness vector. If a new food is produced using equation 3, the algorithm will get constricted in local optima upon explaining difficult multimodal complications. ABC/best/1 is unable to avoid premature convergence. On the basis of the above description, the modified ABC algorithm is as follows:

```
♦ Fixed the population size Q
♦ Creating an initial population
♦ Predicting or calculating the function values of the
    population
♦ For(i=1 to Q)
    {
    New food source is produced utilizing a new search
    process
    }
♦ Select the values of $J_{r1}$ , $J_{r2}$ from the current population.
♦ New food source is generated from $K_i$ using equation 2
♦ If(f($H_i$) < f($J_i$))
    {
    $J_i$=$V_i$
    }
♦ Else
    {
    If rand(0,1) < P
    {
    Choose the value of b from from {1,2,3,…,n}
    }
    }
♦ New food source generation using equation 3
♦ If(f($H_i$) < f($J_i$))
    {
    $k_i$ = $H_i$
    }
♦ End while(max function evaluations attained)
```

IV. Proposed Methodology

The methodology is as follows:

Creation of DAG: Considering the dataset, a DAG is made consisting of 10different jobs.

Fetching Costs: Communication as well as computation cost is fetched respectively.

Assignment of jobs to processors: Assigning jobs to processors considering the sum of two above factors.

Reduction in Energy Consumption: In order to reduce the total energy consumption, Modified ABC is used for optimized list of processors.

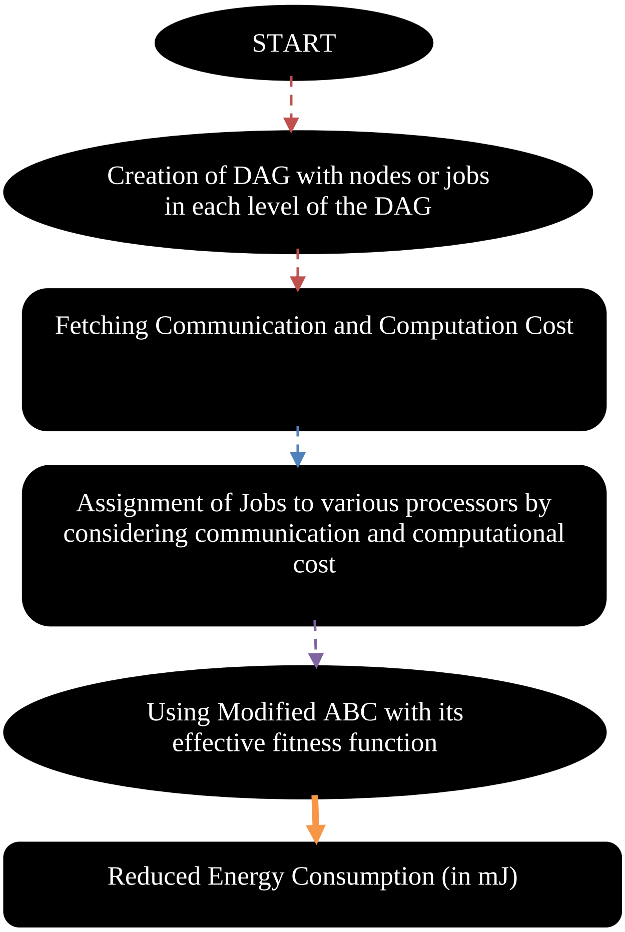
START
Creation of DAG with nodes or jobs in each level of the DAG
Fetching Communication and Computation Cost
Assignment of Jobs to various processors by considering communication and computational cost
Using Modified ABC with its effective fitness function
Reduced Energy Consumption (in mJ)

Figure 3. Proposed Methodology

V. EXPERIMENTAL RESULTS

Result Analysis of Modified Artificial Bee Colony (MABC)

The implementation of Modified ABC is performed using MATLAB 2018, which provides the results shown in the form of the graph as well as a table. The execution cost on Engine 1 is shown in fig 4. The blue color in the graph is for Engine 1. Similarly, the execution cost for Engine 2 and Engine 3 is denoted with blue in figure 5 and figure 6 respectively. The energy consumption is calculated in mJ. The first job consumes 1.525 energy in mJ whereas after applying MABC, the energy consumption reduced to 1.505 mJ. The second job consumes 1.299 mJ , which is reduced to 1.089 mJ. In the same way, the energy consumption gets reduced for each job with either a huge margin or a minor margin. The results with accurate measurement are shown in the table 1. The scheduling can be called Energy-aware scheduling using the Modern approach for ABC.

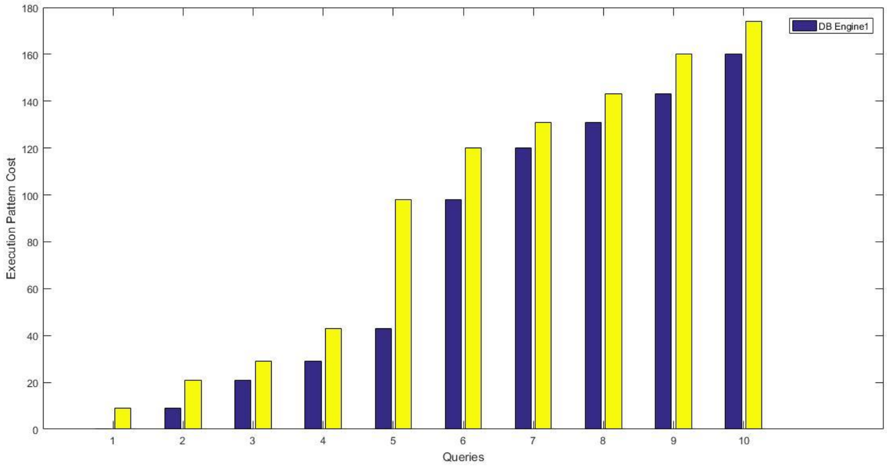

Figure 4. Execution Pattern Cost on Processor 1

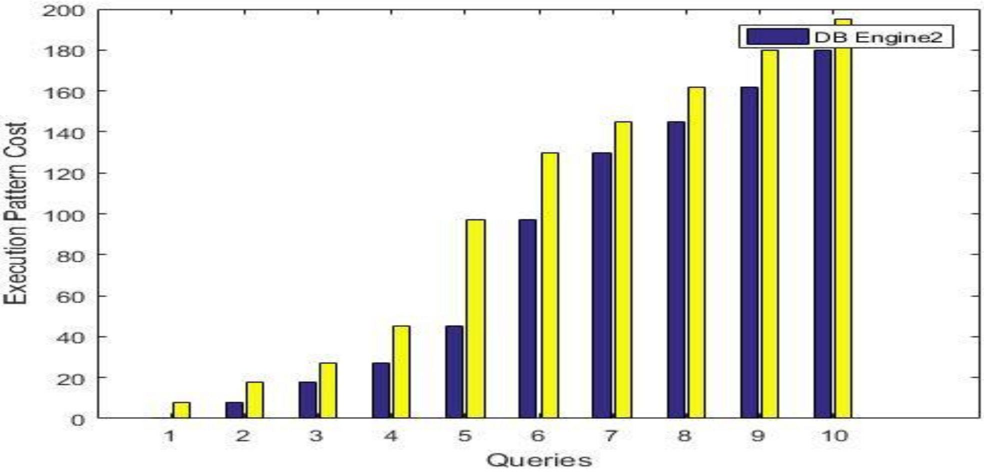

Figure 5. Execution Pattern Cost on Processor 2

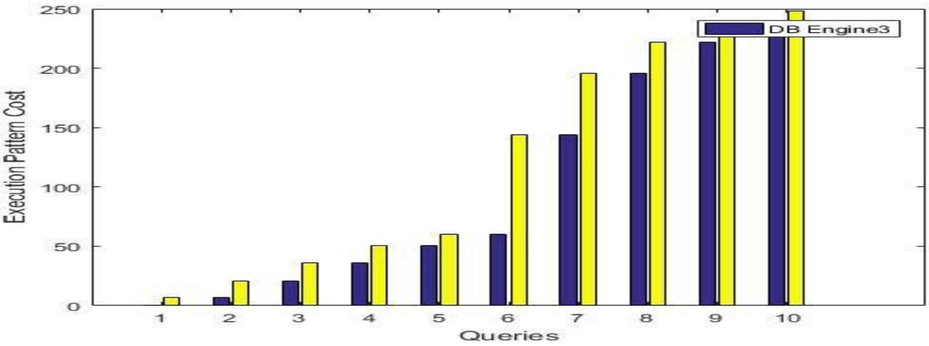

Figure 6. Execution Pattern Cost on Processor 3

Table 1. Energy Consumption in mJ

Job	ENERGY CONSUMPTION (mJ)	
	Without MABC	*With MABC*
1	1.525	1.505
2	1.282	1.033
3	1.278	1.044
4	0.192	0.19
5	0.782	0.78
6	0.782	0.779
7	0.7285	0.7282
8	0.7285	0.4799
9	1.587	1.581

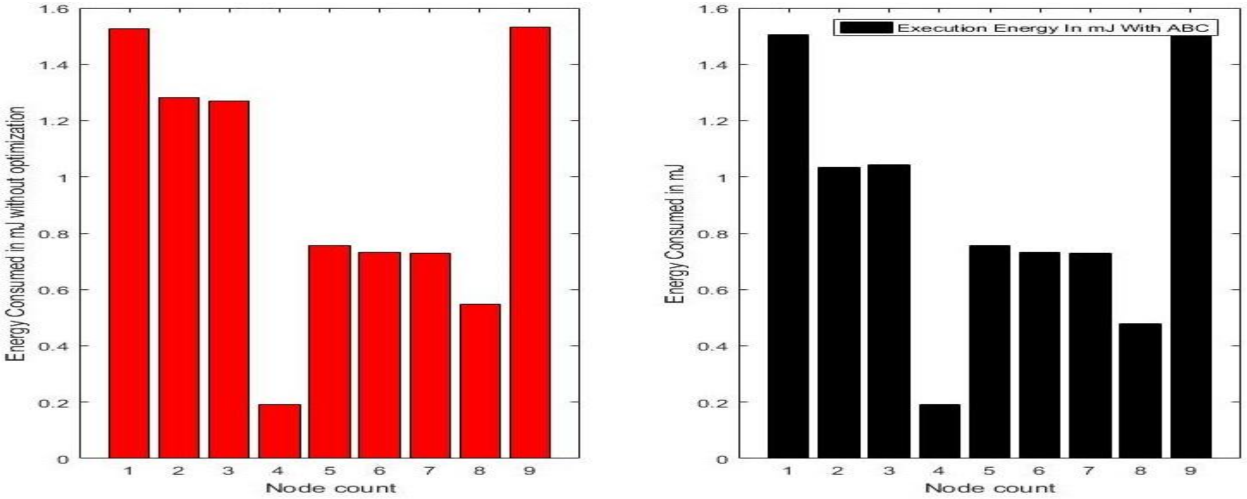

Figure 7. Energy Consumption with and without ABC

VI. DISCUSSION AND CONCLUSION

The energy consumption in task scheduling in a parallel cloud environment can become efficient using different algorithms and technologies in the field of artificial intelligence and machine learning. Considering task scheduling as the most important issue in parallel computing, we have used modified ABC in which network cost and load are taken as an additional parameter of investigation of scheduling. The modified ABC is found effective in reducing energy consumption to a great extent. The proposed methodology gives accurate and effective results that will be found useful in real-life applications. . In this, ABC has various colonies located on dissimilar network hosts as well as the algorithm is accepted in various colonies in parallel fashion. The communication among colonies is approved through exchanging immigrants. In order to determine the communication of colonies with neighbors, a dynamic strategy is followed up. The algorithm is useful in making the parallel environment more efficient by reducing energy consumption. The energy consumption is reduced for each job in the DAG. Scheduling with MABC in the heterogeneous environment becomes easy as well as effective.

REFERENCES

1. Ghanbari, Shamsollah, and Mohamed Othman. "A priority-based job scheduling algorithm in cloud computing." *Procedia Engineering* 50.0 (2012): 778-785.
2. Karthick, A. V., E. Ramaraj, and R. Ganapathy Subramanian. "An efficient multi-queue job scheduling for cloud computing." *2014 World Congress on Computing and Communication Technologies*. IEEE, 2014.
3. Maqableh, Mahmoud, and Huda Karajeh. "Job scheduling for cloud computing using neural networks." *Communications and Network* 6.03 (2014): 191.
4. Javanmardi, Saeed, et al. "Hybrid job scheduling algorithm for cloud computing environment." *Proceedings of the fifth international conference on innovations in bio-inspired computing and applications IBICA 2014*. Springer, Cham, 2014.
5. Paul, Mousumi, Debabrata Samanta, and Goutam Sanyal. "Dynamic job scheduling in cloud computing based on horizontal load balancing." *International Journal of Computer Technology and Applications (IJCTA)* 2.5 (2011): 1552-1556.
6. Kaur, Rajveer, and Supriya Kinger. "Analysis of job scheduling algorithms in cloud computing." *International Journal of Computer Trends and Technology (IJCTT)* 9.7 (2014): 379-386.
7. Bitam, Salim. "Bees life algorithm for job scheduling in cloud computing." *Proceedings of The Third International Conference on Communications and Information Technology*. 2012.

8. Wang, Xiaoli, Yuping Wang, and Yue Cui. "A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing." *Future Generation Computer Systems* 36 (2014): 91-101.
9. Asadzadeh, Leila. "A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy." *Computers & Industrial Engineering* 102 (2016): 359-367.
10. Asadzadeh, Leila, and Kamran Zamanifar. "An agent-based parallel approach for the job shop scheduling problem with genetic algorithms." *Mathematical and Computer Modelling* 52.11-12 (2010): 1957-1965.
11. Adams, Joseph, Egon Balas, and Daniel Zawack. "The shifting bottleneck procedure for job shop scheduling." *Management science* 34.3 (1988): 391-401.
12. Banharnsakun, Anan, Booncharoen Sirinaovakul, and Tiranee Achalakul. "Job shop scheduling with the best-so-far ABC." *Engineering Applications of Artificial Intelligence* 25.3 (2012): 583-593.
13. Kurdi, Mohamed. "A new hybrid island model genetic algorithm for job shop scheduling problem." *Computers & Industrial Engineering* 88 (2015): 273-283.
14. Kannan, V. R., and S. Ghosh. "An evaluation of the interaction between dispatching rules and truncation procedures in job-shop scheduling." *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH* 31.7 (1993): 1637-1654.
15. Sakellariou, Rizos, and Henan Zhao. "A hybrid heuristic for DAG scheduling on heterogeneous systems." *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*. IEEE, 2004.
16. Canon, Louis-Claude, et al. "Comparative evaluation of the robustness of dag scheduling heuristics." *Grid Computing*. Springer, Boston, MA, 2008.
17. Gao, Wei-feng, and San-yang Liu. "A modified artificial bee colony algorithm." *Computers & Operations Research* 39.3 (2012): 687-697.

AUTHOR PROFILE

Sakshi Kapoor is currently doing her Master of Technology in Computer Science and Engineering from **Chitkara University, Punjab, India**. Her research interests are Cloud Computing, Machine learning, Scheduling, Internet of Things (IOT), CloudIot. She currently lives in Punjab, India. She has done her Bachelor of Technology from Sri Sukhmani Institute of Engineering and Technology (SSIET), Derabassi. Her Languages are English, Hindi, Punjabi. She has done her publications in many conferences. She has good knowledge of CloudSim, Matlab.

Dr. Surya Narayan Panda is a Professor in **Chitkara University, Punjab, India.** He is Director Research in Chitkara University Research and Innovation Network **(CURIN), Chitkara University**. His research areas are Internet oF Things(IOT), Network Security, Cloud Computing, CloudIoT. He currently lives in Ambala city. He has done many publications in international journals as well as conferences. He has great knowledge about latest tools in Cloud Computing and Internet of Things (IoT).