International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-6, January 2012

An Efficient Modified Artificial Bee Colony Algorithm for Job Scheduling Problem

Manish Gupta, Govind sharma

Abstract- Swarm intelligence systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. Particle swarm, Ant colony, Bee colony are examples of swarm intelligence. In the field of computer science and operations research, Artificial Bee Colony Algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm. The job scheduling problem is the problem of assigning the jobs in the system in a manner that will optimize the overall performance of the application, while assuring the correctness of the result. In this paper, An Efficient artificial bee colony (ABC) algorithm, where we have used additional mutation and crossover operator of Genetic algorithm (GA) in the classical ABC algorithm. We have added crossover operator after the employed bee phase and mutation operator after onlooker bee phase of ABC algorithm, is proposed in this paper, for solving the job scheduling problem with the criterion to decrease the maximum completion time. The simulated results show that ABC proves to be a better algorithm when applied to job scheduling problem.

General Terms
Algorithms, Experimentation, Verification.

Keyword: Artificial Bee Colony, ABC, Genetic Algorithm, GA, Mutation, crossover.

I. INTRODUCTION

The problem of optimization is the most crucial problem in today's era and a great many work have been done to solve it. Previously a lot of work has been done on GA, ABC and hybridization of various
evolutionary algorithms. There are few literatures available which compares their performance evaluation and suggests the best technique to be opted for specific problems. Artificial Bee Colony (ABC) is one of the most newly defined algorithms by Dervis Karaboga in 2005 [1], provoked by the intelligent behavior of honey bees. It is as easy as Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithms, and uses only common control parameters such as colony size and maximum cycle number. ABC as an optimization tool provides a population-based search method in which individuals called foods positions are customized by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and at last the one with the highest nectar.

Manuscript received December 21, 2011
Manish Gupta, Email: manishgupta.2007@gmail.com
Govind sharma, Email: bobby_mits@yahoo.com

Rescheduling and optimization of logistic processes using GA and ACO is proposed in [2], A Bee Colony Optimization Algorithm for Traveling Salesman Problem is proposed in [3] , An idea based on honey bee swarm for numerical optimization is proposed in [4], A Survey: Algorithms Simulating Bee Swarm Intelligence is proposed in [5], A Study on a Real-Coded Genetic Algorithm is proposed in [6], Genetic Algorithms for Real Parameter Optimization, Foundations of Genetic Algorithms is proposed in [8], Handbook of Genetic Algorithms is in [9], An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms is in [10], Predictive Models for the Breeder Genetic Algorithm is proposed in [11], Genetic algorithm + data structure = evolution program is given in [12], Bases in the crossover landscape is proposed in [13]..

.Yazdani et al.developed a parallel variable neighborhood search (PVNS) algorithm based on six neighborhood structures [14]. Moradi et al. proposed an efficient architecture for scheduling the FJSP with machine availability constraints [15]. Defersha and Chen presented a parallel GA for the FJSP with sequence-dependent setups [16].

Zhang et al. proposed an effective GA for solving the FJSP to minimize makespan time [17]. Li et al. developed a hybrid TS with an efficient neighborhood structure for the FJSP [18]. Most recently, Moslehi and Mahnam [19] proposed a Pareto approach using PSO and local search for the multi-objective FJSP, and the computational results proved the efficiency of the algorithm. Wang XJ et al.[20] proposed A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. Pan et al. [21] proposed a discrete artificial bee colony algorithm based on a self-adaptive strategy to solve the lot-streaming flow shop scheduling problem with the criterion of total weighted earliness and tardiness penalties. The latest research has revealed some good properties of ABC [22,24,25]. Wong et al. [23] developed a novel bee colony optimization with local search to solve traveling salesman problem. a multi-objective genetic algorithm is proposed in [26] for stochastic job shop scheduling problems in which the makespan and the total tardiness ratio should be minimized. To facilitate the understanding of the algorithm, a flow chart [27] in figure 1 is provided. Pezzella et al. [28] introduced a GA integrating different strategies for generating the initial population, selecting individuals for reproduction and
reproducing new individuals to solve the FJSP. Xing et al. proposed a knowledge-based ant colony optimization algorithm (KBACO) for solving the FJSP [29]. In [10], genetic algorithm (GA) was hybridized with variable neighborhood search to solve the multi-objective FJSP.

The problem of task scheduling arises in a situation where there are more tasks than the available resources. Consider a picture wherein there are x, x={1,2,3,4,........X} jobs to be done and there are y, y={1,2,3,4.......Y} resources available. With the condition that the job is not allowed to be migrated between resources.

In such a situation if we have y>x then there is no reason for developing new algorithms for task scheduling because then resources can be allocated to the tasks on first come first serve basis.But if y<x then we need to develop new algorithms for task scheduling because now inefficient resource allocation can greatly hamper the efficiency and throughput of the scheduler.

To formulate the problem, defineTa a={1,2,3,…X } as x independent tasks permutation and R_b b={1,2,3,…y } as y computing resources. Suppose that the processing time$P_{a,b}$ for task a computing on b resource is known. The completion time F(x) represents the total cost time of completion [7].

In this Paper, we have extended the classical Artificial Bee Colony algorithm to the area of optimization problem. Proposed method basically adds an additional step of mutation operator and crossover operator in the Artificial Bee Colony for finding out the optimality. To validate the performance of proposed method job scheduling problem is used in our experiment.

The organization of the paper is as follows section 2 gives brief introduction on Artificial Bee Colony algorithm. Proposed Artificial bee colony with Genetic mutation and crossover is explained in section 3. Experimental Results are given in section 4. Section 5 gives Conclusion.

II. ARTIFICIAL BEE COLONY

In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making decision to choose a food source, is called an onlooker and a bee going to the food source visited by itself previously is named an employed bee. A bee carrying out random search is called a scout. In the ABC algorithm, first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout. The main steps of the algorithm are given below:

- Initialize.
- REPEAT.

(a) Place the employed bees on the food sources in the memory;

(b) Place the onlooker bees on the food sources in the memory;

(c) Send the scouts to the search area for discovering new food sources.

- UNTIL (requirements are met).

In the ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto the food sources and then measuring their nectar amounts; selecting of the food sources by the onlookers after sharing the information of employed bees and determining the nectar amount of the foods; determining the scout bees and then sending them onto possible food sources. At the initialization stage, a set of food source positions are randomly selected by the bees and their nectar amounts are determined. Then, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area within the hive. At the second stage, after sharing the information, every employed bee goes to the food source area visited by her at the previous cycle since that food source exists in her memory, and then chooses a new food source by means of visual information in the neighborhood of the present one. At the third stage, an onlooker prefers a food source area depending on the nectar information dispersed by the employed bees on the dance area. As the nectar amount of a food source increases, the probability with which that food source is chosen by an onlooker increases, too. Hence, the dance of employed bees carrying higher nectar recruits the onlookers for the food source areas with higher nectar amount. After arriving at the selected area, she chooses a new food source in the neighborhood of the one in the memory depending on visual information. Visual information is based on the comparison of food source positions. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one.

In ABC algorithm each food source position represents a candidate solution of optimization problem. In optimization problem each solution is associated with the fitness value on the basis of fitness value it is decided that which solution is better. So the nectar amount of a food source corresponds to the fitness value of the associated solution in ABC algorithm. The number of employed bees or the onlooker bees is equal to the number of solutions in the population. The ABC algorithm generates a random solution or initial population of size NF, where NF denotes the size of population or total number of food source. Each solution is represents the position of food source and denoted as x_{ij}, where i represents a particular solution (i=1,2,…..,NF) and each solution is a D-dimensional vector so j represents a particular dimension of a particular solution (j=1,2,….,D). After initialization of random solution employed bees start their searching. Employed bees search the food source near the previous food source, if the generated new solution is better than the previous solution than new solution replaces the old one. The comparison of food sources or solutions is done on the basis of fitness value or nectar amount of food source.

After all employed bees complete the search process; they share the nectar information of food sources (solutions) and their position information with onlooker bees. Now onlooker bee chooses a food source depending on the probability value P_i associated with the food source. Probability value for each food source is calculated by following equation (1):

$$P_i = \frac{f_i}{\sum_{n=1}^{NF} f_n} \qquad (1)$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Where   is the fitness value of the solution *i* or the nectar amount of food source evaluated by employed bee and *NF* is the number of food source. So after the evaluation of the food source by the employed bees the probability value for each food source is determined which is used by onlooker bees.

To produce the candidate solution from the previous solution artificial bee uses the following equation (2):

$$v_{ij}= x_{ij}+\O_{ij}(x_{ij}-x_{kj}) \qquad (2)$$

Where *j* is a index for dimension (*j=1,2,….,D*), *k* is a index which represents particular individual or solution from the population (*k=1,2,3…..,NF*), and *i* is also a index represents a particular solution (*i=1,2,…..,NF*). The difference between *i* and *k* is that *k* is determined randomly and value of *k* has to be different from *i*. $\O_{ij}$ is a random number between [-1,1]. It controls the production of the neighbor food positions around x_{ij}.. The difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is reduced. After the production of candidate solution v_{ij} , its fitness value is calculated and then it is compared with the fitness of x_{ij}.

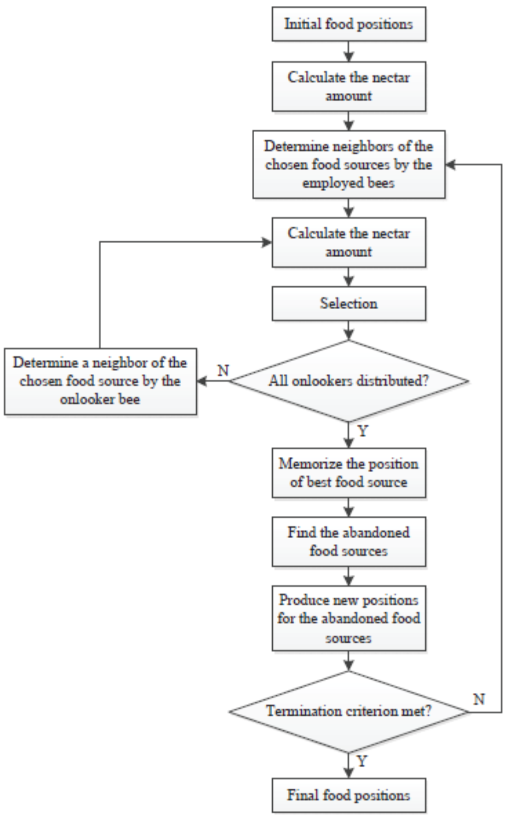

Fig.1 Flow Chart of ABC Algorithm

If the new candidate solution has equal or better nectar or fitness than the old, it is replaced with the old one in the memory. Otherwise, the old is retained. If a solution is not improved further through a predetermined number of cycles then that food source is assumed to be exhausted. Exhausted food source is replaced by new food source generated by scout bees.

III. ARTIFICIAL BEE COLONY WITH MUTATION AND CROSSOVER

In this proposed method, one additional step is added to standard Artificial Bee Colony Optimization is of mutation operator. Mutation operator is added after the employed bee phase of Artificial bee colony algorithm. ABC algorithm has four phases initialization phase, employed bees phase, onlooker bees phase and scout bees phase, we add mutation phase after the employed bee phase. Employed bee phase do the local search and mutation after the employed bee phase explore the search space and do the searching new area of solution space. Through mutation, on the one side, there is a chance of changing the local best position, and the algorithm may not be trapped into local optima. On the other side, individual can make use of the others' advantage by sharing information mechanism. In this method, the mutation step is carried out on the probabilistic way in each food searching operation for each iteration during the life cycle of ABC optimization technique. Selection of food source is done in a random manner. Food Source is selected arbitrarily from the food size and mutation is performed. In mutation, generated offspring's replaces the older offspring's. The mutation operator used in this paper is uniform mutation. When performing mutation, we randomly select one food source x_{ij} and replace its one of the dimension value by random number generated in between lower and upper bound value of the food source.

Below the proposed ABC with mutation after employed bee phase is shown. First phase is initialization phase where individuals are randomly selected from the search space.

Algorithm 1: ABC with Mutation After Employed Bee Phase

[*Initialsation Phase*]

for i=0 to max number of Food source NF **do**

for d=0 to dimension size **do**

Randomly initialize food source positions Xij

end for d

Compute fitness of each food source

end for i

Repeat

[*Employed Bee Phase*]

for i=0 to max no of employed bee **do**

for d= 0 to dimension **do**

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

produce new candidate solution

end for d

Compute fitness of individual

if fitness of new candidate solution is better **than** the existing solution replace the older solution.

end for i

for i = 0 to max number of food source NF **do**

Calculate the probability for each food source.

end for i

[Crossover phase]

If crossover criteria satisfies

For i=0 to maximum no. of food source

Select two random individuals from the current population for crossover operation.

. **Apply** crossover operation

New offspring generated from parents as a result of crossover. Replace the worst parent with the best new offspring if it is better.

End of i

[*Onlooker Bee Phase*]

for i=0 to max no of onlooker bee **do**

choose food source on the basis of probability Pi

For d= 0 to dimension **do**

Produce new candidate solution for food source position Xij

End for d

Compute fitness of individual food source

If fitness of new candidate solution is better **than** the existing solution replace the older solution.

End for i

[*Mutation Phase*]

If mutation criteria is met **then**

Select random particle from current population for mutation operator.

Apply mutation operation to generate new individuals new offspring generated from the result of mutation.

New set of sequence is generated for offspring

Compute the cost for that offspring

Compute the fitness of updated individual

[Scout Bee Phase]

If any food source exhausted **than**

Replace it by new random position generated by scout

bee.

Memorize the best food source so far

Until (Stopping criteria met).

Next is employed bee phase where local search is performed by employed bees. Next we have added additional operator to the ABC algorithm to check whether it improves the performance of algorithm. We have added a mutation operator after employed bee. Mutation is performed on the individual food source or individual if mutation probability satisfied. Onlooker phase and scout bee phase performed on current population or food sources after the mutation phase. At last best food source or individual is considered as global best solution.

IV. EXPERIMENTAL RESULTS

For every algorithm there are some control parameters which are used for its efficient working. Hence, there are some control parameters for Bee Colony with Mutation Algorithm also. We did an extensive literature survey and carried out our own experiments for determining the values of these control parameters. From this we found that the values which we have taken in this experiment are standard values and they are also suitable for this experiment.

The first control Parameter is Maximum cycle number and the value of this parameter we have taken in our experiment as 20,000. The next parameter in our experiment is maximum number of food source and we have taken its value to be 40. Another control parameter is number of runs and we have taken its value in our experiment as 30. It must be noted that each run contains maximum cycle number, which is 20,000 in our experiment. The fourth control parameter is Dimension and its value is taken as 30. In this experiment we are using the feature of uniform mutation operator in the ABC with mutation algorithm. The control parameter for mutation operator is Probability. Therefore we need to find the value of this parameter also. Its value can range from 0.1 to 0.9 we have taken as 0.7 for the mutation.

In this section we analyze the result obtained by our algorithm. To test the efficiency of our algorithm results of ABC is compared with Genetic algorithm (GA) results. In a grid scheduling task we already have the information about the number of resources, number of tasks, and the amount of time that will be taken by a resource to complete a task. We just need to find the sequence which will provide us the optimal results.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

We conducted the experiment by varying the number of resources as well as varying the number of tasks and then we compared our results with that of GA. In particular, we have taken three cases in which we have taken different number of resources and tasks.

Experiment 1: Here, we are assuming there are 5 resources and 17 tasks. Following are the execution time (in units) taken by ABC and GA.

Table2: Execution time calculated by GA and ABC for17 tasks by 5 resources

Genetic Algorithm (GA)	Modified Artificial bee colony(ABC)
3088.0	**3018.0**

The sequence generated by GA is: 14, 9, 0, 15, 4, 13, 6, 1, 3, 8, 11, 16, 12, 10, 2, 7,5.

The sequence generated by our proposed ABC with is: 0, 3, 9, 5, 12, 2, 13, 4, 1, 6, 16, 7, 8, 11, 10, 14, 15.

Experiment 2: Here, we are assuming there are 10 resources and 27 tasks. The execution time (in units) taken by GA and PSO are:

Table3: Execution time calculated by GA and ABC for27 tasks by 10 resources

Genetic Algorithm (GA)	Modified Artificial bee colony(ABC)
5415.0	**5311.0**

The sequence generated by GA is: 24,9,7,2,26,1,8,13,3, 18, 10, 0, 23, 5, 17, 14, 4, 15, 12, 6, 16, 20, 11, 22, 25, 19, 21.

The sequence generated by our proposed ABC is: 0, 5, 15, 9, 23, 4, 19, 8, 7, 3, 26, 6, 1, 14, 16, 2, 12, 11, 21, 17, 22, 25, 18, 13, 24, 10, 20.

Experiment 3: Here, we are assuming there are 12 resources and 30 tasks. The execution time (in units) taken by GA and PSO are:

Table4: Execution time calculated by GA and ABC for 30 tasks by 12 resources

Genetic Algorithm (GA)	Modified Artificial bee colony(ABC)
6134.0	**6015.0**

The sequence generated by GA is: 28, 24, 20, 25, 10, 7, 17, 4, 9, 22, 6, 11, 14, 18, 0, 29, 15, 13, 26, 12, 1, 19, 21, 5, 3, 27, 2, 29, 16, 8.

The sequence generated by our proposed PSO is: 0, 5, 19, 16, 24, 1, 21, 9, 2, 4, 29, 3, 6, 15, 17, 7, 13, 8, 23, 10, 26, 28, 18, 14, 25, 27, 11, 20, 12, 22.

From the table 2, table 3, table 4, it is clear that ABC takes less execution time than the GA algorithm.

V. CONCLUSION

In this paper, real coded mutation and crossover operator is applied to the ABC after the employed bee phase and onlooker bee phase of ABC algorithm. In iteration, with some probabilistic criteria selected food source is altered by mutation operator. The experiments are performed on a job scheduling problem available in the literature. There is no specific value for mutation probability for which we can obtain best results for job scheduling experiments. As future work we have the intention to apply other types of mutation operators and crossover operator in the ABC algorithm.

REFERENCES

1. Dervis Karaboga • Bahriye Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm” J Glob Optim (2007), pp 459-471
2. C.A. Silvaa, lM.C. Sousaa, T.A. Runkler, "Rescheduling and optimization of logistic processes using GA and ACO," Engineering Applications of Artificial Intelligence., vol. 21, pp. 343-352, 2008.
3. L.-P. Wong, M.Y. Hean Low, C.S. Chong, "A Bee Colony Optimization Algorithm for Traveling Salesman Problem," Second Asia International Conference on Modelling & Simulation, 2008, pp. 818-823.
4. D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, Technical Report-TR06, 2005.
5. D. Karaboga, B. Akay, "A Survey: Algorithms Simulating Bee Swarm Intelligence," Artificial Intelligence Review., vol. 31, pp. 68-85, 2009.
6. G-. G. Jin, S-. R. Joo, “A Study on a Real-Coded Genetic Algorithm,” Journal of Control, Automation, and Systems Engineering, vol. 6, no. 4, pp. 268-274, April 2000.
7. le. Biesmeijer, T.D. Seeley, "The use of waggle dance information by honey bees throughout their foraging careers," Behav. Ecol. Sociobiol., vol. 59, pp. 133-142, 2005.
8. Wright, A. “Genetic Algorithms for Real Parameter Optimization, Foundations of Genetic Algorithms”, G. Rswlins(Ed.), Morgen Kaufmann publishers, CA, pp.205-218, 1991.
9. L. Davis, “Handbook of Genetic Algorithms”, Von Nostrand Reinhold, N.Y., 1991
10. C. 2. Janikow and Z. Michalewicz, “An Experimental Comparisonof Binary and Floating Point Representations in Genetic Algorithms,”Pmc. 4th Int. Con$ on Genetic Algorithm, Morgan Kaufiann F’ublishers, CA, 1991
11. H. Miihlenbein and D. Schlierkamp-Voosen,“Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization”, Evolutionary Computation, vol. 1, pp. 25-49, I993
12. Z. Michalewicz, “Genetic algorithm + data structure = evolution program”, Springer-Verlag, Inc., Heidelberg, Berlin, 1996.
13. L. J. Eshelman, R. A. Caruana, and J. D. Schaffer, “Bases in the crossover landscape,” Pmc. 3rd Int. Con$ on Genetic Algorithms, J. Schaffer(Ed.), Morgan Kaufmann Publishers, LA, pp.10-19, 1989.
14. Yazdani M, Amiri M, Zandieh M (2010) Flexible job-shopscheduling with parallel variable neighborhood search algorithm.Expert Sys Appl 37(1):678–687.
15. Moradi E, Fatemi Ghomi SMT, Zandieh M (2010) An efficient architecture for scheduling flexible job-shop with machine availability constraints. Int J Adv Manuf Tech 51(1–4):325–339.
16. Defersha FM, Chen M (2010) A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups. Int J Adv Manuf Tech 49(1–4):263–279.
17. Zhang GH, Gao L, Shi Y (2011) An effective genetic algorithm for the flexible job-shop scheduling problem. Experts Sys Appl 38 (4):3563–3573.
18. Li JQ, Pan QK, Suganthan PN, Chua TJ (2011) A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. Int J Adv Manuf Tech 52(5–8):683–697.
19. Moslehi G, Mahnam M (2011) A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. Int J Product Econ 129(1):14–22.
20. Wang XJ, Gao L, Zhang CY, Shao XY (2010) A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. Int J Adv Manuf Tech 51 (5–8):757–767.

21. Pan QK, Tasgetiren MF, Suganthan PN, Chua TJ (2011) A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf Sci 181(12):2455–2468.
22. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 24(1):108–132.
23. Wong LP, Low MYH, Chong CS (2008) Bee colony optimization with local search for traveling salesman problem. In: Proc. of 6th IEEE International Conference on Industrial Informatics, pp 1019–1025.
24. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl.Soft Comput.* 2008, *8*, 687–697.
25. Karaboga, D.; Akay, B. A survey: algorithms simulating bee swarm intelligence. *Artif. Intell.Rev.* 2009, *31*, 61–85.
26. Lei, D. Simplified multi-objective genetic algorithms for stochastic job shop scheduling. *Appl.Soft Comput.* 2011, doi:10.1016/j.asoc.2011.06.001.
27. Akay, B.; Karaboga, D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J. Intell. Manuf.* 2011, doi:10.1007/s10845-010-0393-4.
28. Pezzella, F.; Morganti, G.; Ciaschetti, G. A genetic algorithm for the flexible job-shop scheduling problem. *Comput. Oper. Res.* 2008, *35*, 3202–3212.
29. Xing LN, Chen YW, Wang P, Zhao QS, Xiong J (2010) A knowledge-based ant colony optimization for flexible job shop scheduling problems. Applied Soft Com 10(3):888–896.
30. Gao J, Sun LY, Gen M (2008) A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Com Oper Res 35(9):2892–2907.

AUTHORS PROFILE

Mr. Manish Gupta, obtained Master of Technology (M.Tech.) in Computer Science from ABV-Indian Institute of Technology & management, Gwalior in 2011. The area of research is Soft Computing and Wireless Network.

Mr. Govind Sharma, obtained Master of Technology (M.Tech.) in Computer Science from ABV-Indian Institute of Technology & management, Gwalior in 2011. The area of research is Wireless Network and Soft Computing.