Comparative Study of GA and ABC for Job Scheduling

V. Selvi, R. Umarani

Abstract- ***In the field of computer science and operation's research, Artificial Bee Colony (ABC) is an optimization algorithm relatively new swarm intelligence technique based on behaviour of honey bee swarm and Meta heuristic. It is successfully applied to various paths mostly continuous optimization problems. Swarm intelligence systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. 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. ABC algorithm, is proposed in this paper, for solving the job scheduling problem with the criterion to decrease the maximum completion time. In this paper, modifications to the ABC algorithm is based on Genetic Algorithm (GA) crossover and mutation operators. Such modifications applied to the creation of new candidate solutions improved performance of the algorithm.***

Keywords: Artificial Bee Colony, Genetic algorithm, Job scheduling.

I. INTRODUCTION

Scheduling jobs has been a popular research topic for many years. There are many different ways to schedule jobs and the threads which make them up. However, only a few mechanisms are used in practice and studied in detail. Users submit jobs in batch to a resource management system queue and a centralized scheduler decides how to prioritize and allocate resources for job execution. To minimize the response time, the scheduling system strategy needs to prioritize competing user jobs with varying levels of priorities and importance and allocate resources accordingly.

Several inhabitants based on algorithm has been proposed to find near-optimal solutions to the difficult optimization problems like scheduling and routing problems. An inhabitant based algorithm refers the inhabitants consisting of possible solutions to the problem are modified by applying some operators on the solutions depending on the information of their fitness values. Inhabitants based on algorithm are classified into two groups: evolutionary algorithms and swarm intelligence-based algorithms. Evolutionary algorithm is one of the most accepted Genetic Algorithm (GA) based on the natural evolution. In the basic GA, a selection operation is applied to the solutions evaluated by the evaluation component. As a relatively new member of swarm intelligence, an artificial bee colony (ABC) algorithm is based on collective behaviour of self-organized systems.

Manuscript received on January, 2013.

V.Selvi, Associate Professor, Department of M.C.A, M.A.M College Of Engineering, Siruganur, Trichy(Dt),Tamil Nadu, India.

R.Umarani, Associate Professor, Department of Computer Science, Sri Saradha College For Women, Salem, Tamil Nadu, India.

So far, ABC algorithm has received interest from researchers in a variety of fields. The ABC algorithm was first proposed to optimize multi-variable and multi-modal continuous functions. Many comparative studies showed that the performance of the ABC algorithm was competitive when compared to other population-based algorithms with the advantage of employing fewer control parameters in the continuous space.

This paper performs an experiment on proposed adaptive ABC with crossover and mutation operations. The crossover and mutation operations are performed to find best food source positions. After the crossover and mutation operation the fitness of the individual food source's best position is compared with that of the two off-springs and the best one is taken as the new individual best food source position. Two food sources are selected as parents through selection process and calculate their fitness values. After selecting crossover and mutation points randomly, new fitness values are generated using crossover and mutation's probabilities. Both the above techniques perform a crossover and mutation by swapping the food source around the crossover and mutation points.

The remainder of this paper is organized as follows: Section 2 briefly reviews the Genetic algorithm. Section 3 presents the basic ABC algorithm and discusses the proposed adaptive ABC technique, which is detailed in its subsections. Section 4 discusses about the Experimental Results and Discussion. Section 5 concludes the paper.

II. GENETIC ALGORITHM

In the computer science field of artificial intelligence, Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Basically it consists of five components: a random number generator, a fitness evaluation unit and genetic operators for reproduction; crossover and mutation operations.

Simple generational genetic algorithm procedure

1. Choose the initial population of individuals
2. Evaluate the fitness of everyone in that population.
3. Repeat on this generation until termination (time limit, sufficient fitness achieved, etc.):
   i. Select the best-fit individuals for reproduction
   ii. Breed new individuals through crossover and mutation operations to give birth to offspring
   iii. Evaluate the individual fitness of new individuals
4. Replace least-fit population with new individuals

Figure 1: The procedure of Genetic algorithm

The initial population required at the start of the algorithm, is a set of food source generated by the random generator.

Each position of a food source represents a possible solution of the optimization problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. A fitness value is a measure of the goodness of the solution that it represents. Essentially the aim of the genetic operators is to transform this set of food source into sets with superior fitness values. The reproduction operator performs a natural selection function known as seeded selection. The crossover and mutation operator chooses pairs of food sources at random and produces new pairs. The simplest crossover and mutation operation is to cut the original food sources nectar amount at a randomly selected point and to exchange. The number of crossover and mutation operations is governed by a crossover and mutation rate. The mutation operator randomly mutates or reverses the values of food source. The number of mutation operations is determined by a mutation rate. A phase of the algorithm consists of applying the evaluation, reproduction, crossover and mutation operations.

2.1 Genetic algorithm for Job scheduling

Step 1. Initialize the population as input number of processors, number of jobs.
Step2. Process started
Step2.1 Evaluate the fitness function (makes pan).
Step2. 2. Perform selection to select best individuals from the current population.
Step 2. 3. Perform two-point crossover. Choose pairs of chromosomes (task).Choose a random point exchange machine assignments from that point until the end of the chromosome.
Step 2.4.Mutation: Randomly select a task. Randomly, reassign it to the new machine.
Step 3. The process is repeated until the stopping criterion is met. (Best fitness, minimum completion time)
Step 4. Stop

Figure 2: The procedure of genetic algorithm for job scheduling

III. THE ARTIFICIAL BEE COLONY ALGORITHM

Inspired by the intelligent foraging behaviours of Honeybee Swarm, an Artificial Bee Colony algorithm is developed, which is a new population-based meta-heuristic approach. In ABC algorithm, three different kinds of foraging bees are involved: employed bees, onlooker bees, and scout bees. The procedure that followed in the ABC algorithm is given in Fig. 3.

Initialization phase
Step 1: Initialize parameters, including the number of food sources or the number of employed bees, number of onlooker bees and number of scout bees.
Step 2: Initialize population of food sources with random solutions.
Step 3: Calculate the objective value of each food source and then determine the best food resource. Employed bee phase
Step 4: For every employed bee, generate a new food source.
Step 5: Calculate the objective value for every new food sources and compute the best food source.
Onlooker bee phase
Step 6: Calculate the probability of selecting food source using Equ. (1).
Step 7: Calculate the number of onlooker bees to be sent to the food source.
Step 8: Every onlooker bee, generate the new food sources.
Scout bee phase
Step 9: Initialize scout bees with random solutions and update the best food sources.
Step 10: Determine the worst employed bees and replace them with the scout bees if the scout bees are better.
Step 11: If a stopping criterion is met, then the best food source is obtained with its objective value.

Figure 3: The basic procedure of ABC algorithm

Employed bees are those bees that are exploiting a food source currently. The working process of the bees is described as follows:

The employed bees bring loads of nectar from the food sources to the hive and then share the food source information with onlooker bees by dancing in a common area in the hive called dance area. The duration of a dance is proportional to the nectar content of the food source currently being exploited by the dancing bee. Onlooker bees need to watch numerous dances before choosing a food source, which tends to choose a food source according to the probability proportional to the quality of that food source. Therefore, the good food sources tend to attract more bees than the bad ones. A scout or onlooker bee may change into an employed bee when it finds a better source. An employed bee associated with a food source may become a scout or onlooker bee when the food source is exploited fully.

In the ABC algorithm, each food source represents a possible solution to the problem under consideration, and the nectar amount of a food source represents the quality of the solution. The ABC algorithm assumes that there is only one employed bee for every food source, i.e. The number of food sources is same as the number of employed bees. The employed bee of an abandoned food source becomes a scout bee and as soon as it finds a new food source, it becomes an employed bee again. The ABC algorithm is an iterative algorithm. It starts by associating all employed bees with randomly generated food sources (solution). Then, every employed bee moves to a new food source in the neighbourhood of its currently associated food source and evaluates its nectar amount (objective value) during iterations. When the employed bees completes the process, they share the nectar information of the food sources with the onlooker bees and the number of onlooker bees to be sent to the food source found by the employed bee is proportional to the nectar amount of that food source.

The probability p_i of selecting a food sources i is determined by the following expression:

$$p_i = \frac{fit_i}{\sum_{i=1}^{l} fit_i} \tag{1}$$

Here, fit_i is the objective value of the solution represented by the food sources, i and l is the total number of food sources. Clearly, good food sources will attract more onlookers than the bad ones.

3.1 Adaptive ABC Algorithm

In adaptive ABC algorithm, new food sources are generated by accomplishing the crossover and mutation operations. The crossover and mutation are the genetic algorithm operations. In this proposed method, crossover and mutation operator is added after the employed bee phase of Artificial Bee Colony algorithm.

ABC algorithm has four phases namely initialization phase, employed bees phase, onlooker bees phase and scout bees phase, adding 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 search for 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. 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, food source x_{ij} is randomly selected and replace its one of the dimension value by random number generated in between lower and upper bound value of the food source.

The procedure that followed in the adaptive ABC algorithm is demonstrated in Fig. 4.

Initialization phase
Step 1: Initialize input parameters, including the number of food sources or the number of employed bees e_b, number of onlooker bees o_b and number of scout bees s_b.
Step 2: Initialize populations by generating random food sources f_s.
Step 3: Calculate the fitness value $F(f_s)$ of each food source f_s and then determine the best food resource bf_s.
Employed bee phase
Step 4: For every employed bee, generate a new food source $Nf_s(e_b)$ by using the crossover and mutation operations.
Step 5: Calculate fitness value $F(f_s(e_b))$ for every newly generated food sources and compute the best food source.
Onlooker bee phase
Step 6: For every onlooker bee, generate a new food source $Nf_s(o_b)$ by using the crossover and mutation operations.
Step 7: Calculate fitness value $F(f_s(o_b))$ for every newly generated food sources and compute the best food source.
Scout bee phase
Step 9: Initialize scout bees with random solutions and compute fitness value $F(f_s(s_b))$ for these random solutions.
Step 10: Find the best scout bee among the randomly generated food sources using the fitness value $F(f_s(s_b))$.
Step 11: The scout bee's best food source $B(f_s(s_b))$, the employed bee's best food source $B(f_s(s_b))$ and the onlooker bee's best food source $B(f_s(s_b))$ are compared based on their fitness values.
Step 12: Among these food sources, the best food source is stored in the scout bee's phase and remaining food sources are given to the next iteration.
Step 13: The process is repeated until the stopping criterion is met. Then, the best food source is obtained with its objective value from the scout bee's phase.

Figure 4: The procedure of Adaptive ABC algorithm

3.2 Proposed Adaptive ABC technique for Job Scheduling

Our proposed technique finds out the way for the assignment of jobs to the resources by achieving minimum completion time. The adaptive ABC algorithm is initiated by generating food sources f_s. The food sources are represented as $f_{si} = \{R_1, R_2, \cdots R_k\}; k \in M$, where i represents the number of generated food sources. In this food source, each resources have the allocated jobs based on their execution and processing times e_n and $p_{n,m}$.

3.2.1 Crossover and mutation in the employed bee phase

In our ABC algorithm, employed bees perform global exploration with multiple different neighbourhoods for promising food sources over the entire region. Since the JSP consists of machine assignment and operation sequence, we will design the crossover and mutation operators to evolve the machine assignment and operation sequence.

3.2.2 Crossover and mutation for machine assignment

To evolve the machine assignment, two crossover and mutation operators are applied with equal probability, i.e., the two-point crossover and mutation and uniform crossover and mutation. These crossover and mutation operators only change the machine assignment but not change the operation sequence. To the two feasible parents, the offspring generated by these crossover and mutation operators is still feasible

3.2.3 Mutation for machine assignment

To enhance the exploration capability in the employed bee search phase, a mutation operator for machine assignment is proposed and embedded in the ABC algorithm. To reduce the computation load, the following mutation procedure is used in the ABC algorithm with a probability 50%:

Step 1. Randomly generate an integer I from 1 to n, where n is the total number of operations
Step 2. Randomly select I positions from the machine assignment vector
Step 3. For each selected position, replace the machine with a different machine randomly chosen from the candidate machine set (no change happens if the set only includes one machine)

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the proposed algorithm is evaluated under five different job datasets and the results are compared against the conventional algorithms such as ABC, GA and, ABC-GA. The results that are obtained under different experiments are given in following Tables I and Figures 5 to 8.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 1: Job Completion Time for Job Dataset I by the proposed with (i) 50 iterations, (ii) 100 iterations and (iii) 200 iterations, ABC, GA and ABC_GA algorithm with the number of 10 jobs and resources are 50,100,150,200.

No of jobs	No of Resource	Test Runs	ABC	GA	ABC_GA
10	50	50	180	190	160
		100	210	185	161
		200	225	160	127
10	100	50	240	210	224
		100	320	222	227
		200	315	218	183
10	150	50	190	150	96
		100	180	112	98
		200	160	98	74
10	200	50	220	120	110
		100	210	110	99
		200	180	110	91

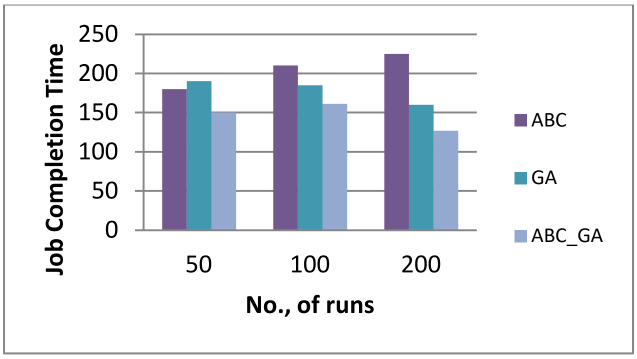

Figure 5: Mean Job Completion Time for individual Job Datasets proposed with 50 iterations, ABC,GA and ABC-GA .

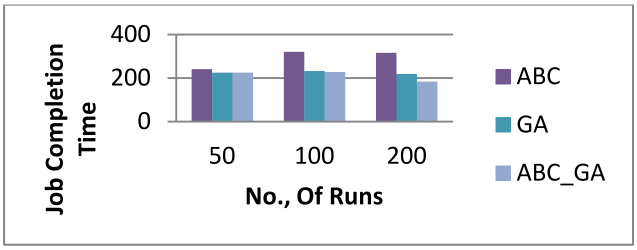

Figure 6: Mean Job Completion Time for individual Job Datasets proposed with 50 iterations, ABC, GA and ABC-GA.

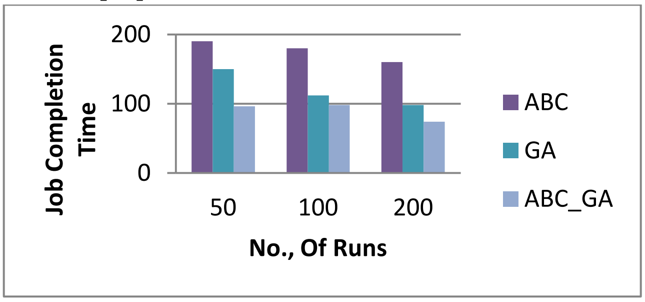

Figure 7: Mean Job Completion Time for individual Job Datasets proposed with 50 iterations, ABC, GA and ABC-GA.

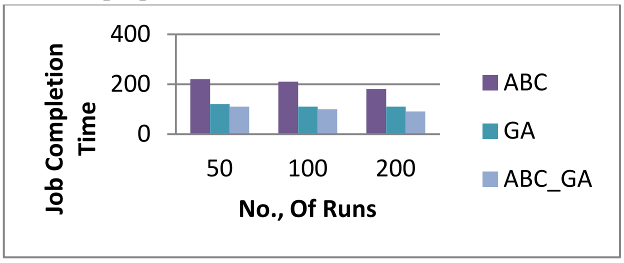

Figure 8: Mean Job Completion Time for individual Job Datasets proposed with 50 iterations, ABC, GA and ABC-GA.

V. CONCLUSION

In this paper, an adaptive ABC technique was proposed to allocate the available jobs to the exact resources. The proposed method has achieved the minimum completion and makes span time. The drawbacks of existing techniques were solved by considering some efficient factors in job scheduling process. Thus, the proposed technique has achieved high performance in allocating the available jobs to the precise resources and also attained a high efficiency. The performance of the proposed job scheduling technique was analyzed with two hybrid techniques namely ABC and GA with experimental results that proves, that the proposed job scheduling technique has attained high accuracy and efficiency than the two hybrid techniques. Hence, the proposed adaptive ABC job scheduling technique is capable of finding the optimal jobs to the resources and also achieving the minimum completion time.

REFERENCES

1. Dror G. Feitelson, Larry Rudolph and Uwe Schwiegelshohn, "Parallel Job Scheduling -A Status Report", In Proeedings of the Conference on JSSPP, pp.1-16, 2004.
2. Ivan Rodero, Francesc Guim and Julita Corbalan, "Evaluation of Coordinated Grid Scheduling Strategies", In Proceedings of 11th IEEE International Conference on High Performance Computing and Communications, Seoul, pp. 1-10, 2009.
3. Oliner, Sahoo, Moreira, Gupta and Sivasubramaniam, "Fault-aware Job Scheduling for BlueGene/L Systems", In Proceedings of 18th International Parallel and Distributed Processing Symposium, 2004.
4. Grudenic and Bogunovi, "Computer Cluster Scheduling Algorithm Based on Time Bounded Dynamic Programming", In Proceedings of the 34th International Convention on MIPRO, 2011, Opatija, pp. 722-726, 2011
5. Abdelrahman Elleithy, Syed S. Rizvi and Khaled M. Elleithy, "Optimization and Job Scheduling in Heterogeneous Networks ", International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, 2008
6. Zhang, Franke, Moreira and Sivasubramaniam, "A Comparative Analysis of Space- and Time-Sharing Techniques for Parallel Job Scheduling in Large Scale Parallel Systems", pp. 1-33, 2008.
7. Surekha and Sumathi, "Solution to the Job Shop Scheduling Problem using Hybrid Genetic Swarm Optimization Based on (λ, 1)-Interval Fuzzy Processing Time", European Journal of Scientific Research, Vol. 64, No. 2, pp. 168-188, 2011.
8. Bin Cai, Shilong Wang and Haibo Hu, "Hybrid Artificial Immune System for Job Shop Scheduling Problem", World Academy of Science, Engineering and Technology, Vol. 59, No. 18, pp. 81-86, 2011.
9. Mohammad Akhshabi, Mostafa Akhshabi and Javad Khalatbari, "Parallel Genetic Algorithm to Solving Job Shop Scheduling Problem", Journal of Applied Sciences Research, Vol. 1, No. 10, pp. 1484-1489, 2011.
10. Manish Gupta, Govind sharma, "An Efficient Modified Artificial Bee Colony Algorithm for Job Scheduling Problem", International Journal of Soft Computing and Engineering (IJSCE), Vol. 1, No. 6, pp. 291-296, January 2012.
11. Hadi Mokhtari, "Adapting a Heuristic Oriented Methodology for Achieving Minimum Number of Late Jobs with Identical Processing Machines", Research Journal of Applied Sciences, Engineering and Technology, Vol. 4, No. 3, pp. 245-248, 2012.
12. Elnaz ZM, Amir MR, Mohammad R, Feizi D (2008). Job Scheduling in Multiprocessor Architecture Using Genetic Algorithm. Proc. IEEE, pp. 248-250.
13. Thanushkodi K, Deeba K (2009). An Evolutionary Approach for Job Scheduling in a Multiprocessor Architecture. CiiT Int. J. Artif. Intell. Syst. Mach. Learn., 1(4).
14. Tung-Kuan L, Jinn- Tsong T, Jyh-Hong C (2005). Improved genetic algorithm for the job-shop scheduling problem. International Journal Advanced Manufacture Technology (Spiringer), pp. 1021-1029.