International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 2, Issue 4(April 2013), PP.39-45
www.irjes.com

Bee foraging behaviour techniques for grid scheduling problem

Sana Alyaseri[1], Ku Ruhana Ku-Mahamud[2]
[1]*Nizwa College of Technology, Oman*
[2]*School of computing, Universiti Utara Malaysia, Malaysia*

ABSRACT: *Grid computing is the infrastructure that involves a large number of resources like computers, networks and databases which are owned by many organizations. These resources are collected together to make a huge computing power. Job scheduling problem is one of the key issues in grid computing and failing to look into grid scheduling results in uncompleted view of the grid computing. Achieving optimized performance of grid system, and matching application requirements with available computing resources, are the objectives of grid job scheduling. Bee colony approaches are more adaptive to grid scheduling due to high heterogeneous and dynamic nature of resources and applications in grid. These algorithms have shown encouraging results in terms of time and cost. This paper presents some resent research activities inspired by bee foraging behavior for grid job scheduling especially ABC and BCO approaches. Different original studies related to this area are briefly described along with their comparisons against them and results. The review summary of their derived algorithms and research efforts is done.*

Keywords: *Artificial Bee Colony (ABC), Bee Colony Optimization (BCO), Computational grid, Gird computing, Grid, Job Scheduling*

I. INTRODUCATION

The demand for computing power is increasing as a result of rapid improvements of sciences like physics, biomedicine and others. Despite the fact that the amount of computing power available to both researches and business is growing at a significant rate, the fulfillment of the requirements of the data and compute intensive applications, and sophisticated problems is still a challenge. For instance, the high energy physics and computational genomics, the volume of increasing data is already measured in terabyte and will soon total petabytes which consequently increase the demand for huge computing power [1].

Basically, the grid is an emerging infrastructure describing the ability to pool and share the information technology resources in global environment. It also enables a seamless, secure, transparent and simple access to a vast collection of many different types of hardware and software resources including compute nodes, software codes, data repositories, storage devices, graphics and terminal devices, and instrumentation and equipment [2].

One of the key challenges for the grid computing researchers is to find the efficient allocation of resources to the jobs submitted by users. Job scheduling is the process of mapping jobs into particular available resources. Therefore, the coordination and allocation of resources for efficient execution of user jobs is the main objective of grid job scheduling problem [3].

Generally a grid scheduling problem is a challenge due to many constraints and optimization criteria in a dynamic environment. Therefore it faces a very complex and computationally hard process. Heuristics have been studied for a large number of optimization problems. They are suitable to deal with different types of grid scheduling like immediate and batch scheduling, multi objective, decentralized and hierarchical complex grids [4].

Honey bees are one of the most well studied social insects. In the last few years a lot of research based on different bee behaviors has been carried out to solve complex combinatorial and numerical optimization problems [5]. Many features of bees' behavior like their memories, navigation systems, group decision making processes, the bee dance (communication) and bee foraging inspired the researchers to mimic them in algorithms [6].

Honey bees are social insects living as colonies and there are three kinds of bees in a colony, namely, the queen bee, the workers and the scouts. Research in bee colony behavior can be divided into two major groups. The first group examined bee colony foraging behavior and produced some approaches like the Artificial Bee Colony (ABC), Bee Colony Optimization (BCO) and Virtual Bee Algorithms (VBA). However, VBA is proposed for solving engineering application therefore it is not included in this study. The second group of research was inspired by the marriage principles in bee colonies and devised several approaches, like Marriage in Honey-Bees optimization (MBO) and honey bees mating optimization (HBMO). Additionally, there are a few studies of approaches based on the evaluation of queen bee which are classified as an improvement of the genetic algorithms [5].

This paper has been organized as follows. In section I the introduction is presented. Job scheduling problem is explained in section II. Section III shows bee colony in nature. In section IV Artificial bee colony is discussed. Then, bee colony optimization is described in section VI. The review summery is clarified in section V. Finally the conclusion and future work is presented in section VII.

II. JOB SCHEDULING PROBLEM

Grid job scheduling is defined as the procedure of making scheduling decisions including assignment of jobs to the resources over multiple grid sites. Moreover grid scheduling can be shown as a whole family of problems, due to the many parameters and criteria intervening in the scheduling [7].

Hence the grid scheduling is a very complicated problem considered as multiple objective functions problem. It's obvious that in grid system two major parties namely the resource customers who send the various applications and the resource providers who share their resources in grid. The resource providers commonly have different motivations when connect to the grid. These motivations are represented by objective functions in scheduling [8].

The objective functions can be divided into two types, namely, application centric and resource centric [9]. The scheduling algorithms adopting the application centric objective plan to optimize the performance of each individual job which leads to reduce makespan. Obviously, most scheduling strategies in grid try to reduce the Makespan, which is commonly used as an objective function in many scheduling algorithms. However, the Makespan is the time spent from the beginning of the first task of the job to the end of the last task of the job [10].

Grid job scheduling depends on several optimization criteria to determine efficient performance of the grid computing including makespan, flowtime, resource utilization, load balancing, matching proximity, turnaround time, total weighted and completion time. However, some of these criteria are opposing to each other, such as minimizing of makespan disagrees with the maximizing of resource usage and response time. The makespan and flowtime are the most common optimization criteria in grid job scheduling problem and extensively studied in the literature [7][11]. The makespan measures the throughput of the grid system, and flow time measures its quality of service (QoS). The scheduling commonly tries to reduce makespan and flowtime.

There are three common types of grid scheduling structures. The first type is the centralized scheduling which includes a single job scheduler and a single point to collect the whole information about the grid. This type of scheduling suffers from lack of scalability and fault tolerance. Therefore, it is not considered in the large scale grids. The hierarchical scheduling is the second type which usually involves two schedulers, one in the global level, i.e., grid level, and the other in the local level, i.e. cluster level. It also suffers from the limited scalability and fault-tolerance but it is scaled well than the first one. The last type is the decentralized scheduling which includes many distributed schedulers interacting with each other to allocate jobs and there is no single point to collect the information. This type of scheduler is more adaptive for the grid systems but less efficient than others [12][7].

Commonly, three main phases or stages are involved in the grid job scheduling process. The first phase is the resources discovery which creates a list of possible resources. The next phase is the resource selecting and scheduling which considers collecting the best set of resources that meet the jobs requirements. The last phase is the job execution which includes file staging and clean up as shown in Figure 1 [13][7]. Nevertheless, the second phase is the concern of this research.

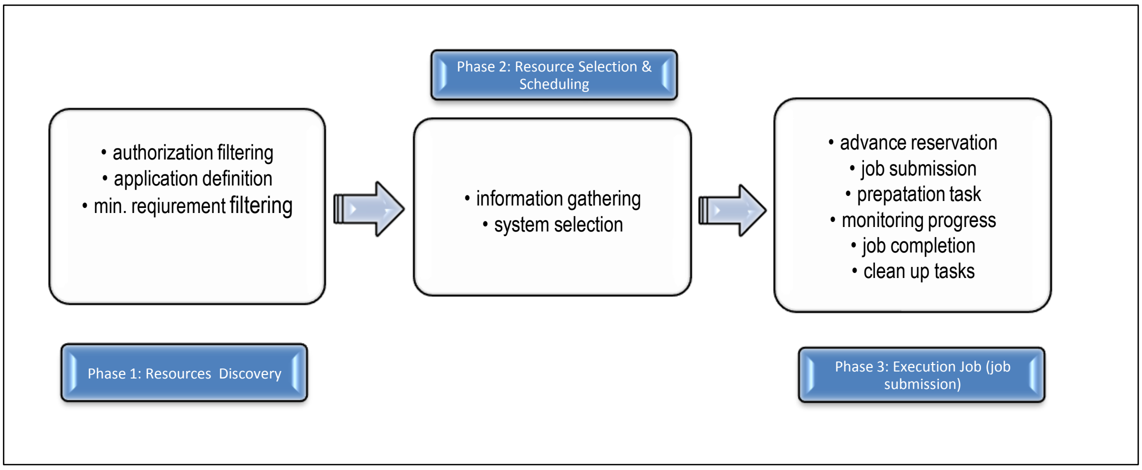

Figure 1: Three Phases of Scheduling Architecture

Generally the tradition parallel and distributed systems have a lot of scheduling problems. Intensive studies have been done in scheduling problems in these systems. These include symmetric multiple processors (SMP), massively parallel processors computers (MPP), and cluster of workstations (COW) [14]. In practice, these traditional scheduling algorithms generate poor grid schedules because of the restrictions imposed on the underlying traditional systems [15]. The first restriction is where all resources are administrated by a single domain. The second highlights that the scheduler controls all resources giving a single system image. Other restrictions are the resource pool is changeless, computation and related data should reside in the same site and the scheduler manages the contention caused by incoming jobs. All these restrictions are not supported by the grid environment. However, grid has many single properties like a dynamic environment, running by different operating systems and a variety of connecting resource types. Therefore the design of scheduling algorithms is a great challenge in the grid [16].

Commonly the key challenge in grid computing involves many constraints and optimization criteria in the dynamic grid environment. Grid scheduling has to make scheduling decisions of resources over multiple administrative domains. This can include searching multiple administrative domains to use a single machine for multiple jobs, or searching multiple resources at a single site or multiple sites for a single job matching. The objective of grid scheduling is to find an optimum schedule in heterogeneous system. It is in general a NP-hard problem as it is a very complex and computational hard process [17].

III. BEE COLONIES IN NATURE

Social insect colonies like bee colonies can be defined as dynamic systems where the bees adjust their behavior according to the information collected from the environment. The individual bees do not perform all the tasks due to the labor division system, which is known in insect colonies [18]. However, the foraging behavior in a bee colony is composed principally of two types of behavior, navigating behavior and recruitment behavior [19].

Bees have already gained knowledge about specific routs and landmarks. By using their celestial compass like sun azimuth, bees can navigate unfamiliar environment searching for food. They also need to be able to discover previously unknown parts of the terrain [20]. Therefore, bees use a procedure called path integration (PI) to measure the directions and distances traveled on each part of the search. PI helps the bees to compute their present location from their past trajectory. As a result, the bees can return to their starting point by choosing the direct path rather than returning via their outbound trajectory [19].

Recruitment behavior in bees includes selecting food sources which are performed by recruiting members of the colony. Bees whose job is to seek for food sources inform their colony mates about the distance and the direction of these food sources by performing their waggling dance. The waggling dance is executed on the vertical combs in the hive see Figure 2 [20]. The waggle dance represents a communication tool among bees. When a bee finds a rich food source, it returns to the hive and starts to dance in a particular pattern. Through this informative dance, the bee has indicated its hive mates the direction and distance of the food source [21].

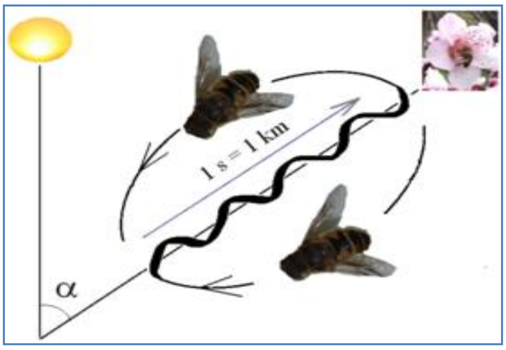

Figure 2: Bees dancing to show position and quality of the food source

IV. ARTIFICIAL BEE COLONY

Artificial bee colony (ABC) is proposed as an approach based on honey bee swarm for solving multidimensional and multimodal optimization problems [22]. The proposed approach simulates the bee foraging behavior and the exchange of information amongst bees about good food sources and uses a population of different types of bees to find the optimal solution. Nevertheless, ABC works partially like and partially differently from real bee colonies in nature [24]. It's obvious that ABC is being used lately by many researchers in job grid scheduling, due to its encouraging results in solving NP-problems like travel salesman problem and classification problems [25].

There are three groups of bees in artificial bee colony (ABC): employed, onlookers and scouts bees. The employed bees randomly search for food source (potential solutions). By dancing in dance area in the hive, the employed bees share the information of food source (the quality of solution). The onlooker bees are waiting in dance area to evaluate various dances before choosing a food-source position, according to the probability proportional to the quality of that food source. The scout bees randomly look for possible new [22].

Basically there are two equations (1,2) support the artificial bee colony algorithm mentioned below [25]:

$$p_i = \frac{fit_i}{\sum_{k=1}^{SN} fit_k} \qquad (1)$$

Where p_i represents the probability of an onlooker selects a solution *i* (food source) based on the probability value associated with that solution (food source), fit_i is the fitness value of solution *i* (food source).

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

Where *i*, *k* € {1,…..,SN}, *j* € {1…..n} are randomly parameters but *k* has to be different from *i* , and v_i is the new candidate solution (food source) generated by using the current food source x_i and a randomly chosen food source x_k from the population. ∂_{ij} is a random number between [-1,1] which is control on the distance of neighbor food source around x_{ij}.

There are also three control parameters in the ABC: *SN* is represented as the number of food sources which is equal to the number of employed or onlooker bees, the value of "limit" and the maximum cycle number (MCN). "limit" is important control parameter and is defined as the value of predetermined number of cycles in ABC which is represented the number of abandoned solutions (food sources). Assume that the abandoned solution (food source) is x_i and *j* € {1…..n}. Then the scout bee finds the new food source to be replace with x_i, this can defined in the equation(3).

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \qquad (3)$$

The main steps of algorithm are following [5]:

```
1: Initialize Population
2: REPEAT
3: Place the employed bees on their food sources
4: Place the onlooker bees on the food sources depending on their nectar amounts
5: Send the scouts to the search area for discovering new food sources
6: Memorize the best food source found so far
7: UNTIL requirements are met
```

V. BEE COLONY OPTIMIZATION

Further improvement and generalization are incorporated with the bee system to generate bee colony optimization meta-heuristic (BCO) for solving deterministic combinatorial problems as well as uncertainty combinatorial problems. The basic concepts of BCO were introduced by Lucic and Teodorovic as a general algorithmic framework for solving NP-problems [26] [27] [28]. As a new heuristic approaches to the others the BCO has shown its superiority than other heuristics approaches in solving large and complex real world problems like routing and scheduling [29].

The framework of BCO includes a colony of artificial bees searching collaboratively for the optimal solution of a given problem. Each artificial bee generates one solution to the problem. Forward pass and backward pass are two alternating phases constituting a single step in the BCO algorithm. In the forward pass the artificial bee flies to create one partial solution. The artificial bees have to exploring the search space and choose new partial solutions by the roulette wheel [23].

When the backward pass starts the bee returns to the hive for dancing used as a communication technique to exchange information about the quality of the obtained bee solutions. Depending on the information obtained from dancing some bees become recruits and the others are uncommitted bees. The recruits fly to the next stage as the problem already has been divided into stages for extending their partial solutions. The uncommitted bees decline their partial solutions and fly on the same partial path of the recruits then expanding this path by selecting new partial solutions in the next stage [30][31].

After forward pass is completed and all bees come back to the hive (dancing area), they will start exchange their information to other bees. In dancing area the amount of each bee's loyalty (loyalty decision to be recruiting) could be calculated by equation (4):

$$P_b^{u+1} = e^{-\frac{O_{max}-O_b}{u}} \qquad (4)$$

Where b=1,2…….B and B is the number of bees, *u* the forward pass counter taking value 1,2……*NC*, *NC* is the number of forward (backward) in each iteration.
O_b € [0,1] is calculated by equation (5):

$$O_b = \frac{Y_{max} - Y_b}{Y_{max} - Y_{min}} \qquad (5)$$

Where Y_b is the partial solution result of *b*, Y_{max} and Y_{min} are the largest and smallest partial solution results.
The probability recruitment for partial solution by any uncommitted bee is calculated by equation (6):

$$P_b = \frac{O_b}{\sum_{k=1}^{R} O_k} \qquad (6)$$

Where *b*=1,2…….*R*, and *R* is number of uncommitted bees.
The main steps of BCO algorithm are following [31]:

```
B: the number of bees involved in the search.
NC: the number of forward (backward) passes in a single iteration.
Do
1- Initializing: every bee is set to empty solution
2- Repeat
    //forward pass
      Set b=1;
   a)    Repeat
            (1) Evaluate all possible moves;
            (2) Choose one move using the roulette wheel.
             (3) b=b+1;
          Until b >B
         //backward pass start, all bees are back to the hive.
    b) For each bee
          Evaluate (partial/complete) solution for bee b;
     c) Sort the bees according their objective function value
     d) For each bee
           The Loyalty decision is taking by using the roulette wheel;
     e) For each bee
        If (bee is follower), choose a recruiter by the roulette wheel.
3. Evaluate all solutions and find the best one.
While stopping criteria is not satisfied
```

VI. REVIEW SUMMARY

Based on the above explanation of the resent bee colony approaches which are inspired by bee foraging behavior, some derived algorithms are tabularized in Table (1) along with some their features. Different techniques are added to improve the work of these derived algorithms for solving grid scheduling. Most developments have involved improving the local search of the original approaches. Although, BCO approaches are widely used to solve different scheduling problems, most studies are paid attention to ABC approaches. The majority of derived bee colony approaches have not concerned to the common dynamic environment of grid. Moreover the scheduling problem in the large grids is not totally integrated in the surveyed approaches. However the grid scheduling problem includes different issues related to the nature of grid environment like requirements of users, the dead line of complete task, the priority of tasks, cost and security issues. These issues need much attention when develop a new algorithm based on bee colony approaches for grid scheduling.

Approaches	Papers/ derived approaches	improvements	objectives	Environment/ features	Comparison with others
ABC	BABC, EBABC1 and EBABC2 [32]	using the binary representation, after that used flexible ranking strategy (FRS) to enhance BABC and generate EBABC	achieve a balance between diversification and convergence of the search process	grid computing/ independent jobs	genetic algorithm (GA), simulated annealing (SA) and particle swarm optimization (PSO)
	ABC for grid scheduling [33]	Applying single shift neighborhood (SSN) strategy and double shift neighborhood (DSN) strategy to each employed bee according to the fitness value of bees. Ejection chain Neighborhood (ECN) also has applied to employed bees based on probability value.	Makespan, cost and reliability	grid computing/ independent jobs	The results have compared with ACO in different task length and resource speed.
	An efficient modified ABC [34]	Mutation and crossover operators are added after employed phase and onlooker phase of the original ABC respectively.	Decrease the maximum completion time	grid computing/ independent jobs/ small grid	Genetic algorithm (GA)
	MOABC [35]	Numerical optimization techniques based on population are added	Minimize the cost and time execution	Grid computing / dependent jobs (workflow)	With scheduler DBC and WMS
BCO	JDS-BC [36]	Original BCO is improved methodically in this research to meet the special needs of scheduling, computational grid and data intensive grid	Minimize makespan and total datafile transfer time	Applying both computational grid and data intensive grids, independent jobs	DIANA grid scheduler and FLOP scheduler
	Bee colony task scheduling algorithm [37]	Used FCFS, SJF and LJF to improve the original BCO	Minimize makespan, satisfy the deadline and priority requirement	Computational grid	FCFS, SJF and LJF priority rule algorithm.

Table 1: the approaches of ABC and BCO in different factors and parameters

VII. CONCOLUTION AND FUTURE WORK

The objective of grid computing is to solve a complex task in shorter time and utilizes the hardware efficiently. To get this objective best job scheduling strategies have to be employed. Heuristics based population algorithms have shown significant results in solving many optimization problems. The bee colony approaches as member in this group also have presented superior results in solving a lot of optimization problems such as scheduling. The brief survey is given for the ABC and BCO approaches for grid scheduling in order to show the researchers general features of these algorithms. Although, the techniques show very promising results, the topic is still in its infancy. However, ABC and BCO need more improvement to give significant results to solve grid scheduling especially in large dynamic grid. The future work will be concerned with the development of significant scheduling algorithm based on bee colony behavior which is heterogeneous and works in dynamic environment of grid.

REFRENCES

[1] I. Alexandru and E. Dick H J, Grid Computing Workloads, Internet Computing, IEEE, 15(2), Digital Object Identifier: 10.1109/MIC.2010.130, 2011, 19 – 26.
[2] B. Wikinson, Grid computing: Techniques and applications (CR press LLC), 2008.
[3] K. Christodoulopoulos, V. Sourlas, I. Mpakolas and E. Varigos, A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in grid network. Computer communication, 32, 2009, 1172-1184.

[4] A.Chandak, B. Sahoo & A. Turuk, Heuristic task allocation strategies for computational grid. International Journal advanced networking and applications, 5(2), 2011, 804-810.

[5] S. Bitam, M. Batouche and E. Talib, A survey on bee colony algorithms. 24th IEEE International Parallel & Distributed Processing Symposium 2010 NIDISC Workshop, Atlanta, Georgia, USA, 2010.

[6] D. Karaboga and B. Akay, A survey: algorithms simulating bee swarm intelligence. Artificial Intelligent Review, 31(1-4), 2009, 61-85.

[7] F. Xhafa and A. Abraham, Meta-heuristics for grid scheduling problems. F.Xhafa, A. Abraham (Eds.). Meta for scheduling in distributed Computer Environment, SCI 146, 2008, pp. 1-37, springer-verlag Berlin Heidelberg.

[8] F. Pop, and V. Cristea, Optimization of scheduling process in grid environments. U.P.B. Sci. Bull., Series C, 71(4), 2009.

[9] M. Abdullah, M. Othman, H. Ibrahim & S. Subramaniam, Improved Genetic Algorithm for Scheduling Divisible Data Grid Application. Proceedings of the 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, 14-17 May 2007, Penang, Malaysia.

[10] F. Xhafa, J. Kolodziej, L. Baroll and A. Fundo. GA+TS hybrid algorithm for independent batch scheduling in computational grids. In 2011 international conference on network based information systems, 2011.

[11] R. Sahu and A. Chaturvedi, Many-Objective Comparison of Twelve Grid Scheduling Heuristics. In International Journal of Computer Applications, 13(6), 2011,0975-8887.

[12] R. Kumar, and N. Kaur, Job Scheduling in Grid Computers. In Proceeding of the National Conference on Challenges & Opportunities in Information Technology, COIT 2007, Mandi Gobindgarh. March 23, 2007.

[13] J. Schopf , A General architecture for scheduling on the grid. Special issue of JPDC on grid computing, 2002,1000-1002.

[14] F. Dong, and S. Akl, Scheduling algorithms for grid computing: state of the arts and open problems, Technical Report No. 2006-504, School of Computing, Queen's University Kingston, Ontario, January 2006.

[15] F. Berman, High-performance schedulers. In Ed. By Foster I. & Kesselman, C. Kaufman, M. publishers, The grid: blue print for a future computing infrastructures, 1998.

[16] M. Raju, T. Divya, and T. Rao, T, Efficient resource scheduling in data grid. Proceedings of the international conference on information science and applications ICISA, 2010, Chennai, India

[17] G. Kaur, Makespan optimization algorithm for scheduling in grid. Msc thesis, Thapar university, Patiala, India, 2008.

[18] D. Teodorovic and M. Dell'Orco, Bee colony optimization - a cooperative learning approach to complex transportation problems. In Advanced OR and AI Methods in Transportation, Proceedings of the 10th Meeting of the EURO Working Group on Transportation, 2005, 51–60, Poznan, Poland.

[19] N. Lemmens, S. Jong, K. Tuyls, and A. Nowe, A bee algorithm for multi-agent systems: recruitment and navigation combined. In Proceedings of ALAg, an AAMAS workshop, 2007, 66-70, Honolulu, Hawaii, USA.

[20] F. Dyer, Spatial memory and navigation by honeybees on the scale of the foraging range. The journal of experimental biology, 1996, 147-154.

[21] L. Wong, M. Low and C. Chong, Bee colony optimization with local search for traveling salesman problem. In Proceeding of the 6th IEEE International Conference on Industrial Informatics, 2008, 1019 -1025.

[22] D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[23] T.Davidovic, M. Selmic and D. Teodorovic, Scheduling Independent Tasks: Bee Colony Optimization Approach. In Proceeding 17th Mediterranean Conference on Control and Automation, 2009,1020-1025, Makedonia Palace, Thessaloniki, Greece, June 24-26, 2009.

[24] D. Karaboga and C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm. Applied soft computing, 11, 2011,652-657.

[25] S. Kim, J. Byeon, H. Liu, A. Abraham and S. McLoone, Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization, Springer-Verlag Berlin Heidelberg 2012.

[26] P. Lucic and D. Teodorovic, Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, 2001, 441–445, Sao Miguel, Universidade dos Agores.

[27] P. Lucic and D. Teodorovic, Transportation modeling: an artificial life approach. In Proceedings of the 14th IEEE 'International Conference on Tools with Artificial Intelligence IEEE Computer Society, 2002, 216–223, Washington, DC, 4–6 November 2002.

[28] P. Lucic, and D. Teodorovic, computing with bees: attacking complex transportation engineering problems, International Journal on Artificial Intelligence Tools, 12, 2003, 375–394.

[29] X. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley, first edition, June 10, 2011.

[30] T. Davidovic, M. Selmic & D. Teodorovic, Bee colony optimization for scheduling independent tasks. In Proceedings of the Symposium on Information Technology, 2009, YUINFO 2009, (on CD 116.pdf), Kopaonik, Serbia, March 08-11, 2009.

[31] D. Teodorović, T. Davidović, and M. Šelmić, Bee Colony Optimization: the application survey. ACM transition on computational logic, 2011, 1-20.

[32] K. Vivekanandan, D. Ramyachitra and B. Anbu, Artificial bee colony algorithm for grid scheduling. Journal of Convergence Information Technology, 6(7), July 2011.

[33] S. Kim, J. Byeon, L. Hongbo, A. Abraham and S. McLoone, Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization, In soft computing, 2012,1-16.

[34] M. Gupta & G. Sharma, An Efficient Modified Artificial Bee Colony Algorithm for Job Scheduling Problem, International Journal of Soft Computing and Engineering (IJSCE), 1(6), 2012, ISSN: 2231-2307.

[35] M. Arsuaga-Rios, Multi-objective Artificial Bee Colony for Scheduling in Grid Environments, 11th International conference on Intelligent Systems Design and Applications (ISDA), 2011.

[36] J. Taheri, Y. Lee, A. Zomaya and H. Siegel, A Bee Colony based optimization approach for simultaneous job scheduling and data replication in grid environments, Computers and Operations Research ,2011, doi:10.1016/j.cor.2011.11.012

[37] Z. Mousavinasab, R. Entezari-Maleki, A. Movaghar, A Bee Colony Task Scheduling Algorithm in Computational Grids, Digital Information Processing and Communications, Communications in Computer and Information Science, 188, 2011, 200-210.