

# Population based Heuristic Approaches for Grid Job Scheduling

Sana Alyaseri

Information Technology Department  
Nizwa College of Technology, Oman

Alaa Aljanaby

Computer Science Section  
College of Arts and Sciences  
University of Nizwa, Oman

## ABSTRACT

Several strategies and approaches have been proposed to provide quality solutions for the grid job scheduling problem. Recently population based heuristics approaches are used widely to solve this problem. These approaches have demonstrated a surprising degree of effectiveness for handling combinatorial optimization problems. In this paper, the population based approaches for grid job scheduling have been studied. The focus was on investigating the criteria that help in the selection of the best scheduling algorithm for a certain type of grid and also shedding the light on how to improve the available population based approaches.

## General Terms

Swarm Intelligence, Grid Job Scheduling

## Keywords

Genetic algorithms, Particle swarm optimization, ant colony optimization, artificial bee colony, grid job scheduling.

## 1. INTRODUCTION

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 [1]. The grid scheduling is an NP-complete problem and heuristic optimization approaches are the best methods to tackle the NP-complete problems. The heuristic approaches are classified into four categories in task scheduling, namely: population based heuristics, meta-heuristics, hybrid heuristics and economic based heuristics [2].

Population based heuristics are frequently employed in grid job scheduling and they are based on the biological behavior of the members of a population and employ Swarm Intelligence (SI) techniques. SI is defined as the group of techniques using a population of workers/particles/agents working together to find an optimal (or near optimal) solution to the problem in hand. Genetic Algorithms (GA), Artificial Bee Colony (ABC), ant colony optimization (ACO), genetic algorithm (GA) and particle swarm optimization (PSO) are successful applications of SI where the members of group are simulated honey bees, ants, genes and birds respectively. Self-organization and the division of labor are two fundamental bases that are sufficient to produce swarm intelligent behavior [3].

In this paper, a comparative study has been conducted between the population based heuristic approaches for job scheduling in batch mode. The rest of the paper is organized as follows. Section 2 provides some background for the grid job scheduling problem. The basic concept and the description

of the heuristic technique of the genetic algorithms, particle swarm optimization, ant colony optimization and artificial bee colony have been presented in Section 3 to Section 6 respectively. Section 7 presents a comparison between the mentioned population-based heuristic approaches. The conclusion is given in Section 8.

## 2. GRID 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 [4].

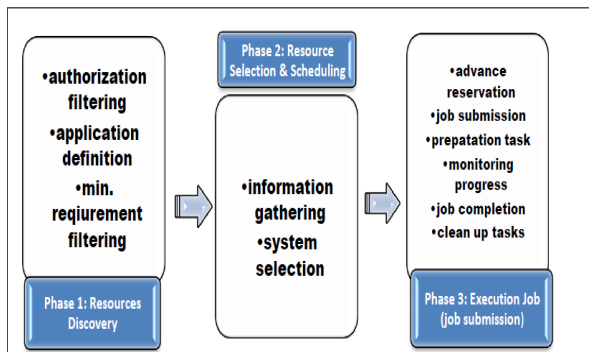
There are three common types of grid scheduling structures. The first type is the centralized scheduling which includes a single job scheduler and a single point to collect the whole information about the grid. This type of scheduling suffers from lack of scalability and fault tolerance. Therefore, it 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., the grid level, and the other in the local level, i.e., the 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 [5][4].

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 [6][4]. Nevertheless, the second phase is the concern of this paper. Figure 1 shows the phases of the scheduling process.

Generally the traditional parallel and distributed systems have a lot of scheduling problems. Intensive studies have been done on scheduling problems in these systems. These include symmetric multiple processors (SMP), massively parallel processors computers (MPP), and cluster of workstations (COW) [7].

In practice, these traditional scheduling algorithms generate poor grid schedules because of the restrictions imposed on the underlying traditional systems [8]. The first restriction is where all resources are administrated by a single domain. The second restriction is 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 environment [9].



**Figure 1: Three phases of scheduling architecture**

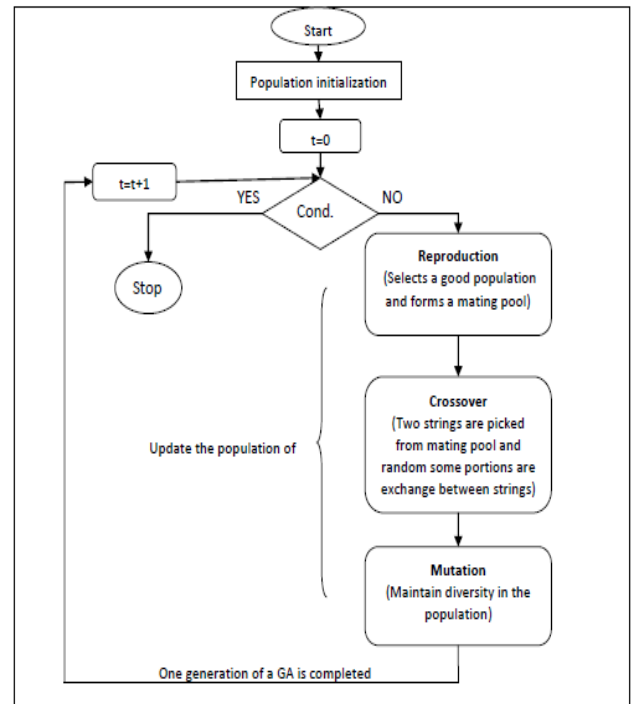
Commonly the key challenge in grid scheduling is the involvement of many constraints and optimization criteria in the dynamic grid environment. Grid scheduler has to make decisions of scheduling resources over multiple administrative domains. This may include searching multiple administrative domains to use a single machine for multiple jobs, or searching multiple resources at one or more sites for a single job matching. The objective of grid scheduling is to find an optimum schedule in a heterogeneous system. Grid scheduling is an NP-hard problem as it is a very complex and computationally hard process [10].

### 3. GENETIC ALGORITHMS

Genetic Algorithms (GAs) are stochastic search methods from evolutionary computing which are based on a biological evaluation system. Commonly GAs explore a solution search space to get the optimal solution [11]. Genetic algorithm techniques as shown in Figure 2 are represented using a string coding of fixed length where the solutions in the search space of GA are similar to the natural chromosomes and GA operators are similar to the genes operators [12].

One of the early works using GA in a computational grid for solving job scheduling was in 2000 by Abraham et al [13]. They have presented a study on using heuristics approaches inspired by nature including a GA hybridize with the Tabu Search (TS) algorithm and the Simulating Annealing (SA) algorithm individually. The results demonstrated that a hybrid GA is better than pure GA. Moreover, a GA hybridize with SA has given better convergence than a GA hybridize with the TS. However, GAs deal with a population of solutions whereas TS and SA are heuristic approaches deal with one solution at time.

Genetic algorithms have been widely used as approaches to improve the performance of job scheduling in grids. A novel genetic algorithm has been proposed to address the scheduling of decomposable data grid applications considering computation and communication at the same time. It depends on balancing data load automatically without requiring pre-decomposed jobs [14].



**Figure 2: GA technique**

In another research, the data intensive applications as target can be decomposed into multiple independent subtasks according to the decomposed data and be executed in parallel across multiple sites. This data is decomposed into multiple smaller data objects of arbitrary size and processed onto multiple virtual sites. The computation starts after the assigned data is totally transferred to the site. Then, the result of decomposed data and subtasks are gathered and returned the target. Therefore, the time cost includes the input data transfer time, the computation time and the output data transfer time [14].

GAs is also employed in job scheduling for large scale grid applications. Different GA's operators are examined in this approach to find which one performs the best. The work showed a robustness of implementation and an improvement of performance of static instances [15].

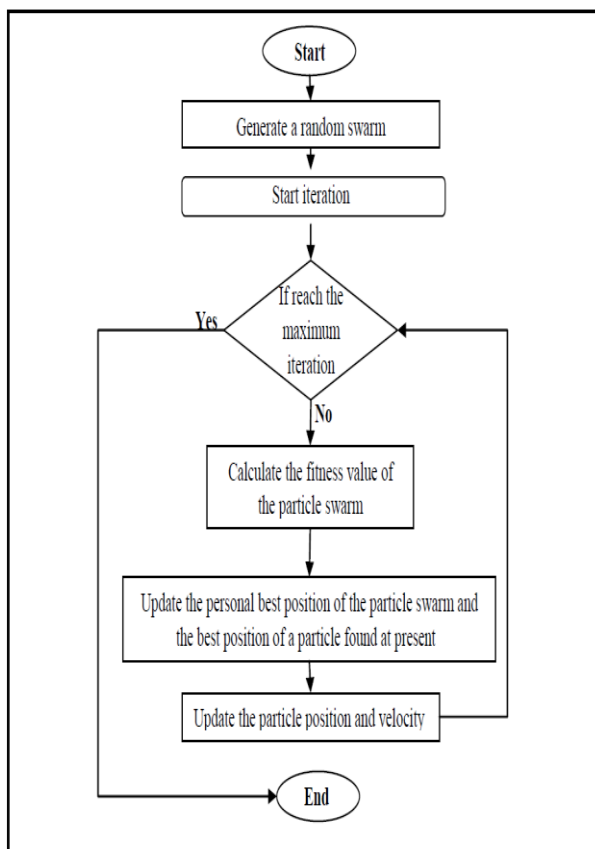
Besides, GA shows different results in data grid scheduling and much research has been done on it. It is employed in data grid scheduling by using co-scheduling that integrates the information about the data set and their replicas needed by the jobs to reach better performance in data grid. In different scenarios, the approach has been compared with the TS method and Max-Min method. The results showed that TS and GA are better than the Max-Min algorithm. However, the research indicated to the impact of including the information of replicas in the performance of scheduling algorithms in data grid [16].

GAs have also been integrated with SA and TS individually to improve their performance. The results showed that hybrid methods are better than the basic GA and have different performance rates amongst themselves [17]. Another research have employed GA in a new task scheduling algorithm to assign tasks to grid resources with the objective of minimizing makespan which leads to increase the throughput of the grid. The results were compared to the traditional algorithms like min-max, suffrage and min-min. the obtained results showed that the proposed algorithm finds the best suitable

task/resource pairs with lower makespan overcoming the weakness of the other algorithms [18].

#### 4. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is defined as one of latest evolutionary optimization techniques inspired by nature, it simulates the behavior of swarms like birds flocking and fish schooling [19]. It is an adaptive method that can be used to solve optimization problems. PSO as outlined in figure 3 conducts searches using a population of individual called particles, where, each particle flies in a problem search space to find an optimal or near optimal solution. The position of each particle is adjusted according to its own experience and the experience of its neighbors. PSO approach contains a local search method which is based on self experience and global search method which is based on neighbor experiences. By adapting these two methods, PSO attempts to balance between exploration and exploitation [20].



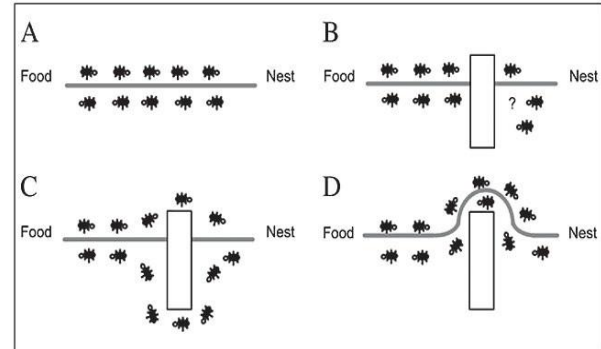
**Figure 3: Particle swarm optimization algorithm**

Many approaches based on PSO are introduced for grid job scheduling. PSO is presented as a novel approach to grid scheduling and has achieved a high convergence rate through the use of modified inertia parameters. The new algorithm was applied in the grid job scheduling and has performed better than the original approach [22].

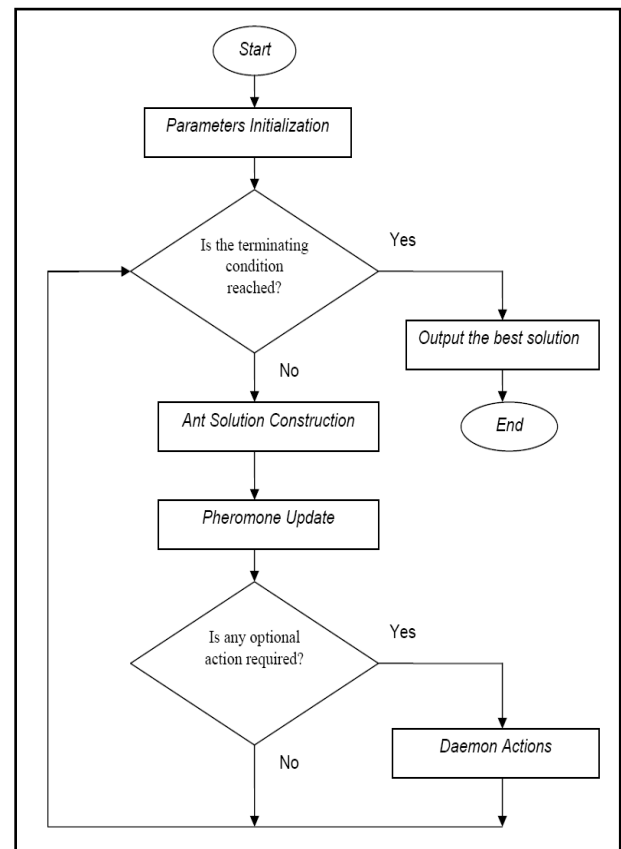
Furthermore, PSO is applied to solve the scheduling problems in a grid environment, the results show that PSO is efficient for job scheduling in computational grids [20]. The comparison between the performance of PSO and GA has been conducted and the results show that PSO is better than GA under the same conditions [23].

#### 5. ANT COLONY OPTIMIZATION

Ant System Algorithm was presented in 1996 by Dorigo [24] as a new heuristics predictive scheduling algorithm. The algorithm was inspired by real ants in their search for food, see Figure 4. The ACO algorithm has been applied to solve many NP-problems such as Travelling Salesman Problem (TSP) and Job Shop Scheduling (JSS) [25][26][27].



**Figure 4: Real ants search technique**



**Figure 5: Ant colony algorithm**

ACO has a number of sensitive parameters which should be defined according to the problem conditions [27]. ACO methods designed for job scheduling in computational grids have given a real time distributed system with no global control for the scheduler. Besides, the ACO can freely adopt with the system environment at run time [23].

A new heuristic algorithm for task scheduling has been designed based on the ant algorithm. The authors also designed simple grid simulation architecture for resource management and task scheduling to be used to test the new

algorithm. The results achieved were good in terms of response time, average resource utilization and task fulfillment proportions [28].

An efficient scheduling algorithm is still an urgent need because of the rapid growing of the grid computing technology which leads to more complex scheduling problems. Therefore, a general framework of grid scheduling based on ACO to improve the performance of grid scheduling was proposed in 2007. The traditional algorithms with the proposed approach were compared. The proposed approach developed the decision of scheduling by selecting the suitable processor for each job which led to minimizing the job tardiness time and efficient utilization of processors [29].

In large grids, ACO is employed by using a new heuristic technique for scheduling problems to provide an optimal or near optimal solution [25]. The proposed scheduler assigns an application to a resource from a pool of available resources by selecting the best match. The authors also presented in other research a further improvement of their approach by adding a local search [26][30]. Moreover, the free time of the resources and the execution time of the jobs to achieve efficient resource utilization and scheduling are taken into consideration in the modified approach. The results of the algorithm were compared with the Max-min algorithm and shows that the proposed algorithm is capable to give a high quality job scheduling.

Another modified ACO approach for grid scheduling has been achieved by modifying pheromone updating rule in the original ACO. This modifying led to a greater percentage of effective solutions than that of the existing ACO [31]. Another research work is also focused on defining the pheromone trail for producing a new version of the ACO. This implementation enables the grid scheduling to get the fastest machine resources to make short scheduled tasks [32].

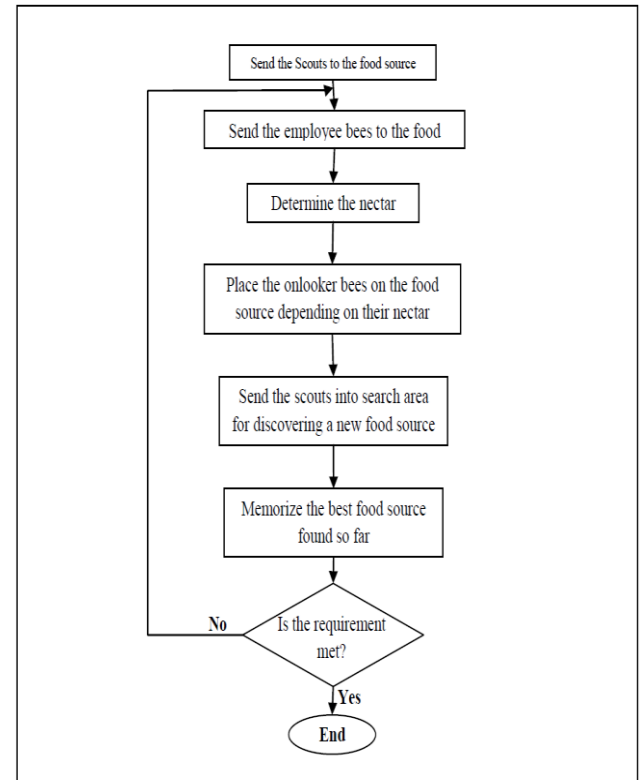
## 6. ARTIFICIAL BEE COLONY OPTIMIZATION

Artificial bee colony optimization is proposed as an approach based on honey bee swarm for solving multidimensional and multimodal optimization problems called artificial bee colony (ABC). The proposed approach simulates the bee foraging behavior and the exchange of information amongst bees about good food sources. There are three collections of bees in ABC: employed bees, onlooker bees and scouts. The bees in the colony are divided into two equal groups. The first one contains the employed bees and the second one contains the onlooker bees. For every employed bee there is only one food source and the employed bee whose food source has been over used by other bees becomes a scout. Although the algorithm was applied on very limited (only three) test functions, the results were very robust, at least for these functions [33], see figure 6.

ABC uses a population of different types of bees to find the optimal solution. The behavior which is used to find a food source is determined by the type of bee. The bee that is waiting in the dance region to make a decision about the best food source is called onlooker. Indeed, the scout bee is defined as the bee which flies unexpectedly in space looking for the food source and the employed bee is defined as the bee which goes to the selected food sources [34]. Nevertheless, ABC works partially like and partially different from real bee colonies in the nature [35][36].

It's obvious that ABS 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 [37]. Therefore, ABC algorithm have been implemented to solve job scheduling problems in grids. The proposed approach aims to reduce the makespan and utilize the resources in an efficient way. Besides, it has given a better QoS than ACO as well as reached optimal solutions. For transferring the jobs to resources, Single Shift Neighborhood (SSN), Double Shift Neighborhood (DSN) and Ejection Chain Neighborhood (ECN) techniques are applied and compared in this approach [38].



**Figure 6: Artificial bee colony optimization**

Some improvements have been made to the classical ABC to tackle job scheduling in grids by adding crossover and mutation operators as in the genetic algorithm. The improvements consist of using an additional crossover operator after the employed bee phase and mutation operator after onlooker bee phase of ABC. The new algorithm aims to reduce the maximum completion time. The results were encouraging after comparison with GA [39].

## 7. COMPARISON AND DISCUSSION

In fact, most genetic algorithms which were used for solving scheduling problem in grids have considered the assignment of scheduling several pre-decomposed tasks onto several computing nodes. It was found that the long execution time is one of the drawbacks of proposed GA [15]. The scheduler schema in computational grids takes longer to create when the GA approach is employed than other approaches like ACO and PSO [40].

**Table 1: Population based approach comparison**

criteria	GA	PSO	ACO	ABC
application	Wide	Limited	Moderate	Limited
Environment	Data grid & grid computing	Grid computing	Grid computing	Grid computing
Size of grid	Different size	Small	Large	Small
Objectives	Minimum makespan, minimum flow time and good load balance	Minimum completion time	Minimum Makespan, load balance	Minimum makespan
disadvantages	Long execution time	Converge prematurely	Stagnation problem when all jobs require or assigned to the same resources	Poor converge
advantages	Use the information generated in the past as well as the present estimates	Few control parameters, converge in fast rate	Adapted to the changes	Fewer compared with others

Moreover, PSO has better integrated performance levels when employed in grid job scheduling problem than other approaches like ACO and GA [40]. Although PSO has many advantages and has achieved significant results. It suffers from some problems like premature convergences and slow convergence problems, so improvements in this regards are needed [41]. However, PSO is applied in many areas like neural network training, control system analysis, task scheduling of computational grid and others. PSO is better than genetic algorithm and simulated annealing due to little number of parameters it uses [21]. Many heuristics algorithms based on ACO have been used to solve grid scheduling problem. The results and experiments showed that ACO has given a better solution in a large grid environment when compared with the genetic algorithms and traditional approaches [25][26][27]. Table 1 shows a comparison between GA, PSO, ACO and ABC.

## 8. CONCLUSION

The complexity of scheduling increases with the size of the grid. Therefore, a new area of research has been appeared which is based on heuristics techniques that give optimal or near optimal solutions for large grids. This paper attempts to study the four popular population based heuristic approaches namely: GA, PSO, ACO and ABC for grid job scheduling and compare their performance based on certain criteria. The comparison result shows that GA has been widely applied in solving grid job scheduling and it is better when dealing with data grid. ACO, PSO and ABC deal with computational grid and PSO has better convergence than other approaches. As for the size of the grid ACO has shown good performance when applied on big grids. In fact, a hybridization of the good features of these approaches is necessary to have a new population based scheduling approach that can show a consistent performance when applied on different grid types.

## 9. REFERENCES

- [1] Dabas, P. & Arya, A. (2013). Grid computing: An introduction. *International journal of advanced research in computer science and software engineering*, 3(3), 2013.
- [2] Chandak, A., Sahoo, B. & Turuk, A. (2011). Heuristic task allocation strategies for computational grid. *International Journal advanced networking and applications*, 5(2), 804-810.
- [3] Kaur, A. & Goyal, S. (2011). A survey on the applications of bee colony optimization techniques. *International journal on computer science and engineering (IJCSE)*, 8(3) August 2011.
- [4] Xhafa, F. & Abraham, A. (2008), 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.
- [5] Malarizhi, N. & Uthariaraj, V. (2012). Comparison of resource scheduling in centralized, decentralized and hybrid grid environment. *International journal of emerging technology and advanced engineering*, 2(7), 382-388, July 2012.
- [6] Schopf, J. (2002). A General architecture for scheduling on the grid. Special issue of JPDC on grid computing, 2002,1000-1002.
- [7] Dong, F. & Akl, S. (2006). 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.
- [8] Berman, F. (1998). High-performance schedulers. In Ed. By Foster I. & Kesselman, C. Kaufman, M. publishers, the grid: blue print for a future computing infrastructures, 1998.
- [9] Raju, M. & Divya, T. and Rao, T. (2010). Efficient resource scheduling in data grid. Proceedings of the international conference on information science and applications ICISA, 2010, Chennai, India
- [10] Kaur, G. (2008). Makespan optimization algorithm for scheduling in grid. Msc thesis, Thapar university, Patiala, India, 2008.
- [11] Koulouzis, S., Wood, T. & Groen, D. (2006). An Investigation into the Use of Genetic Algorithms for Grid Scheduling.

- [12] Deb, K. (1997). Genetic Algorithm in Search and Optimization: The Technique and Applications. Proceeding of International Workshop on Soft Computing and Intelligent System, pp. 58-87, 1997.
- [13] Abraham, A., Buyya, R. & Nath, B. (2000). Nature's heuristics for scheduling job on computational grids. *In Proceeding of 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000)*.
- [14] Kim, S. & Weissman, J. B. (2004). A genetic algorithm based approach for scheduling decomposable data grid applications. *In Proceeding of International Conference on parallel processing ICPP 2004*. pp 406-413
- [15] Carretero, j. & Xhafa, F. (2006). Use of genetic algorithms for scheduling jobs in large Scale grid applications. *Technological and economic development of economy*, 6(1), 11–17. <http://www.tede.vgtu.lt>.
- [16] Elghirani, A., Subrata, R., Zomaya, A. & Al Mazari, A. (2008). Performance Enhancement through Hybrid Replication and Genetic Algorithm Co-Scheduling in Data Grids. IEEE.
- [17] Bhana, S. & Gopalan, N. (2008). A hyper-heuristic approach for efficient resources scheduling in grid. *International Journal of computers, communications & control*, 3(3), 249-258.
- [18] Entezari-Maleki, R. & Movaghar, A. (2011). A Genetic Algorithm to Increase the Throughput of the Computational Grids. *International Journal of Grid and Distributed Computing*, 4(2), June, 2011.
- [19] Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. *In Proceeding of International Conference on neural network*. pp 1940-1948.
- [20] Izakian, H., Ladani, B., Zamanifar, K. & Abraham, A. (2010). A Novel Particle Swarm Optimization Approach for Grid Job Scheduling.
- [21] Zhang, L., Chen, Y., Sun, R., Jing, S. & Yang, B. (2008). A Task Scheduling Algorithm Based on PSO for Grid Computing. *In International Journal of Computational Intelligence Research*, 4(1), 37–43.
- [22] Mathiyalagan, P. Dhephthie, U. & Sivanandam, S. (2010). Grid scheduling using enhanced PSO algorithm. *International Journal on Computer Science and Engineering (IJCSSE)*, 2(2), 140-145
- [23] Shakerian, R., Kamali, M., Hedayati, M. & Alipour, M. (2011). Comparative study of ant colony optimization and particle swarm optimization for grid scheduling. *TJMCS* 3(2), 469-474.
- [24] Dorigo, M. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 26(1), 1996,1-13.
- [25] Kousalya, K. & Balasubramanie, P. (2008). An enhanced ant algorithm for grid scheduling problem. *International Journal of Computer Science and Network Security IJCSNS*, 8(4), April 2008.
- [26] Kousalya, K. & Balasubramanie, P. (2009). To improve Ant algorithm's grid scheduling using local search. *International Journal of computational cognition* , 4(7), Dec. 2009.
- [27] Semnani, S. H., Zamanifar, K., Nematbakhsh, N., (2009). New Heuristic in Ant Colony Optimization to Solve Job Scheduling Problem in Grid.
- [28] Xu, Z., Hou, X. & Sun, J., (2003). Ant algorithm-based task scheduling in grid computing. *Electrical and Computer Engineering, IEEE CCECE 2003. Canadian Conference* , 2, 1107-1110, 4-7 May 2003, 10.1109/CCECE.2003.1226090.
- [29] Lorpunmanee, S., Sap, M. N., Abdullah, A. H., Chompoo-inwai, C. (2007). An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment. *World Academy of Science, Engineering and Technology*, 29, 2007.
- [30] Kousalya, K. & Balasubramanie, P. (2008). Ant Algorithm for Grid Scheduling Powered by Local Search. *International Journal Open Problems Computer, Mathematics*, 1(3), December 2008.
- [31] Mathiyalagan, P., Suriya, S., Sivanandam, S. (2010). Modified ant colony algorithm for grid scheduling. *International journal computer science and engineering*. 2(2), pp 132-139.
- [32] Maruthanayagam, D. & Umarani, R. (2010). Enhanced ant colony algorithm for grid scheduling. *International journal computer technology applications*, 1 (1), 43-53.
- [33] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, October 2005.
- [34] Akbari, R., Zeighami, V. & Ziarati, K. (2010). Artificial bee colony for resource constrained project scheduling problem. *Growing science Ltd*.
- [35] Davidovic, T., Selmic, M. & Teodorovic, D. (2009). Bee colony optimization for scheduling independent tasks. *In Proceedings of the Symposium on Information Technology*, YUINFO 2009, (on CD 116.pdf), Kopaonik, Serbia, March 08-11, 2009.
- [36] Davidović, T., Šelmić, M. & Teodorović, D. (2009). Scheduling Independent Tasks: Bee Colony Optimization Approach. *In Proceeding 17th Mediterranean Conference on Control and Automation*, (pp.1020-1025), Makedonia Palace, Thessaloniki, Greece, June 24-26, 2009.
- [37] Karaboga, D. & Ozturk, C. (2011). A novel clustering approach: artificial bee colony (ABC) algorithm. *Applied soft computing* 11, 652-657.
- [38] Vivekanandan, K., Ramyachitra, D. & Anbu, B. (2011). Artificial bee colony algorithm for grid scheduling. *Journal of Convergence Information Technology*, 6(7), July 2011.
- [39] Gupta, M., & Sharma, G., (2012). An Efficient Modified Artificial Bee Colony Algorithm for Job Scheduling Problem. *International Journal of Soft Computing and Engineering (IJSCE)*, 1(6), January 2012, ISSN: 2231-2307.
- [40] Meihong, W. & Wenhua, Z. (2010). A comparison of four popular heuristics for task scheduling problem in computational grid. *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010. pp1-4.
- [41] Li, H., Wang, L. & Liu, J. (2010). Task scheduling of computational grid based on particle swarm algorithm.