I.J.Computer Network and Information Security, **2014, 2, 16-22**
Published Online January 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2014.02.03

Scheduling in Grid Systems using Ant Colony Algorithm

Saeed Molaiy
Department of Computer Engineering, Germi branch, Islamic Azad University, Germi, Iran.
saeed_molaiy@yahoo.com

Mehdi Effatparvar
ECE Department, Ardabil branch, Islamic Azad University, Ardabil, Iran.
effatparvar@iauardabil.ac.ir

***Abstract* —** Task scheduling is an important factor that directly influences the performance and efficiency of the system. Grid computing utilizes the distributed heterogeneous resources in order to support complicated computing problems. Grid can be classified into two types: computing grid and data grid. Job scheduling in computing grid is a very important problem. To utilize grids efficiently, we need a good job scheduling algorithm to assign jobs to resources in grids. This paper presents a new algorithm based on ant colony optimization (ACO) metaheuristic for solving this problem. In this study, a proposed ACO algorithm for scheduling in Grid systems will be presented. Simulation results indicate our ACO algorithm optimizes total response time and also it increase utilization.

***Index Terms* —** Grid systems, scheduling, response time, utilization.

I. INTRODUCTION

Grid is defined as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements. Grid technology is defined as the technology that enables resource virtualization, on-demand provisioning and resource sharing between organizations. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration.

The term Grid computing originated computer power as easy to access as an electric power grid. Grid computing is the process of applying the resources of many computers in a network to a single problem at the situations like usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. It involves the use of software that can divide and form out pieces of a program to as many as several thousand computers. It looks as distributed and large-scale cluster computing and as a form of network distributed parallel processing.

Grid computing is an emerging computing model that provides the ability to perform higher throughput computing by taking advantage of man networked computers. They use the resources of many separate computers connected by a network to solve large scale computation problems. They provide the ability.

- To perform computations on large data sets, by breaking them down into many smaller ones.
- To perform many more computations.

Grid computing appears to be a promising trend for three reasons:

1. Its ability to make more cost-effective use of a given amount of computer resources.
2. A way to solve problems that cannot be approached without an enormous amount of computing power.
3. It suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as collaboration towards a common objective.

Scheduling is a key concept in computer multitasking operating systems, multiprocessing operating system design and in real-time operating system design. In modern operating systems, there are typically many more processes running than there are CPUs available to run them. Scheduling refers to the way processes are assigned to run on the available CPUs. This assignment is carried out by software known as a scheduler or is sometimes referred to as a dispatcher.

Grid Scheduling, Mapping of jobs to resources as per their requirements and properties. It is proposed to solve large complex problems. Grid scheduling is an intelligent algorithm capable of finding the optimal resource for processing a job. Major function is to find the optimal resources and to allocate them to each individual job. The grid scheduler is shown in figure1. The objectives of a grid scheduler are overcoming heterogeneousness of computing resources, maximizing overall system performance, such as high resource, utilization rate and supporting various computing intensive applications, such as batch jobs and parallel applications. The grid scheduler is mainly concerned with the following: CPU utilization to keep the CPU as busy as possible, throughput number of process that complete their execution per time unit , turnaround time amount of time to execute a particular process and response time amount of time it takes from when a request was submitted until the first response is produced.

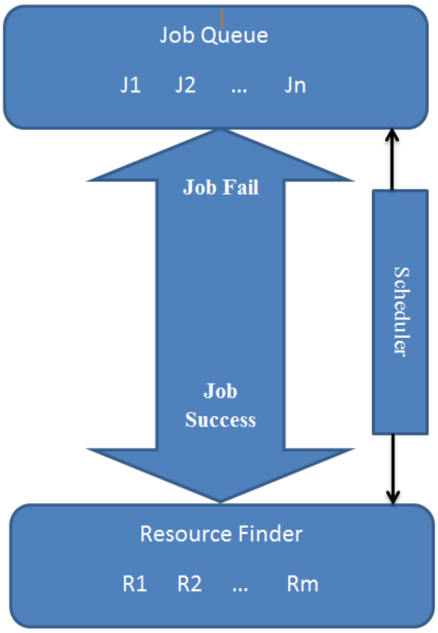


Figure 1: Grid scheduler Architecture with its Components

Artificial swarm intelligence, in particular Ant Colony Optimization (ACO), is a relatively new computational and behavioral paradigm for solving optimization and combinatory problems and hence it can be used for scheduling; it is based on the principles that control the behavior of natural systems. In such a simulation model, many distributed agents evolve and interact with each other in order to reach a global goal, such as ant colonies and bird flocks. This approach emphasizes the distributed structure of the problem, direct or indirect interactions among relatively simple agents and also among the agents and their environment.

The application of swarm intelligence to networks problems arises when a group of autonomous programs (agents) are working together. This is referred to be “Ant Colony Optimization” ACO or multi-agent system. Each individual or program or autonomous module can be represented as an agent, these multi-agents can be used for network applications such as finding the shortest path, routing, scheduling, load balancing and management and so on.

Some related works to using ACO inshceduling and load balancing are the use of multi-agents system, two algorithms has been proposed by (Heusse et al., 1998) [1]: The first one is based on round trip routing agents that update the routing tables by backtracking their way after having reached the destination. The second one relies on forward agents that update the routing tables directly as they move toward their destination. On the other hand, Salehi and Deldari (2006) [2] presents an echo system of intelligent, autonomous and cooperative ants. The ants in this environment can procreate and also may commit suicide depending on existing conditions. A new concept called Ant level load balancing and scheduling is presented for improving the performance of the mechanism. Sim and Sun (2003a)[3] has presented a Multiple Ant Colony Optimization (MACO) approach for load balancing in circuit-switched networks. MACO uses multiple ant colonies to search for alternatives to an optimal path. One of the impetuses of MACO is to optimize the performance of a congested network by routing calls via several alternatives paths to prevent possible congestion along an optimal path. In MACO, each group of mobile agents corresponds to a colony of ants and the routing table of each group corresponds to a pheromone table of each colony (Sim and Sun, 2003)[4] . By adopting the MACO approach, it may be possible to reduce the likelihood that all mobile agents establish connections using only the optimal path (Sim and Sun, 2003) [4]. The advantage of using MACO in circuit-switched routing is that it is more likely to establish connections through multiple paths to help balance the load but does not increase the routing overhead (Sim and Sun, 2003)[4]. Bulancea et al. (1996) [5], used mobile agents to encapsulate tasks, in distributed memory message-passing computers. In order to increase system flexibility and to balance an arbitrary graph topology computing environment, in the proposed model, the agents have the possibility to explore, learn and share information about the load. The proposed model was compared with deterministic DASUD and evolutionary algorithms. The allocation on physical processors of the proposed agent algorithm was generally better than that given by the DASUD algorithm when the task’s time requirement is increased.

This study is divided into the following sections: In section 2 an overview of the Ant colony optimization algorithms. The proposed method is described in Section 3. Results of the study are analyzed in Section 5. Finally, Section 6 presents the conclusions.

II. ANT COLONY OPTIMIZATION ALGORITHMS

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs [1-12].

The intelligent group-behavior characterizes the whole colony of social insects, such as ants; examples of that emergent behavior include foraging and nest building. This collective behavior can be viewed as a powerful problem-solving system that can solve problems such as scheduling and load balancing. Starting from simple interacting agents with rules of interaction among individuals and between individuals and the environment, the ant colony can provide intelligence far away from any individual capability. Properties associated with their group behavior, such as self-organization, flexibility and robustness, can be shown as characteristics that should exist in complex system for control, optimization and problem solving techniques [8-13].

The problem of how almost blind ants could cooperate in order to find the shortest path to the food source has attracted ethologists for several years. It was found that they communicate by placing pheromone trails, a chemical substance that attracts other ants, along with their movements towards the source of food [6, 14]. An ant will prefer with high probability to follow the path having more pheromone, thus enforcing the trail with its own pheromone. This collective behavior is called autocatalytic behavior, in which it is defined to be a positive feedback process in which a process reinforces itself to enhance its performance, this feedback enforces the process towards a rapid convergence towards the final solution (Dorigo et al., 1996)[6, 7].

The ants do not forage only for finding food, rather they forage in order to wander, search, return to home, attract to a target, trace pheromone and carry food [7]. The pheromones also evaporate over time. As a consequence, the pheromone will become less detectable after a while and the longer trails will be less attractive to other ants (Tarasewich and Patrick, 2002)[8]. Ants can construct the shortest path from their nest to the source of food, through the use of pheromone trails. The ant leaves some quantities of pheromone on the ground while it walks. The next one will sense it and based on a probability proportional to the amount of pheromone, it will choose its path.

However, artificial ants have some major differences with real ones: First, ants have a memory. Second, ants are not completely blind, finally, the time of artificial ants is discrete (Krohn, 2001) [9].

Generally, Ant Colony Optimization (ACO) is a general-purpose heuristic algorithm, which can be used to solve different combinatorial optimization problems [6]. In ACO, the search activities is distributed over artificial ants, which mimic the behavior of real ants. The advantages of that system are positive feed-back, distributed computation and the use of a constructive greedy heuristic. Positive feed-back refers to the ability to rapid discovery of good solutions. The ACO is also a population-based approach in which parallization can easily be achieved [6].

One of the main important concepts added by ACO is that finding the solution is an emergent process of the cooperation and the interaction of simple agents. Another concept is the use of indirect stygmergetic communication by changing the environment. Ant algorithms can be considered multi-agents systems that exploit artificial stigmergy as a means for coordinating artificial ants for the solution of computational problems.

The collective activities of social insects are self-organizing, meaning that complex group behavior emerges from the simple interactions of individuals. The results of self-organization are of global nature, but come basically from local information and interactions [8]. The interaction within a society of insects can take one of the two forms: direct and indirect. Direct interactions can take the form of bodily contact, visual contact and food exchange. Indirect interaction is also important, it can occur when agents exchange information through the environment in which they exist. Thus the storage of information occurs at the colony level as well as individual level. This cooperation through modification is called stigmergy (Krohn, 2001) [9].

Basic operations of ant colony optimization systems: In this part; the basic and necessary operations that should exist in any ACO system, including our proposed model are summarized and discussed. In ACO, the ants are adaptive, i.e., if the environment changes, the ants will look for a better solution. The ACO is suitable to discrete optimization problems. The main characteristics of AC system are positive feedback, distributed computation and a constructive greedy heuristic (Dorigo et al., 1996) [6]. In the following; a short description will be given to the basic operations involved in ACO.

Stigmergy: The indirect communications via the pheromones is an example of positive feedback called autocatalytic behavior. Conversely, the negative feedback is introduced through the evaporation of pheromone trail [9]. The characteristics of the swarm intelligence model of ACO are the new added concepts of self-organization and stigmergy. In a distributed system such as Ant System, communication between agents is of the great importance. The form of communication is indirect. This communication can be viewed as a space deformation of the system in which ants reside [9]. The behavior of the whole ant colony is highly structured, interactions is based on a very simple flows of information (Colorni et al, 1992) [10]. The general idea of the ACO algorithm is that two opposite forces enforce the optimization method to reach the solution. The first force is autocatalytic process that drives the good solutions to emerge. The other force is the greedy force that always selects the first shortest path to find a solution. Neither of the two forces can reach the optimal solution alone. But when they are working together, it seems that the greedy force can give the right suggestions to the autocatalytic force and let it converge to the optimal value very quickly [10]. However, one of the problems of ACO is that whenever there is a significant change in the environment, it takes some ime before ants discover it and modifies their information.

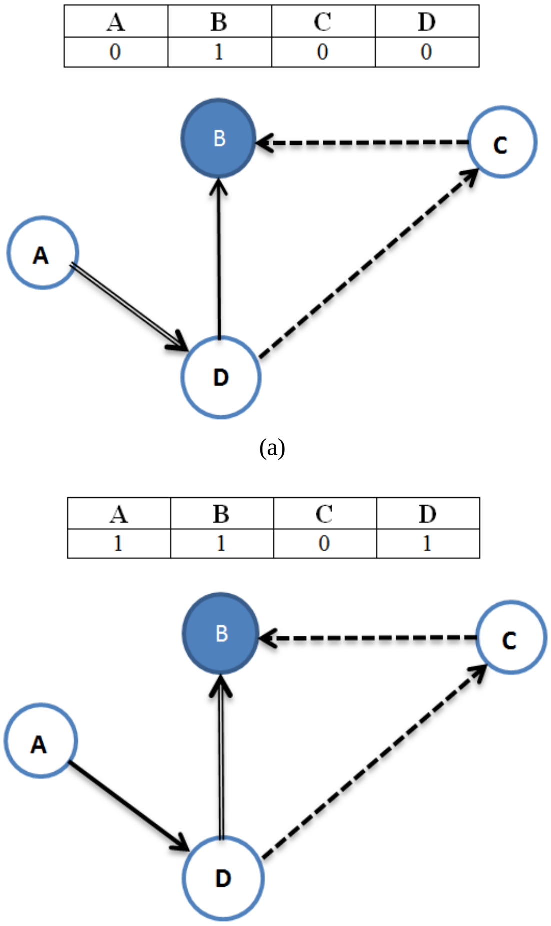
Pheromone evaluation: The pheromone that each ant lays attracts the following ants so that they will likely search in the same region of the search space. In general, it is assumed that pheromone evaluation is done locally by the ants. Merkle et al. [11] proposed to extend the local view of the ants by a look-forward strategy. Merkle et al. showed that the behavior of an ant algorithm can be improved scientifically when the ants use more information than just the local information values for their decision.

Information exchange in ant colony optimization: Information exchange between ant colonies is one of the most important factors that influence the optimization behavior. It was shown that several colonies could perform very effectively with a little exchange of information (Middendorf et al., 2000) [12]. Improvement is achieved by adding a little bit of information exchange. It was also shown exchanging the local best solution only and not to often is sufficient to have a very good performance.

In [12], different information exchange policies are investigated. It was shown that exchanging a small amount of information significantly affects the performance of the optimization performance. This can be done by permitting the other colony to make benefit from good solutions obtained by other colonies.

III. SCHEDULING USING ANT COLONY ALGORITHMS IN GRID SYSTEMS

The scheduling strategy, in this work, is time dependent and follows the natural dynamics of the pheromone in real life. At a specified time period, each resource will act like a nest and sends number of ants (the number of ants depends on the loading status of each resource; the overloaded and underloaded resources will send more ants).

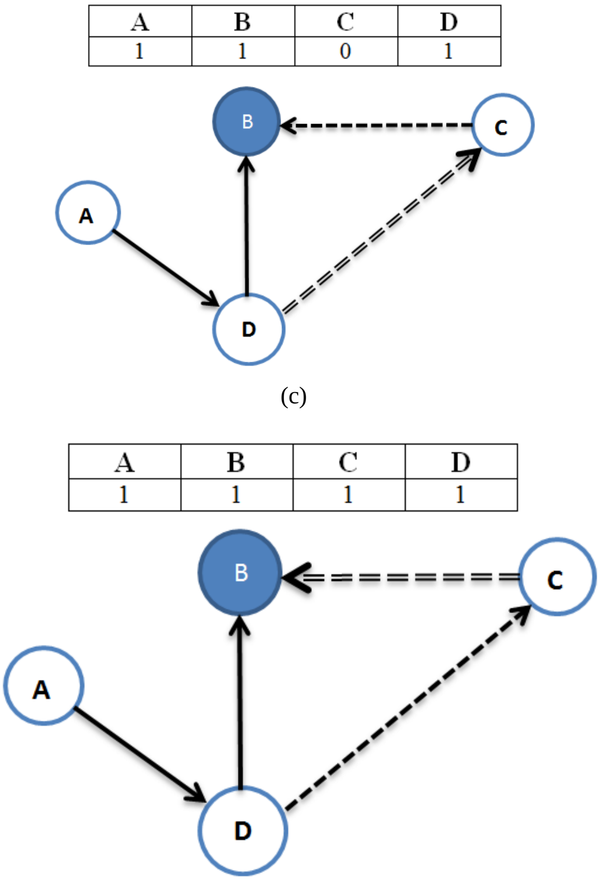


A	B	C	D
0	1	0	0

(a)

A	B	C	D
1	1	0	1

(b)



A	B	C	D
1	1	0	1

(c)

A	B	C	D
1	1	1	1

(d)

Figure 2: Example of the operations of the model. (a) An ant moves from A to D; (b) The ant moves from D to B; (c) Another ant moves from D to C; (d) The ant moves from C to B.

Each ant will travel a tour (the tour length depends on the size of the system and the loading status of the resource). Finally, the proposed algorithm will be compared with the standard work-stealing algorithm.

Our model is fully distributed, i.e., each resource behaves independently as well as each ant or agent, this mean that each resource or ant is autonomous. Figure 2 represents the attached information to each node or ant.

In our model, each node contains information about other nodes in the system (Figure 2). At the initial state, the table entries are Null.

In each ant tour, the ant will carry the updated information about all nodes that the ant has been passed through. Upon arrival of the ant at each node, the following actions will be done:

1. If the node does not have the information contained in the ant table, these information will be passed to the node table without any update.
2. If the node contains information that does not exist in the ant's table, the ant table will be updated.
3. If both of them share the same information, the newly updated one will replace the other.

IV. EVALUATION AND SIMULATION RESULTS

In this section, we present and discuss the experimental results of the proposed scheme. All simulations were performed using MATLAB software.

The proposed strategy was simulated and tested, the number of nodes in the Grid system was assumed to be 30, an ant is assumed to travel from one resource to another in 1 time step, each task is assumed to take 40 steps. In order to emphasis the efficiency of the proposed algorithm, we consider the case when the Grid system is very irregular; it is assumed that resource number 1 is busy with 60 tasks and the other nodes are idle, Figure 3 shows the efficiency of both the work-stealing approach and ant-colony approach. It is clear that the efficiency of the ant-colony approach comes from the ability to distribute loading information through all the nodes, the tour of ants was randomly chosen, however the cleverness of the ants to carry the new loading-status to each nodes increases the chance of each node to quickly find a good food source or a busy node.

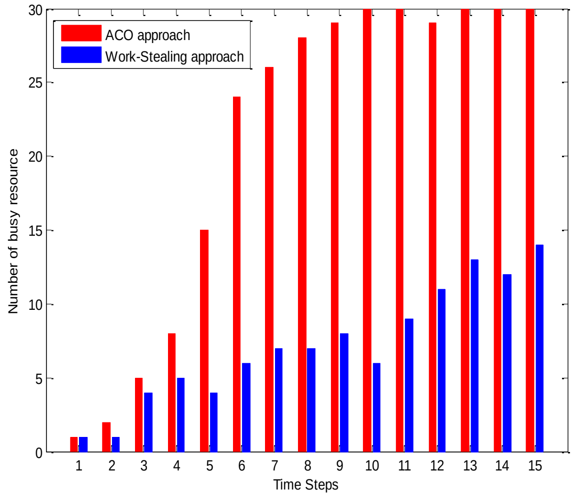


Figure 3: Load distribution: The number of busy nodes in the time-steps

Figure 4 compares the speed-up of proposed model versus the standard work-stealing approach. Based on the above settings, the proposed model achieved a speed-up of 12.6, while the work-stealing approach's speed-up was 5.1.

Figure 5 and Figure 6 studies the effect of enlarging the number of nodes against the number of steps needed to raise the load distribution up to 50%. It is shown that as the number of nodes becomes larger, the time-steps needed for the work-stealing approach increases dramatically, while the proposed model remains almost constant.

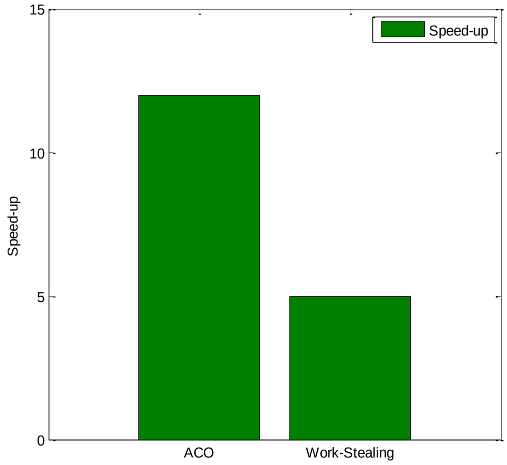


Figure 4: A comparison of the speed-up of the proposed model and the work-stealing model.

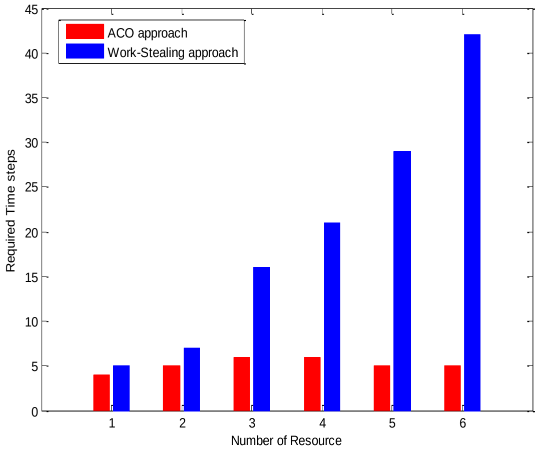


Figure 5: Load balancing dynamics: The required time steps needed to reach a load distribution of 50% versus the number of resources.

This approach and its results give an example that highlights the importance of the swarm system in the decision-making process in general, where each agent can play a small role and the global behavior could be robust and reliable.

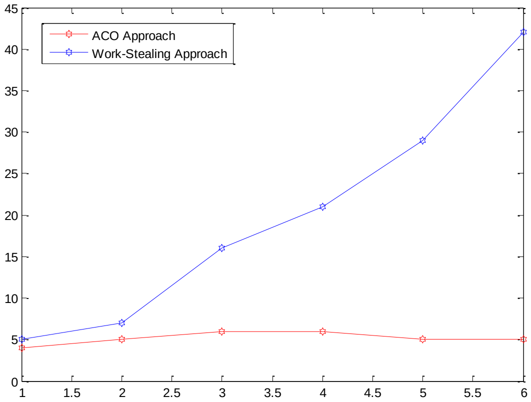


Figure 6: Load balancing dynamics: The required time steps needed to reach a load distribution of 50% versus the number of resources.

V. CONCLUSION

It has been convincingly proved in the recent research papers that task scheduling on computational grids is best solved by heuristic approach. The project tried to cover the state-of-the-art studies about one such heuristic namely Ant Colony Optimization (ACO) algorithm and its applicaion to grid systems. In this study, we proposed method use of multiple nests, or ant colonies in the search process, helped in raising the rate of information exchange all over the resources in the system. Besides, the dynamical information exchange and the fully distribution are other main characteristics that distinguishes our approach.

Results have shown the efficiency of the proposed model compared with the standard work-stealing algorithm in terms of number of busy nodes and the elapsed time to reach an efficiency of 50%.

REFERENCES

[1] Heusse, M., S. Guerin, D. Snyersy and P. Kuntz, 1998. A new distributed and adaptive approach to routing and load balancing in dynamic communication networks. University of Washington.

[2] Salehi, M.A. and H. Deldari, 2006. Grid load balancing using an echo system of intelligent ants. Proceedings of the 24th IASTED International Conference on Parallel and Distributed Computing and Networks, Feb. 14-16, ACTA Press, Innsbruck, Austria, Anaheim, CA, USA., pp: 47-52.

[3] Sim, K.M. and W.H. Sun, 2003b. Ant colony optimization for routing and load-balancing: Survey and new directions. IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum., 33: 560-572.

[4] Sim, K.M. and W.H. Sun, 2003a. Multiple ant colony optimization for load balancing. Lecture Notes Comput. Sci., 2690: 467-471. DOI: 10.1007/b11717.

[5] Bulancea, C., B. Paechter and A. Carter, 1996. Using ACO metaheuristics on load balancing algorithm. Trans. Syst. Man Cybernet.

[6] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[7] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992.

[8] Tarasewich, P. and P.R. McMullen, 2002. Swarm intelligence: Powers in numbers. Commun. ACM, 45: 62-67.

[9] Krohn, J., 2001. Ant algorithms and the swarm intelligence model of problem solving. Proceedings of the Conference on the UMM Computer Science Discipline Seminar, (CCSDS'01), UMM University of Minesota Morris, Morris, pp: 1-6.

[10] Colorni, A., M. Dorigo and V. Maniezzo, 1992. Distributed optimization by ant colonies. Proceeding of the 1st European Conference on Artificial Life, (ECAL'92), Elsevier Publishing, Paris, France, pp: 134-142.

[11] Merkle, D., M. Middendorf H. Schmeck, 2000. Pheromone evaluation in ant colony optimization. Proceedings of the 26th Annual IEEE International Conference on Industrial Electronics, Control and Instrumentation, Oct. 22- 28, IEEE Xplore Press, Nagoya, Japan, pp: 2726-2731.

[12] Middendorf, M., F. Reischle and H. Schmeck, 2000. Information exchange in multi colony ant algorithms. Lecture Notes Comput. Sci., 1800: 645-652. DOI: 10.1007/3-540-45591-4_87.

[13] Siriluck Lorpunmanae, Mohd Noor Sap, Abdul Hanan Abdullah and Aboamama Atahar Ahmed, "Multi-Constraint Dynamic Scheduling of Indepent jobs onto Grid Environment", Jurnal Teknologi Maklumat, 2007.

[14] Kousalya.K and Subramanie.P, "Ant algorithms for Grid Scheduling Powered by local search", IEEE International Journal, 2008.

[15] Huiyan, Xue-Qin-Shein, Xing Li, Ming-Hui Wu, "An Improved Ant Algorithm for Job Scheduling in Grid Computing", 18-21 August 2005.

[16] Li Liu, Yi Yang, Lian Li and Wanbin Shi, "Using Ant Colony Optimization for SuperScheduling in Computational Grid", Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06), 2006.

[17] H. Yan, X. Shen, X. Li and M. Wu, "An Improved Ant Algorithm for Job Scheduling in Grid Computing", In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 18-21 August 2005.

[18] D. Klus´aˇ cek, L. Matyska, and H. Rudov´a, "Grid scheduling simulation environment", Submitted to MISTA, 3rd Multidisciplinary International Scheduling Conference: Theory and Applications, France, 2007.

[19] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen and R. F. Freund (2001), "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing. Vol.61 (6): Pages 810-837.

[20] C. Ernemann , V. Hamscher and R. Yahyapour, "Benefits of Global Grid Computing for Job Scheduling", Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04), 2004.

[21] R. Buyya and M. Murshed, "GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing", The Journal of Concurrency and Computation: Practice and Experience (CCPE), 14:1175-1220, 2002.

[22] P. Fibich, L. Matyska, and H. Rudov´a, “Model of Grid Scheduling Problem”, In Exploring Planning and Scheduling for Web Services, Grid and Autonomic Computing, Papers from the AAAI-05 workshop. Technical Report WS-05-03, AAAI Press, 2005.

[23] R. Baraglia, R. Ferrini, and P. Ritrovato, “A static mapping heuristics to map parallel applications to heterogeneous computing systems”, Research articles. Concurrency and Computation: Practice and Experience, 17(13):1579–1605, 2005.

[24] Shih-Tang, L., C. Ruey-Maw, H. Yueh-Min and W. Chung-Lun, 2008. Multiprocessor system scheduling with precedence and resource constraints using an enhanced ant colony system. Expert Syst. Appli., 34: 2071-2081. DOI: 10.1016/j.eswa.2007.02.022.

[25] Braun, D.T., H.J. Siegel, N. Beck, L.L. Boloni and M. Maheswaran et al., 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. J. Parallel Distributed Computer, 61: 810-837. DOI: 10.1006/jpdc.2000.1714.

Saeed Molaiy, born in 1989, He is MSc student in Islamic Azad University of Germi, Iran. His research interests include computer networks and operating systems.

Mehdi Effatparvar (Ph.D) is member of computer engineering department in Islamic Azad University of Ardabil, His research interests include computer networks, distributed systems and operating systems.

How to cite this paper: Saeed Molaiy,Mehdi Effatparvar,"Scheduling in Grid Systems using Ant Colony Algorithm", IJCNIS, vol.6, no.2, pp.16-22, 2014. DOI: 10.5815/ijcnis.2014.02.03