

Proposing a scheduling algorithm to balance the time and cost using a genetic algorithm

Ali Akbar Faraj
Department of Computer
Science and Research Branch
Islamic Azad University Kish, Iran

Ali Haroon Abadi
Member of Science Board of
Computer Group in Azad Islamic
University of Tehran Center

Abstract: Grid computing is a hardware and software infrastructure and provides affordable, sustainable, and reliable access. Its aim is to create a supercomputer using free resources. One of the challenges to the Grid computing is scheduling problem which is regarded as a tough issue. Since scheduling problem is a non-deterministic issue in the Grid, deterministic algorithms cannot be used to improve scheduling.

In this paper, a combination of genetic algorithms and binary gravitational attraction is used for scheduling problem solving, where the reduction in the duty performance timing and cost-effective use of simultaneous resources are investigated. In this case, the user determines the execution time parameter and cost-effective use of resources. In this algorithm, a new approach that has led to a balanced load of resources is used in the selection of resources. Experimental results reveals that our proposed algorithm in terms of cost-time and selection of the best resource has reached better results than other algorithm.

Keywords: Grid computing, Static timing strategies, Genetic algorithms, Local search algorithm following the binary gravitational attraction, Optimizing Time

1. INTRODUCTION

Grid is a form of distributed computing systems which is available on a network and appears to the user as a large virtual computing system [4]. One of the most important parameters in the Grid environment is grid scheduling and load balancing services. Therefore, a correct and reliable scheduler is needed to increase the efficiency of the grid. Unfortunately, the dynamic nature of the Grid resources and users' demands has led to complexity of Grid scheduling problem and optimized assigning of tasks to resources. This would have prompted researchers to use innovative algorithms for solving this challenge. A scheduler in grid environment should provide a time and cost effective scheduler for users through using users' parameters, without engaging users in the complexity of such an environment. Genetic algorithm is one of the best innovative algorithms due to the fact that it generally investigates the problem from several different directions at once. Therefore, genetic algorithm is used to solve many optimization problems [7]. In this paper, after reviewing the strengths and weaknesses of previous approaches, the combination of genetic algorithm and local search following the gravitational force will be examined with an aim of reducing the simultaneous implementation of tasks time and costs. In this algorithm, a new approach has been used in the selection of resources.

The following sections of this paper are as follows: Section 2 describes the literature. Section 3 introduces the related work on scheduling and Section 4 is dedicated to the algorithm proposed. Performance evaluation of the proposed algorithm is presented in Section 5. Section 6 presents the conclusions.

2. BACKGROUND

2.1 Scheduling Method

The scheduling task issue is considered as a tough challenge which is composed of n tasks and m resources. Each task

must be processed by a machine and does not stop until the end of the performance. We used ETC matrix model described in [2]. The system assumes that the expected execution time for each task i , on every resource j is predetermined and is located in the matrix ETC, ETC $[i, j]$. Here, makespan is regarded as the maximum completion time in Completion-Time $[i, j]$, calculated in the following equation (1)[10]:

$$\text{Makespan} = \text{Max}(\text{Completion-Time}[i, j] | 1 \leq i \leq N, 1 \leq j \leq M) \quad (1)$$

In the above equation, Completion-Time $[i, j]$ is equal to the time when the task i on the source j is completed and it is calculated in equation (2):

$$\text{Completion-Time}[i, j] = \text{Ready}[M] + \text{ETC}[i, j] \quad (2)$$

2.2 Genetic Algorithm

Inspired by the evolution of organisms in nature, genetic algorithm is optimizing problems. Based on the evolution of living organisms, species can survive in nature. Genetic algorithm randomly selects chromosomes. Combination is a process that displaces determined sequence of the chromosomes. Mutation is a process that changes the determined sequences of chromosomes with multiple mapping functions, new to the current population. Combination and mutation operations are done randomly. After this operation, a new population is generated, then the population will be evaluated and the process will be repeated several times until the accomplished process criteria are authenticated [5].

2.3 Local search algorithms following binary gravitational attraction

Voudouris, et al. proposed GELS algorithm for the first time in 1995 to surf in a search and tough problem solving space [11]. In 2007, Webster offered it as a powerful algorithm and called it the Gravitational Attraction Algorithm. This algorithm is based on the principle of gravity force and imitates this natural process to surf within a search space. It assumes that there are only the gravity and motion laws governing. Each response has different neighbors which can be classified based on a criterion related to the problem. Obtained neighbors in each group are called neighbors in that dimension. Each dimension has got an initial velocity. After an initial velocity is defined for each dimension, the greater velocity the dimension has, the better response will be provided to the problem. In each dimension, the neighbors responses to simillar methods are obtained from the current response. It is so-called obtaining neighbor's response from the current response in a specified dimension. For each dimension of response, an initial velocity is defined. The greater the velocity is, the more appropriate response will be provided for the issue [5]. The mass of an object is determined based on its fitness, according to equation (4). The $fit_i(t)$ is the the suitability of mass i , $worst(t)$ is the worst fitness at time t , and $best(t)$ is the best fitness at time t .

$$M_{ii}(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \quad (3)$$

$$M_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (4)$$

The mass of dimension d is accelerated in a way that is proportional to the force exerted on the mass in that (F_{id}) direction divided on the inertial mass or inertia (M_{ii}), expressed in equation (5):

$$a_i^d(t) = \frac{F_{id}^d(t)}{M_{ii}(t)} \quad (5)$$

Elitist gravitational search algorithm is introduced for solving continuous problems in search space. In fact, V_{id} has the likelihood of zero or one of X_{id} instead of indicating the speed in binary version. Its formula uses equation (6) to update the position of each dimension [10]

$$X_{id}^d(i, j) = \begin{cases} \sim X_{id}^d(t) & \text{if } rand < S(V_{id}^d(t+1)) \\ X_{id}^d(t) & \text{if } rand \geq S(V_{id}^d(t+1)) \end{cases} \quad (6)$$

Each mass velocity is the sum of the ratio of its current mass speed and mass acceleration, updated according to equation (7). It should be noted that because of determination of the range for the initial velocity, if adding momentum to an entry of the initial velocity vector cause its value to exceed from the specified range, its value would be set in this range [9].

$$V_i^d(t+1) = rand_i \times V_i^d(t) + a_i^d(t) \quad (7)$$

3. Previous Research

Depending on the size and dynamics of the Grid, deterministic algorithms cannot be useful for solving the scheduling problems. This has led researchers to research innovative algorithms for these problems.

A valuable technique for the design of effective scheduling is proposed using genetic algorithm and aims to minimize the response time tasks [12]. The results show that the scheduler has a very high speed and decreases time response. Grid users pay the costs to holders of financial resources for the sake of using such resources [1]. This brings about an incentive for resource owners to share their resources. This model leads to significant economic costs and benefits for users and owners of the resources. As a result, the costs and economic benefits are considered in the objective function of scheduler in most of the scheduling algorithms [11]. A new algorithm has been suggested by combining genetic algorithms and gravitational attraction to solve the scheduling task problems in grid environment. The cost of using this algorithm is ignored [8].

In reference [5], a genetic algorithm, in which disorganized variables are used instead of random variables to produce chromosomes, is proposed. This makes the solution in the search space spread, prevents the algorithm premature convergence, and brings about better solutions and product in a shorter time. This paper uses an algorithm for scheduling tasks in grid gravitational attraction [3].

Most of these procedures have considered completion time and a few other implementation costs. However, they failed to establish a balance and balanced load among them. Thus, in Section 4, a new method will be discussed to investigate the balance and balanced load in a Grid environment.

4. . THE PROPOSED ALGORITHM

The proposed scheduling uses the combination of genetic algorithms and binary gravitational attraction for scheduling tasks in grid. Since the performance of genetic algorithms depends largely on how chromosomes are encoded, we will further investigate it. In this way, a chromosome shows a solution. Each solution in this algorithm is encoded as a row of natural numbers matrix. For example, we have assumed a set of n tasks $T = \{T1, T2, T3, \dots, Tn\}$ and a set of m resource $P = \{P1, P2, P3, \dots, Pm\}$. In this method, each gene represents a cell as a function and the chromosomes' lengths are considered as the number of input functions. The contents of each cell of the matrix is as the resource and its value in chromosomes can be random between 1 and m . Figure 1 shows an example of the chromosomes encoding.

can enter the time (XT and XC) method. These the range [0,1] and equal one. For equal to .7, this



values of cost and in the proposed values are placed in sum of them is example, if Xc is means that the user is 70% concerned with costs and 30% concerned with completion task time. The scheduler should find a resource for performing tasks and that resource should improve costs up to 70% and time up to 30%. The most important task scheduling objective is to minimize the time and cost of completing tasks. Given the above definitions, a chromosome fitness function can be calculated in equation (15):

Figure 1. The chromosomes encoding in BGAGSA algorithm

4.1 The objectives and fitness function in the proposed algorithm

The first objective of the proposed algorithm is to reduce the longest completion task time of all processors in the system with makespan. In the proposed method, the user is allowed to enter the maximum completion time and maximum costs of task performance. Here suppose that L_e represents the task i and R_j is the processing speed of the resource. The execution time of task i on j resource can be calculated by the equation (8).

$$T(i, j) = \frac{L_e i}{R_j} \quad (8)$$

The completion time of task i on resource j can be calculated based on equation (9).

$$CompleteTime(i, j) = T(i, j) + TransferT + wait(i, j) \quad (9)$$

In the above equation, $TransferT$ and $wait(i, j)$ are respectively the data transfer time and waiting execution time. Thus, the makespan system can be calculated using equation (10), where A_j is a set of tasks allocated to resource j .

$$CompleteTime(j) = \frac{\sum_{k \in A_j} CompleteTime(k)}{R_j} \quad (10)$$

$$F_{Time} = Max\{CompleteTime(j)\} \quad 1 \leq j \leq M \quad (11)$$

The second objective of our algorithm is the total cost which must be minimized. Suppose that R_j representing a fixed unit price for resource j and is considered fixed and $TransferC$ is the cost data transfer. The cost of task i on resource j would be calculated by equation (12).

$$cost(i, j) = (T(i, j) \times P_j) + TransferC \quad (12)$$

The cost of task i accomplishment on resource j can be computed using equation (13).

$$Cost(j) = completeTime(j) \times P_j \quad (13)$$

Therefore, the overall cost of a solution (chromosome), which represents the cost associated with a chromosome in the population, can be expressed using equation (14).

$$F_{cost} = \sum_{j=1}^M Cost(j) \quad (14)$$

4.2 First fitness function

When offering tasks, some users are concerned with the task cost and others with completion time. Accordingly, the user

$$Fitness = X_t \times \frac{1}{F_{Time}} + X_c \times \frac{1}{F_{Cost}} \quad (15)$$

Where FTime and FCost are the longest completion time of tasks and the overall cost of a solution. As the equation shows, the less the values of FTime and FCost are, the more the value of the fitness function will be and shows that the more suitable solution for the scheduling problem it is. In the proposed algorithm, to select chromosomes (parents) from competition operator and intersection operations from partially mapped crossover operator in mutation phase after selecting a chromosome in previous stage, a gene is randomly selected and its source field value randomly changes between 1 and m. The main purpose of these mutations is changing the resource parameter of a function in order for a task to be applied on better processing resources.

4.3 Second fitting Function

Since the main objective of the proposed scheduling algorithm is to minimize the total length of time and reduce the cost of implementing tasks, several chromosomes are found and their total scheduling length is the same. Thus, a chromosome with more balanced load is selected in the second fitness function. To calculate the balanced load, we calculate the sum of the standard deviation of productivity. As a result, we must calculate the efficiency of resources. Productivity is calculated in equation (16). We initially calculate the greatest performance time.

$$U = \text{Max}\{T_{ij} + t_{ij}\} \quad (16)$$

In the above equation, T_{ij} is the sub-task processing time and t is data transmission time. They are calculated to obtain the greatest performance time to be put into the variable U. If the efficiency of resource i in the execution of the task j is shown by E the relationship, it can be calculated in equation (17).

$$E_i = \frac{T_{ij} + t_{ij}}{U} \quad (17)$$

In the amount of U is then calculated using equation (16). According to equation (17), the efficiency of each source will be obtained. To obtain the standard deviation, we calculate the mean total productivity using equation (18).

$$E = \frac{\sum_{i=1}^m E_i}{m} \quad (18)$$

To do this, we need to add all the productivity of resources together and then divide it by the number of sources. Then, we calculate the load balance using equation (18) at which the efficiency of all resources is computed. Finally, if we have $\mu_2 = 0$, the maximum load reduction will be obtained.

4.4 Local search algorithm parameters imitating the binary gravitational attraction

At this stage, because the genetic local search algorithm is weak, some solutions, which are similar in terms of overall length of scheduler, are given to the GELS algorithm to have a solution to their neighbors. Each current solution has different neighbors. Each of them is obtained based on a change in the current solution, that is, shift towards a neighbor solution [4]. In the proposed method, a solution represents a chromosome and solution dimensions are considered as each gene from chromosome (Dimensions of the solution are the

neighbor solutions which are obtained through changing the current solution). To determine the representation of solutions with the aim of finding an appropriate mapping, it is performed between solutions and GELS factors. Solutions are encoded in the form of $m \times n$ where m represents the source and n represents the task and the element $X_k(i, j)$ indicates that task j is assigned to an agent k in the resource i . For example, tasks 4 and 5 are executed simultaneously in a resource. Each task can only be assigned to a single resource. Each resource can have multiple tasks in sequential form. The encryption is performed using equation (19) to update the position of each dimension [13].

$$X_k(i, j) = \begin{cases} 1 & \forall_k(i, j) = \text{Max}\{V_k(q, j)\} \\ & \forall_q \in \{1, 2, \dots, \dots, m\} \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

In the above equation, a resource is dedicated to a task X_k and the relevant element in V_k is not assigned to any other resource. The number of entries of initial velocity vector has been considered equal to the number of solutions. The initial velocity vector entries are initially assigned random numbers between 1 and maximum initial velocity [5]. The search space is considered as a set of N mass. The position of each mass in the search space is a point of the space that is a solution to the problem. Neighbor solution of each current solution is equal to a solution where the source assigned to task changes. This is done in a way that an initial velocity is given to each dimension of the solution. This work is done randomly and its value is between one and maximum speed. The gene with the highest speed is selected and its value will change randomly (with a number between 1 and M) [5].

After obtaining the solution of the current neighbor, the fitness value of the solution of neighbor is calculated using the equation (15). If the neighboring solution is improved in comparison with the current solution, it replaces its parent chromosome in the new population. Otherwise, it will not be copied into the new population, and its value is maintained to calculate the mass of each particle. Then, the value of acceleration is calculated between the neighboring solution and the current solution, and its value is added to the initial velocity vector, related to the dimension from which the neighboring solution is obtained, in order for the initial velocity vector to be updated. It should be noted that due to the determination of the range for the initial velocity, if by adding the gravitational force to an entry in the initial velocity vector, its value exceeds the specified range, its value is set in this range interval. Given the above, the mass of neighboring solutions is calculated through equation (20) [13]

$$M_k = \frac{f(X_k) - \text{worst}}{\sum_{i=1}^N f(X_i) - N \times \text{worst}} \quad k = 1, 2, \dots \quad (21)$$

Where $F(X_k)$ is the fitness value of particle, and worst is the worst value among all the particles. The acceleration matrix of K factors is calculated through equation (22) [13].

$$a_k(i, j) = \sum_{j \in K_{best}} G \times \frac{\text{rand}_q M_q}{R + \epsilon} (X_q(i, j) - X_k(i, j)) \quad (22)$$

According to the above equation, the mass of i with an acceleration equal to a is drawn to the mass of j ($a_k(i, j)$), in

which Kbest is a collection of first K factor with the best performance value and the greatest mass. M_q is the mass of Q factor and G is the gravitational constant. The pseudo-code of the proposed algorithm is presented in figure 2.

```

step1: initialization
step2: 2.1. Generate the k number of random Chromosome
with length n
        2.2. Speed_vector[1..n] = The initial velocity for
each dimension
        2.3. Location_vector[1..n] = The value of the initial
position of each particle
        2.4. Setting the Maximum: Maximum-Time (Mt)
and Maximum-cost (Mc) according to the user's
requirement.
step3: Compute Makespan and cost for all chromosomes.
step4: repeat
        4.1. Evaluate all individuals using formula 15
        4.2. Select the P/2 members of the combined population
based on minimum fitness, to make the population the
next generation.
        4.3. Crossover & Mutation
step5: The best chromosomes from the genetic algorithm as
a solution to it current GELS will be producing a
neighboring chromosomes.
step6:
        6.1. direction = max(speed_vector[...])
        6.2. change the velocity_vector[index] of
current_solution with random integer between 1 and Max
Velocity.
        6.3. Fitness of the neighboring chromosomes by
equation (15) is calculated and saving the worst fitness
        6.4. if direct chromosomes < neighbor chromosome
Neighbor chromosome is selected as the best solution
step7:
        7.1. Calculated the mass of each particle with using
formula (22)
        7.2. Calculate Acceleration Matrix for 'k' factors with
using formula (21)
step8:
        8.1. Update the factors by using (19)
        8.2. Update Velocity_Vector for each dimension by
acceleration matrix of chromosome with using formula (7)
until a terminating criterion is satisfied
    
```

Figure 2.Pseudocode the algorithm proposed

5. DISCUSSION

The results of the evaluation of the proposed algorithm with the two algorithms of [6] GA and [7] Genetic-Variable neighborhood search for scheduling independent tasks are presented in this section. All experiments have been done on a system running Windows XP operating system with configuration of 3 GHz CPU and 4GB of RAM. Table (1) indicates the amounts of parameters of genetic and binary gravitational emulation local search algorithms.

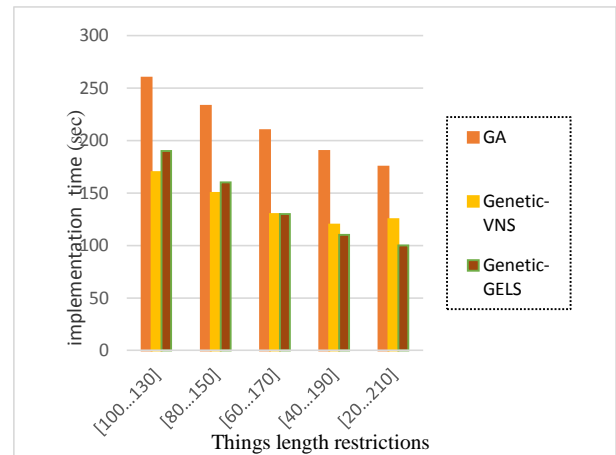


Figure 3. Comparison of makespan of algorithms

GA	Number of generations	100
	Selection	Contesting
	Combined rate	0.85
	Mutation rate	0.02
GSA	Initial position	Between 1 and n
	Initial velocity	Between 1 and the maximum speed
	Maximum speed	The size of the input functions
	Neighborhood radius (R)	1
	Gravitational constant (G)	6.672

To evaluate the proposed algorithm, a series of simulations was considered on 10 sources based on the change in the number of users' tasks and payable budgets by the customers. In the first experiment, aimed at optimizing the time, we have experimented the proposed algorithm with two other scheduling algorithms through applying a change in the number of tasks and a fixed budget. The number of 20 iterations have been performed in this algorithm in order to obtain the execution time of task using existing algorithms, as the results have been uncertain and the results of each performance can be different to some extent from the previous one. The results of the first experiment (figure 3) indicate that the proposed algorithm has a higher performance than the other two algorithms in terms of task completion time. Furthermore, by increasing the number of iterations, the overall performance time increases in all algorithms.

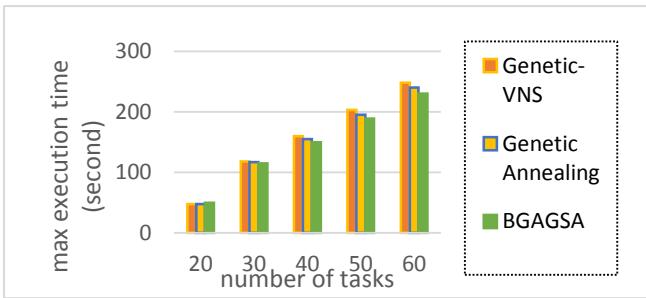


Figure 4. Results of time-optimization algorithms with different budgets

Having the purpose of optimizing time, the second experiment is compared through providing different budgets. The proposed algorithm is evaluated along with several other scheduling algorithms. The budget parameter has been considered as the variable. As it can be seen in figure 4, Genetic-GELS algorithm functioned better than other algorithms in all cases, and by increasing the budget, the amount of execution time is reduced. In fact, the more specified budget set by the user, the less time is required for algorithms to perform practical tasks.

In the third experiment, the effect of heterogeneity of tasks on the performance of the proposed algorithm and other targeted algorithms was investigated. The length of the tasks is considered variable. As it can be seen in figure 5, the time changes differently due to an increase in the heterogeneity of the tasks in each of the algorithms. A good algorithm is expected to be able to make use of the heterogeneity of tasks and reduce the execution time of tasks. As it is shown clearly in the figure, the proposed algorithm has achieved a better performance compared to other algorithms.

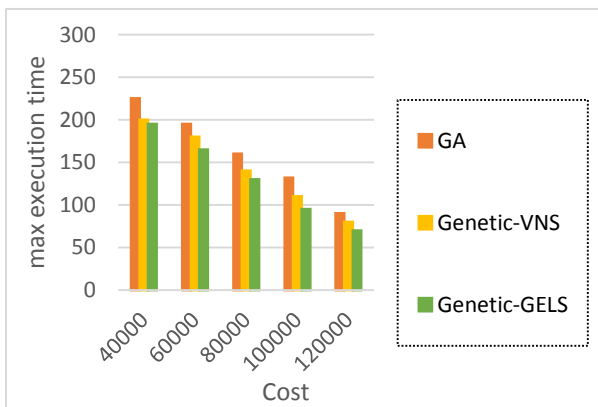


Figure 5. Results of time-optimization algorithms with different heterogeneities

7. REFERENCES

[1]Buyya,R., Giddy J., Abramson,D., A case for economy grid architecture for service-oriented grid computing , in: 10th IEEE Internat. Heterogeneous Computing Workshop , San Francisco, CA, April 2001.

[2]Braun, T. D., Siegel, H. J , A taxonomy for describing matching and scheduling heuristics for mixed machine

6. CONCLUSIONS

In the proposed method, the researchers have combined genetic algorithm with binary gravitational emulation local search algorithm which has been used for scheduling in grid environment. The proposed algorithm, in which a weighted objective function is used considering the degree of importance of time and cost of user’s projects, gives more freedom for specifying the time and cost of users’ projects. In the use of this algorithm, similar to the objective function, the time and the cost, along with their weight, are considered based on the user’s perspective. The purposes of the proposed scheduling are to minimize completion time and cost of implementation of tasks for different tasks of users simultaneously.

The proposed scheduling is compared with two algorithms of [6] GA and [7] Genetic-Variable neighborhood search with regard to the parameters of time and cost. The results are investigated based on cost and user requests in different charts. They show that if we have limited number of resources as well as high number of duties, the proposed scheduler has the best performance compared to the other three schedulers in reducing the overall time and cost of scheduling. The results of experiments show that the hybrid genetic algorithm and the gravitational attraction can reach a high performance regarding creation of a balance between cost and tasks implementation scheduling. Further research can be conducted with regard to the practice of resource allocation to works in the proposed algorithm through using an approach based on fuzzy logic and using of fuzzy inference system.

heterogeneous computing systems, Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems, pp. 330-335, 1998.

[3]Cruz-Chavez, M., Rodriguez-Leon, A., Avila-Melgar, E., Juarez-Perez, F., Cruz-Rosales, M. and Rivera-Lopez, R,“Genetic-Annealing Algorithm in Grid Environment for Scheduling Problems, Security-Enriched Urban Computing and Smart Grid Communications in Computer and Information Science”, Springer, Vol. 78, pp. 1-9, 2010.

[5]Ghaedrahmati V.,Enayatallah Alavi S.,Attarzadeh I.,“A Reliable and Hybrid Scheduling Algorithm based on Cost and Time Balancing for Computational Grid "ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 3, No.9 , May , ISSN : 2322-5157, 2014.

[6]Gharooni fard. G, Moein darbari. F, Deldari. H, Morvaridi. A, "Scheduling of scientific workflows using a chaos- genetic algorithm", International Conference on Computational Science ICCS, pp. 1439- 1448, 2010.

[7]Holland, j. “Adaptation in Natural and Artificial Systems”, AnnArbor,MI: Univ. of Michigan Press. 220, no. 4598, pp. 671- 680. 1975.

[8]Kardani-Moghaddam, S., Khodadadi, F., Entezari-Maleki, R., and Movaghar,A., “A Hybrid Genetic Algorithm and Variable Neighborhood Search for Task Scheduling Problem in Grid Environment”, *Procedia Engineering*, 29, pp. 3808-3814,2012.

[9]Pooranian, Z., Harounabadi, A., Shojafar, M., Hedayat,.N, “New hybrid algorithm for task scheduling in grid computing to decrease missed task”, *World Acad Sci Eng Technol* 55,924–928,2011.

[10]Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.,"BGSA: binary gravitational search algorithm". *Nat. Comput.* 9(3), 727–745, 2010.

[11]Shojafar, M., Pooranian, Z., Abwajy,J.H., Meybodi, M., ,”An Efficient Scheduling Method for Grid Systems Based on a Hierarchical Stochastic Petri Net” ,*Journal of Computing Science and Engineering*,Vol. 7, No. 1, March, pp. 44-52,2013.

[13]Young,L., McGough,S., Newhouse,S. and Darlington,J., Scheduling Architecture and Algorithms within the ICENI Grid Middleware, in *Proc. of UK e-Science All Hands Meeting*, pp. 5-12,

Nottingham, UK, September 2003.

[14]Zhang, L., Chen, Y., Sun, R., Jing, S., Yang, B., “A Task Scheduling Algorithm Based on PSO for Grid Computing”, *International Journal of Computational Intelligence Research*, Vol 14. , No.1 , pp 37-43,2008.

[15]Zarrabi A.,Samsudin K.,"Task scheduling on computational Grids using Gravitational Search Algorithm",*Department of Computer and Communication Systems Engineering,University Putra Malaysia,Volume 17,issue 3, December,ISSN: 1573-7543,pp 1001-1011. 2013.*