www.ijarp.com ISSN 2456-9992 Page: 01-14

International Journal Advanced Research Publications

Research Article

Volume: 02 Issue: 05

TASK SCHEDULING USING HYBRID CUCKOO AND GENETIC APPROACHES IN GRID ENVIRONMENT

Dr. Rachhpal Singh, Dr. Rupinder Singh

AP, Khalsa College, Amritsar (Punjab) –India.

Article Received: 27 March 2026, Article Revised: 17 April 2026, Published on: 07 May 2026

***Corresponding Author: Dr. Rachhpal Singh**

AP, Khalsa College, Amritsar (Punjab) –India.

DOI: https://doi-doi.org/101555/ijarp.8269

***ABSTRACT*:**

The grid environment uses distributed resources to solve complex business, industrial, and technical issues. Therefore, a competent scheduling technique is required to achieve the grid's goals. The cuckoo optimization algorithm (COA) and genetic algorithm (GA) for work scheduling in a grid computing environment served as the inspiration for the development of the Hybrid Cuckoo Genetic method (HCG). In order to efficiently schedule numerous tasks with shorter schedules and load balancing, this HCG is used on a grid. In this case, the number of job sets is used to evaluate transmission time. The results demonstrate the effective management of complexity, load balance, and resource use.

KEYWORD: *Grid computing, Cuckoo optimization algorithm, Load Balance, Transfer time, Genetic algorithm, Make span time, Schedule length.*

1. INTRODUCTION:

Fame of internet with accessibility of high bandwidth structure of networks requires dispersed multi-user machines. Current searches in computer science launched a recent concern known as "Grid Computing" [1]. Grid computing is collection of machines or systems or computers or system resources from several distributed locations to achieve a common objective. Grid develops a system having non-interactive workloads which has a large number of folders or files or both. It is a method in resolving comprehensive struggles in commerce, technical science (engineering), medical science and computer science [2]. This computing engages sharing and management of sources, management and description of information, task scheduling, and load balancing so on. The purpose of this type of computing is to make use of the resources that are available for compound calculations by webs that are geographically scattered. This system's main idea is to share resources between software coats for ease of use and privacy. It is responsible for virtualization with direct running requests and resource search. In computational forms, grids feature distributed computations and simple computing logic. Logic is taken into account with simple virtual machines that have a large, robust system that can manage a large number of machines. Heterogeneous machines connected to one another with a range of resources in a shared format make up such a set.

Scheduling is the process of allocating shared resources over time to test grid system performance. It has been the focus of a significant amount of literature in the field of actions search. Examining computers and their scheduling issues, where tasks reference performance and systems mention resources, has been important.

In parallel computing, the task scheduling problem in a directed acyclic graph (DAG) is comparable to an NP Complete problem. Particle Swarm Optimization (PSO) in grid scheduling is one of the heuristic or meta-heuristic strategies that yield superior outcomes with suitable computing time [20]. In heterogeneous computing scenarios, heuristic scheduling techniques [3] [4] [5] are commonly employed. A resource may be utilized or processed in a shared or exclusive manner. Heuristic scheduling is the method used to determine whether or not a group of jobs can be scheduled in such a context and to identify the optimal option for creating the best scheduling system. Such algorithms can be used to systematically investigate a set of heuristics because this kind of problem is known as NP complete. Simple heuristics don't work well because of how complicated the problem is. In contrast to an ideal algorithm that requires exponential time complexity, an algorithm that employs combinations of these basic heuristics performs exceptionally well. Implementation workload and processing time are not detectable in dynamic grid systems. Thus, prognostic models are needed for grid scheduling.

Xin-she Yang and Suash Deb created the search mechanism and optimization method known as Cuckoo in 2009. The obligatory brood parasitism of certain cuckoo species, in which they lay their eggs in the nests of other host birds (of other species), promoted cuckoo. In this case, the Hybrid Cuckoo Genetic Algorithm (HCG) is a meta-heuristic method for resolving grid system task scheduling issues. Cuckoo search and genetic mechanisms both stimulate it. Compared to GA, this has the capacity for non-local pointing with fewer restrictions. The results show that HCG is effective for work scheduling in grid computing with a sequential imitated setup (experiment). Additionally, HCG avoids capturing non-global resources.

In Second Section, relevant research on parallel system architecture and its different models is presented. Cuckoo optimization and genetic approach methodology are discussed in Section third. The experimental evaluation and other relevant parallel processing concerns are examined in the fourth section and at end is conclusion.

2. Literature Survey:

Following is the Table showing the Study of the research:

Researchers	**Research Description**
Ajith Abraham et. al [3]	They have suggested computational grids and followed up with a succinct description of three natural heuristics: Tabu Search, Genetic Algorithm, and Simulated Annealing.
Fatos Xhafa and Ajith Abraham [4]	By evaluating certain key concepts from grid computing with scheduling dilemmas and decisions utilizing heuristic and meta-heuristic methods, they proposed this work.
Masoud Yaghini and Mohammad Rahim Akhavan Kazemzadeh [5]	They have introduced DIMMA, a design and implementation technique for meta-heuristic algorithms. The suggested approach consists of three primary stages, each of which involves a number of steps in which the performance that should be given up is explicitly identified in this document.
Jennifer M. Schopf [6]	They have put out a standard grid scheduling architecture. Source selection decisions should be made by the grid scheduler in the surrounding environment.
Javier Carretero and Fatos Xhafa [7]	They have suggested that GAs use schedulers to effectively allocate jobs to sources within the grid system. One of the main challenges in creating computational methods, such Grid and P2P, is scheduling in order to maximize their vast computing capacity.
Lei Zhang et. al [8]	In order to solve the problem of task scheduling in grid systems, they have developed a projected heuristic strategy based on the particle swarm optimization technique.
P. Mathiyalagan et. al [9]	This heuristic ant colony technique has been projected to work well in grid scheduling environments with stigmatic communication. Compared to the existing ant colony algorithm, the proposed ant colony algorithm in this study addresses the grid scheduling problem more effectively thanks to a customized pheromone update regulation.
Raksha Sharma et. al [10]	They have put forth the goal of scheduling in order to attain the primary viable environment throughput and to provide a comparable function that requires the availability of computing resources. The review's objective is to convince recreational grid computing researchers.
Pinky Rosemarry et. al [11]	They have suggested doing this in order to achieve the most likely environment with maximum throughput and to achieve the same

	function requirements with accessible calculation sources.
Xin-She Yang and Suash Deb [12]	They have suggested developing Cuckoo Search, a novel meta-heuristic technique for issue optimization. It is based on the obligatory brood parasitic performance of a few cuckoo genera combined with the levy flight performance of certain fruit flies and birds.
Jean-Paul Watson [13]	He has made predictions on local searches. He claims that the best methods for finding high-quality solutions to a variety of combinatorial optimization problems—that is, obtaining the best outcomes in job shop scheduling—are always local search strategies.
Liang Sun, Xiaochun Cheng and Yanchun Liang [14]	They suggest an algorithm for the selection process that uses hypermutation values and an extended life span approach. They used an adaptive penalty function as the fitness function in this unique search process, which yields the best results in both feasible and impractical areas of the solution space.
Hedieh Sajedi and Maryam Rabiee [15]	They suggested scheduling jobs. One of the most challenging NP-hard problems encountered in practice, the examination is conducted within the framework of the well-known job-shop scheduling problem.
J.C. Beck, T.K. Feng and J.P. Watson [16]	This work, which they have proposed, is an important step toward a theory of local search. They develop theoretical models of the performance of four popular local search algorithms—a random walk, tabu search, iterated local search, and simulated annealing—using observed techniques.
Liang Sun et. al [17]	For work shop scheduling, they have suggested using GA with a penalty mechanism. Here, a life span-extensive policy and a clonal assortment dependent on hypermutation are designed. In order to allow the method to explore both feasible and infeasible portions of the solution space, an adaptive penalty function is used.
Said Fathy El-Zoghdy [18]	He resolves the grid computing load balance and work migration issue. This load balancing policy is intended to manage the many jobs on a heterogeneous cluster system in grid computing.
Saeed Molaiy [21]	He created a solution to address the work scheduling issue and suggested that, by allocating the heterogeneous resources using the ant colony optimization technique, task scheduling is a crucial component that enhances the effectiveness and performance of grid computing systems.

3. METHODOLOGY:

Followings are the explanation of both techniques and their hybrid approach:

A. Cuckoo Search (CS):

Xin-she Yang and Suash Deb created the optimization technique known as "cuckoo search" [12]. A few species of cuckoo were forced to engage in clutch parasitism by depositing their eggs in the nests of other host birds. Few host birds are able to establish a connection by clashing with cuckoos that interfere. A host bird will either throw these unidentified eggs away from its nest or construct a new nest somewhere else if it determines that the eggs are not within its grasp. Cuckoo optimization uses nests as resources, cuckoo as a grid broker, and cuckoo eggs as freshly arrived tasks. The host's eggs are regarded as jobs in a queue, with egg features serving as restrictions. A newly arrived task is chosen for execution if it guarantees the limitations of tasks in holdings. If not, the work is eliminated from that resource, and another optimal resource is chosen for that task. The steps of the suggested algorithm (HCG or Hybrid Cuckoo Genetic Algorithm) are as follows:

1. Using random particle cuckoo's habitat is initialized and computed the value of random particles.
2. Calculate ELR for each cuckoo as in equation (1) below:

$$ELR = \alpha * \frac{NE}{TE} * (var_{hi} * var_{low}) \quad (1)$$

Where ELR- Egg laying radius, NE- Current eggs numbers and TE- Eggs total number.

1. Cuckoo will lay egg in their subsequent space. This process will continue till required number of eggs lay down in the nest. Host birds recognize the eggs must be slayed and ensured that the security is in the full form.
2. Eggs shade, chick mature and that creates one more optimized form of results. Estimate habitat of every recently grown-up cuckoo.
3. Apply limit constraint on cuckoo's maximum number in surrounding. Also destroy all live in the pits habitats. Cuckoos locate finest group and choose objective habitat and this help in finding fitness function.
4. Allow new cuckoo population shift toward objective habitat that is another form of success.
5. If condition met then its ending, otherwise go to step 2. It means the procedure is complete.

B. Genetic Approach (GA):

The well-known heuristic method, the genetic algorithm, is effective and helpful, but the challenge is enormous and challenging. Natural selection, which "selects the best and discards the worst" in order to find a satisfactory result, is supported by genetic algorithms. The genetic algorithm achieves evolution by continually applying the set of genetic operators after starting with the population generation of solutions for the problem. Crossover and mutations are two fundamental processes used in each iteration of GA to produce more generations. GA begins with the formation of a started population, and further created populations undergo iterative evolution. Additionally, if the current generated population has a higher fitness value than the current generated population, an objective function known as the fitness function is calculated in each iteration with each individual. Additionally, the previous population is replaced by a new one, and this process continues until either n generations occur or the stopping criteria are satisfied [22]. Selection, crossover, and mutation are the three search operators employed in the genetic algorithm that greatly aid in finding the best answer. In order to obtain the optimal solution, GA often comprises of the following components:

Population:

The main set utilized in this procedure is the population. A population is defined as a collection of chromosomes. Random variables are used to generate it. The selection operator uses it, and the crossover operator uses it for swapping, and the mutation operator uses it for changing its shape. This phase is also known as the goal-achieving phase because the production of chromosome sets is its primary activity.

Fitness function:

The goal function is the fitness function. Here, the fitness function plays a crucial role in carrying out the subsequent processes. To determine the precise ideal solution, fitness is calculated for each chromosome. Results are simply attained with the aid of the fitness feature. The fitness value is obtained using a number of calculations and reasoning.

Selection:

The fundamental criteria used on the population is called selection. Ranking metrics are the basis for selection. The chromosome with the lowest fitness value will be swapped out for another chromosome once each chromosome's fitness is evaluated. As a result, a new population is transformed. Selection is comparable to choosing a cricket team from a list of the nation's athletes.

Crossover:

Crossover refers to the application of population data switching following a selection process. The partial variables in the chromosome are switched with one another using probability in the crossover procedure. Either a single point crossover or a two point crossover uses this.

Mutation:

Mutation operation is rarely used. Mutation is applied to preserve genetic diversity from one generation of a population of chromosomes to next one i.e. invert the randomly selected bits based on the mutation probability. Mutation is creation of different spices in the genetic approach.

COA and GA are combined to create the Cuckoo genetic algorithm. The genetic approach's disadvantage is that it takes a fair amount of time to estimate and requires expanding a trial in order to characterize the best features. Its quick data transfer rate between particles, which raises the likelihood of being in a non-global optimal, is another disadvantage.

In recent years, there has been a growing interest in using meta-heuristic algorithms to solve issues relating to redundancy and dependability in resource assignment. To tackle the optimization of allocation problem, a recently created meta-heuristic optimization algorithm called cuckoo search (CS) is hybridized with the well-known genetic algorithm (GA). The balance between exploration and exploitation capabilities was further enhanced by incorporating genetic operators into standard cuckoo search, and the algorithms' performance revealed greater search space [19].

To address the aforementioned flaw, this suggested method combines the cuckoo search methodology with a genetic approach. There are two stages to HCG. One uses cuckoo optimization, while the other uses a genetic technique. A population of cuckoo eggs in a region with the greatest potential for expansion is fed to the second stage of HCG at the conclusion of the first stage. One point crossover is consumed at the subsequent stage of HCG. Additionally, new populations will be impacted by mutation. When a cuckoo undergoes a mutation, it is randomly selected from the population and replaced by another member of the population. The process of mutation creates a cuckoo population, where the best-performing generation is hoarded, and so on.

The process of allocating resources for tasks that have some significance is known as scheduling, and it results in the completion of the work in its entirety. Virtual computing components such as threads, processes, or data flows could constitute the task. Cuckoo search and genetic techniques use job scheduling to develop a strategy that yields the best possible outcome. The goal of HCG's scheduling efforts is to assign jobs to clients in a way that minimizes completion time, maximizes resource efficiency, and accelerates convergence. HCG is employed when dealing with complex hypotheses [15]. This provides accurate and effective solutions. The following are the steps:

Developing initial population:

Spawning initial population is pedestal of beginning in this approach. Population of cuckoos are recorded into a set of tasks. Every cell of habitat has mapped to possessions which are situated in process of every task.

Items arriving here, together with *n, n* is no. of job sets, *j* the no. of tasks (because every task set has many tasks), *r* is no. of resources, *V n*r* is order of tasks for every task set and time required for every task, this approach tries to search best result. It is assumed that no. of resources with no. of operations is equal: *r=j*.

Choose an optimal sequence of systems:

Sequence of run-time tasks on each resource is different. So for each job *a[i]r*r,* minimum execution time should be resulted. Nevertheless, presentation of a resource to do a task alone is not determinative, but all resources have minimum time to achieve a task with postulation that a resource should not be used more than once.

Calculation of Fitness Function:

Here, it is assumed that each job can only be carried out on a system where there is a one-to-one relationship between a system and a task's operations. Therefore, the algorithm's requirement is satisfied by the fitness function applied here. Finding the conjecture and consequence coefficient related to the fitness function (FF1) is necessary to ensure this. Equation (2) defines the fitness function:

$$FF_1 = \sum_{j=1}^{n} min\,(ET) + FL \qquad (2)$$

Where j is no. of tasks, *FL* is penalty coefficient and *ET* is the execution time of tasks.

Once the entire work sequence has been obtained, it is allocated to the resources by examining the load balance, execution time, and transfer time.

Operations based on the input sequence:

After adjusting task execution times and determining the optimal task sequence for resources, resources will be allocated according to input succession. A number of tasks will be reviewed; the user must determine which task should be sent to which system. In this case, operations are assigned to resources using operation-based management. The number of tasks perfectly corresponds to the number of resources since each task precisely executes on each resource once.

4. Experimental Analysis:

Analysis was done on the performance of many aspects of work scheduling for parallel systems with the optimization of minimal make span time, minimized total transfer time, and efficient load balance. The optimization function with the specified task schedule length is used to analyze the make span time, total transfer time, and effective load balancing.

Although a variety of tools, such as SimGrid, Bricks, Gang Sim Arena, OptorSim, GridSim, Griddingcomp, and Alea, are available for building and developing simulating scheduling algorithms in grid computing systems, the developed simulation is unique and superior to all of them. It has all the tools needed to simulate and model every event and operation that occurs in grid computing environments, including managing heterogeneous resources, load balancing resources, a help box for system users, applications that simulate resource management, and the development of DAG between jobs and processors. It facilitates the development and computation of load-balancing algorithms. A heterogeneous grid environment was constructed utilizing various resource specifications in order to calculate the performance of the suggested load balancing technique.

Keep in mind that the distinction only applies to Random Access Memory, the central processing unit's processing speed, and the operating system. The system's performance in the grid computing environment is impacted by this. Simulated heterogeneous computers or machines in the grid system are created based on a specified set of workloads, resources, and the number of processors or machines in order to evaluate the performance of HCG.

MATLAB 13a version with an advanced optimization tool is used to build the experiment or system model. First, the system resources and jobs are chosen, after which the resources are split into a number of resource sets and the jobs are separated into job sets. The objective of the experimental study is to optimize the optimal resources for the job sets using the HCG algorithm by dividing 100 jobs into 14 job sets and 13 processors into 100 resource sets. All of the simulation calculations were carried out on a computer running the most recent version of Windows 8.1, with a core 2 dual processor (C2D) operating at 3.9 GHz and 4GB of internal memory (RAM). The bandwidth speed is selected between 50 and 100 Mbps, and the time unit is seconds.

The HCG algorithm and its mechanism are the sole optimal schedule or method for resolving any task or work scheduling issue. When compared to previously calculated methods such as the ant colony algorithm, particle swarm optimization algorithm, or basic genetic algorithm, it is the best. Therefore, schedules derived from different algorithms can be compared and computed in order to determine the optimal schedule utilizing the HCG algorithm.

However, for very large scheduling issues (i.e., problems with numerous jobs or tasks with multiple processors), finding an optimal plan using the HCG technique is extremely difficult, if not impossible, due to the combinatorial nature of the scheduling problem. In a parallel system or grid computing environment, a load balance approach is utilized to distribute the load across the processors. Task balance or load balancing between tasks or jobs between heterogeneous machines or processors may not be satisfied in some of the strategies or approaches that yield the best schedule length or smallest makespan with optimized outcomes or solutions.

The suggested mechanism or technique manages load balance modifications with updated procedures between tasks and processors, nodes, or machines to obtain the shortest schedule length (minimum makespan) and fulfills the load balance in the task-processor relationship in order to prevent such situations or problems. During load balancing between tasks and processors, the simulator calculates the transmitting time of specified job sets. The simulator goes through several rounds in order to evaluate the necessary optimized outcomes and determine the optimal solution.

Fig. 1: Number of Job Sets and their Transfer Time.

The number of job sets and the associated transfer time are displayed in Fig. 1. As previously mentioned, the MATLAB simulator is used to calculate these numbers. This experiment used a grid environment, a cuckoo mechanism, and a genetic approach. This will establish a connection between the specified set of work sets and calculate the transmitting time based on the specified parameters. In order to prevent duplication during processing, the jobs are carried out on separate processors. This was put up in the simulator.

The definition of load balancing is the division of total work so that a machine, node, or processor must be set up between two or more machines in order to complete a significant amount of work in the allotted time period or interval and set the optimal manner to get the best answer. This will ensure that every user receives the fastest possible service. The number of job sets and load balancing with resources are displayed in Fig. 2. By changing the job sets and the various values, the MATLAB simulator computes these values.

Fig. 2: Number of Job Sets and their Load Balancing.

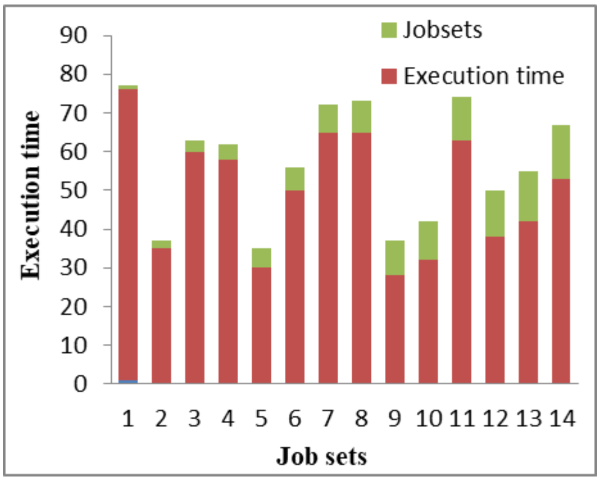

Fig. 3: Number of Job Sets and their Execution time.

This is calculated using a separate range of load balancing readings. Job distribution and load balancing are crucial in this grid computing system. There should be an imbalance between resources or processors and jobs or job sets if a processor or resource is frequently used (repetitively) in this model or simulator. This will lead to inaccurate outcomes. Simulator procedures were therefore configured to prevent recurrence in order to avoid this issue. Execution time is the amount of time needed to run an algorithm on a set of grid computers or parallel machines for a given input. Fig. 3 displays the execution time or make span time of the specified work sets. Because separate job sets contain different types of jobs, their execution times vary. As we wrap up our performance study, we would like to examine the system's performance in the future by looking at its scalability and how well it uses the resources available with the variety of jobs.

5. CONCLUSION:

Here, HCG is suggested as a solution to the work scheduling conundrum in a grid setting. This method comes before COA and GA's drawbacks. By choosing the best resource and calculating the least amount of time needed to complete tasks on parallel computer systems or grid computing environments, resource sets and job execution based on the Cuckoo-genetic algorithm with optimum scheduling of two work sets are investigated. The efficient design of our suggested system leads to the conclusion that task scheduling generates efficient scheduling outcomes in parallel systems. Managing load balancing with reduced makespan time and total transfer time is the primary emphasis of the study. Future discussions aim to expand our optimization technique to assess the parallel system in order to increase scalability.

REFERENCES:

1. I. Foster, C. Kesselman, "*The Grid: Blueprint for a Future Computing Infrastructure*", Morgan Kaufman Publishers, USA, 1999.
2. I. Foster, C. Kesselman, "*The Grid 2: Blueprint for a New Computing Infrastructure*", 2nd ed., Morgan Kaufmann, 2004.
3. Ajith Abraham, Rajkumar Buyya and Baikunth Nath, "Nature's Heuristics for Scheduling Jobs on Computational Grids", *IEEE international conference* on advanced computing and communications, 2000.
4. Fatos Xhafa and Ajith Abraham, "Meta-heuristics for Grid Scheduling Problems, *Springer*, 2008.
5. Masoud Yaghini and Mohammad Rahim Akhavan Kazemzadeh, "DIMMA: A Design and Implementation Methodology for Meta heuristic Algorithms - A Perspective from Software Development, *International Journal of Applied Meta heuristic Computing*, 1(4), 58-75, October-December 2010.

6. Jennifer M. Schopf, "Ten actions when grid scheduling".
7. Javier Carretero and Fatos Xhafa, "Genetic algorithm based schedulers for grid computing systems", *International Journal of Innovative Computing*, Information and Control ICIC International, Vol. 3, 2007
8. Lei Zhang, Yuehui Chen, Runyuan Sun, Shan Jing and Bo Yang, "A Task Scheduling Algorithm Based on PSO for Grid Computing", *International Journal of Computational Intelligence Research*, Vol.4, pp. 37–43, 2008.
9. P. Mathiyalagan, S.Suriya and Dr. S. N. Sivanandam, "Modified Ant Colony Algorithm for Grid Scheduling", *International Journal on Computer Science and Engineering*, Vol. 02, pp. 132-139, 2010.
10. Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra and Prachet Bhuyan, "A Survey of Job Scheduling and Resource Management in Grid Computing", *World Academy of Science*, Engineering and Technology, Vol.4, 2010
11. Pinky Rosemarry, Payal Singhal, and Ravinder Singh, "A Study of Various Job & Resource Scheduling Algorithms in Grid Computing", *International Journal of Computer Science and Information Technologies*, Vol. 3, pp. 5504-5507, 2012.
12. Xin-She Yang and Suash Deb, "Cuckoo Search via L´evy Flights", *IEEE Publications,* pp. 210-214, 2009.
13. Jean-Paul Watson, "*Empirical modeling and analysis of local search algorithms for the job-shop scheduling problem*", 2003.
14. Liang Sun, Xiaochun Cheng and Yanchun Liang, "Solving Job Shop Scheduling Problem Using Genetic Algorithm with Penalty Function", *International Journal of Intelligent Information Processing*, Vol.1, 2010.
15. Hedieh Sajedi and Maryam Rabiee, "A Meta heuristic Algorithm for Job Scheduling in Grid Computing", *I.J. Modern Education and Computer* Science, vol.5, pp. 52-59, 2014.
16. J.C. Beck, T.K. Feng and J.P. Watson, " A Hybrid Constraint Programming / Local Search Approach to the Job-Shop Scheduling Problem", *INFORMS Journal on Computing*, Vol. 23, No. 1, pp. 1–14, 2011.
17. Liang Sun, Xiaochun Cheng , Yanchun Liang , "Solving Job Shop Scheduling Problem Using Genetic Algorithm with Penalty Function", *IJIIP: International Journal of Intelligent Information Processing*, Vol. 1, No. 2, pp. 65 ~ 77, 2010
18. Said Fathy El-Zoghdy, "A Hierarchical Load Balancing Policy for Grid Computing Environment" *J. Computer Network and Information Security***,** 2012, 5, 1-12 Published

Online June 2012 in MECS, (http://www.mecs-press.org/) DOI: 10.5815/ijcnis.2012.05.01

19. G. Kanagaraj, S.G. Ponnambalam, N. Jawahar, **“**A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems”, *Computers & Industrial Engineering***,** Volume 66, Issue 4, December 2013, Pages 1115–1124.
20. Ritu Garg, Awadhesh Kumar Singh, “Enhancing the Discrete Particle Swarm Optimization based Workflow Grid Scheduling using Hierarchical Structure”, *I. J. Computer Network and Information Security*, 2013, 6, 18-26, Published Online May 2013 in MECS (http://www.mecs-press.org/),
DOI: 10.5815/ijcnis.2013.06.03.
21. Saeed Molaiy, Mehdi Effatparvar, “Scheduling in Grid Systems using Ant Colony Algorithm”, *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
22. Chandrashekhar Azad1, Vijay Kumar Jha, “Genetic Algorithm to Solve the Problem of Small Disjunct In the Decision Tree Based Intrusion Detection System”, *J. Computer Network and Information Security,* 2015, 8, 56-71, Published Online July 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.08.07