IJCTA, 9(10), 2016, pp. 4779-4786
© International Science Press

Role of GA, GSO and SLPSO algorithms in task scheduling

***Nazneen Taj **Dr. Anirban Basu**

Abstract : Cloud computing provides services to the cloud user on the basis of pay as use model similar to our utility services. Not only this it also helps us to reduce the overall information technology cost and improve the economy of IT sector.Due to these remarkable features of cloud many IT sectors have come forward to use cloud to run their application and improve the economy of their company.Infrastructure asa Service (IaaS) has become the foundation forhigher level services like Platform as a Service (PaaS)and Software as a Service (SaaS) in cloud computing. Amazon EC2 and IBM Smart Cloud Enterprise which are the IAAS providers rent resources like virtual machines to the users of cloud.

In this paper we are comparing three task scheduling algorithms to calculate the average memory utilization and CPU utilization in-turn we will find the maximum profit and average run time of all the three algorithms.

Keywords : Cloud computing, virtual machine, task scheduling, genetic algorithm, group search optimization algorithm, GGSO, Infrastructure as a Service (Iaas), Quality -of-Service (QoS)

1. INTRODUCTION

Cluster computing and grid computing paradigms came forward to provide services to the clients based on pay as you use modal with reference to these, cloud computing is the new trend in the networking.The word cloud refer to as global model from which clients and user can access applications, platforms, resources and so on from any part of the world.Therefore,cloud computing made the software services to be accessible and utilized by the common man rather being confined to the computers of IT sectors[1]. Three aspects are new in CloudComputing.

1. Users of cloud need to plan ahead for provisioning as infinite resources are not available in cloud.
2. Allow companies to increase hardware resources only when it is need, thus eliminate up front commitment of cloud users.
3. Making sure to use the resources by paying only till they are actually required and releasing them back so that the enough resources are available in cloud[2].

Three sorts of administrations are conveyed: Platform as a Service (Paas), Infrastructure as a Service (Iaas) and Software as a Service (Saas). Cloud clients utilize these administrations at whatever point required by interest utilizing pay-every use model. Iaas suppliers, for example, Amazon EC2 and IBM Smart Cloud Enterprise [3]. Dynamic schedulers can be distinguished as Pre-emptive and Non pre-emptive schedulers.In Pre-emptive scheduling the task can be interrupting during its execution but in the later it can be interrupted. By making the task to adhere deadlines pre-emptive scheduling algorithms can be used to implement real time systems and also to remove deadlocks [4].Thus if we want to save energy consumption we need to select an algorithm which uses lower run time but fast in computation speed with less resource utilization and of course more profit to the service provider and the customers of the cloud.

* Asst. Prof., Dept. Of CSE KNSIT, Bangalore-560064 nazneentaj@rediffmail.com
** Prof. Dept. Of CSE APSCE, Bangalore, India abasu@anirbanbasu.in

2. REVIEW OF LITERATURE

Comparing the bee swarms Chong and Low (2006) developed computation based on Bee Colony Optimization (BCO) to solve JSSP.They considered the state transition rule to do the scheduling.Improved artificial bee colony algorithm (IABC) was developed by to enhance the search performance.To avoid local optima and explore search space the mutation operation was utilized in this[5].

The fundamental issue of planning was the method to allocate clients' assignments to increase the advantages of Iaas supplier simultaneously guaranteeing the Qos. The problem was designed as an integer programming (IP) show, and unravelled by flexibility towards the self-adaptive learning particle swarm optimization (SLPSO)-based scheduling method in. Their technique was not found appropriate for elevated issue example e sorts in view of the needed implementation of computational duration. These investigations, particularly XingquanZuo et al. [6].with a specific end goal to minimize the expense of the transforming the creator LizhengGuo et al.[7] have clarified the Task Scheduling employing the optimization strategy (PSO) which is focused around little position quality principle. Raj Kumar et al. proposed that when we are using the cloud services we need to provide security to the cloud customers thus he proposed security solutions like continuation mechanism, data security and virtualization security[8].

3. PROPOSED COMPARISON

In this paper we have compared three algorithms based on the resource utilization and average run time, first we start with the explanation of each algorithm as follows:

A. Genetic algorithm

This was first launched by Holland in 1975 [9], as an iterative stochastic in which the natural appraisal is employed to the search technique. The GA techniques can be employed to tackle the optimization issues by replicating the genetic function of the biological organism [10, 11]. As indicated by the name, the GA approaches follow the evolutionary theory in nature to tackle the optimization issues. A general genetic algorithm performs three genetic functions such as the selection, cross over and mutation. In the selection, certain solutions from the populations are chosen as parents where as in the crossover the parents are crossbred to generate the offspring. In the case of the mutation the offspring is organized in accordance with the mutation rules. In the genetic algorithm, a solution is termed as the individual and the iterations of the algorithm are labelled as the generation. Further, several genetic algorithms utilize the elitism. In other words, a lot of the best individuals are copied to the succeeding generation.

Quite different from various conventional search methods, the GAs employs multiple search nodes concurrently. Each and every search node relates to one of the existing solutions and is indicated by a sequence of symbol, which is known as the chromosome, whereas the symbol forming part of the sequence is called the genes. Each chromosome possesses a related value called the fitness value, which is assessed by means of the objective function (fitness function) value $f(x)$. In a GA, only superior chromosomes possessing high fitness values survive and create the offspring communicating their biological heredity to new generations. By evolving the chromosome incessantly, the solutions corresponding to the search nodes are steadily enhanced. A set of chromosomes at a particular stage of the GA is known as the pop. The number of chromosomes (individuals) in a population is known as the pop size. The elitism size represents the number of fitness individuals that are copied directly to the succeeding generation. Algorithmically, the fundamental genetic algorithm (GAs) is explained as follows:

Step 1 : Initialize the random population of chromosomes which is the appropriate solution for the problem.

Step 2 : Evaluation of Fitness function of chromosome $f(x)$ in the population.

Step 3 : Repeat the following steps until new population is got.

- According to their fitness Select two parent chromosomes from a population.Good fitness, the bigger change to be selected to be the parent.
- Form new offspring from parents with cross over probability if not then offspring will be the exact copy of the parents.

- At each locus mutate new offspring with mutation probability.
- In the new population include the new offspring.

Step 4 : Repeat the same with the new generated population.

Step 5 : If the end condition is met, discontinue, and return the best solution in the existing population.

Step 6 : Repeat from step 2.

B. Group search optimization algorithm

The GSO technique is at first proposed by [11] [12]. It has often paved the way for the adoption of two foraging techniques within groups: 1) creating and looking for food; and 2) scrounging, joining resources exposed by others. With a view not to entrap in local minima, the GSO further deploys the "ranger" foraging techniques. The population of the GSO approach is known as a group and each individual in the population is termed a member.

The algorithm is as follows :

1. Set K = 0
2. Randomly initialize the position Y_1 and head angle ϕ_1 of all members & Calculate the fitness value of the initial members: $f(Y_1)$.

   WHILE (the termination conditions are not met)

   FOR (each members *i* in the group)
3. **Choose producer :** Find the producer Y_p in the group;

Perform producing :

1. Producer scans at degree and then scans randomly sampling three points in the sampling field.
2. It will jump to the new point with the best resource and if it doesn't get the best resource it will stay back at the old point and turns its head to new angle.
3. Producer will turn its head back to zero degree if it does not find the better area iteration .
4. **Perform scrounging :** To perform scrounging randomly select best members from rest 80%.

Perform ranging : Ranging will be performed to the residual members.

1. generate a random head; and 2) choose a random distance l_i from the Gaussian distribution move to the new point.

   Verify feasibility : Each of the group member should not violate the constraints. If it does, it will move back to the previous position to guarantee a feasible Solution.

   END FOR
5. **Set $k = k + 1$;**

   END WHILE

C. SLPSO (self-adaptive learning particle swarm optimization)

All of the works mentioned highlighted the neighbourhood assets allotment in a lonely Iaas cloud, and do not include the scheduling assignments among distinctive clouds. Further, the Priority was a critical issue of task scheduling in cloud scenarios. In these cases, Iaas suppliers declined assignment requirements when its assets were not enough to complete those assignments. Conversely, it has a detrimental impact on its assured Quality-of-Service (Qos) and infamy. In order to address this problem, XingquanZuo et al. [13] launched a Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid base as an administration (Iaas) Cloud. The fundamental issue of planning was the method to allocate clients' assignments to increase the advantages of Iaas supplier simultaneously guaranteeing the Qos. The problem was designed as an integer programming (IP) show, and unravelled by flexibility towards the self-adaptive learning particle swarm optimization (SLPSO)-based scheduling method in [13]. Still, their technique was not found appropriate for elevated issue example sorts in view of the needed implementation of computational duration. These investigations, particularly XingquanZuoet al. [13]) motivated me to go on with my evaluation for task planning issue aided by a hybrid optimization technique.

4. RESULT AND DISCUSSION

These comparisons are performed on the Mat lab version (7.12), employing windows machine having Intel Core i5 processor with speed 1.6 GHz and 4 GB RAM.

- **Experimental design :** Three problem examples are planned. Example 1 contains 8 applications and its parameters are demonstrated in Table 4. VM instance type demanded by each application is randomly selected from the above three VM types. The deadline of each application is a consistently disseminated random integer between 1 h and 5 h with an eye on restricting the search space. To guarantee that the deadline of each application exceeds its runtime, the latter is chosen as a consistently apportioned integer between 1h and its deadline. With a view to reproduce the situation of resource shortage we take the number of CPU as 20 and the size of memory as 40 GB. Instances 2 and 3 comprise 5 and 10 applications respectively and their parameters are vividly exhibited in Table 5.

Table 1. VM Instance types

Name	*CPUs*	*Memory*
Small	1	1.7
Large	4	7.5
X large	8	15

Table 2. Private Cloud's Cost and Price

	Cost	*price*
Small	0.03	0.08
Large	0.12	0.32
Xlarge	0.24	0.64

Table 3. EC' price

ECs	*Small*	*Large*	*Xlarge*
A	0.085	0.34	0.68
B	0.070	0.30	0.70
C	0.100	0.40	0.72

Table 4. Parameters of problem instance 1

Application		*Cloud resources*	
Parameters	*Values(integer)*	*Resources*	*Number*
Number of tasks	~unit[1,5]	CPU	20
VM instance type	~unit[1,3]	Memory	40GB
Deadline (hours)	~unit[1,5]		
Runtime (hours)	~unit[1,Deadline]		

Table 5. Parameters of problem instance 2 and 3

Application		*Cloud resources*	
Parameters	***Values(integer)***	***Resources***	***Number***
Number of tasks	~unit[1,50]	CPU	512
VM instance type	~unit[1,3]	Memory	1024GB
Deadline (hours)	~unit[1,168]		
Runtime (hours)	~unit[1,Deadline]		

1 **Problem Instance 1 :** The suggested approach attained maximum profit in 24 assessments for this small size instance, which is high compared with other algorithms like GSO,GA and SLPSO. The standard profit acquired in the 24 runs of SLPSO, GA and GSO and their average runtime are specified in Table 6.

2. **Problem Instance 2 and 3 :** Problem instances 2 and 3 contain more tasks and devour more resources than instance 1. The suggested approach attained maximum profit in 24 evaluations for this large size instance, which is high compared with other algorithms like SLPSO, GSO and GA. The average profit attained in the 24 runs of SLPSO, GSO and GA and their average runtime are specified in Table 6.

Table 6. Comparison of profit and avg. Time for problem instance 1, 2 and 3

	Problem instance 1		*Problem instance 2*		*Problem instance 3*	
	Maximum	***Avg.***	***Maximum***	***Avg.***	***Maximum***	***Avg.***
Algorithms	***Profit***	***run time (ms)***	***profit***	***run time (ms)***	***proft***	***run time (ms)***
GA	2.18	14	3784	5485.4	2897.94	5435.54
GSO	4.5	15.5	4890	3945.56	4284.34	4734.85
SLPSO	4.9	29.55	3545	2934.37	2974.45	4300.28

Table 7. Comparison for CPU utilization rate

Average CPU Utilization Rate			
Algorithm	***Instance I***	***Instance II***	***Instance III***
GA	0.445632	0.45335	0.46332
GSO	0.55621	0.54224	0.613325
SLPSO	0.71245	0.68441	0.72335

Table 8. Comparison for memory utilization rate

Average Memory Utilization Rate			
Algorithm	***Instance I***	***Instance II***	***Instance III***
GA	0.41225	0.38665	0.574412
GSO	0.547784	0.532241	0.63225
SLPSO	0.68356	0.657741	0.732256

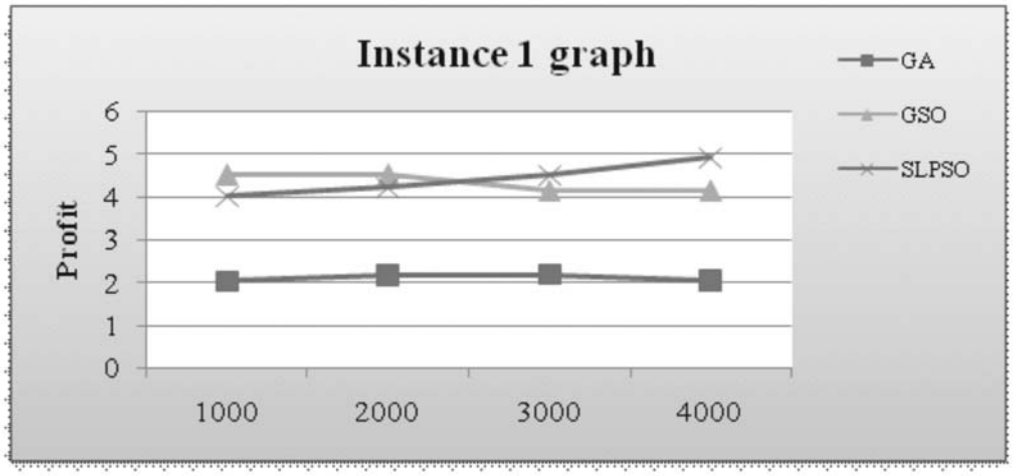

Fig. 1. Performance of profit plot for proposed against existing algorithm for instance 1 From the above graph it is clear that profit for instance 1 will be higher in SLPSO algorithm.

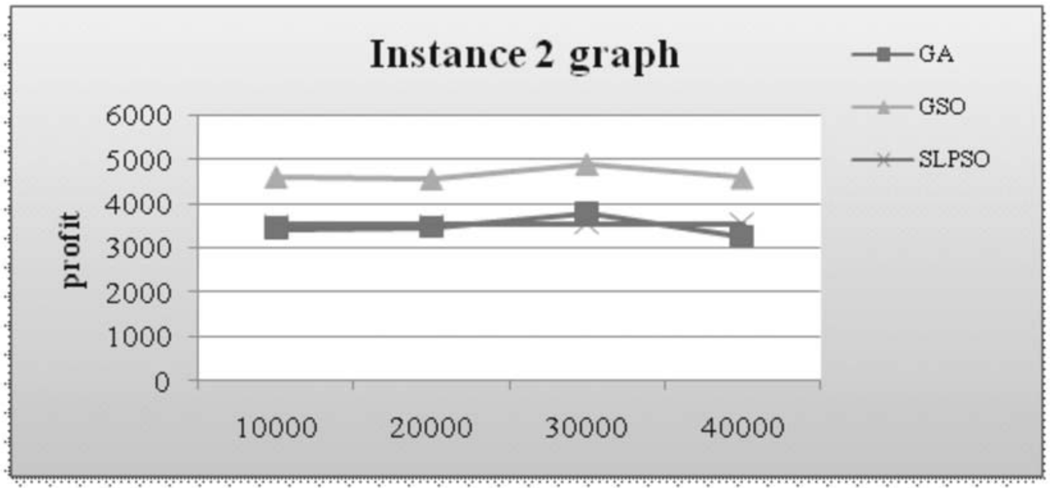

Fig. 2. Performance of profit plot for proposed against existing algorithm for instance 2 This graph shows that GSO will generate a higher profit for instance 2.

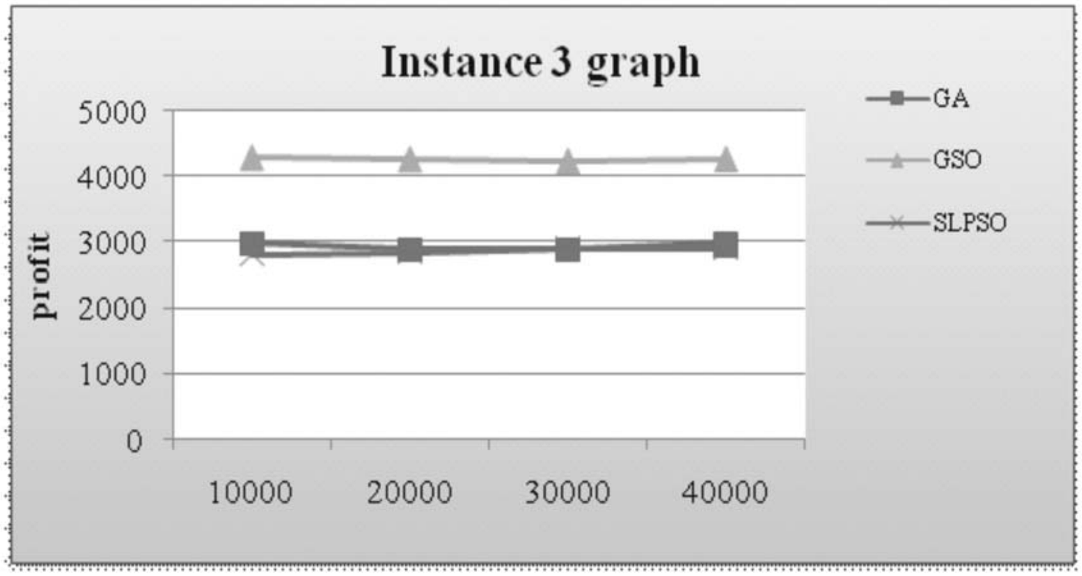

Fig. 3. Performance of profit plot for proposed against existing algorithm for instance 3. This graph shows that for instance 3 GSO will yield higher profit.

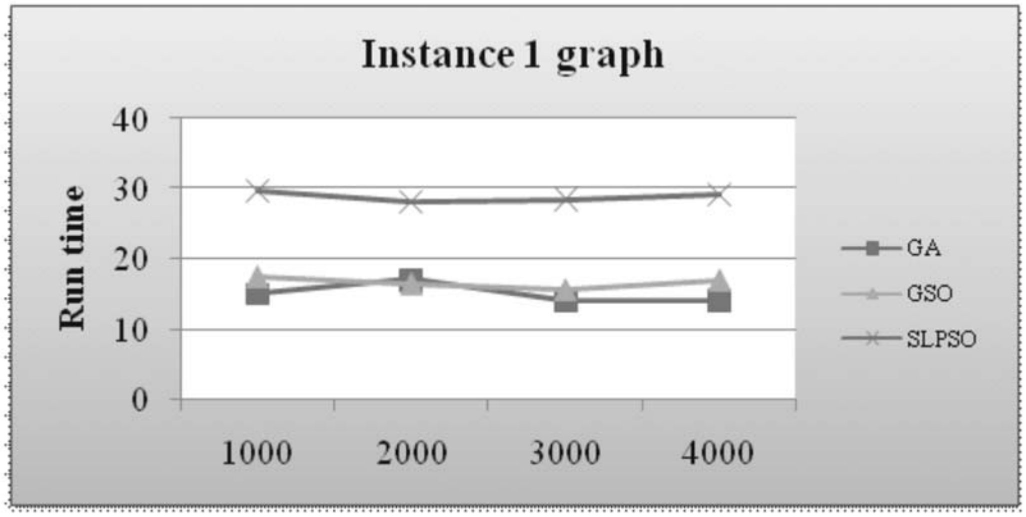

Fig. 4. Runtime performance of instance 1

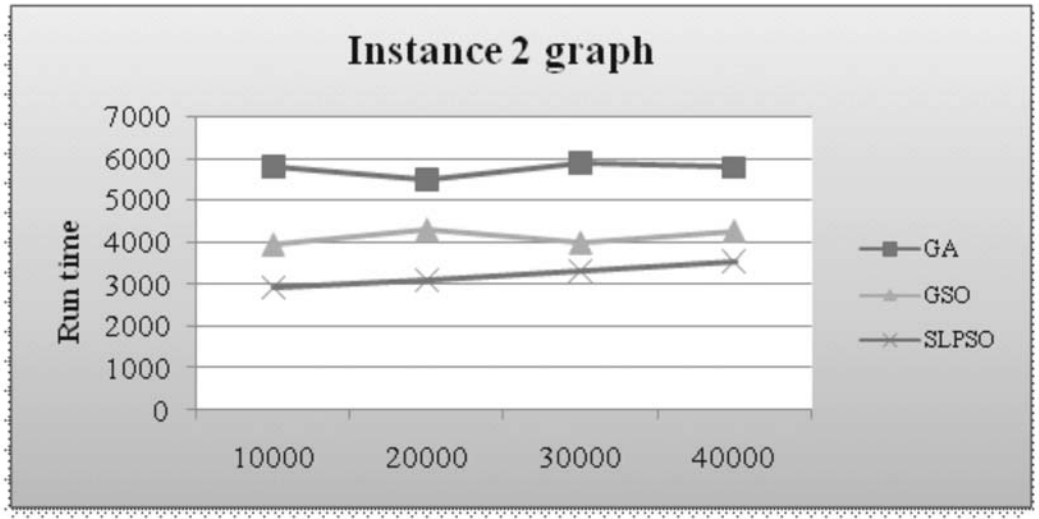

Fig. 5. Runtime performance of instance 2

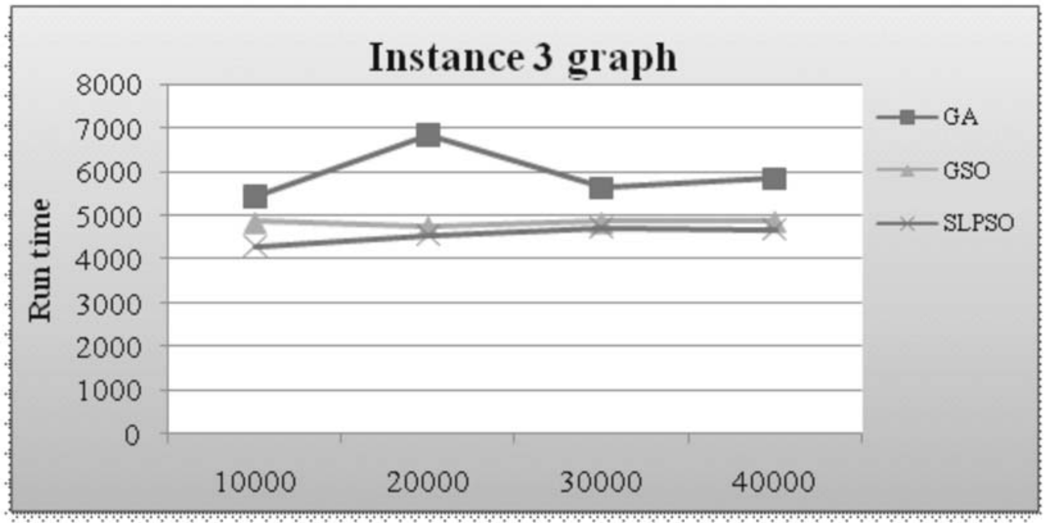

Fig. 6. Runtime performance of instance 3

From figs. 4,5& 6 SLPSO have more run time for instance 1, GA has more run time for instance 2 andGA has higher run time for instance 3.

5. CONCLUSION

In this paper, for the resources allocation problem of an IaaS cloud in a hybrid cloud environment an integer programming model is established.The main matter is how to assign users' tasks to maximize the revenue of Infrastructure as a Service (IaaS) provider while guaranteeing Quality-of-Service (QoS). From the above graphs we find that SLPSO is comparatively a better algorithm in case of average run time, CPU utilization and the profit. Hybridization of these algorithms can improve the performance of the cloud still better.

6. REFERENCE

1. R.Buyya, C.S Yeo, S.Venugopal, J. Broberg, and I. Brandic, “Cloud Computing and Emerging IT platforms”, Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Journal of Future Generation Computer Systems, Vol.25, No. 6, pp. 599-616, 2009.
2. M. Armbrust, et al., “A View of Cloud Computing,” Communication of ACM, vol. 53, pp. 50-58, 2010.
3. R. Buyya, C. S. Yeo, S. Venugopal, and J. Broberg, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Comput. Syst., vol. 25, no. 6, pp. 599-616, 2009.
4. Fatma A. Omara, Mona M. Arafa, “Genetic algorithms for task scheduling problem,” J. Parallel Distrib. Comput.Vol.70 ,pp.13-22, 2010. [29] Jiayin Li, MeikangQiu, JianweiNiu, WenzhongGao, ZiliangZong, Xiao Qin, “Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems”, IEEE, 2010.
5. Fatma A. Omara, Mona M. Arafa,“Genetic algorithms for task scheduling problem,” J. Parallel Distrib. Comput.Vol.70 ,pp.13-22, 2010.
6. XingquanZuo, Guoxiang Zhang, and Wei Tan, “Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud”, IEEE transactions on automation science and engineering, Vol. 11, No. 2, pp. 564-573, 2014.
7. LizhengGuo,ShuguangZhao,ShigenShen and Changyuan Jiang, “Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm”, Journal Of Networks, Vol. 7, No. 3, 2012.
8. Raj Kumar,“Research on cloud computing security threats using data transmission”, IJARCSSE,vol. 5, issue 1, Jan 2015.
9. J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, USA, 1975.
10. A. Zomaya, C. Ward, B. Macey, Genetic scheduling for parallel processor systems: comparative studies and performance issues, IEEE Trans. Parallel Distributing System, vol. 10, no.8, pp.795-812, 1999.
11. I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, “Effective leadership and decision-making in animal groups on the move,” Nature, vol. 434, pp. 513-516, 2005.
12. S. He, Q. H. Wu and J. R. Saunders, “A Group Search Optimizer for Neural Network Training” Lecture Notes in Computer Science, vol.3982, pp.934-943, 2006.
13. XingquanZuo, Guoxiang Zhang, and Wei Tan, “Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud”, IEEE transactions on automation science and engineering, Vol. 11, No. 2, pp. 564-573, 2014.