A Novel Twin Space Crowding Genetic Algorithm for Workflow Scheduling in Cloud Computing Environment

T. Sounder Rajan and Dr.M. Selvam

Abstract--- *Cloud Computing proffers wide computation and resource facilities for execution of various application workflows. A workflow is an ordered set of interrelated tasks that define the operational aspect of a process or procedure. Many different resources used in execution of single workflow. Cloud Computing offers highly dynamic environment in which the system load and status of resource are frequently varied. As the workload increases with increase in Cloud Services and clients there is a necessity to handle these requests or jobs. This work used a niche genetic algorithm based on a novel twin-space crowding approach for effectual work flow scheduling. The proposed Twin-space Crowding (TC) method is designed in a parameter-free paradigm. That is, when cooperatively exploring solutions with GAs, it does not require prior knowledge related to the solution space to design additive problem-dependent parameters in the evolutionary process. Experimental results show the better outcome of the proposed method.*

Keywords--- *Cloud Computing, Workflow Scheduling, Niche Genetic Algorithm, Twin-space Crowding, Hill–valley Functions.*

I. INTRODUCTION

Cloud Computing is a big-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted realistic, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet. The on-demand, self-service, pay by use features of cloud computing is also an elaboration of set features. Workflow is stated as the automation of a business process, complete or in segments, during which documents, data or labor are exchanged between one participants for action, in accordance with a set of procedural rules (Singh &Pushpendra, 2013).When talk about cloud, the major advantage of cloud is its application scalability or snapping. This snapping nature of cloud facilitates changes of resource and characteristics at run time. The cloud can be capable of doing workflow management systems to readily meet the quality of service requirements of application.

The WfMC (Workflow Management Coalition) describes workflow, recognizing the interfaces within this structure which permits products to collaborate at a sort of levels. Workflow scheduling includes mapping of network functions with given resources in a way that some preset criteria is met. Workflow scheduling is famous NP-omplete problem and significant cause in workflow management system. Oncoming workflow to cloud computing permits one to exploit the advantages of cloud for workflow implementation. (Kaur et al , 2011) scheduling could be

T. Sounder Rajan, Assistant Professor, Department of Computer Science Engineering, K S R Institute for Engineering and Technology, Namakkal, Tamilnadu. E-mail:tsounderrajan@gmail.com

Dr.M. Selvam, Principal, Thangavelu Engineering College, Chennai, Tamilnadu. E-mail:ammselvam@gmail.com

multi objective too. The multi objective feature of scheduling is difficult to solve. Hence , this analysis initiated a modern niche genetic algorithm relying on twin-space crowding. Niche techniques have been formed to measure down the genetic drift effects resulting from the replacement operator in the simple GA. By including scaling fittingness or by altering the fittingness competence rule , niche techniques change the easy GA by mentoring convergence such that multiple loci always provides niche Gas the toughness and efficient required to investigate optima in several multi model optimization problems (Chen et al , 2014). When availed to solve optimization problems, so, most niche techniques need prior knowledge like the niche radius or the distance threshold. To address these setback, this study suggests Twin-space Crowding (TC) , a fictional niche technique intentional in a parameter free paradigm; (i.e.) it doesn't need prior knowledge of practical optimization problems , and the suggested crowding mechanism does not need additional customized parameters. A simple workflow is shown in Figure 1 which represents a workflow schedule. In a workflow, the execution order of interdependent tasks is disciplined by their dependence, e.g. a task is always executed after its immediate parent tasks. The entry task is task 0, 1 and 2 and the exit task is task 7. The child tasks 3, 4, 5 and 6 are executed only after the execution of their parent tasks.

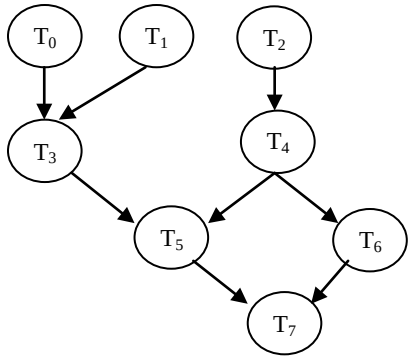

Figure 1: A Sample Workflow

The rest of the work is organized as follows, section 2 discusses about the related concepts, section 3 provides the research methodology, section 4 evaluates the proposed method in cloud environment and section 5 concludes the research.

II. LITERATURE SURVEY

Cloud Computing provides highly dynamic environment in which the system load and status of resource are frequently varied. The execution of cloud workflows faces many unknown factors in allocating and scheduling workload. The first step is to provide an effective workflow allocation model by considering the client's requirements given by Singh and Pushpendra (2013). Lin &Shiyong (2011) firstly formalize an ideal of a Cloud situation and a workflow graph indication for such an environment. Then, suggested the SHEFT workflow forming algorithm to form a workflow loosely on a Cloud computing situation. Bittencourt*et al.* (2011) handles with this problem, presenting HCOC: The Hybrid Cloud Optimized Cost scheduling algorithm. Hybrid Cloud optimized cost scheduling algorithm finalizes which infrastructure resources should be hired from the public cloud and aggregated to the private cloud to give adequate processing power to carry out a workflow within the stipulated period.

Bessai*et al.* (2012) suggest three compactable bi-criteria proposals for scheduling workflows on scattered Cloud resources, on account of overall execution time and the cost incurred by using a set of resources. Bala&Inderveer (2011) have surveyed different types of workflow scheduling algorithms and tabulated their different parameters along with tools, scheduling factors and so on.

Abrishami *et al.* (2013) adapt the PCP algorithm for the Cloud environment and propose two workflow programming algorithms: a one-phase algorithm known as Infrastructure as a Service (IAAS) Cloud Partial Critical Paths (IC-PCP), and a two-phase algorithm which is inferred as Infrastructure as a Service (IAAS) Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2).Wang *et al.* (2011) initially suggested a reliability-driven (RD) reputation, which is time base, and can be availed to measure the integrity of a resource in broadly distributed systems. They then propose a look-ahead genetic algorithm (LAGA) which avails the RD fame revamp both the make span and the reliability of a workflow application. Xue& Wu (2012) gives a QoS-based hybrid particle swarm optimization (GHPSO) to schedule applications to cloud resources. Mao & Marty (2011) present an approach whereby the basic computing elements are Virtual Machines (VMs) of different sizes/costs, jobs are specified as workflows, users specify performance requirements by assigning (soft) deadlines to jobs, and the goal is to assure all jobs are finished within their deadlines at minimum financial cost. Fard*et al.* (2012) proposed a common skeleton and heuristic algorithm for varied dimensional static scheduling of scientific workflows in haterogeneous computing environments.

Qing *et al.* (2008) proposed a novel genetic algorithm which combines crowding and clustering for multimodal function optimization, and examine convergence properties of the algorithm. Qu*et al.* (2012) integrates a novel local search method with some existing PSO based multimodal optimization algorithms to enhance their local search ability. An adaptive niche hierarchy genetic algorithm (ANHGA) is proposed by Ye *et al.* (2011). The algorithm is based on the adaptive mutation operator and crossover operator that alter the crossover rate and frequency of mutation of each individual, and adopts the gradient of the individualist to decide their mutation value.

Some of the multimodal optimization problems,a simple genetic algorithm cannot efficiently and simultaneously maintain multiple global or local optima. So, the population is easily trapped in a limited number of solutions, which is a condition that results in early solutions with no capability to obtain better outcomes. Hence, niche methods have been developed to minimize genetic drift effects which leads from the substitute operator in the simple GA. It is used to solve the optimization problems, but, most niche methods needs beforehand wisdom like niche radius or the distance threshold. Hence to resolve this limitation, this study suggested twin-space crowding which is parameter free paradigm and it does not need prior knowledge about practical optimization problems. The suggested method is explained in the following sections.

III. RESEARCH METHODOLOGY

A cloud computing is a huge network where the millions of users accessing thousands of servers all times. Scheduling the tasks at the server end is very difficult for a server, because the number Job requesting is very large in number, which required some resources to execute. This section provides the research methodology for workflow scheduling in cloud computing. The proposed Niche Genetic algorithms provide robust search techniques that allow

a high-quality solution to be derived from a huge search space in polynomial time, by applying the principle of evolution.

Twin-Space Crowding Genetic Algorithm

similar to a simple GA, a general niche GA initially introduces and measures the population to choose the best fitting chromosomes for the mating pool. It then employs crossover and mutation operators to generate and measure the new offspring (Chen *et al*., 2014).When it evolves a new generation, the common niche GA does not merely reserve individuals with the best fitness. It normally employs a certain substitute operator to help in choosing new parents.

Niche Family

One of the modern niche mechanisms is pre selection, in which an offspring could only substitute one of its parents. A crowding method known as Niche Crowding (NC) was then suggested for enhancing the pre selection process during which an offspring substitutes the most alike individual taken from a randomly chosen subpopulation depending on the Crowding Factor (CF) attribute.. In this method, the evaluation of likeness betwixt two real-code chromosomes is usually depending on the Euclidean distance. The NC tends to cause a large genetic shift in multimodal executions, specifically when the chosen subpopulation dimension is minimum (Chen et al., 2014).

Another famous niche technique is the sharing execution, which minimizes the fitness of the maximum alike individuals within the population depending on the mentioned niche radius (Goldberg & Jon, 1987). A practical risk of the sharing function technique is selecting an sufficient radius attribute, which is usually problem-dependent and requires that the numbers and shapes of peaks must be known. An identical condition applies to the choosing of certain parameter attributes in different niche algorithms, e.g., the choosing of clearing radius(distance) in the clearing technique and the choosing of species distance in the species conservation technique (Li et al., 2002).

But, in all most all industrial applications, a minimum beforehand wisdom about the fitness landscape is available during parameter setting. Hence, some crowding techniques employ a parameter-free paradigm in algorithm design.

Algorithm 1: Deterministic Crowding Algorithm

Step 1: Randomly select two different parents, p1 and p2.
Step 2: Apply crossover, mutation and other operations togenerate two offspring, c1 and c2.
Step 3: If$(|p1, c1| + |p2, c2|) \leq (|p1, c2| + |p2, c1|)$
Then
If Fitness(c1) is better than Fitness(p1) Then
Replace p1 with c1
If Fitness(c2) is better than Fitness(p2) Then
replace p2 with c2
Else
If Fitness(c2) is better than Fitness(p1) Then
Replace p1 with c2
If Fitness(c1) is better than Fitness(p2) Then
Replace p2 with c1
End.

In Algorithm 1, the DC method use a specific alternate functioner to cluster the nearby offspring and parent, and a parent could be alternated merely by improved offspring that are clustered jointly. though, the DC reduces the selected error, inherited move harms could still avoid the algorithm starting efficiently recognized several loci.

Twin-Space Crowding

Normally, two input functions are absolutely executed in a variety of crowding techniques: the first function make sure that person with the most excellent strength in the populace are typically spends with the maximum continued existence probability; the second function is a capability competition amongst the populace those situated in the similar niche. though, owing to the inadequate information of the actual resolution gap, mistaken finding about niche locality can avoid great accomplishment of the two function in the crowding algorithms (Chen et al., 2014).

To get rid the risk of detecting niches, this research suggests a modern mechanism for dynamically tracing niche coverage by doing various competence tournaments to maximize the survival chances of high-fitness individuals and to erase unnecessary individuals within a pair of niches.The pseudo steps of Algorithm 2 are specified as follows.

Algorithm 2: Twin-space Crowding Algorithm

Definition:

$x \succ y$: x dominatesy (has better fitness compared to y).

Input:

(1) The whole parent population $P = \{P_1, \dots, P_m\}$withsize m.

(2) The whole offspring population$O = \{O_1, \dots, O_n\}$with size n.

(3)Dimension number of any population member. (ornumber of genes in any chromosome).

Output: Updated parent population P.

Initialization: Define a temporary variable i, where i←1.

[Phase One]

Step 1: Find$P' \in P$where $|P'O_i| = min_{j=1}^{m}\{|P_jO_i|\}$

Step 2: If$O_i \succ P'$ then$P' \leftarrow O_i$ and go to step 9.

[Phase Two]

Step 3: Compute the set

$$T = \{T'|T' \in P, (|T'P'| \leq |P'O_i|) \wedge O_i \succ T'\}$$

Step 4: If$T = \theta$ then go to step 9 (i.e. ignoring this offspring).

Step 5: Define a temporary member Q.

Let$Q_j = \frac{O_{i,j}+P_j'}{2}, 1 \leq j \leq d$.

Step 6: If $Q \succ P'$, then$P' \leftarrow Q$, and go to step 9.

Step 7: If $Q \succ O_i$, then go to step 9 (i.e. ignore this offspring).

[Phase Three]

Step 8: Randomly choose a member $\bar{T}$from T, and let$O_i \leftarrow \bar{T}$

Step 9: Let i=i+1. If i≤ n, then go to step 1 (next offspring).

Otherwise, stop the algorithm.

The suggested TC algorithm has three key phases. For every offspring member, e.g.,O_i, the first phase finds the closestparent, which is designated P'. If O_i dominates P', P' is straightly substituted by O_i. Specifically, the competence law availed in this phase replicates that availed in NC technique. However, the offspring that is misplaced in the first phase but has better fitness attributes than the other parents, have another chance to enter the later phases.

The second phase avails a new version of the Hill-Valley (HV) function. The HV function is initially executed to get a temporary member Q and to evaluate whether members O_i and P' are located at different loci (niches). The modified algorithm replaces P' with Q if Q is better than P' (since Q is also better than O_i in this situation). If Q is not better than P' and O_i, the third phase is done.

The third phase fixes the competence between O_i and the parents found in the circled region where the circle center is P' and the radius is$\left|P'O_i\right|$. In the region, one parent that is poorer than$O_i$is randomly selected and substitutede by O_i. This process is needed as, under this condition, the selected parent is not onlypoorer but also adjacent (more crowding) to P' in comparison with O_i. The third phase could be seen as a self-adaptive niche investigational technique, and the competence is held in a dynamically circled region. That is, the suggested algorithm dynamically chooses the crowding void area for the survival competence.

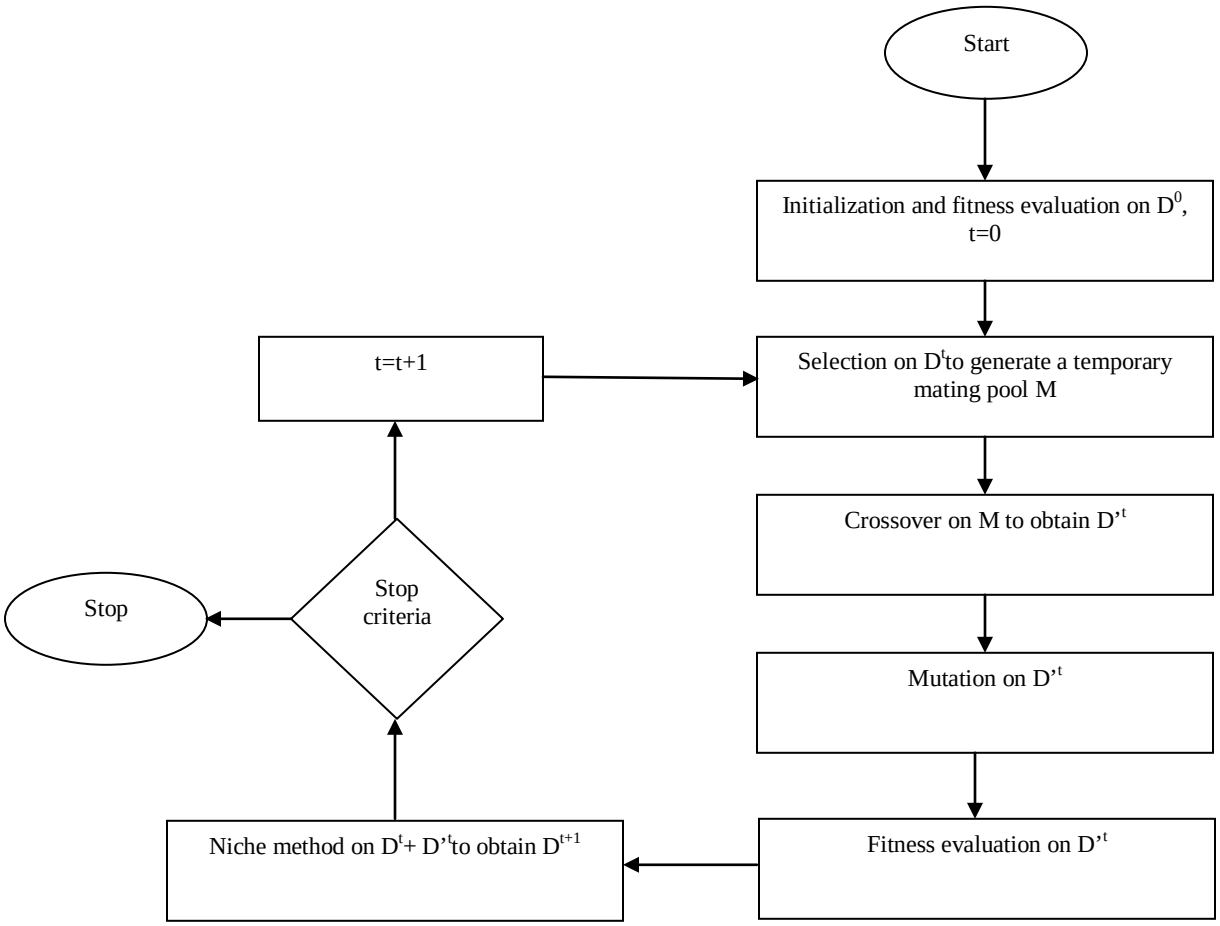

Figure 2: Flowchart of a General Niche-based Genetic Algorithm

Fig. 2 shows the classic genetic flow of a general niche-basedGA. That is, when incorporated in the GA, the proposed nichemethod (TC) is inserted into the flow as the replacement operatorto generate a new parent population before a new generation isexecuted.TCGA is implemented to solve the task scheduling problem. Tasks arrives into the system randomly then group them as batches. The batch of tasks generated is given as input to the TCGA. The TCGA schedules these tasks onto the available resources. TCGA generates new solution and test the solution by evaluating the fitness. Each solution is scored and either accepted or rejected before considering it for the next generation.

IV. EXPERIMENTAL RESULTS

This section provides the experimental evaluation of the proposed method. In order to verify our algorithm, metrics such as deadline, cost and makespan are used. Form the table and figure it is proved that our proposed method provides the better result. Table 1 shows the maximum number of tasks, computing capacity and maximum bandwidth capacity available on 4 data centers and Table 2 shows the comparison of GA and TCGA for the available resources using deadline, cost and deadline respectively. Figure 3 and 4 shows the deadline and cost comparison of the GA and TCGA.

Table 1: Available Resources in 4 Data Centers

Data Center #	*1*	*2*	*3*	*4*
Maximum Number of Tasks	128	256	512	1024
Computing Capacity	8	16	32	64
Available network bandwidth	8	4	10	15

Table 2: Comparison of GA and TCGA

Scheduling algorithm	*Resource matrix*	*Makespan*	*Cost (Rs)*	*Deadline (%)*
GA	128 x 8	1068912.31	1761254.67	71.59
TCGA		1035489.22	1486193.22	78.32
GA	256 x 16	1076312.58	1793516.11	76.25
TCGA		1046654.73	1593145.63	81.54
GA	512 x 32	1101245.45	1826548.15	78.93
TCGA		1086541.23	1615795.34	85.43
GA	1024 x 64	1125486.12	1856479.25	82.14
TCGA		1104863.65	1648751.35	88.71

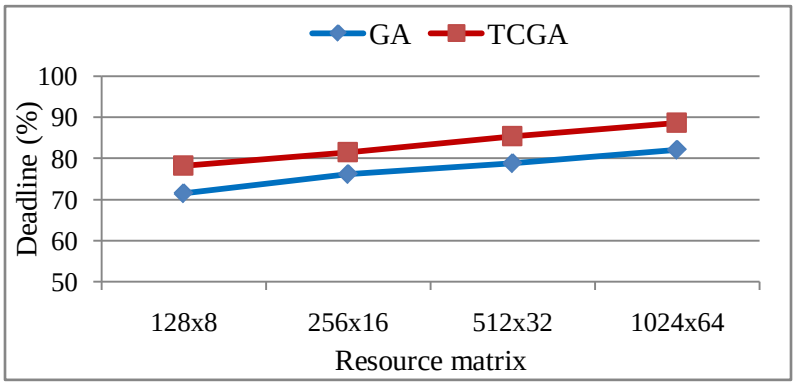

Figure 3: Deadline Comparison

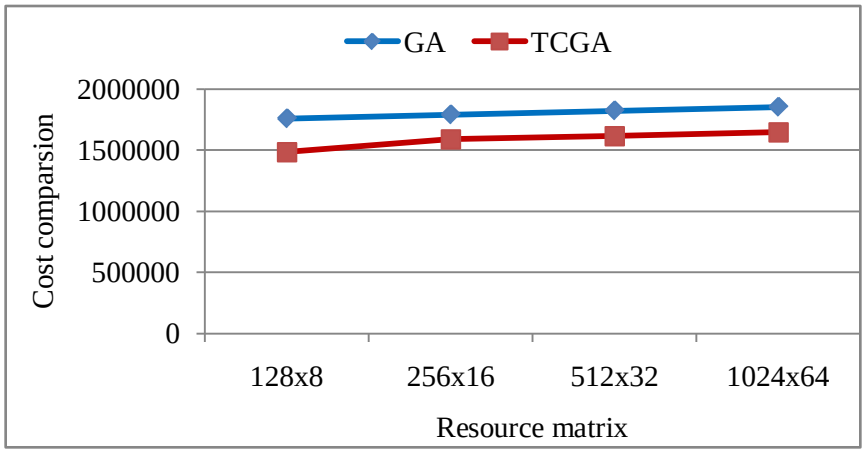

Figure 4: Cost Comparison

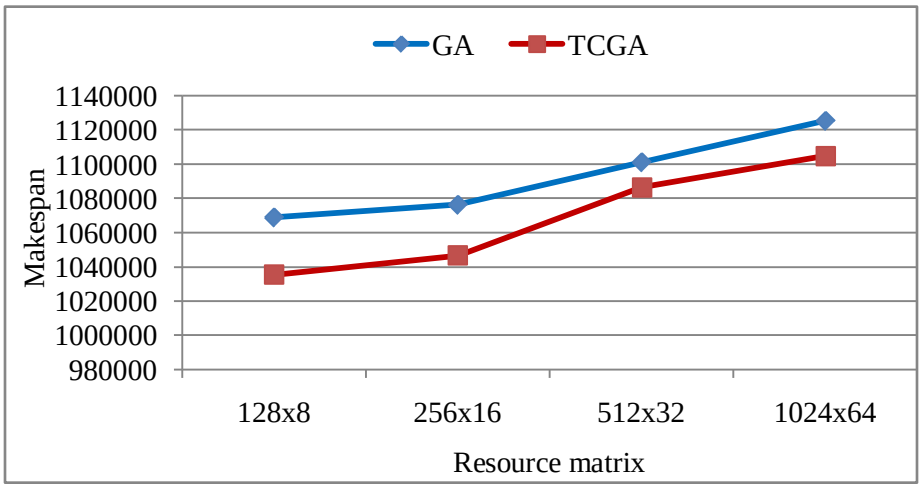

Figure 5: Makespan Comparison

Fig. 5 shows performances of TCGA for makespan. This result indicates TCGA is scalable than original genetic algorithm, which can indirectly extend its practicability. Hence the suggested TC method not required earlier information of the result hole to support the crowding device, use of the TCGA is identical to that of the simple GA. The suggested niche method (TC) is put into the surge as the substitute machinist to generate a novel close relative inhabitants before a novel production is implemented this makes the proposed method very efficient.

V. CONCLUSION

This effort urbanized a novel TC NC technique for Workflow Scheduling in Cloud Computing. The main idea for proposing this model is to improve the cost based on QoS requirement and grouping of resources and jobs over the cloud. Hence the suggested algorithm does not need earlier comprehension of the result liberty to help the crowding device, the suggested TCGA is fit for resolve workflow harms. The value and competence of the suggested TCGA were established using a limit, cost comparison and makespan metrics. The investigational outcome illustrate that the suggested TCGA acquires improved resolution.

REFERENCES

[1] S. Abrishami, M. Naghibzadeh and D.H. Epema, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds", Future Generation Computer Systems, Vol. 29, No. 1, Pp. 158-169, 2013.

[2] A. Bala, and I. Chana, "A survey of various workflow scheduling algorithms in cloud environment", 2nd National Conference on Information and Communication Technology (NCICT), Pp. 26-30, 2011.

[3] K. Bessai, S. Youcef, A. Oulamara, C. Godart and S. Nurcan, "Bi-criteria workflow tasks allocation and scheduling in Cloud computing environments", 5th International Conference on Cloud Computing (CLOUD), Pp. 638-645, 2012.

[4] L.F. Bittencourt and E.R.M. Madeira, "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds", Journal of Internet Services and Applications, Vol. 2, No. 3, Pp. 207-227, 2011.

[5] C.H. Chen, T.K. Liu and J.H. Chou, "A Novel Crowding Genetic Algorithm and Its Applications to Manufacturing Robots", IEEE Transactions on Industrial Informatics, Vol. 10, No. 3, Pp. 1705-1716, 2014.

[6] H.M. Fard, R. Prodan, J.J.D. Barrionuevo and T. Fahringer, "A multi-objective approach for workflow scheduling in heterogeneous environments", 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Pp. 300-309, 2012.

[7] D.E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization" Proceedings of the Second International Conference on Genetic Algorithms and their applications, Pp. 41-49, 1987.

[8] N. Kaur, T.S. Aulakh and R.S. Cheema, "Comparison of workflow scheduling algorithms in cloud computing", International Journal of Advanced Computer Science and Applications, Vol. 2, No.10, 2011.

[9] J.P. Li, M.E. Balazs, G.T. Parks and P.J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization", Evolutionary computation, Vol. 10, No. 3, Pp. 207-234, 2002.

[10] C. Lin and S. Lu, "Scheduling scientific workflows elastically for cloud computing", IEEE International Conference on Cloud Computing (CLOUD), Pp. 746-747, 2011.

[11] S.W. Mahfoud, "Niching methods for genetic algorithms", Urbana, 51(95001), Pp. 62-94, 1995.

[12] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows" Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, 2011.

[13] L. Qing, W. Gang, Y. Zaiyue and W. Qiuping, "Crowding clustering genetic algorithm for multimodal function optimization" Applied Soft Computing, Vol. 8, No. 1, Pp. 88-95, 2008.

[14] B.Y. Qu, J.J. Liang and P.N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization", Information Sciences, Vol. 197, Pp. 131-143, 2012.

[15] R. Singh and P.K. Petriya, "Workflow Scheduling in Cloud Computing", International Journal of Computer Applications, Vol. 61, No. 11, 2013.

[16] X. Wang, C.S. Yeo, R. Buyya and J. Su, "Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm", Future Generation Computer Systems, Vol. 27, No. 8, Pp. 1124-1134, 2011.

[17] S.J. Xue and W. Wu, "Scheduling workflow in cloud computing based on hybrid particle swarm algorithm", TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 10, No. 7, Pp. 1560-1566, 2012.

[18] F. Ye, W. Qi and J. Xiao, "Research of Niching Genetic Algorithms for Optimization in Electromagnetics", Procedia engineering, Vol. 16, Pp. 383-389, 2011.