I.J. Information Technology and Computer Science, **2015, 08, 37-43**
Published Online July 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.08.06

Cost Minimized PSO based Workflow Scheduling Plan for Cloud Computing

Amandeep Verma
University Institute of Engineering and Technology, Panjab University, Chandigarh, India
E-mail: amandeepverma@pu.ac.in

Sakshi Kaushal
University Institute of Engineering and Technology, Panjab University, Chandigarh, India
E-mail: sakshi@pu.ac.in

Abstract—Cloud computing is a collection of heterogeneous virtualized resources that can be accessed on-demand to service applications. Scheduling large and complex workflows becomes a challenging issue in cloud computing with a requirement that the execution time as well as cost incurred by using a set of heterogeneous cloud resources should be minimizes simultaneously. In this paper, we have extended our previously proposed Bi-Criteria Priority based Particle Swarm Optimization (BPSO) algorithm to schedule workflow tasks over the available cloud resources under given the deadline and budget constraints while considering the confirmed reservation of the resources. The extended heuristic is simulated and comparison is done with state-of-art algorithms. The simulation results show that extended BPSO algorithm also decreases the execution cost of schedule as compared to state-of-art algorithms under the same deadline and budget constraint while considering the exiting load of the resources too.

Index Terms—Workflow, Bi-Criteria Scheduling, Resource Reservation, HEFT, PSO, Priority

I. INTRODUCTION

Workflows constitute a common model for describing a wide range of scientific applications in distributed systems. Workflow scheduling is a process of mapping inter-dependent tasks on the available resources such that workflow application is able to complete its execution within the user's specified Quality of Service (QoS) constraints such as deadline and budget [1]. The grid workflow scheduling algorithms attempt to minimise the execution time without considering the cost of accessing resources. But, in case of cloud, different resources are available at different cost. Normally, faster resources are more expensive than the slower one. Therefore, workflow scheduling in cloud, requires both time and cost constraints to be satisfied as specified by the user [2]. A good heuristic tries to balance both these values and still obtain a near optimal solution [3].

In our previous work, we proposed Bi-Criteria Priority based Particle Swarm Optimization (BPSO) to schedule workflow tasks over the available cloud resources that minimized the execution cost and the execution time under given the deadline and budget constraints[4]. But while creating a schedule plan, we did not consider the existing load on the resources. So, the schedule plan created may conflict with the tasks already running on cloud resources. In this paper, to avoid the resource reservation conflict, we further extend BPSO algorithm to schedule workflow tasks over the available cloud resource while considering the confirmed reservation of resources. The extended BPSO algorithm is evaluated using simulation with five different real world workflow applications. The remaining paper is organized as follow: Section II presents the related work in the area of workflow scheduling. The problem description is presented in section III. The extended BPSO algorithm is evaluated and compared with state-of-art algorithms in section IV. Section V concludes the paper.

II. RELATED WORK

Scheduling of workflows is an NP – complete problem [5]. Many heuristic algorithms such as Minimum Completion Time, Sufferage, Min-min, and Max-min are used as candidates for best-effort based scheduling strategies [6]. Heterogeneous Earliest Finish Time (HEFT) [7] is a popular list based scheduling algorithms in which the priority is assigned to the workflow tasks and a task with higher priority is scheduled before a task with lower priority. But all of these heuristics just try to minimize the makespan without considering the monetary cost of executing the workflow tasks. So these methods are mainly suitable for Grid environment.

Only few works in the past considered bi-objective (time and cost mainly) criteria to schedule workflow tasks over grid and cloud resources. The Multi-Objective Evolutionary Algorithms (MOEAs) [8] are the effective way to solve multi-objective optimization problems like scheduling in grid. An MOEA approach produces Pareto optimal set of solutions, which is the set consisted of all non-dominated solutions. Cost and deadline constrained workflow scheduling in IaaS clouds was discussed in [9]. But, the resource model considered in the proposed algorithms consists of homogeneous resources. Zheng W. and Sakellariou R. [10] proposed two scheduling

heuristics LOSS and GAIN (based upon HEFT) that either tried to optimized time or cost, to meet the user's specified budget. So at a time, only one of the objectives *i.e.* either time or cost is optimized.

Now-a-days, meta-heuristic techniques such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization have been gaining popularity. This is due to the fact that these are easy to implement, have a faster convergence speed and give an approximate solution in much lesser time as compared to traditional methods [11-12]. In our previous work, we had proposed deadline and budget constrained heuristic based Genetic Algorithms [13-16] and bi-criteria priority based particle swarm optimization, (BPSO) [4] to schedule workflow tasks over the cloud resources without considering the existing load of the resources.

The major key issue with all these heuristics and meta-heuristic techniques is that none of them considered the existing load of resources and thus tend to schedule the tasks which may lead to reservation conflicts *i.e.* they may overlap with the tasks that are already executing on those resources. To the best of our knowledge, only Zheng W., and Sakellariou R. [16] proposed Budget and Deadline Constrained BHEFT which is the extension of HEFT that gives BDC plan to check whether a workflow request should be accepted or not while considering the existing load of the resources. So in this paper, we have further extended our proposed BPSO heuristic to create a schedule plan that tends to minimize the execution cost and time under the user's specified deadline and budget simultaneously while considering the reservation conflicts of the resources too.

III. Problem Description

A workflow application is modelled by a Directed Acyclic Graph (DAG), defined by a tuple $G\ (T, E)$, where T is the set of n tasks $\{t_1, t_2, ..., t_n\}$, and E is a set of e edges, represent the dependencies. Each $t_i\ \varepsilon\ T$, represents a task in the application and each edge $(t_i .... t_j)\ \varepsilon\ E$ represents a precedence constraint, such that the execution of $t_j\ \varepsilon\ T$ cannot be started before $t_i\ \varepsilon\ T$ finishes its execution [13]. A task with no parent is known as an *entry* task and a task with no children is known as *exit* task. The information associated with each task (t_i) are: the service type (y_i) that task wants to use and task size (z_i) in Million of Instructions *(MI)*.

There is a group of service types $S= \{S_0, S_1, ...\}$ and a set of heterogeneous resources that are fully interconnected. The different resources may have different processing power expressed as Million of Instruction per Second (MIPS).

It is assumed that a resource r_p is able to provide all the service types. For each service type S_x, a parameter β_x is given to depict its standard execution time, which is used to estimate the execution time of a task which uses this service type.

The execution time, $ET_{(i,p)}$, of a task t_i on a resource r_p is calculated using (1).

$$\boldsymbol{ET_{(i,p)}=(Z_i * \beta_x) / MIPS\ of\ r_p} \quad (1)$$

and the execution cost $EC_{(i,p)}$ is given by (2) .

$$\boldsymbol{EC_{(i,p)}=\mu_p * ET_{(i,p)}} \quad (2)$$

where μ_p is the price unit for resource r_p. Moreover, all resources are assumed to be in same physical region, so data storage cost and data transmission costs are assumed to be zero. Only, time to transmit data between two depend tasks *(ct)*, which are mapped to different resources is considered during experiment.

The BPSO algorithm [4] first assign the priority to all workflow tasks using the bottom level which is same as defined in HEFT[7] and is given by (3).

$$\boldsymbol{blevel(t_i) = w_i + \max_{t_j \varepsilon\ succ(t_i)} \{d_{ij} + blevel(t_j)\}} \quad (3)$$

where w_i is the average execution time of the task on the different computing machines. $succ(t_i)$ includes all the children tasks of t_i. d_{ij} is the data transmission time from a task t_i to t_j. If a node has no children, its b-level is equal to the average execution time of the task on the different computing machines.

In this paper, to avoid the resource reservation conflict, we further extend BPSO to schedule workflow tasks over the available cloud resource while considering the confirmed reservation of resources. For every resource, the exiting load is denoted by the set of pairs $L=\{(bt_0,et_0\ ),(bt_1,et_1),...(bt_n,\ et_n)...\}$, where bt denotes the beginning time of reservation slot and et denotes the end time of that reservation.

A. Workflow Scheduling Model

The different entities in our workflow scheduling model are: User, Scheduler and Resource Provider (RP).

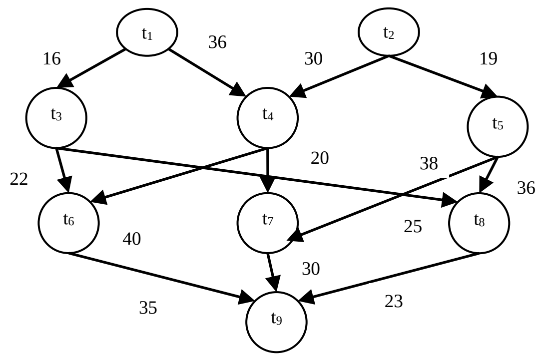
- The **Resource Provider** has a set of resources with different capabilities and provides particular services. The information related to set of available resources and list of services provided is publicly published. The RP responds to the queries from the scheduler about the availability of requested time slot on its resources.
- The **user** is requesting services from the resource provider to run a particular application along with budget and deadline constraint.
- The **scheduler** decides how to execute workflow tasks over available resources while considering the current reservation slots of resources.

The whole steps are summarized as follows:

1. A user submits a workflow application along with budget and deadline constraint to the scheduler.
2. On receiving such request, the scheduler tries to find out suitable allocation of workflow tasks over the resources provided by RP. For this purpose, the scheduler first sends a query *Free Slot Query (FSQ)[16]* to RP. The FSQ is of the form: $FSQ\ (t_i, r_p, dt_{(i,p)}, ET_{(i,p)}) = \{\ min(a,b)\ |\ (a,b) \cap\ L_p\ and\ a>=dt_{(i,p)}\ and\ b>= a+ET_{(i,p)}\}$, where t_i, is the task that we want to execute on resource r_p, $dt_{(i,p)}$, is the time at which whole data required to execute t_i, on r_p, is available and $ET_{(i,p)}$ is its execution time. $ET_{(i,p)}$ and L_p is the existing load set of r_p. For example, if $L_1 =\{(0,10),(12,14),(20,40)\}$ and for task t_2, its $ET_{(2,1)}=3$ and $dt_{(2,1)}=5$, then $FSQ(2,1,5,3)= (14,17)$.
3. As soon as, the RP sends a reply to FSQ, the scheduler makes a scheduling plan to execute workflow tasks. If

that plan is under user's specified constraints, then the execution of workflow starts otherwise the workflow execution request will be rejected.

The extended BPSO used this scheduling model is illustrated with an example shown in Fig. 1.



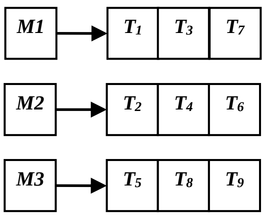
(a) A Sample DAG

Task	R_0	R_1	R_2	b-level	Order of execution according to b-level
T_1	22	42	10	140.07	1
T_2	26	34	12	137	2
T_3	28	40	14	104.53	4
T_4	24	34	11	101	5
T_5	30	40	14	104.6	3
T_6	22	38	10	62	8
T_7	26	44	12	64	7
T_8	30	50	20	67.2	6
T_9	24	36	14	24.67	9

(b)Estimated Execution Time and b-level of tasks

Resource	Price
R_0	0.40
R_1	0.29
R_2	0.92

(c)Price unit of Resources



(d) Schedule according to b-level

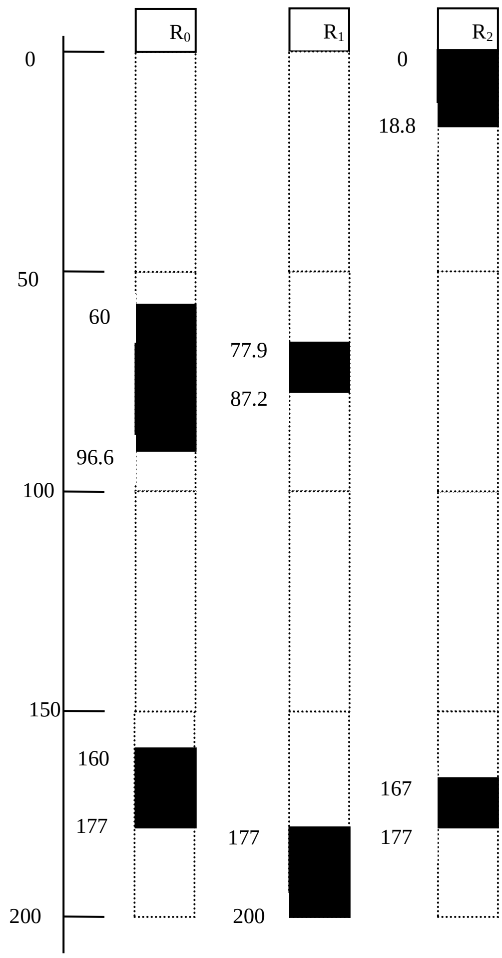
Fig. 1. An example of b-level

Fig. 1(a) shows the structure of DAG. Each edge is representing the amount of data to be transferred (in MB) between the dependent tasks. Fig. 1(b) shows the execution time of these tasks on three different available resources along with their b-level computed using (3). Then the tasks are sorted in descending order of their b-level. The tasks are sent to different machines according to their order of execution for completion of workflow application. The price for running a task on different resources is shown in Fig. 1(c) and the schedule generated according to blevel of a DAG is shown in Fig. 1(d). It is assumed the bandwidth between the resources is 20Mbps and all resources are able to provide all the required services. Assume a deadline of 200 time units and budget of 110 price units. Fig. 2(a) shows the existing loads on the resources represented by dark slots and Fig. 2(b) shows the assignment of different workflow tasks according to the b-level schedule as shown in Fig. 1(d). The total time taken by this schedule while considering the existing reservation of different resources is 151.2 seconds and cost is 99.4 price units, both are within user specified deadline and budget. This blevel schedule is then inserted into initialization of BPSO.

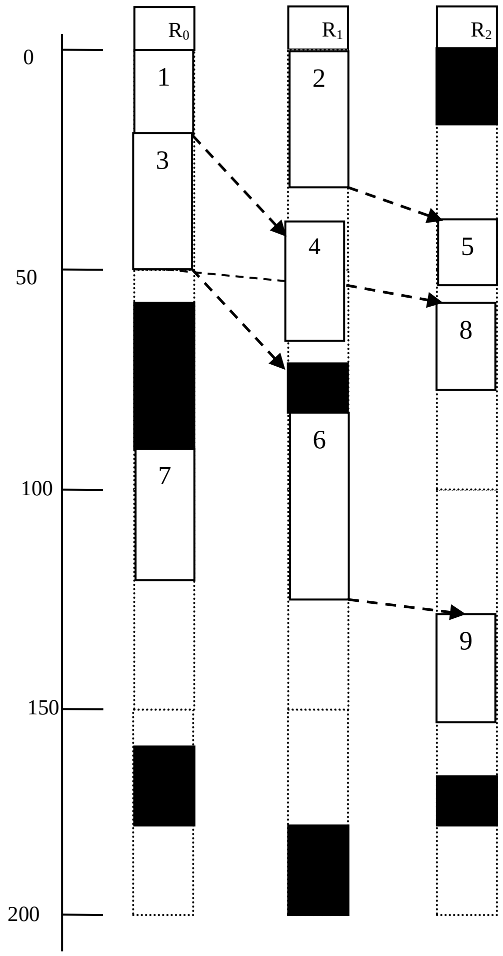
IV. Simulation and Analysis

To evaluate the workflow scheduling algorithm, we used five synthetic workflows based on realistic workflows from diverse scientific applications, which are:

- Montage: Astronomy
- Genome: Biology
- CyberShake: Earthquake
- LIGO: Gravitational physics
- SIPHT: Biology



(a)



(b)

Fig. 2. (a) Existing Load of Resources (b) Assignment of workflow task on different resources

The detailed characterization for each workflow including their structure and data and computational requirements can be found in [18].

A. Experiment Setup

The cloud used in our simulation is same as in [4] and is consisting of six resources with different processing speed and hence with different prices; similar to Amazon EC2 services [18]. There are four service types having a standard execution time of 1.0, 1.5, 2.5 and 3.0 respectively. For our experiment, we have extended the CloudSim[19] library. The processor speeds of different resources are selected randomly in the range of 1000-6000 MIPS. The power ratio, α_p of these resources is calculated using (1). The reasonable values for deadline D, and Budget B are generated as:

Deadline D= LB_D + k_1 * (UB_D -LB_D), where LB_D= M_{HEFT} (makespan of HEFT), UB_D= 5* M_{HEFT} and k_1 is a deadline ratio in range from 0 to 1.

Budget B= LC_B + k_2 * (UC_B –LC_B), where LC_B is the lowest cost obtained by mapping each task to the cheapest service and UC_B is the highest cost obtained conversely and k_2 is a budget ratio in range from 0 to 1.

The fitness function used in extended BPSO is as described in (4):

$$\boldsymbol{Fitness = \alpha * Time + (1-\alpha) * Cost} \tag{4}$$

where *Time* is the total execution time and *Cost* is the total execution cost of a generated workflow schedule and α is the cost-time balance factor in a range of [0,1] which represents the user preference for execution time and execution cost.

The existing load of resources is randomly generated for simulation. The procedure for generating existing load is same as given in [16]. The performance metric chosen for the comparison is Normalized Schedule Cost (NSC). The NSC of a schedule is calculated using (5)

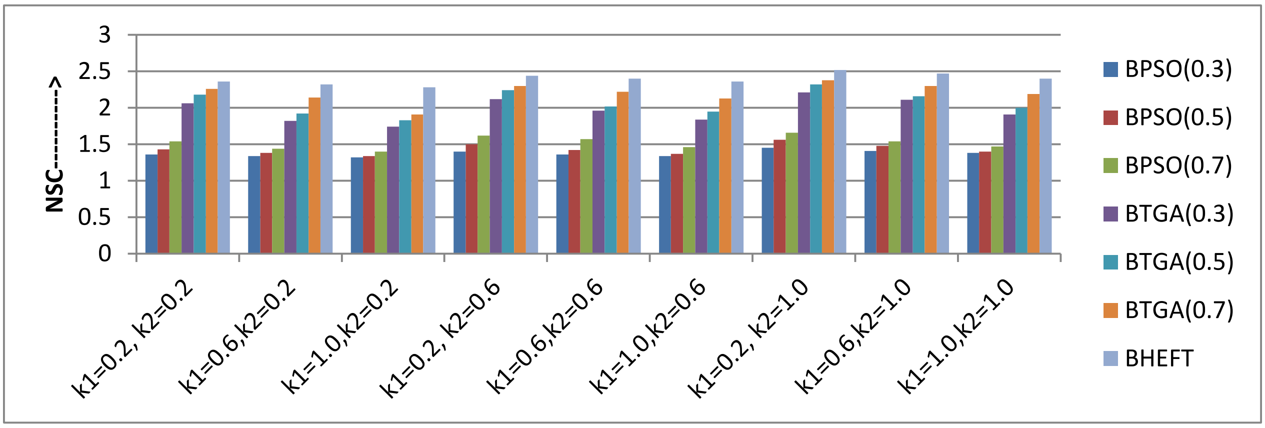
$$\boldsymbol{NSC = \frac{Total\ Cost}{C_c}} \tag{5}$$

where C_c is the execution cost of the same workflow by executing all the tasks on the fastest service, according their precedence constraints.

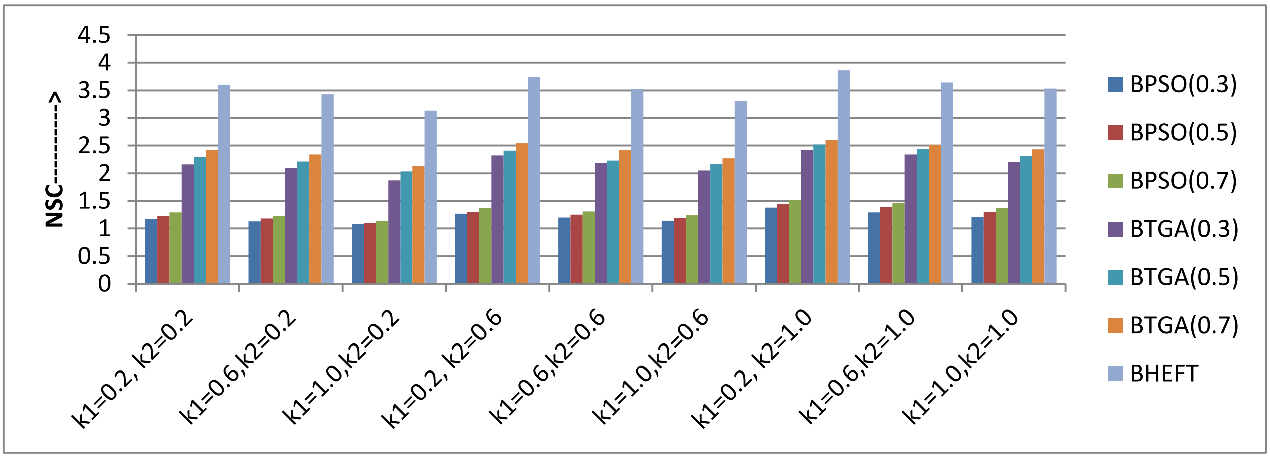
B. Experiment Results

To evaluate the performance of extended BPSO heuristic, we modified our previously proposed heuristic BTGA [14, 15]. In [14], BTGA heuristic is used to schedule workflow tasks over different available resources under the used specified deadline constrained using Genetic algorithm. Similarly, in [15], BTGA heuristic is used to schedule workflow tasks over different available resources under the used specified budget constrained using Genetic algorithm. In our current simulation, to modify BTGA heuristic, we have generated the schedule according to the fitness function as described in (5) while considering the existing load of the resources and BPSO heuristic is compared with modified BTGA heuristic and BHEFT heuristic[16] with respect to the monetary cost. As in case of BPSO, BTGA and BHEFT, the services are assigned randomly to different tasks of workflows and existing loads are also created randomly, so for all these algorithms, simulation is carried out for 50 times and average value of NSC is used for comparing the performance of BPSO, BTGA and BHEFT. In our experiments, we have chosen three different values of cost-time balance factor *i.e.* **α = (0.3, 0.5, and 0.7).** So while comparing, we are representing BPSO(0.3), BPSO (0.5) and BPSO(0.7), BTGA(0.3), BTGA(0.5) and BTGA(0.7), just to represent three different variants of BPSO and BTGA respectively corresponding to the values of α. Fig. 3 shows the average NSC of scheduling different workflows with BPSO, BTGA and BHEFT for three different values of k_1 (0.2, 06, and 1.0) and three different values of k_2 (0.2, 0.6, and 1.0), in total 9 combination. It shows that all variants of BPSO heuristic outperform the BHEFT heuristic significantly by reducing the execution cost of schedule under the same Deadline and Budget Constraint and using same pricing model in all cases. Even extended BPSO heuristic is generating the schedules which are cheaper than schedules created by BTGA heuristic. This is due to the fact that PSO has a faster convergence rate than GA. Also, it has fewer primitive mathematical operators than in GA (e.g. reproduction, crossover, mutation).

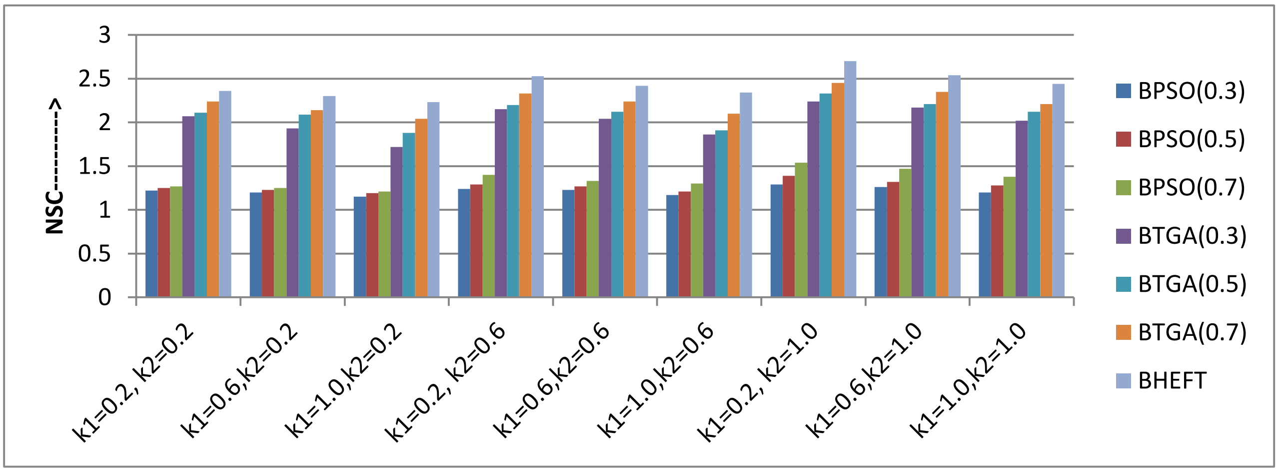
The variation in average NSC for three different values of α is due to the fact that when the user sets α = 0.3, then the user gives more preference to minimize the total execution cost as compared to minimize the total execution time (according to (5)), for complete workflow. As the value of α increase, the user preference for minimizing the execution cost is decreasing and the value of NSC also increases respectively. At a fixed budget ratio, *i.e.*, k_2, the deadline is relaxed by increasing the value of deadline ratio, *i.e.*, k_1 from 0.2 to 1.0. As a result, the scheduler is able to choose the cheaper services for assigning workflow tasks. The NSC of created schedule plan is reduced under the same budget ratio as shown in Fig. 3. Similarly, by fixing the deadline ratio, *i.e.*, k_1, and by varying the value of budget ratio, *i.e.*, k_2 =0.2, 0.6, and 1.0, the budget is relaxed for each of these values, respectively and the scheduler is able to choose the expensive services for assigning workflow tasks.



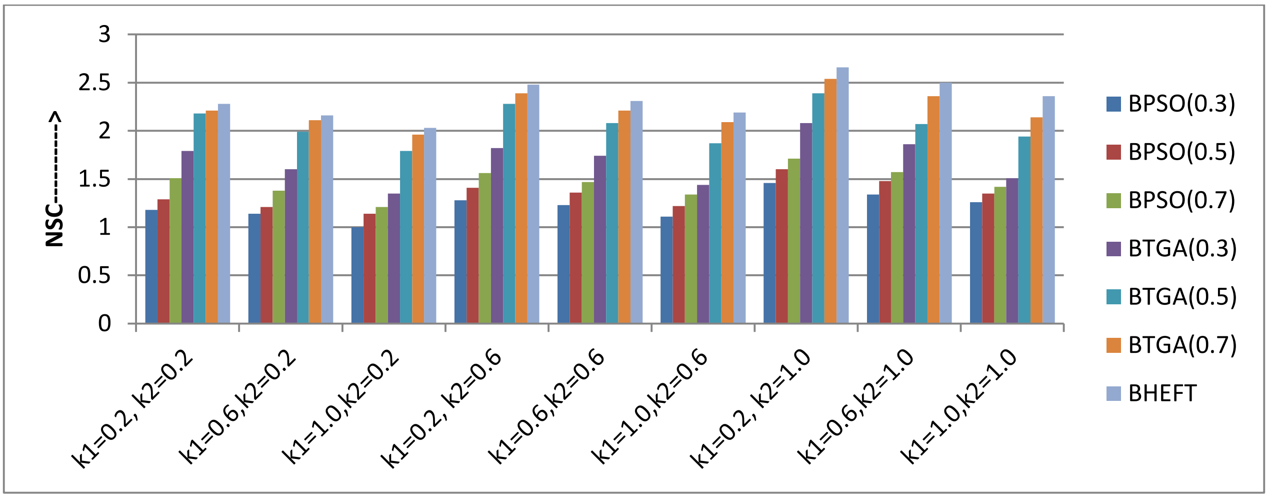
(a) Montage,25 nodes



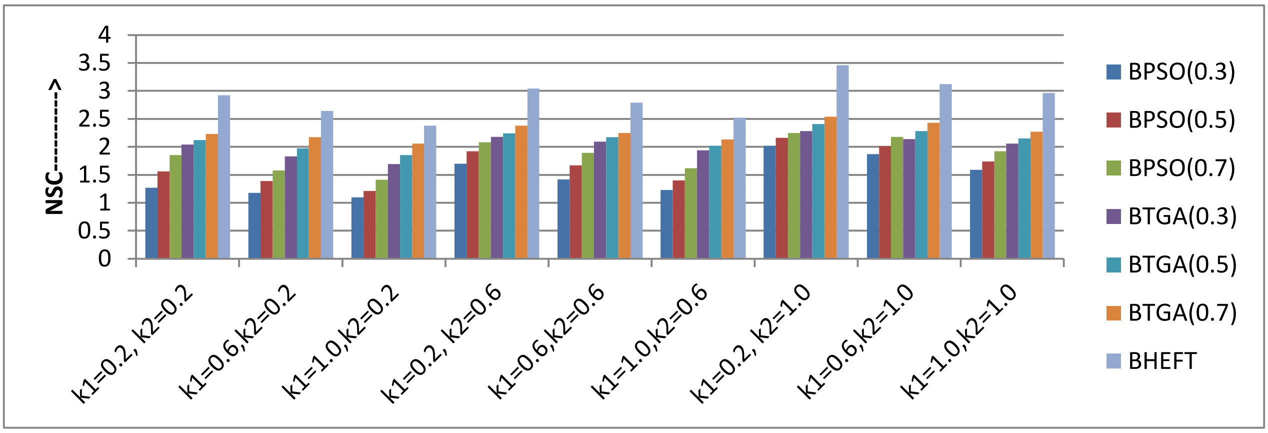
(b)Genome,24 nodes



(c)CyberShake,30 nodes



(d) SIPHT,29 nodes



(e)LIGO,30 nodes

Fig. 3. Average NSC of different Workflows

V. Conclusion and Future Work

In this paper, we have extended our previously proposed heuristic, namely, Bi-Objective Priority based Particle Swarm Optimization (BPSO) algorithm to schedule workflow applications to cloud resources that minimizes the execution cost while meeting the deadline and budget constraint for delivering the result. Each workflow's task is assigned priority using bottom level. These priorities are then used to initialize the PSO. The fitness of the generated schedule using extended BPSO is evaluated based upon the cost-time balance factor. The extended heuristic is evaluated with synthetic workflows that are based on realistic workflows with different structures and different sizes. The comparison of extended BPSO algorithm is done with BHEFT algorithm (with considering the existing load of resources) and BTGA algorithm under same deadline and budget constraint and pricing model. The simulation results show that the proposed algorithm has a promising performance as compared to BHEFT algorithm and BTGA algorithm by considering the reservation slots of the resources. In future, we intend to further improve our work by merging BPSO with other dynamic heuristics and then comparing with other existing multi-objective heuristics techniques in literature.

References

[1] I. Taylor, E. Deelman, D. Gannon, and M Shields., "Workflows for e-science: scientific workflows for grid", 1st Edition, Springer, 2007.

[2] S. Pandey, "Scheduling and management of data intensive application workflows in grid and cloud computing environment", PhD Thesis, University of Melbourne, Australia, 2010.

[3] L. Ke, J. Hai, C. X. L. Jinjun, Y. Dong, "A compromised time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on cloud computing platform", *Journal of High Performance Computing Applications* , vol.24, no.4, pp: 445–456, 2010.

[4] A. Verma, and S. Kaushal, "Bi-criteria priority based particle swram optimization workflow scheduling algorithm for cloud", *Proceeding of International Conference on Recent Advances in Engineering and Computational Sciences (RAECS)*,pp:1-6, March 2014 .

[5] J. Yu, and R. Buyya, "Workflow scheduling algorithms for grid computing", *In: Xhafa F, Abraham A (eds) Metaheuristics for scheduling in distributed computing*

environment, Springer, Berlin, 2008, ISBN: 978-3-540-69260-7.

[6] J. Yu, and R. Buyya, "Taxonomy of workflow management systems for grid computing", *Journal of Grid Computing*, vol.3, no.1-2, pp: 171–200, 2005.

[7] T. Haluk, H. Salim, and M.Y. Wu, "Performance-effective and low-complexity task scheduling for heterogenous computing", *IEEE Transaction on Parallel and Distributed Systems*, vol. 13, no. 3, pp: 260-274, 2002.

[8] Z. Aimin, Q. Bo-Yang, L.C. Hui, Z. Shi Zheng, N.S. Ponnuthurai, Qingfu Z., "Multiobjective evolutionary algorithms: A survey of the state of the art", *Journal of Swarm and Evolutionary Computation*, vol. 1, no.1, pp:32–49, 2011.

[9] M. Malawki, G. Juve, E. Deelman, J. Nabrzyski, " Cost and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds", *in proceeding of IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, UT, pp:1-11, 2012.

[10] R. Sakellariou, H. Zhao, E. Tsiakkouri, M.D. Dikaiakos, "Scheduling workfloes with budget constraint", *In: Gorlatch S, Danelutto M.(eds.) Integrated Research in GRID Computing*,pp:189-202, Springer, 2007.

[11] S. Pandey, W. Linlin, M.G. Siddeswara, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments", In. *Proceeding International Conference on Advanced Information Networking and Applications* Perth, WA, pp: 400-407, April 20-23, 2010

[12] W. Zhangjun, L. Xiao, N. Zhiwei, Y. Dong, and Y. Yun, "A market-oriented hierarchical scheduling strategy in cloud workflow systems", *Journal of Supercomputing*, vol.63, no.1, pp: 256-293, Jan 2013.

[13] A. Verma, and S. Kaushal, "Deadline and budget distribution based cost-time optimization workflow scheduling algorithm for cloud", *In. IJCA Proceeding of International Conference on Recent Advances and Future Trends in IT*, Patiala, India, pp: 1-4, 2012.

[14] A. Verma, and |S. Verma, "Deadline constraint heuristic based genetic algorithm for workflow scheduling in cloud", *Journal of Grid and Utility Computing*, vol. 5, no. 2, pp:96-106, 2014.

[15] A. Verma, and S. Kaushal, "Budget constraint priority based genetic algorithm for workflow scheduling in cloud", *In. Proceeding of IET International Conference on Recent Trends in Information, Telecommunication and Computing*, India pp: 8-14, 2013.

[16] S. Kaur and A. Verma, "An efficient approach to genetic algorithm for task scheduling in cloud computing environment",*IJITCS*,vol.4, no. 10, Sept. 2012.

[17] W. Zheng, and R. Sakellariou, "Budget-deadline constrained workflow planning for admission control", *Journal of Grid Computing*, vol. 11, no. 4,pp: 633-651, December 2013.

[18] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.H. Su, K. Vahi, "Characterization of scientific workflows", *In. Workshop on Workflows in Support of Large Scale Science*, CA, USA, pp:1-10, 2008.

[19] Amazon Elastic Compute Cloud (Amazon EC2) Online Available: http://aws.amzon.com/ec2/.

[20] N. C. Rodrigo, R. Ranjan, B. Anton, A.F.D.R. Cesar, R. Buyya, "**C**loudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Journal of Software: Practice and Experience (SPE)*, vol. 41, no. 1, pp: 23-50, 2011.

Authors' Profiles

Amandeep Verma is an Assistant Professor (IT) in UIET, Panjab University, Chandigarh, India. She is currently pursuing her PhD in workflow scheduling in cloud computing.. Her area of interest includes parallel and distributed computing, computer networks and cloud computing.

Sakshi Kaushal is currently an Associate Professor at Department of Computer Science and Engineering in UIET, Panjab University, Chandigarh, India. She received her PhD in Computer Science and Engineering from Thapar University, Patiala, India in 2009. Her research interests include computer networks and cloud computing. She has published various papers in the field of mobility in wireless networks, analysis and implementation of mechanisms to optimise network performance in high speed networks and cloud environment.

How to cite this paper: Amandeep Verma, Sakshi Kaushal,"Cost Minimized PSO based Workflow Scheduling Plan for Cloud Computing", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.8, pp.37-43, 2015. DOI: 10.5815/ijitcs.2015.08.06