IJCTA, 9(27), 2016, pp. 43-50 ISSN: 0974-5572

Differential Evolution Algorithm for Workflow Scheduling (DEWS) in Public Cloud

D. I. George Amalarethinam* **and T. Lucia Agnes Beena****

ABSTRACT

Infrastructure as a Service (IaaS) is one of the common Cloud service models, which is most used by the scientific applications. As the users are charged only for the usage of resources based on the Service Level Agreements (SLA), the users are attracted towards the IaaS. Workflow scheduling is a complex issue in IaaS because multiple scheduling parameters are to be considered to satisfy the Quality of Service parameters. Workflow applications comprises of various sub-tasks, which are to be executed in a particular method. These tasks have parent child relationship. The parent task needs to be executed before its child task. Workflow scheduling algorithms are supposed to preserve dependency constraints implied by their nature and structure. Resources are allocated to various sub-tasks of the original task by keeping into account these constraints. The role of workflow scheduling algorithm is to find the schedule which satisfies the SLA document which is written between a cloud user and a cloud service provider. Many heuristic algorithms were proposed in the literature, targeted only a single parameter for scheduling. But the user may require multiple objectives to be satisfied such as cost optimization, makespan optimization, reliability, deadline constrained, budget constrained etc. Hence, it is the responsibility of the Scheduling algorithm to find the optimal schedule that satisfies the SLA. The proposed algorithm uses Differential Evolution technique to optimize the scheduling parameters such as execution time of the application and Cost of executing the application in the Cloud. The proposed algorithm is compared with the Genetic Algorithm and the results outperform the Genetic Algorithm.

Keywords: Multi-objective optimization, Differential Evolution Algorithm, Mask mutation, Recombination, Workflow scheduling.

1. INTRODUCTION

In this Information era, large volume of data are transferred and processed over Internet. Scientific applications need large amount of resources to execute their simulations. The suitable platform to handle this ever-increasing data and analysis of the scientific applications is the Cloud Computing. Public Cloud uses the pricing models based on utility computing with pay-as-you-go principle. The Cloud's extraordinary features such as Scalability, Flexibility and Cost efficiency grant fine solution for the Scientific Workflows.

One of the important issues in Scientific Workflows is scheduling. Good scheduling algorithms need to produce optimal results according to objective functions in a very short time without consuming too many resources. The challenge involved in Workflow Scheduling is Quality of Service (QoS). QoS involves various parameters such as Budget, Deadline, Reliability, Availability, Minimizing the makespan, Supporting Service Level Agreement, Security and Load Balancing [1]. Among these parameters, minimizing the makespan and Cost of executing the workflow forms the important concern of the workflow scheduling. Executing the workflow with economic cost and minimum makespan in the IaaS, is a multi-objective problem. In Multi-Objective Optimization (MOP), there is no single optimal solution with respect to all objectives, but there is a set of tradeoff solutions known as Pareto front [2]. The main benefit of MOP is that the user can choose the solution which suits their requirement.

* Dean of Science & Director – MCA, Jamal Mohamed College, Trichy, *Email: di_george@ymail.com*

** Head, Dept. of Information Technology, St. Joseph's College, Trichy, *Email: jerbeena@gmail.com*

Evolutionary techniques are now widely used for tackling complex MOP problems. Differential Evolution (DE) is one such evolutionary technique which has gained a reputation of a very effective global optimizer. The aim of this paper is to optimize the makespan and economic cost of the workflow using DE. Since minimizing makespan and minimizing economic cost are contrasting objectives, DE is the best technique to get the optimal solution. Also DE technique satisfies the Cloud users by converting their choice on economic cost or makespan into the objective function. Thus the optimized result favors the user satisfaction.

2. MOTIVATION

Many heuristic and Meta-heuristic algorithms were proposed for the scheduling problems in the literature. The heuristic algorithm doesn't search the entire solution space to find the solution. Also, it fits only for a particular type of problem. In contrast to heuristic algorithms, Meta-heuristic methods find a near optimal solution by improving the initial solution based on the quality parameters. In Cloud Workflow Scheduling, quality parameters form an important role. The quality parameters are Deadline, Budget, Security, Availability, Reliability, Makespan and Cost [3]. Among these parameters Makespan and Cost need more attention in Cloud Scheduling. The parameters Makespan and Cost are inversely related. So the workflow scheduling can be down with multi-objective optimization techniques. The most common Multi-objective workflow scheduling technique blends the multiple objectives in a single function and optimizes that function. There are many scheduling algorithms proposed by the researchers using the hybrid approach. Combining the list scheduling algorithm with the meta-heuristic algorithm, they try to optimize the multi-objective workflows. A new Pareto-based list scheduling heuristic proposed by Juan et. al. [2] provides the user with a set of tradeoff optimal solutions. The user has to choose the one that better suits their requirements manually. Ajeena et. al. [4] proposed the bi-objective task scheduling algorithm used weighted sum approach for pareto-optimality and Particle swarm algorithm to solve the independent task scheduling. A task scheduling algorithm in cloud computing with the goal of the minimum completion time, maximum load balancing degree and the minimum energy consumption using improved differential evolution algorithm was proposed by Jing et. al.[5].

The optimization technique of DE has been used for solving the multi-objective parameters in Grid scheduling also. Jayasudha et. al. [6] improved the DE technique to solve the multi-objective parameters of makespan and flowtime in the Grid environment. Bessai et. al. [7] used three different approaches such as Aggregation approach, õ-approach and Pareto approach to solve the bi-criterion allocation and scheduling strategy. These approaches tried to optimize the workflow application completion time and cost incurred for the resource utilization. Udomkasemsub et. al. [8] proposed a scheduling framework for Cloud Data Analytics. In the scheduling plan Artificial Bee Colony method is applied. To solve the conflicting objectives, Pareto based technique is adopted. Leena et. al. [9] proposed a bi-objective optimization algorithm using NSGA II, to optimize the execution time and cost of scheduling in Hybrid Cloud. Hence, the Cloud Workflow Scheduling should be carefully coordinated and optimized in order to achieve the minimum Cost and minimum Makespan. This paper applies Differential Evolution Algorithm for the workflows to optimize the makespan and the cost.

3. THE PROPOSED WORK

Workflow scheduling focuses on the resource allocation and execution of dependent task. Hence, the workflow applications are modeled as a Directed Acyclic Graph (DAG) representing the different tasks and the interdependency among the tasks. In Cloud, to schedule the workflow applications, many parameters are to be optimized. To get an optimal schedule, satisfying different parameters Multi-Objective optimization is needed. This paper makes an attempt to provide an optimal schedule for workflow applications using Differential Evolution Algorithm named Differential Evolution Workflow Scheduling (DEWS). The advantages of Differential Evolution Algorithm (DEA) are its simple structure, ease of use, speed and robustness. The DE algorithm is a population based algorithm like genetic algorithm using the similar operators, crossover, mutation and selection. The algorithm uses mutation operation as a search mechanism and selection operation to direct the search toward the prospective regions in the search

Table 1
Steps of Differential Evolution Algorithm

Initialization
Evaluation
Repeat
    Mutation
    Recombination
    Evaluation
    Selection
Until *(termination criteria are met)*

space. By using the existing population members to construct a new population, the recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space. The main steps of the DEA [10] are given in Table 1.

3.1. Initialization

As an initial population, 20 schedules were produced by generating individuals (chromosomes) with the list –based heuristic algorithms such as HEFT [11], CFCSC [12] and LBTP [13]. The remaining individuals are generated randomly. The schedules are checked for the precedence constraints of the tasks. Based on the number of tasks in the input DAG, the resources needed for the DAG is decided using the equation 1.

$$r = \lfloor n \rfloor \tag{1}$$

where r is the number of resources needed for the input DAG and n represents the number of tasks in the DAG [14]. Consider a sample DAG with 10 tasks, namely, T0 through T9 and the resources to be used are represented as 0, 1 and 2. Hence the individual chromosome is represented as

P = (T0, 1) (T1, 0) (T3, 1) (T2, 2) (T4, 0) (T5, 1) (T6, 2) (T7, 0) (T8, 0) (T9, 1)

T0 represent the first sub task of the DAG and it is allotted to resource 1. To generalize, an individual is denoted as

$$P_i^G = \left(P_{i1}^G, P_{i2}^G, P_{i3}^G, \ldots P_{ij}^G \ldots P_{in}^G\right) \tag{2}$$

where G in equation 2 denotes current generation, i = 1, 2, …, p_s and p_s denotes population size. The P_{ij}^G (j = 1, 2, ..., n) includes the number of tasks. Once the initial population is generated, the fitness value of each individual is evaluated. Each of the individual undergoes mutation, recombination and selection. The initial values of the other parameters are given in Table 2.

Table 2
Experimental Setup

Initial population	20
Alpha	0.6
Crossover rate (CR)	0.6
Number of Generations (G)	100

3.2. Fitness Measure

A fitness function is used to measure the quality of the solutions according to the optimization objectives. The scheduling parameters for optimization of the schedule considered in the proposed algorithm DEWS are Makespan

(MS) and Monetary Cost (C). Considering the Makespan and the Monetary Cost, the fitness function for the proposed algorithm DEWS is given in equation 3.

$$f(x) = \text{Alpha} * (MS_{maximum} - MS_{current}) + (1 - \text{Alpha}) * (C_{maximum} - C_{current}) \quad (3)$$

where $MS_{maximum}$ is the highest makespan value in the current generation, $MS_{current}$ is the makespan of the current schedule, $C_{maximum}$ is the monetary cost of the schedule whose makespan is the highest in this generation and $C_{current}$ is the monetary cost of the current generation. Alpha is a cost-efficient factor that represents the user's preference for the makespan and the monetary cost. The value of Alpha ranges between 0 and 1. For the proposed algorithm the Alpha value is varied from 0.5 to 0.8 in steps of 0.1 and it is found that the 0.6 for Alpha optimizes schedule for the given DAG.

3.3. Mutation

There are different mutation techniques which are very popular in the literature [15]. One of them is given in equation 3.

$$Q_a^G = P_b^G + F\left(P_c^G - P_d^G\right) \quad (4)$$

where P_b^G, P_c^G and P_d^G are randomly selected from the population such that a, b, c and d belong to {1, 2, 3, …, p_s} and a # b # c # d. The mask mutation operator r is used in equation 4, since the mutation scaling factor F in DEA is not applicable for workflow scheduling problems [15]. Hence the equation 3 is modified as represented in equation 5.

$$Q_a^G = P_b^G \Theta F\left(P_c^G \Theta P_d^G\right) \quad (5)$$

where r is a mask mutation factor, which is from an integer set V corresponding to the number of resources. The V is randomly divided into the two sets V_1 and V_2, where $V_1 \cap V_2 = r$ and $V = V_1 U V_2$. To preserve the precedence constraint of task execution, the order of tasks remains unchanged in mask mutation. The algorithm for mask mutation is given in Table 3. For example, if a DAG with Ten tasks executed on Three Virtual Machines, the steps of mask mutation is illustrated below. Consider that the schedules P_1 and P_2 selected randomly in the place of P_c and P_d, then

P_1 = (T0, 1) (T1, 1) (T3, 2) (T6, 1) (T4, 1) (T2, 1) (T7, 1) (T5, 2) (T8, 2) (T9, 2)

P_2 = (T0, 1) (T1, 1) (T3, 1) (T6, 1) (T2, 2) (T4, 2) (T9, 2) (T7, 1) (T5, 1) (T8, 2)

Table 3
Algorithm for Mask Mutation [16]

```
Begin
    for x =1 to 10
        if (x^th element of P1) ∈ V1 then
            V1 and V2 ← x^th element of P1
        end
        if (x^th element of P2) ∈ V2 then
            V1 and V2 ← x^th element of P2
        end
    end
End
```

Consider that the set V comprising of three Virtual Machines (VMs), V = {0, 1, 2}. The set V is randomly divided into two sets V_1 and V_2, such that V_1 = {2} and V_2 = {0, 1} respectively. The mask mutation algorithm in Table 3 is applied to P_1 and P_2 which produce schedules Q_1 and Q_2 as given below.

Q_1 = (T0, 1) (T1, 1) (T3, 1) (T6, 1) (T4, 2) (T2, 2) (T7, 1) (T5, 1) (T8,2)(T9, 2)
Q_2 = (T0, 1) (T1, 1) (T3, 1) (T6, 1) (T2, 2) (T4, 2) (T9, 2) (T7, 1) (T5, 1) (T8, 2)

3.4. Recombination

Each mutant vector Q_i^G recombines with its respective parent P_i^G through crossover operation to produce its final offspring schedule R_i^G. The schedule R_i^G is produced based on the crossover rate CR which is between 0 and 1.

$$R_1 = \begin{cases} Q_1 \; if\, rand\,(0,\ 1) \Leftarrow CR \\ P_1 \; if\, rand\,(0,\ 1) > CR \end{cases} \tag{6}$$

For example, P_1 and Q_1 recombine to form R_1 following the equation 6. The resultant recombined schedule is given below.

P_1 = (T0, 1) (T1, 1) (T3, 2) (T6, 1) (T4, 1) (T2, 1) (T7, 1) (T5, 2) (T8, 2) (T9, 2)
Q_1 = (T0, 1) (T1, 1) (T3, 1) (T6, 1) (T4, 2) (T2, 2) (T7, 1) (T5, 1) (T8, 2) (T9, 2)

When CR = 0.6 and a random sequence of ten numbers within the range [0, 1] are generated as follows: **0.67**, 0.24, 0.35, 0.46, **0.78**, 0.54, 0.48, 0.03, 0.25 and 0.5. By applying the equation 5, the resultant R_1 is generated as below.

R_1 = **(T0, 1)** (T1, 1) (T3, 1) (T6, 1) **(T4, 1)** (T2, 2) (T7, 1) (T5, 1) (T8, 2) (T9, 2)

3.5. Selection

Each generated new individual is evaluated with the fitness function and based on the fitness value either the new individual R_1 or the Parent P_1 is selected for the next generation as given in the equation 7.

$$P_i^{G+1} = \begin{cases} R_i^G \text{If f } (R_i^G) < f(P_i^G) \\ R_i^G, \qquad \text{otherwife} \end{cases} \tag{7}$$

The fitness value for the individual P_i^G & R_i^G are 15.35 and 13.2 respectively. As R_i^G is less than P_i^G, R_i^G is assigned to P_i^{G+1}. This process is repeated till the termination criteria are met.

Table 4
Makespan (Sec.)

No. of Tasks	*No. of Resources*	*Algorithms*	
		WSGA	*DEWS*
10	3	56.2	46.28
20	4	61.26	45.66
50	7	126.8	110.19
100	10	148	134.27
150	12	153.68	151.28
200	14	176.95	170.55

4. RESULTS AND DISCUSSION

The proposed algorithm DEWS is developed in Java in the Netbeans IDE 7.1. The input for the WSGA is the arbitrary task graph generated by a program developed in Java [17]. This program generates the needed virtual

Table 5
Cost ($)

No. of Tasks	*No. of Resources*	*Algorithms*	
		WSGA	*DEWS*
10	3	3.8	2.74
20	4	3.9	3.39
50	7	16.32	7.6
100	10	25.33	22.3
150	12	46.14	40.12
200	14	75.25	51.9

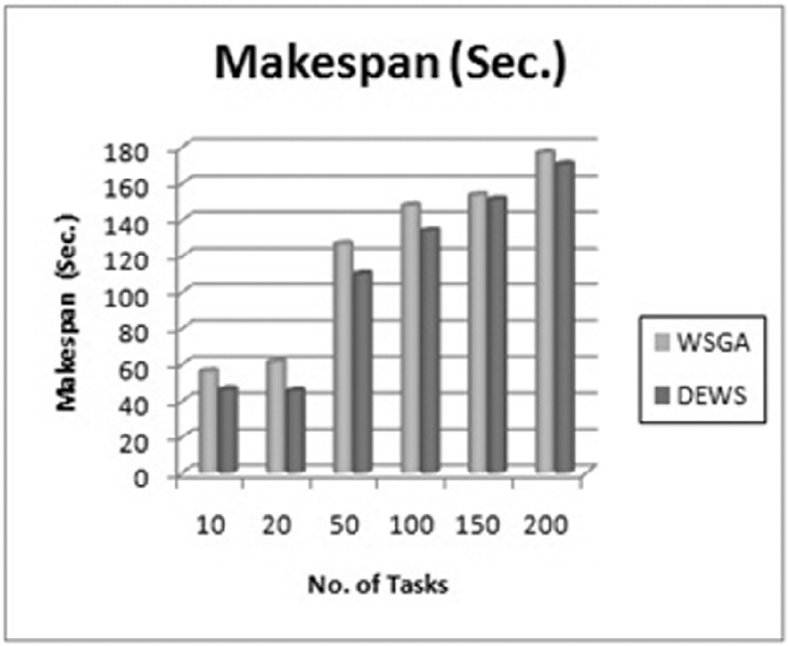

Figure 1: Graphical representation of Makespan

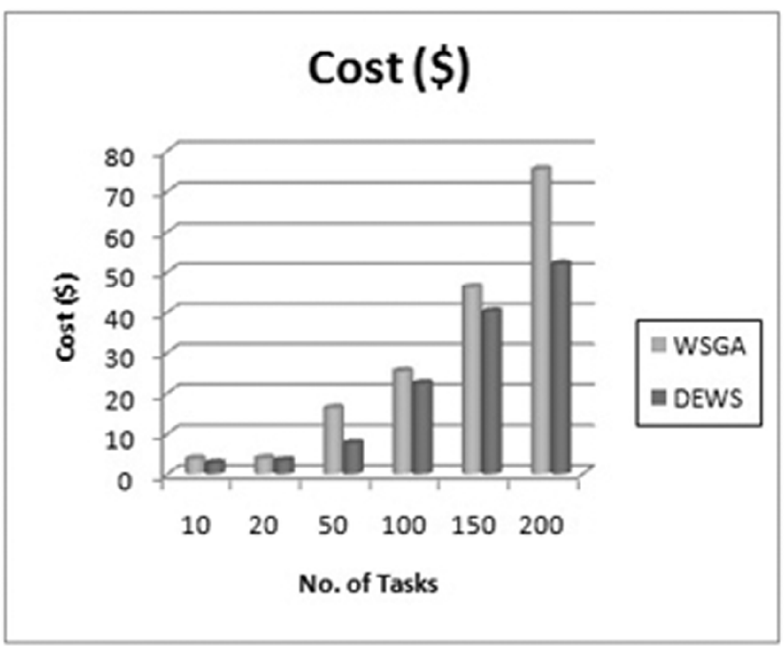

Figure 2. Graphical representation of Cost

machine instance with various speeds randomly. Given the number of tasks to be generated and the number of virtual machines, the program generates the arbitrary task graphs. The arbitrary task graphs are given as input to the heuristics algorithm to form some initial individuals and other needed individuals are spawned randomly. The virtual machine instance is charged based on the Google AppEngine scheme[18]. In Google AppEngine the virtual machine instance is charged per minute usage but the proposed algorithm charges for per second usage. The number of tasks in the arbitrary task graph is varied from 10 to 200, the scheduling parameters (makespan, monetary cost) are observed. By repeating the experiment from 100 generations to 2000 generations, it was found that the optimal schedule for the given arbitrary task graphs is achieved in the 100th generation. The results are tabulated in Table 4 and Table 5.

The scheduling parameters makespan and monetary cost are compared with the Genetic Algorithm WSGA [19]. The proposed algorithm DEWS outperforms the WSGA. The graphical representation of the result is revealed in Figure 1 and Figure 2.

5. CONCLUSION

Cloud computing is popular because of its pay-as-you-go model. Therefore the Cloud users concentrate on the Cost of using the resources and Cost becomes the vital parameter. The Workflow scheduling in Cloud has to focus on more than one scheduling parameter, in order to provide the optimal schedule for the workflow. This paper proposed a multi-objective algorithm for workflow scheduling using Differential Evolution technique. The proposed DEWS is tested with the arbitrary task graphs and compared with the Genetic algorithm.

The results gave optimal solution with two conflicting scheduling parameters the makespan and the cost, when compared with the Genetic algorithm. As a future effort, the performance of the algorithm DEWS has to be tested under the simulated environment CloudSim.

REFERENCES

[1] Mohammad Masdari, Sima ValiKardan, Zahra Shahi, Sonay Imani Azar, “Towards workflow scheduling in cloud computing: A comprehensive analysis”, *Journal of Network and Computer Applications*, **66**, 64–82, May 2016.

[2] Durillo, Juan J., and Radu Prodan. “Multi-objective workflow scheduling in Amazon EC2.” *Cluster Computing*, **17**, 169-189, 2014.

[3] Alkhanak, E.N., Lee, S.P. and Khan, S.U.R., Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems*, **50**, 3-21, 2015.

[4] Beegom, AS Ajeena, and M. S. Rajasree. “A particle swarm optimization based pareto optimal task scheduling in cloud computing.” *In Advances in Swarm Intelligence, Springer International Publishing*, 79-86, 2014.

[5] Xue, Jing, Liutao Li, SaiSai Zhao, and Litao Jiao. “A Study of Task Scheduling Based on Differential Evolution Algorithm in Cloud Computing.” *In Computational Intelligence and Communication Networks CICN*, IEEE, 637-640, 2014.

[6] AR Jayasudha, and T. Purusothaman. “Grid Scheduling using Differential Evolution [DE] for solving multi-objective optimization parameters”, *IJCSE International Journal on Computer Science and Engineering*, **2**, 2322-2327, 2010.

[7] Bessai, Kahina, Samir Youcef, Ammar Oulamara, Claude Godart, and Selmin Nurcan. “Bi-criteria workflow tasks allocation and scheduling in cloud computing environments.” *In Cloud Computing CLOUD, 2012 IEEE 5th International Conference*, 638-645, 2012.

[8] Udomkasemsub, Orachun, Li Xiaorong, and Tiranee Achalakul. “A multiple-objective workflow scheduling framework for cloud data analytics.” *In Computer Science and Software Engineering JCSSE, 2012 International Joint Conference*, 391-398, 2012.

[9] Leena, V.A. and Rajasree, M.S., “Genetic Algorithm Based Bi-Objective Task Scheduling in Hybrid Cloud Platform”. *International Journal of Computer Theory and Engineering*, **8**, 2016.

[10] KARABOĐA, DERVÝÞ, and Selçuk Ökdem. “A simple and global optimization algorithm for engineering problems: differential evolution algorithm.” *Turkish Journal of Electrical Engineering & Computer Sciences*, **12**, 53-60, 2004.

[11] Selvarani, S., and G. SudhaSadhasivam. Improved cost-based algorithm for task scheduling in cloud computing. *In Computational Intelligence and Computing Research ICCIC, 2010 IEEE International Conference,* 1-5, 2010.

[12] George Amalarethinam D. I., Lucia Agnes Beena T. Customer Facilitated Cost-based Scheduling algorithm [CFCSC] in Cloud. *Procedia Computer Science. Elsevier Publications* **46C**, 660-667, April 2015.

[13] George Amalarethinam D. I., Lucia Agnes Beena T. Level Based Task Prioritization Scheduling for Small Workflows in Cloud Environment, *Indian Journal of Science and Technology*, **8**, 1–7, 2015.

[14] George Amalarethinam D. I., Lucia Agnes Beena T. Scheduling Framework for Regular Scientific Workflows in Cloud, *International Journal of Applied Engineering Research*, **10(82)**, 267 – 271, 2015.

[15] Abazar Shamekhi, An Improved Differential Evolution Optimization Algorithm, *International Journal of Research and Reviews in Applied Sciences*, **15**, 2013.

[16] Tsai, Jinn-Tsong, Jia-Cen Fang, and Jyh-Horng Chou. “Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm.” *Computers & Operations Research*, **40**, 3045-3055, 2013.

[17] George Amalarethinam D.I, Joyce Mary G. J. DAGEN – A tool to generate arbitrary Directed Acyclic Graphs used for Multiprocessor Scheduling. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, **2**, 782 – 787, 2011.

[18] Google AppEngine. Available from: URL: https://cloud.google.com/compute/pricing/04.03.2016

[19] George Amalarethinam D. I., Lucia Agnes Beena T., Workflow Scheduling for Public Cloud using Genetic Algorithm (WSGA), *IOSR Journal of Computer Engineering (IOSR-JCE)*, **18(3), V**, 23-27, May-Jun. 2016.