ISSN 2278-3091
Volume 8, No.6, November – December 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse93862019.pdf
https://doi.org/10.30534/ijatcse/2019/93862019

Performance Evaluation of Task Scheduling using Hybrid Meta-heuristic in Heterogeneous Cloud Environment

Faten A. Saif [1], MN Derahman [2], Ali A. Alwan [3], Rohaya Latip[4]
[1]University Putra Malaysia, Malaysia, f-saif500@hotmail.com
[2]University Putra Malaysia, Malaysia, mnoord@upm.edu.my
[3]International Islamic University Malaysia, Malaysia, aliamer@iium.edu.my
[4]University Putra Malaysia, Malaysia, rohayalt@upm.edu.my

ABSTRACT

Cloud computing is a ubiquitous platform that offers a wide range of online services to clients including but not limited to information and software over the Internet. It is an essential role of cloud computing to enable sharing of resources on-demand over the network including servers, applications, storage, services, and database to the end-users who are remotely connected to the network. Task scheduling is one of the significant function in the cloud computing environment which plays a vital role to sustain the performance of the system and improve its efficiency. Task scheduling is considered as an NP-complete problem in many contexts, however, the heterogeneity of resources in the cloud environment negatively influence on the job scheduling process. Furthermore, on the other side, the heuristic algorithms have satisfying performance but unable to achieve the desired level of the efficiency for optimizing the scheduling in a cloud environment. Thus, this paper aims at evaluating the effectiveness of the hybrid meta-heuristic that incorporate genetic algorithm along with DE algorithm (GA-DE) in terms of make-span. In addition, the paper also intends to enhance the performance of the task scheduling in the heterogeneous cloud environment exploiting the scientific workflows (Cybershake, Montage, and Epigenomics). The proposed algorithm (GA-DE) has been compared against three heuristic algorithms, namely: HEFT-Upward Rank, HEFT – Downward Rank, and HEFT – Level Rank. The simulation results prove that the proposed algorithm (GA-DE) outperforms the other existing algorithms in all cases in terms of make-span.

Key words: Cloud computing, GA-DE, hybrid meta-heuristic, task scheduling.

.

1. INTRODUCTION

Cloud computing is a ubiquitous technology that has rapidly increased in deploying services by small, medium and large-scale firms and individual through the Internet. Cloud technology enables on-demand network access to share resources remotely such as servers, applications, storage, services, and database to external customers who are distributed geographically and enable them to just pay for the service which they used. The essential role of cloud computing is to exploit the distributed resources that can be grouped in order to accomplish a high level of throughput. Moreover, cloud computing paradigm allows users to be able to take up with the large-scale computation challenges [1]-[3]. The popularity of Internet services such as the Amazon Web Services, Google App Engine, and Microsoft Azure has drawn a lot of attention to the cloud computing paradigm in the recent decade. The main theme of cloud computing is to offers access to remote computation resources in the form of Virtual Machines (VMs) [2]-[4]. This type of services, called Infrastructure as a Service (IaaS), are implemented in Amazon EC2 [5]. Software and platforms are also offered as services in the cloud and they are called Software as a Service (SaaS) and Platform as a Service (PaaS) respectively [2]-[3], [6]. Cloud computing gain its popularity due to its tremendous advantages for a wide range of corporations as it shift the capital expenditure to operational expenditure. Thus, this results into more focus on providing essential services to end user including but not limited to elastic scalability, system reliability, simple and pervasive access methods, system transparency, and pay-as-you-go mechanisms [2]-[3]. The interest in adopting cloud computing technology by many small and large-scale firms has been dramatically increases in the recent years. This is due to various reasons such as the rapid advancement in computer processors in a form of multi-core processors. Besides, the cost of system hardware has also been reduced significantly. Lastly, the increasing cost of energy required to operate them. In consequence, cloud computing has risen to be the top IT revolutionary technology in the world. One of the core function of the cloud computing is the scheduling which it mapping tasks to proper resources in polynomial time to maintain the desired quality of service (QoS) and maintain the highest level of satisfaction of the service-user.

Scheduling is considered as an NP-complete problem particularly for large and complex tasks. Thus, this type of problem requires an approximate solution with the existing of the constraints to optimize the objectives of the scheduling. These objectives include reduce the completion time, minimize the communication cost, maximize the throughput, decrease the energy consumption, fault tolerance, resources utilization, deadline, laxity, load balancing and tardiness.

Generally, there are two common classification algorithms for task scheduling in the heterogeneous resources in the cloud system [7], namely: heuristic algorithms and meta-heuristic algorithms. The heuristic algorithms provide solution which is almost considered as an optimal solution. This encompasses Minimum Completion Time (MCP), Heterogeneous Earliest Finish Time (HEFT), Graham algorithm, Critical Path On a Processor (CPOP). In contrast, the meta-heuristic algorithms which gain the popularity to solve NP-hard problem and provides good solution with less effort of computational that make it proper for large or complex task. Among the most remarkable meta-heuristic algorithms are Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Honeybee. However, the heterogeneous resources in the cloud environment have made a challenge for scheduling the tasks and it has become more complicated. Hence, both algorithms (heuristic and meta-heuristic) are inefficient to be used. This is due to the fact that both type of algorithms unable to produce a solution for large-scale or complex task in cloud environment that ensure the required level of efficiency when scheduling the tasks. This issue attracts the attention of the researchers in the cloud computing community. Several approaches have been proposed based on hybrid meta-heuristic-based algorithms to provide a maximum rate of efficiency and approximate optimal solution that enhances the performance of the system. This idea has become a recent trend attempting to incorporate two or more of the meta-heuristic algorithms aiming at utilizing the strength of these algorithms leading to provide a better optimal solution. However, the existing studies have not used the standard dataset to verify the effectiveness of the algorithm performance. Thus, this paper focuses on the issue of mapping tasks to cloud computing resources and attempt to examine how to scale the resources in a way to meet jobs requirements in heterogeneous cloud computing environment. Furthermore, the work also emphasizes on evaluating the effectiveness of hybrid meta-heuristic algorithms (genetic algorithm along with differential evolution (DE) algorithm) using the scientific workflows (Cybershake, Montage, and Epigenomics). The proposed technique exploit the operators of genetic algorithm and differential evolution method aiming at determining the best task scheduling strategy that leads to introduce the optimal solutions that guarantee the minimum make-span for tasks execution in the cloud environment. The detail steps of the proposed approach are further explained throughout the paper. Several experiments have been conducted to reflect the impact of the proposed algorithm on scheduling jobs in heterogeneous cloud computing environment.

The remainder of this paper is orderly as follows. Section 2 explain the related work that focuses on explaining the process of task scheduling in cloud. Furthermore, some relevant task scheduling algorithm in cloud environment have been presented and well-discussed. Also, we critically reviewed related work to this study concentrating on the strengthens and the weaknesses of each technique. In Section 3, the methodology of the proposed model is illustrated. Section 4 describes the experimental settings and the performance evaluation then followed by simulation results. Finally, the conclusion and the discussion of the future work have been reported in Section 5.

2. RELATED WORK

Cloud computing incorporate both parallel and distributed systems in one platform focuses on utilizing virtual machine rather than a physical machine to enables manipulate, configure and facilitates sharing computer resources to the end user over the Internet instead of using supercomputer machines that demand expensive maintenance cost [8]. In addition, cloud computing plays a vital role in mapping and allocate a numerous number of resources over the internet to the end-users. Allocating resources to the tasks of application needs to be scheduled that raises effectiveness of the performance [9]. This section explains the task scheduling process highlighting the most critical factors that influence the scheduling process. Besides, we also describe the impact of the scheduling operation in terms of execution time, make-span, communication delay and cost. Furthermore, this section also explains and examines the previous works relevant to task scheduling in cloud computing environment. The discussion concentrates on three types of task scheduling in cloud, namely: heuristic algorithms, meta-heuristic algorithms and hybrid algorithms. Further explanation on these algorithms are given in the following subsections.

2.1 Task scheduling

Scheduling is a process of allocating a variety of tasks on processors aiming at various objectives including increase the speed of execution, decrease the make-span, and reduce the communication delay and cost. The main goal of scheduling in cloud computing is the efficiency that optimizes the overall system performance and minimizes the overhead problems [10]. According to the work presented in [11], a scheduling process is to guide the processing or the execution of interdependent tasks (workflows) on distributed resources. It assign the proper resources to workflow tasks that complete the requested tasks from the users which select the most suitable resources for the task collaborate to improve the performance of the system. From the literature, we have observed that there are two main workflow scheduling models, namely: best-effort based and QoS based. The best-effort based scheduling focuses on reducing the make-span rate while the OoS based emphasizes on reducing the performance under the constraint. For instance, minimizing the execution time under budget constraint. Workflow is a set of related tasks that are connected together in a dependent fashion (i.e., task2 cannot starts execution before task1 is finished). This simplifies the complexity of execution of scientific applications that have to be deploying over heterogeneous systems that represented by a directed acyclic graph (DAG) [7]. The efficiency and the performance of the scheduling process can be measured using numerous metrics including make-span, execution time, tardiness, laxity, utilization, deadline and total cost [12].

2.2 Task scheduling algorithms

In the literature, we conclude that task scheduling under cloud computing have been conducted from different perspectives. There are a variety of jobs have been studied on the environments of cloud computing. Out of those studies, many projects have been done to implement the provisioning process under the cloud computing paradigm. Different task scheduling algorithms and various task types have been tested and evaluated on the cloud model by many researchers [2]-[3]. In this section, there are three common classification algorithms for workflow-scheduling strategies in the heterogeneous resources in the cloud system based on previous studies, namely: heuristic algorithms, meta-heuristic algorithms and hybrid algorithms [7]. Further details pertaining to these scheduling strategies are explained in the following subsections.

2.2.1 Heuristic algorithms

Heuristic-based algorithms attempt to provide almost an optimal solution. Among the typical heuristic-based algorithms are Minimum Completion Time (MCP), Heterogeneous Earliest Finish Time (HEFT), Graham algorithm, Critical Path On a Processor (CPOP) [7]. The study introduced by [13] has proposed a new heuristic-based algorithm for offloading of multi-site in mobile cloud computing. This work is considered as a first attempt discussing the problem of tasks scheduling in the context of mobile cloud computing. The proposed solution concentrates on reduce the energy consumption, computation cost and the make-span when proposing any task scheduling plan. The idea is to convert the multi-objective into single objective then scheduled the tasks by the heuristic algorithm. The result demonstrate the effectiveness of the algorithm by achieving a high quality solutions in a reasonable time. The work presented by [14] has suggested an approach called Improved HEFT intends to schedule the tasks as DAG which chunking the workload over the processor. The experimental result proves that the proposed modified algorithm is better than HEFT in optimizing the completion time in the task scheduling in the heterogeneous cloud computing environment.

2.2.2 Meta-Heuristic algorithms

This type of scheduling algorithms has gained its popularity due to its ability in solving the NP-hard problem and provides solution with less effort of computational that make it proper for large or complex task (NP-complete) problem. This type of algorithms includes Genetic algorithm (GA), Honeybee, Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) [7]. The work presented in [15] proposed a new algorithm which called Completion Time Driven Hyper-Heuristic (CTDHH). The CTDHH algorithm designed with the aim of optimizing the cost of completion time in scientific workflow scheduling in cloud computing environment. The algorithm utilizes four population based on meta-heuristic as a low-level heuristic algorithm (LLH). The algorithm has proven through the experimental results it usefulness in optimizing the cost of workflow scheduling in cloud computing paradigm. The work reported in [16] has proposed an improved genetic algorithm for task scheduling in a cloud environment called N-GA. The N-GA algorithm exploits the benefits of genetic algorithm along with heuristic methods. The algorithm evaluate its correctness and verified it behavioral based on the model checker NuSMV and process analysis Toolkit (PAT). The purpose of using PAT model is to be as a platform to convert the proposed algorithm to SMV code while the NuSMV model checkers is the official base that is suitable for the modeling of the concurrent and distributed systems. Even more, this model has implemented to tackle the gap between formal verification approaches and real applications. The authors have also involved several behavioral models to enable selecting the algorithm that produce the best performance. The simulation results demonstrates that the proposed algorithm outperforms the other existing meta-heuristics algorithms in many aspects. Lastly, the work introduced in [17] have proposed an enhanced Genetic algorithm to assign the static task scheduling to the processor in a heterogeneous computing environment. The proposed algorithm introduces a new operator that guarantee sample diversity and stable coverage for all space. The proposed strategy replaces the random initial population with optimized solutions to reduce the repetitions in GA. The results of the experiments has shown that the proposed solution achieved a significant enhancement in terms of performance and processing cost.

2.2.3 Hybrid strategy

The critical issue of task scheduling on cloud computing attracts the attention of the researchers to propose several hybrid meta-heuristic-based solutions to find an optimal solution that enhances the performance of the system. This strategy has become the recent trend that combines more than one of the meta-heuristic algorithm exploiting their strength to provide a better optimal solution. There are many attempts that have implemented to enhance the performance of the task scheduling in cloud computing environment. The work reported in [18] have proposed an approach which incorporate meta-heuristic and HEFT algorithms to schedule the workflow and is based on a new factor called cost time to make real bi-objective. The simulation result demonstrates that the proposed algorithm has achieved the optimal perform comparing to the other algorithms according to reducing the cost of the execution time and the makespan. The work presented in [19] proposed a new hybrid meta-heuristic-based algorithm to schedule the incoming tasks which are known as a bag of task (BOT) application in the interconnected cloud environment. The algorithm exploits the benefits of simulated annealing algorithm and tabu search meta-heuristic algorithm to reduce the scheduling cost and improve the performance of the scheduling process. The proposed algorithm has been compared with the Fastest Processor Largest Task based on the arrival and the running time. The results of the experiment have proven the effectiveness of the proposed solution in optimizing the cost and the performance of the scheduling operation. Furthermore, The work presented in [20] also discuss the issue of task scheduling in cloud computing environment. They have proposed a new approach aiming at provisioning the resources in multi-tier in cloud computing utilizing the meta-heuristic algorithm. The proposed strategy also involves the Particle Swarm Optimization along with simulated annealing and a hybrid algorithm that consists of (PSO) and (SA) which contribute to accelerating the resources provisioning in cloud computing. The work introduced in [21] have address the issue of scheduling in cloud computing context. They have proposed an algorithm that combines ant colony optimization and particle swarm optimization to tackle the issue of scheduling in Virtual Machines in cloud context. The algorithm utilizes the previous feedback to forecast the workload of new input requests to conform the dynamic environment without an extra information of the task. The algorithm ignores the unsatisfied request before scheduling to minimize the computing time. The result of the experiments indicates that the proposed algorithm can maintain the load balance in a dynamic environment. Last but not least, the work in [22] have proposed two algorithms based on hybrid meta-heuristic and Dynamic dispatch Queues (TSDQ). The hybrid meta-heuristic uses the Fuzzy Logic with Particle Swarm Optimization algorithm (TSDQ-FLPSO), while TSDQ involves the simulated annealing with Particle Swarm Optimization algorithm (TSDQ-SAPSO). The simulation result shows the effectiveness TSDQ-FLPSO to find the optimal result compared to the others algorithms in terms of waiting time, queue length, execution time, cost, resource utilization, and load balancing.

3. GENETIC ALGORITHM – DIFFERENTIAL EVOLUTION METHOD

The proposed approach presents in this paper has been inspired by the work presented in [23]. The idea of the proposed approach relies on combining the operators of Genetic Algorithm and Differential Evolution Method to identify the optimal solutions that minimize the make-span of the tasks execution and find the best task schedule in the cloud environments. The directed acyclic graph which is represented by G(V, E), where the V (Vertices) is a set of nodes in the graph and E (Edges) denotes the precedence relations between the tasks. The nodes are weighted by the value of its computation time and the edges are weighted with the cost of communications between two tasks. If two tasks are assigned on the same processor the communication cost is zero. The directed acyclic graph contains the tasks T_0, T_1, T_2, . . . , T_{10}, and T_0, is entry task and T_{10} is exit task [23]. Figure 1 illustrates the directed acyclic graph [23]. The proposed algorithm works as follows. First, the Genetic Algorithm (GA) is applied including their selection, crossover and mutation operations to produce a set of solutions. These derived solutions are considered as an initial population to Differential Evolution (DE) method. Then, the DE operations applied on these derived solutions to produce a solution as next population to GA. The process continues until the DE terminate the operation and the updated list of all solutions is retrieved. It must be sorted from left to right, but when the new solutions priority is violated after the production of children.

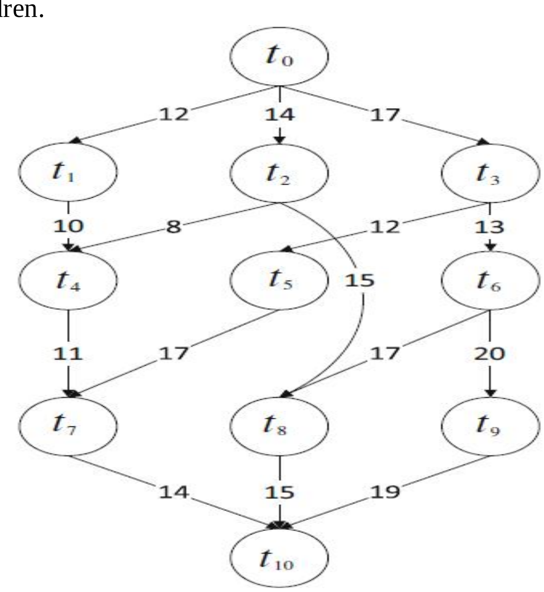

Figure 1: The Proposed Simple Directed Acyclic Graph

In the proposed approach the premature convergence is avoided by comparing the produced children with their parents, if the fitness value has achieved better value than the parent's value, then the parents are replaced by their children; otherwise, it is aborted. Figure 2 illustrates the flowchart of the GA-DE presented in this paper [23]. When the initial population is created, the evaluation of the fitness values must be done. The termination condition must be checked in every iteration of the algorithm. The optimal solutions will be produced unless the termination condition has achieved.

3.1. Production of initial population

The first step in the proposed approach is production of the initial population which contains individuals and chromosomes with fix sizes. This step is important to ensure having a better solution from the previous heuristic priority method. The priorities found from the three methods HEFT-Upward rank, downward rank and level rank for the DAG tasks are considered as the initial population. The remaining of the individuals are generated randomly as depicted in Table 1 [23]. In the chromosome, created the first and last places that linked with start and exit nodes, respectively. Creating the remaining tasks randomly and order the chosen tasks according to their priorities which are not breaking the constraints of the precedence.

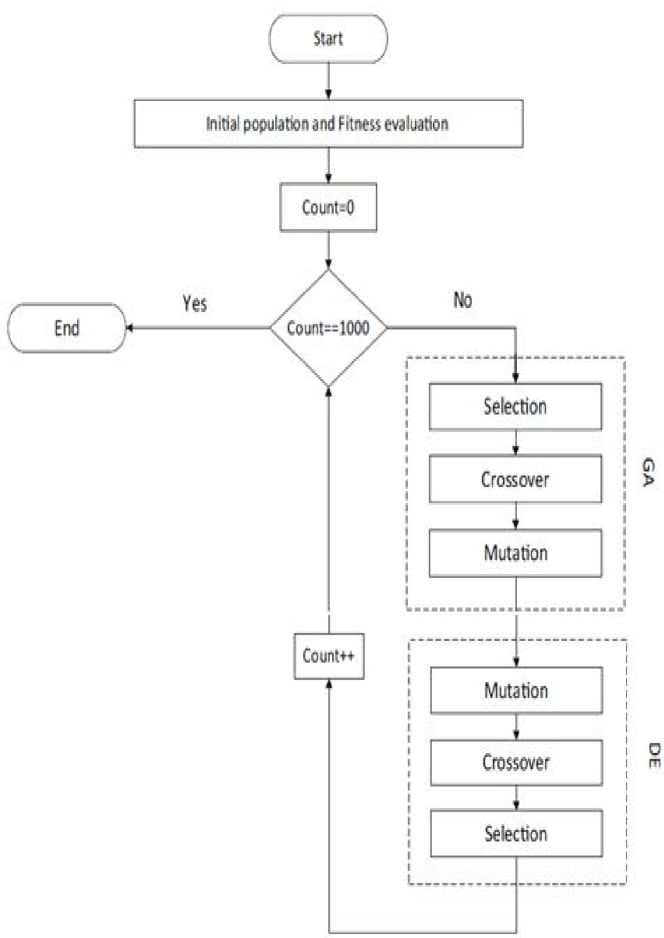

Figure 2: The Flowchart of The GA-DE Algorithm

Table 1: List of Tasks with Their Priorities

Tasks	$rank_u(t_i)$	$rank_d(t_i)$	Level	$rank_u(t_i) + rank_d(t_i)$
t_0	123	0	0	123
t_1	81	22	1	103
t_2	76	24	1	100
t_3	96	27	1	123
t_4	62	41	2	103
t_5	63	45	2	108
t_6	77	46	2	123
t_7	39	69	3	108
t_8	36	75	3	111
t_9	45	78	3	123
t_{10}	11	112	4	123

3.2. Make-span

The next step in the proposed approach for task scheduling in cloud computing is computing the make-span for all individuals. To do so, tasks have to be assigned and scheduled in a proper processor or a machine. In our work to calculate the make-span of each individual HEFT processor allocation method have been applied.

3.3. Fitness Function

The make-span is used as a fitness function of the proposed algorithm. The finish time of the exit task of the directed acyclic graph has been exploited as the make-span of the application.

$$\text{Fitness}\ (T_i) = \text{make-span}\ (T_i)$$

Where T_i represent the task i, and $i = 0, 1, \ldots, n$

3.4. Selection Operator

The roulette wheel selection function is used to select an individual for a good fitness values and the genetic operations are implemented on the selected individuals. The individual which has high fitness value gets more chances to be selected the genetic operations (crossover and mutation) according to their fitness.

3.5. Crossover Operator

The main purpose of this stage is exchanging some genes of one individual with the other genes to generate two valid offspring. The crossover operator utilizes a random single-point which can be created between land n. Then, the crossover is implemented if the genes of both parents from the entry node to a crossover point are not corresponding. The crossover single point which is equivalent to 5 produces the two new offspring. The children from the left side inherit genes of parents in the place of the same gene then chosen genes are deleted from the parent and the residual genes are imported to the child from left to right. Thus, the created children will be effective and their fitness is gained utilizing the fitness function. The fitness values of the children are comparing with parents, the fitness values of the children will be replaced as well as the values are better than parents. Figure 3 demonstrates the crossover operator process [23].

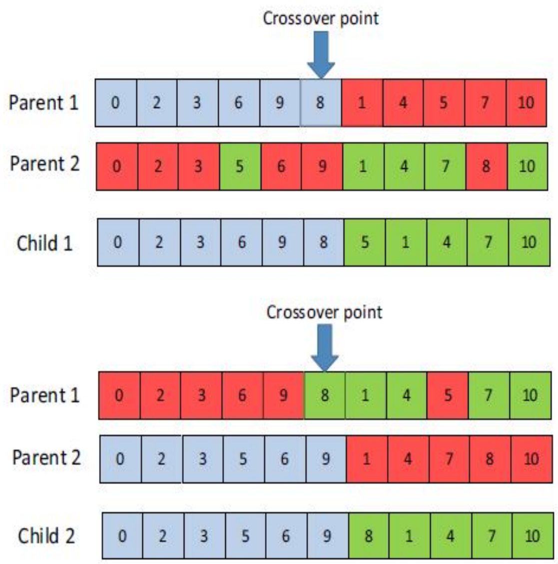

Figure 3: The Crossover Operator

3.6. Mutation Operator

This component is responsible to obtain a new chromosome by changing two genes in a way that precedence constraint is not violated. The process starts as follows. First, a gene is selected randomly and then the first successor of the selected task (tj) from the mutation point to the end is found. If there is m^{th} gene which it is a member of [i+1, j-1] and the predecessors of t_m are not in front of t_i, t_i and t_j can be swapped with each other. If in mutation function these conditions do not happen, then the mutation operator algorithm will be run from the beginning. Finally, the fitness value of a child is measured for the child and if the fitness value of the child better than the parent's fitness then the child will replace the parent. Figure 4 illustrates the detail process of the mutation operator which has been adopted from [23].

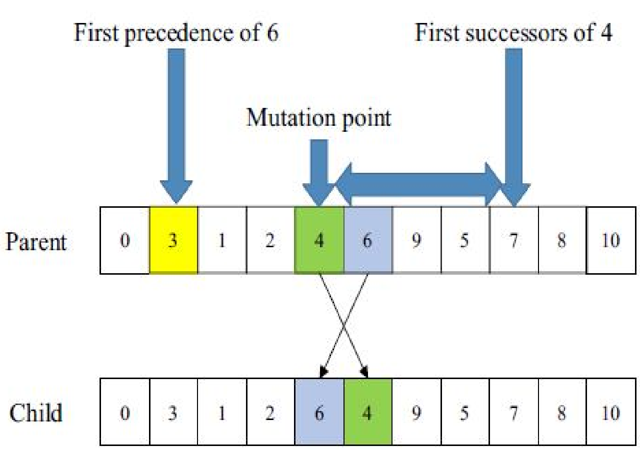

Figure 4: The Mutation Operator

3.7. Termination Conditions

The genetic and DE algorithms are both belong to the evolutionary algorithms category. It has been known that this type of algorithms can have infinite execution time and to adopting such algorithms a predefined termination condition need to be considered in order to stop the process of generating the solution. In this paper, some of the predefined policies have been proposed to ensure the termination of the algorithms after producing the most appropriate solution. Among the factors that have been considered to terminate the algorithms are the fitness evaluations, the running times of the system, and the population diversity. In our proposed approach, the algorithm is terminated after at least 1000 iterations.

4. EXPERMINTAL ENVIRONMENT

This section elaborates the experimental settings of the simulation study that have been carried out to fairly evaluate the performance of the algorithms considered in this work, namely: HEFT upward rank, HEFT downward rank, HEFT-Level rank), and the hybrid meta-heuristic algorithm (GA-DE). Besides, this section also explains and discusses the experiments' results that have been obtained from the simulation study. Further details pertaining the experimental settings and the performance evaluation are explained in the following subsections.

4.1. Experimental settings

The Cloudsim simulator has been used to implement and evaluate the task scheduling algorithms considered in this paper. All experiments have been carried out on Pentium Core i3 with 4GB memory and Windows 10 platform with one Data center and one Data center broker. The experimental comparison among GA-DE, HEFT upward rank, HEFT downward rank, and HEFT-Level rank has been done with statistical analysis according to the make-span. The details of the parameters which have used in the simulation are depicted in Table 2. The table describe the number of Cloudlets, the number of VMs, the VM MIPS, the number of datacenters, the bandwidth, the number of CPUs, the size of the real memory (RAM), the number of host(s), the storage capacity of the of the host machine, and the amount of the real memory of the host machine, and the bandwidth of the host machine respectively.

Table 2: The parameters setting of the simulation study

Parameter	Values
Cloudlets	30-50-100
Number of VMs	30
VM MIPS	1000
Number of datacenters	2
Bandwidth	1000 bps
Pes Number (Number of CPUs)	1
RAM	512 MB
Number of Host(s)	5
Host(s) Storage	1,000,000 MB
Host(s) RAM	10240 MB
Host(s) bw	100000

4.2. Experimental Results

The evaluation of the three heuristic algorithms (HEFT upward rank, HEFT downward rank, and HEFT-Level rank), and the hybrid meta-heuristic algorithm (GA-DE) with 5 hosts and 30 VMs. The generation of the DAGs was on various randomly on cloudsim with (CyberShake_30.xml, CyberShake_50.xml and CyberShake_100.xml) tasks in 50 iterations based on the (makespan) metrics. The simulation result can be shown in Table 3 and Figure 5.

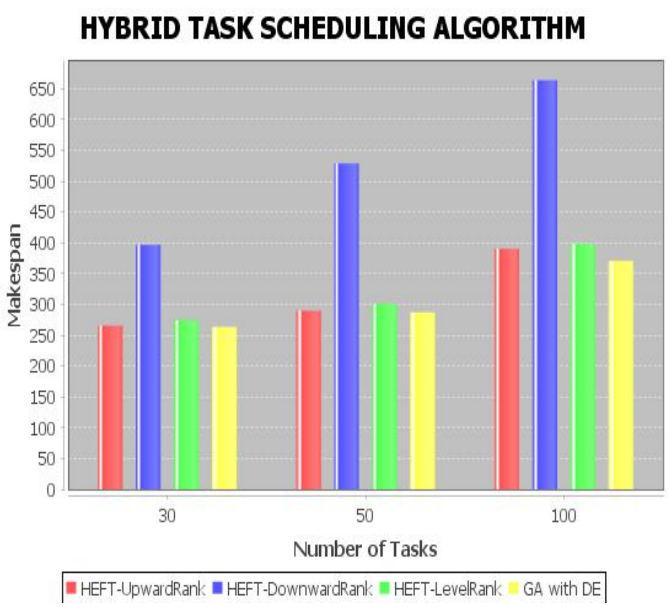

Figure 5: The Effect of the Number of Tasks on the Make-span

Table 3: The Result of Total Executed Jobs

Algorithm	Make-span		
	Cybershake_30	Cybershake_50	Cybershake_100
HEFT upward rank	265.49	290.01	390.31
HEFT downward rank	396.71	528.88	663.719
HEFT-Level rank	274.484	300.85	397.99
GA-DE	263.48	287.10	370.69

In Figure 5 a graph is drawn between the number of different tasks on *x*-axis and the corresponding makespan of the applications on *y*-axis. The Genetic Algorithm-Differential Evolution algorithm is compared against the other heuristic such as HEFT-Upward rank, downward rank and level rank) scheduling algorithms in terms of the makespan by randomly generated DAGs with 10, 50 and 100 tasks. The number of tasks increases in the graph, the proposed algorithm shows the better optimal makespan values compared to other heuristic algorithms.

Table 4: The Result of Total Executed scientific workflows on GA-DE

Algorithm	Make-span		
	Cybershake_100	Epigenomics _100	Montage_100
GA-DE	336.64	42148.66	55.05

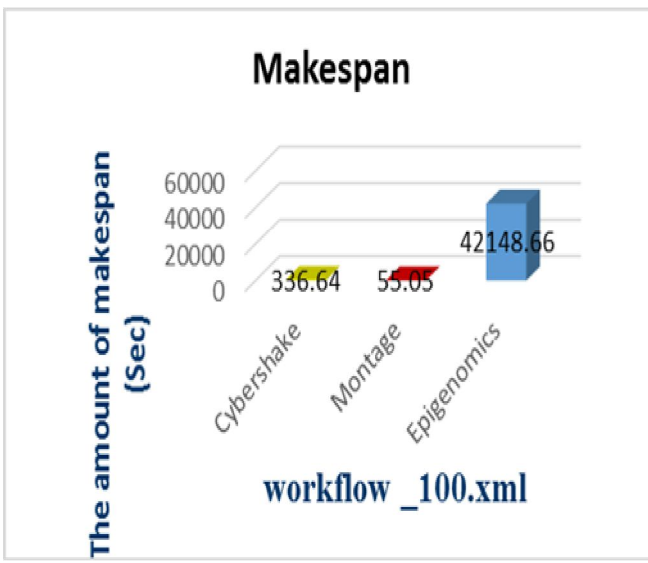

Figure 6: The make-span on Various Scientific Workflows on GA-DE

In Figure 6 graphs are drawn among the different scientific workflows on *x*-axis and the corresponding makespan of the applications on *y*-axis. The comparison among Cybershake, Epigenomics, and Montage in terms of the makespan by randomly generated DAGs wit 100 cloudlets and repeated 50 times until converged values are collected.
The experimental results indicate that Epigenomics is the worst option to utilize in execution tasks with respect to the makespan. On the other hand, using Montage workflow in implementing GA-DE scheduling algorithm executes the optimal performance with generating the minimum makespan compared to others.

5. CONCLUSION

This paper investigates and evaluates the performance of the hybrid meta-heuristic GA-DE using standard data set and compared it with three heuristic algorithms (HEFT-Upward rank, downward rank, and Level rank). A discrete event cloud simulator based on the CloudSim framework for modelling and cloud computing simulation has been used. The adopted algorithms have been measured in terms of the make-span with respect to the number of tasks assigned to the cloud system. The simulation results demonstrate that the GA-DE algorithm outperforms the other algorithms in terms of make-span. In addition, we also conclude that Montage workflow is the most appropriate workflow for executing tasks in the proposed algorithm (GA-DE). In the future work, we plan to incorporate other type of meta-heuristic algorithm with Genetic algorithm and evaluate the effectiveness of the proposed Hybrid-meta heuristic algorithm taking into consideration various relevant performance metrics.

REFERENCES

1. B. P. Rimal, E. Choi, and I. Lumb. **A Taxonomy, survey, and issues of cloud computing ecosystems**, *Cloud Computing Principles, Systems and Applications*, pp. 21-46, May 2010. https://doi.org/10.1007/978-1-84996-241-4_2
2. N. A. Mehdi, H. Ali, A. A. Alwan, Z. T. Abdul-Mehdi. **Two-phase provisioning for hpc tasks in virtualized datacenters,** in *Proc. the Int. Conf. on Emerging Trends in Comp. and Elec. Eng. (ICETCEE'2012)*, Dubai, March 2012, pp. 29 – 35.
3. N. A. Mehdi, A. Mamat, A. A. Alwan, and Z. T. Abdul-Mehdi. **Minimum completion time for power-aware scheduling in cloud computing**. In Proc. *the Int. Conf. of Dev. in e-Systems Eng. (DeSE2011)*, Dubai, December, 2011. https://doi.org/10.1109/DeSE.2011.30
4. H. A. Lagar-Cavilla, J. A. Whitney, A. M. Scannell, P. Patchin, S. M. Rumble, E. D. Lara, M. Brudno and M. Satyanarayanan. **SnowFlock: rapid virtual machine cloning for cloud computing**. In *Proc. the 4th ACM European Conf. on Comp. Sys. EuroSys '09*. Nuremberg, Germany, April 2009, pp. 1-12. https://doi.org/10.1145/1519065.1519067
5. Amazon EC2. https://aws.amazon.com/ec2/
6. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. **A view of cloud computing**, *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, April 2010. https://doi.org/10.1145/1721654.1721672
7. A. Gupta and R. Garg. **Workflow scheduling in heterogeneous computing systems : A survey**. In *Proc. Int. Conf. on Computing and Comm. Tech. for Smart Nation (IC3TSN)*, Gurgaon, India, Oct. 2017, pp. 319–326. https://doi.org/10.1109/IC3TSN.2017.8284499
8. Margaret Rouse. **What is cloud computing? - Definition from WhatIs.com**. Retrieved July 5, 2018, from https://searchcloudcomputing.techtarget.com/definition/cloud-computing
9. P. Mangla and S. K. Goyal. **Heuristic vs meta-heuristic approaches for load balancing in cloud environment**, *International Journal of Computer & Mathematical Sciences*, vol. 6, no. 4, pp. 297-301. Aug. 2017.
10. R. Tyagi and S. K. Gupta. **A survey on scheduling algorithms for parallel and distributed systems**. *Silicon Photonics & High Performance Computing*, vol. 718, pp. 51-64, Dec. 2017. https://doi.org/10.1007/978-981-10-7656-5_7
11. A. Vashistha, R. Porwal , A. K. Soni. **A taxonomy of scheduling algorithms for cloud computing**. *International Journal of Computer Science Issues*, vol. 12, issue 1, no. 2, pp. 67-71, Ja. 2015.
12. S. M. Jaybhaye and V. Z. Attar. **A review on scientific workflow scheduling in cloud computing**. In *Proc. the 2nd Int. Conf. on Comm. and Elec. Sys. (ICCES)*. Coimbatore, India, Oct. 2017, pp. 218-223. https://doi.org/10.1109/CESYS.2017.8321269
13. N. I. Md Enzai and M. Tang. **A heuristic algorithm for multi-site computation offloading in mobile cloud computing**, *Procedia Computer Science*, vol. 80, pp. 1232-1241, June 2016. https://doi.org/10.1016/j.procs.2016.05.490
14. K. Dubey, M. Kumar and S. C. Sharma. **Modified HEFT algorithm for task scheduling in cloud environment**, *Procedia Computer Science*, vol. 125, pp. ages 725-732. Jan. 2018.
15. E. N. Alkhanak and S. P. Lee. **A hyper-heuristic cost optimisation approach for scientific workflow scheduling in cloud computing**, *Future Generation Computer Systems*, vol. 86, pp. 480-506, September 2018. https://doi.org/10.1016/j.future.2018.03.055
16. B. Keshanchi, A. Souri and N. J. Navimipour. **An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, simulation, and statistical testing**,

Journal of Systems and Software, vol. 124, pp. 1-21, February 2017.

17. M. Akbari, H. Rashidi, and S. H.Alizadeh. An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. Engineering Applications of Artificial Intelligence, vol. 61, pp. 35-46, May 2017.
18. A. Choudhary, I. Gupta, V. Singh and P. K.Jana. **A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing**. *Future Generation Computer Systems*, vol. 83, pp. 14-26, June 2018.
19. I. A. Moschaki and H. D. Karatz. **A meta-heuristic optimization approach to the scheduling of bag-of-tasks applications on heterogeneous clouds with multi-level arrivals and critical jobs**. *Simulation Modelling Practice and Theory*, vol. 57, pp. 1-25, September 2015. https://doi.org/10.1016/j.simpat.2015.04.009
20. M. H. Eawna, S. H. Mohammed, E. M. El-Horbaty. **Hybrid algorithm for resource provisioning of multi-tier cloud computing,** *Procedia Computer Science*, vol. 65, pp. 682-690, 2015. https://doi.org/10.1016/j.procs.2015.09.012
21. K. Cho, P. Tsai, C. Tsai and C. Yang. **A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing**, *Neural Computing and Applications*, vol. 26, Issue 6, pp 1297–1309, August 2015. https://doi.org/10.1007/s00521-014-1804-9
22. H. B. Alla, S. B. Alla, A. Touhafi and A. Ezzati. **A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment**. *Cluster Computing*, vol. 21, Issue 4, pp. 1797–1820, Dec. 2018. https://doi.org/10.1007/s10586-018-2811-x
23. A. Kamalini and A. Ghaffari. **Hybrid task scheduling method for cloud computing by genetic and DE algorithms**, *Wireless Personal Communications*, vol. 97, Issue 4, pp. 6301–6323, Dec. 2017. https://doi.org/10.1007/s11277-017-4839-2