

RESEARCH ARTICLE

 OPEN ACCESS

Received: 25.10.2021

Accepted: 05.01.2022

Published: 31.01.2022

Citation: Hanabaratti KD, Rodd SF (2022) Evolutionary Computing based Web Service Composition Technique for Scheduling of Workload under Cloud Environment. Indian Journal of Science and Technology 15(2): 69-80. <https://doi.org/10.17485/IJST/v15i2.1806>

* **Corresponding author.**kdhanabaratti@git.edu**Funding:** None**Competing Interests:** None

Copyright: © 2022 Hanabaratti & Rodd. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](#))

ISSN

Print: 0974-6846

Electronic: 0974-5645

Evolutionary Computing based Web Service Composition Technique for Scheduling of Workload under Cloud Environment

Kavita D Hanabaratti^{1*}, Sunil F Rodd²

¹ Assistant Professor, Department of Computer Science and Engineering, KLS Gogte Institute of Technology, Belagavi

² Professor, Department of Computer Science and Engineering, KLS Gogte Institute of Technology, Belagavi

Abstract

Objectives: The objective of this model is to design a technique to schedule the workload for the computation of web services in the heterogeneous cloud environment. As the Web Service Composition (WSC) for the execution of scientific workload in a heterogeneous cloud computing environment is a challenging task. Modern workload requires dynamic resource provisioning technique as there exist parallelization among sub-tasks and different task demands different Quality of Service (QoS) requirement. **Methods:** This study presents an Evolutionary Computing based Web Service Composition (EC-WSC) technique to execute a large-scale scientific workload in a heterogeneous cloud environment. A multi-objective metric for improving energy efficiency and resource utilization is modelled. Then, an improved searching mechanism for the dragonfly evolutionary computing algorithm is modelled. **Findings:** Experiment outcomes show EC-WSC model attains superior performance in execution time performance analysis and energy efficiency performance analysis when compared with existing resource provisioning models of workload service composition such as Deadline and Budget-Aware Workflow Scheduling (DBAWS)⁽¹⁾, Evolutionary Computing Multi-objective optimization for Hybrid Clouds (EC-MOH)⁽²⁾, Web Service Composition (WSC)⁽³⁾, and Evolutionary Multi-Objective Optimization for clouds (EMOC)⁽⁴⁾ in terms of heterogeneous computing, workload size, multi-objective optimization, QoS metric, and optimization strategy. Our model EC-WSC has proved to be more efficient in terms of energy efficiency by a reduction of 52.13% and also reduction in execution time by 71% when compared with the WSC⁽³⁾ existing Web Service Composition model. **Novelty:** Existing resource provisioning predominantly focused on reducing computation cost and time; however, induces task execution latency and energy overhead. However, EC-WSC is modelled to utilize resources more efficiently and meet task QoS requirements by assuring energy minimization constraints.

Keywords: Cloud Computing; Heterogeneous Computing Environment; Multiobjective optimization problem; Resource Provisioning; Workload Scheduling

1 Introduction

Cloud computing environment offers virtual computing nodes (i.e., instance) for executing workflow tasks submitted by the user. The user can use several cloud instances in a scalable manner for executing tasks. Generally, these instances/services are given to users based on predefined SLA's. The SLA defines a set of agreements between the service provider and service consumers for meeting desired QoS. The users are charged based on QoS and SLA defined. Considering these benefits, grid-based workflow providers such as Pegasus and ASKALON have started supporting cloud-based workflow execution. Thus, it is a challenging task for finding suitable resources (i.e., web service composition mechanism) for executing workload in a multi-processing framework. Generally, the standard web service composition technique hypothesis is that users submit the workload with constraints to the cloud platform. Cloud platform checks and informs its scheduler to find web service available and its configuration for executing workload as per QoS defined. Here the scheduler can allocate resources to workload statically or dynamically. From extensive survey varied, it can be seen an effective web service composition technique generally reduce overall make span with improved resource utilization. However, the modern workload with green computing norms demands the task must be completed within a certain deadline and to meet energy constraints. Existing web composition technique for executing workload focused just for minimizing energy; thus, leads to higher service violation.

Recently, extensive work has been completed on the workload resource provisioning issue in homogeneous processing conditions. In ⁽¹⁾ because of the workload issues and resource provision in cloud computing and for the storage requirements, an optimized method using the multi-objective has been proposed. This method mainly focuses to reduce the cost and time required for the execution of the scientific cloud problems. In ⁽⁵⁾ a survey on various researches has been done to show the improvement in the allocation of virtual machines considering all the researches from 2015 to 2021. This survey comprises the data about the various attributes and parameters required for the computation of the scientific workloads. In the conclusion of this survey, they have presented the major role of machine learning algorithms in cloud computing to balance the workload in virtual machines and the advantages of virtual machines in the cloud network. In ⁽⁶⁾ to improve the QoS requirement and to perform the execution of the workload with less cost and time, an optimized method to allocate the resources using the genetic algorithm has been proposed. The results have been tested using the different workloads deployed in the Amazon Web Services (AWS). In any case, ⁽⁵⁾ ⁽⁶⁾ didn't consider limiting energy utilization for the execution of DAG application tasks. In this manner, incites greater expense of execution. For tending to energy effectiveness, proficient execution, and dependable handling prerequisite of present-day Bigdata computational model/systems ⁽⁷⁾ thinks about that parallel computational platform structure can self-sufficiently enhance the exhibition boundary for uniquely allowing calculation measure as per client QoS or SLA necessities, asset openness, energy limitation, and execution prerequisite. This adaptive crosses directly from the application, for example, planning and mapping of the task to respective computing platforms through power gating and frequencies optimization. Cloud computing platform offers computation of workload in a distributed manner. These platforms are heterogeneous as they offer computation for the large-scale storage and computation process for the workload execution ⁽⁸⁾. These techniques include enormous computational costs and effects climate. This is because of high energy dispersal at

various levels of task execution and storage⁽⁹⁾. In⁽¹⁰⁾ for the scheduling of the tasks and also to estimate the workload of the cloud, an evolutionary method has been proposed based on the hybrid task scheduling. This method uses the PSO scheme and PTP scheme to schedule the resources required by the task and decrease the execution time. This method has also used the machine learning algorithm which has been employed using the GDM scheme. In⁽¹¹⁾ to increase the usage of the resources with less make span and also to reduce the SLA violation, a load balancing scheduling method, ALTS, has been proposed. In this method, the algorithm is mapped to the incoming tasks and these tasks are then mapped to the respective free virtual machine for the execution of the scientific workflow. In any case, these assessments are kept to self-confined tasks. As heterogeneous systems continue being upgraded, DAG applications tasks with selectivity are also continuing to rise. The task scheduling on heterogeneous computation platforms is generally NP-hard.

Different meta-heuristic strategies, for instance, annealing, Ant Colony Optimization (ACO), GA are extensively used in DAG applications task scheduling⁽¹²⁾. In⁽¹³⁾, examined that by upgrading execution as far as decreasing calculation cost for the execution of image-based task; they zeroed in on blending prefetching and task provisioning. The response makespan minimization, execution cost, and enhancement issues are demonstrated through heuristic strategy. In⁽¹⁴⁾ exhibited the critical difficulties that occur during heterogeneous task resource provisioning under heterogeneous computation environment. Here tasks are scheduled to private and hybrid cloud-based cost minimization model. To guarantee prevalent asset usage, they introduced a heterogeneous resource provisioning model under a private Cloud platform. Additionally, for guaranteeing the responsibility execution is finished inside cutoff time imperative a resource provisioning technique is introduced utilizing Back Propagation Neural Network (BPNN) under heterogeneous computing platform. Similarly, in⁽²⁾ addressed DAG task is scheduled within the monetary financial plan and processing efficiency requirement considering heterogeneous computation environment. The main resource provisioning scheme is planned to utilize the Single Objective (SO) metric to be specific Deadline Constrained Cost Optimization (DCOH) model. The DCOH is planned by lessening the financial spending plan of task execution with the cutoff time essential. Second, they introduced a Multi-Objective (MO) resource provision technique specifically Multi-Objective Optimization Method (MOH). The MOH strategy is intended in balancing the financial spending plan and execution time for DAG task computation. The drawback of DCOH is it assumes the makespan of DAG tasks are fixed in nature, and resource provisioning models are designed considering such fix makespan assumption leading to poor performance. Further, it is important to reduce energy consumption and meet the application QoS prerequisites. This paper presents an improved QoS metric to improve resource utilization and reduce energy consumption for DAG task execution. The optimization process is through the development of an improved evolution computing model which is discussed in the next section.

For addressing the tradeoffs problem here is an efficient web service composition technique that considers minimizing execution time and energy consumption by employing multi-objective improved dragonfly algorithm by effectively utilizing resources in multi-core cloud framework.

The contributions of the work are as follows:

- Presented an evolutionary computing-based WS provisioning technique to execute data-intensive workload in the heterogeneous cloud platform.
- The EC-WSC technique executes workload by minimizing energy consumption with high resource utilization.
- The EC-WSC reduces execution time and energy consumption for executing workload of varied size and complexity when compared with the existing WSC technique.

The manuscript is arranged as described. In section II, an evolutionary computing-based web service composition technique is presented. In section III, the EC-WSC model performance is compared with the standard WSC technique. In section IV the research is concluded and also future work in improving the WSC model is discussed.

2 Evolutionary Computing based Web Service Composition Technique for Scheduling Workload in Heterogeneous Cloud Environment

This section presents an Evolutionary Computing-based EC-WSC technique for scheduling workload in the heterogeneous cloud computing environment. First, the workload execution model is presented. Second, presents a system model used for executing the scientific workload. Third, energy consumption for executing scientific workload is presented. The resource provisioning constraint and scheduling metrics to reduce energy dissipation and utilize resources more efficiently for workload execution has been presented. Finally, the dragonfly evolutionary computation algorithm is used for obtaining optimal resource provisioning performance.

2.1 Scientific Workload Model

The data-intensive workload can be described using the following equation

$$X = (x_1, x_2, x_3, \dots, x_n), \tag{1}$$

and $x_j (j = 1, 2, \dots, n)$ represent j^{th} workload. The workload x_j can be expressed using the following equation

$$x_j = (b_j, e_j, H_j), \tag{2}$$

where b_j depicts workload arrival time, e_j represent workload deadline time, and H_j depicts workload structure. Different workload has a different structure which can be defined as a directed acyclic graph as

$$H_j = (U_j, F_j) \tag{3}$$

where U_j defines the set of tasks in workload x_j and it is expressed as follows

$$U_j = (u_1^j, u_2^j, \dots, u_{|U_j|}^j) \tag{4}$$

and F_j defines data dependencies between the workload x_j and it is expressed as follows

$$F_j \subseteq U_j \times U_j \tag{5}$$

An edge $f_{qk}^j \in F_j$ describes the beginning of the task u_k^j that relies on the output information of the ongoing task u_q^j ; thus u_k^j is called a forthcoming task and u_q^j depict the preceding task.

2.2 System model for workload execution on heterogeneous cloud environment

This system model used in this work is shown in Fig. 1. The modern heterogeneous cloud computing environment is generally composed of a huge number of dynamic physical machine I and is described as follows

$$I = (I_1, I_2, I_3, \dots, I_o), \tag{6}$$

where o defines the number of physical machines present in a heterogeneous cloud environment. For respective physical machine $I_l \in I$, can be exhibited as follows

$$I_l = (st_l, n_l, q_l^\uparrow, o_l, (g_l, w_l), V_l), \tag{7}$$

where st_l depicts storage size, n_l represent memory capacity, q_l^\uparrow defines peak energy level, o_l bandwidth, capacity, (g_l, w_l) depicts corresponding voltage and frequency level and V_l represent the virtual machine presented in the physical machine. The discrete voltage and frequencies of the physical machine I_l can be expressed as follows

$$(g_l, w_l) = ((g_l^1, w_l^1), (g_l^2, w_l^2), \dots, (g_l^\uparrow, w_l^\uparrow)) \tag{8}$$

Similarly, the virtual machine in the physical machine I_l is exhibited as follows

$$V_l = (v_{l,1}, v_{l,2}, \dots, v_{l,|V_k|}) \tag{9}$$

Further, each virtual machine can be represented as follows

$$v_{l,m} = (g_{l,m}, n_{l,m}, st_{l,m}) \tag{10}$$

where $g_{l,m}$ depicts virtual machine frequencies level, $n_{l,m}$ represent the virtual machine memory capacity, and $st_{l,m}$ defines storage size. Here the hardware resource can be shared among virtual machines and can migrate between different physical machines.

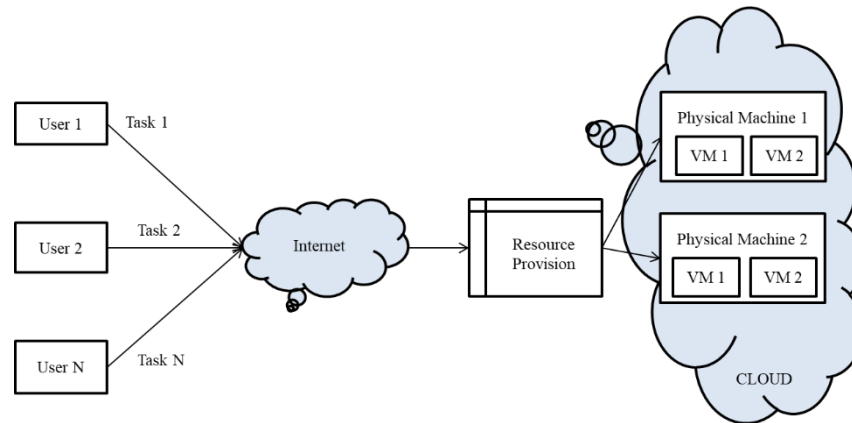


Fig 1. Proposed system model⁽³⁾

2.3 Energy consumption model of physical machines

Let consider t_l as the static energy consumption and q_l^\uparrow represent maximum energy of the physical machine i_l . Using the model described in⁽¹⁵⁾ the energy consumption of physical machine i_l can be expressed as follows

$$I_l = t_l * q_l^\uparrow * z_l^u + \frac{(1 - t_l) * q_l^\uparrow}{(g_l^\uparrow)^3} * (g_l)^3, \tag{10}$$

where $z_l^u \in (1, 0\}$ defines whether a physical machine i_l is active or not at time u , g_l represent actual CPU frequencies at time instance u , and g_l^\uparrow represent maximal CPU frequencies at time instance u . From this parameter, the energy consumption of o physical machines considering different time instances (i.e., xt to yt) can be estimated using the following equation

$$E = \sum_{j=1}^o \int_{xt}^{yt} \left(t_l * q_l^\uparrow * z_l^u + \frac{(1 - t_l) * q_l^\uparrow}{(g_l^\uparrow)^3} * (g_l)^3 \right) dt, \tag{11}$$

where z_l^u and g_l varies with respect to time; thus, are time-dependent; and the other parameter doesn't vary with respect to time; thus, they are static in nature.

2.4 Workload task scheduling resource provisioning metric modeling

Let $y_{k,lm}^j$ represent the mapping association among task u_k^j and virtual machine $v_{l,m}$ on physical machine i_l . The value of $y_{k,lm}^j$ will be equal to 0 if u_k^j is mapped to the virtual machine $v_{l,m}$, otherwise, it will be 1; which can be mathematically described using the following equation

$$y_{k,lm}^j = \begin{cases} 0, & \text{if } u_k^j \text{ is not mapped to } v_{l,m} \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

The data dependencies between the tasks in different scientific workloads should overcome the following resource provisioning constraints

$$gu_{q,lm}^j + uu_{qk}^j \leq st_{k,lm}^j, \quad \forall f_{qk}^j \in F_j \tag{13}$$

where $gu_{q,lm}^j$ represent task u_q^j 's completion time, and uu_{qk}^j depicts time taken for transmission among task u_q and u_k . After completing the execution of all the tasks within the x_j to respective virtual machines, the maximum time taken for completing it is described using the following equation

$$gu_j = \max_{u_k^j \in U_j} \left(gu_{k,lm}^j \right). \tag{14}$$

Further, the resource provisioning model should meet task deadline requirement as described in the below equation

$$gu_j \leq e_j, \quad \forall x_j \in X. \tag{15}$$

Along with, because the restricted resources available with physical machines restrict the number of virtual machines to be used; thus, makes resource provisioning even more difficult considering constraints defined below

$$g_l^\uparrow - \sum_{m=1}^{|V_l|} g_{l,m} \geq 0, \quad \forall i_l \in I; \tag{16}$$

$$n_l - \sum_{m=1}^{|V_l|} n_{l,m} \geq 0, \quad \forall i_l \in I. \tag{17}$$

This work aimed at satisfying the constraint defined in Eq. (13), (15) (16), and (17) for minimizing energy dissipation in resource provisioning for minimizing for executing scientific workload as described in below equation

$$\text{Min} \sum_{l=1}^o \int_{xt}^{yt} \left(t_l * q_l^\uparrow * z_l^u + \frac{(1-t_l) * q_l^\uparrow}{(g_l^\uparrow)^3} * (g_l^e)^3 \right) dt. \tag{18}$$

where o defines physical machine size, xt defines start time for workload execution and yt represents end time for workload execution. Further, this work aimed at using resources more efficiently; thus, the following objective functions are defined

$$\text{Max} \left(\sum_{j=1}^n \sum_{k=1}^{|U_j|} cp_u_k^j * T_k^j \right) / \left(\sum_{l=1}^o g_l^\uparrow * A_l \right), \tag{19}$$

where n defines the workload size of X and $(U_j|)$ represent the task size of the workload x_j , $cp_u_k^j$ represent task u_k^j 's CPU frequencies prerequisite, T_k^j represent time take to complete workload execution, defines physical machines size in a heterogeneous cloud environment, and A_l depicts an active time of the physical machine i_l 's. Scheduling these tasks on multiple computing machines is an NP-Hard problem. An extensive survey carried out in ⁽³⁾ shows dragon fly evolutionary computing algorithm is efficient in solving problems. The evolutionary computing-based WSC technique for scheduling scientific workload in a heterogeneous cloud computing environment is shown in Figure 2.

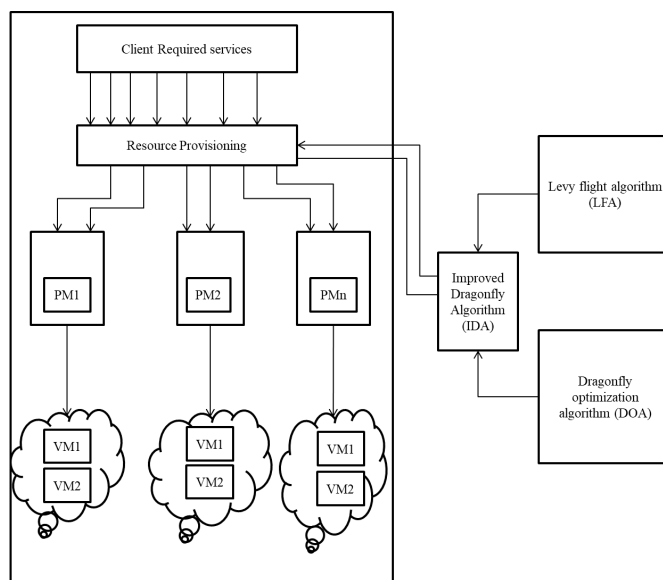


Fig 2. Proposed improved dragon fly-based WSC technique for heterogeneous cloud environment⁽³⁾

2.5 Evolutionary computing scheduling resource provisioning metric modeling

This section presents an evolutionary computing-based WSC technique for workload execution in the heterogeneous cloud computing environment. The objective of this work is to assign resource i.e., virtual machine using an improved dragonfly algorithm (IDA) that maximize resource utilization with minimal energy dissipation. Reducing energy dissipation plays a very important role in reducing the overall cost of provisioning service. The proposed work aimed in bringing good tradeoffs between energy minimization and resource utilization maximization by employing multi-objective-based workload scheduling using IDA. Dragonfly algorithm is a metaheuristic model designed using static and dynamic swarming behaviors of dragonflies in nature⁽³⁾.

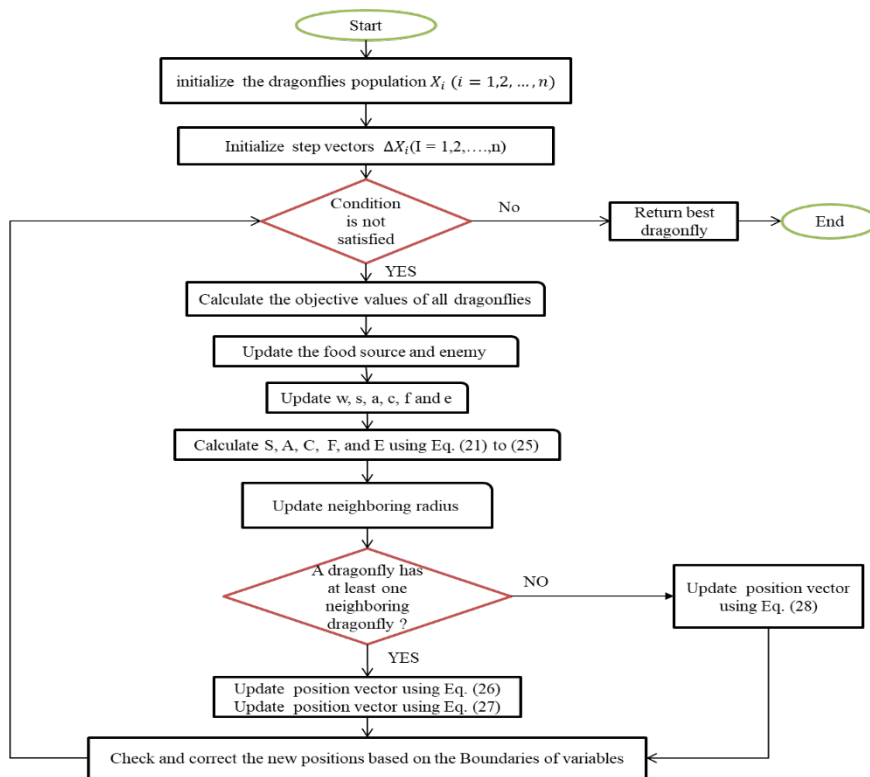


Fig 3. Improved dragonfly algorithm

Dragonfly has two main objectives, Hunting (static) and migration (dynamic). In the migration phase, a huge number of dragonflies move long distances toward diverse territorial regions; this results in the exploration phase. In the static hunting phase, the dragonfly moves with a huge herd and along one bearing with nearby developments and abrupt changes in the flying way, which is reasonable in the exploitation stage. To improve the searching ability and avoid the local optimum, here, improved DA is modeled. The entire process of the improved dragonfly algorithm is shown in Figure 3. The step-by-step process of the proposed multi-objective workload scheduling is given below:

Initialization phase and fitness computation: The population of the dragonflies is randomly initialized in this step. The position of i^{th} dragonfly is established using the following equation

$$X_i = (x_i^1, x_i^2, x_i^3, \dots, x_i^d) \tag{20}$$

where $i = 1, 2, \dots, N$, x_i^d defines the i^{th} dragonfly position within the d^{th} dimension of searching space and N represent the searching agent's size. The population is consisting of a set of solutions. The solution is created based on several user tasks and virtual machines (VM). Initially, the tasks are assigned to VM randomly. Then, based on the fitness function the solutions are updated. Here the weights are initialized randomly for the different operators. The position and velocity of dragonflies are updated using Eq. (21) to Eq. (24).

Updating solution: The dragonfly algorithm is composed of five elements such as separation alignment, cohesion, attraction toward food, and distraction from the enemy. The separation is computed using the below equation

$$SP_i = -\sum_{j=1}^N X - X_j \tag{21}$$

where X defines the current individual position, X_j defines the j^{th} neighborhood individual position and N represent the size of the adjacent individual. The alignment is computed using the following equation

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \tag{22}$$

where V_j depicts the speed of the j^{th} adjacent individual. The cohesion is computed using the following equation

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \tag{23}$$

Attraction toward food is computed using the following equation

$$F_j = X^+ - X \tag{24}$$

where X^+ defines the food source position. The distraction from the enemy is computed using the following equation

$$E_i = X^- - X \tag{25}$$

where X^- defines the enemy position. Then, the speed vector is computed using the following equation

$$\Delta X_{k+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_i \tag{26}$$

wheres define distinct weights, S_i defines the distinct weight of i^{th} individual, c and C_i depicts the cohesion weight and distinct weight of i^{th} individual, respectively. The position vector is computed using the following equation

$$X_{k+1} = X_k + \Delta X_{k+1} \tag{27}$$

where k represents the present iteration. If it doesn't find a dragonfly within its neighborhood the position of the dragonfly is optimized using the levy flight equation defined in Eq. (28). This aid in enhancing randomness behavior and global searching mechanism for achieving obtaining optimal solution as described in below equation

$$X_{k+1} = X_k + Levy(d)X_k \tag{28}$$

$$levy(x) = \alpha * \frac{r_1 * \mu}{(r_2)^{1/\beta}} \tag{29}$$

$$\mu = \left(\frac{\tau(1 + \beta) * \sin(\pi\beta/2)}{\tau((1 + \beta)/2) * \beta * 2^{((\beta-1)/2)}} \right)^{1/\beta}, \tag{30}$$

$$\tau(x) = (x - 1)!, \tag{31}$$

where d depicts the number of position vectors considered, r_1 and r_2 defines the random number ranging between 0 and 1, α and β is a small constant value equal to 0.01. In this work, the fitness is computed using position and updating the velocity of the objective. The position is updated till it obtains optimal solution or it reaches the termination conditions. The EC-WSC technique is very efficient in minimizing task execution time with minimal energy dissipation which is proved through experimental study.

3 Results and Discussion

Here experiment is carryout to evaluate the performance of the evolutionary computing-based Web Service Composition Technique and existing web service composition technique^(2,3) for executing data-intensive workload on the cloud computing environment. Both EC-WSC and existing WSC techniques are implemented using the CloudSim simulator. Energy efficiency and execution time are metrics used for validating the performance of different web service composition models. The Inspiral workload is used for validating the performance of EC-WSC and the existing WSC model. A sample representation of Inspiral workload is shown in Figure 4. The Inspiral workload is memory and CPU-intensive in nature and is used for analyzing binary neutron stars and black holes. More details of workload can be obtained from⁽¹⁶⁾. Here the Inspiral workload size is varied to 30, 50, 100, & 1000 and experiments have been conducted.

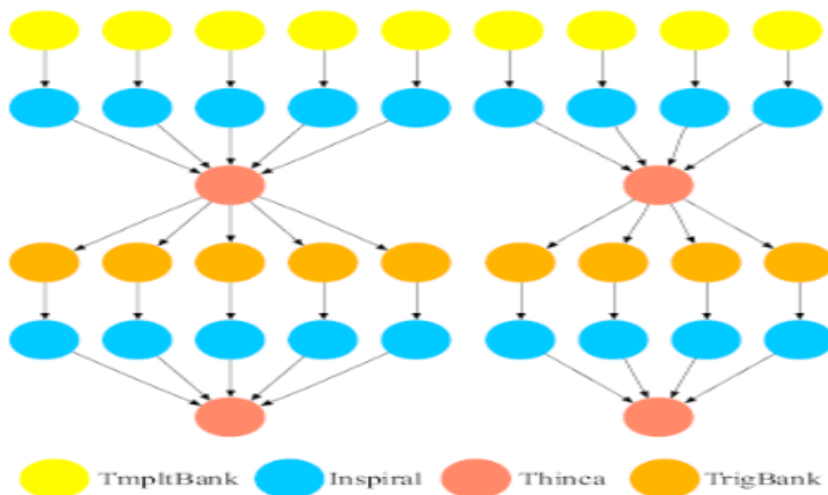


Fig 4. Inspiral workflow

3.1 Execution time performance analysis

Here experiment is conducted by varying Inspiral workload size to 30, 50, 100, & 1000 and execution time achieved using EC-WSC and existing WSC^(2,3) techniques are graphically shown in Fig. 5. From the figure, it can be seen the EC-WSC technique reduces execution time in comparison with the existing WSC technique for both smaller workload sizes and also for larger workload sizes. An average execution time reduction of 83.39% is achieved using EC-WSC in comparison with standard WSC techniques. An important to think from the result we can note that EC-WSC techniques are very efficient for executing larger workload sizes. Further, we evaluate the average execution time efficiency per job using EC-WSC and existing WSC⁽³⁾ as shown from Fig 6. An average execution time reduction of 30.257% is achieved using EC-WSC in comparison with the standard WSC technique. Similarly, we evaluate the average execution time efficiency per job using EC-WSC and existing WSC^(2,3) as shown from Figure 7. An average execution time reduction of 88.28% is achieved using EC-WSC in comparison with the standard WSC model.

3.2 Energy efficiency performance analysis:

Here, the experiment is conducted by varying Inspiral workload size to 30, 50, 100, & 1000 and energy efficiency achieved using EC-WSC and existing WSC⁽³⁾ techniques is graphically shown in Fig. 8. From the figure, it can be seen the EC-WSC technique reduces energy consumption in comparison with the existing WSC technique. An average energy consumption reduction of 41.35% is achieved using EC-WSC in comparison with standard WSC techniques. An important to think from the result we can note that the EC-WSC technique is very efficient for executing both smaller and as well as larger workload sizes.

The Table 1 describes the comparatives analysis of EC-WSC concerning other WSC models such as EC-MOH⁽²⁾, WSC⁽³⁾, DBAWS⁽¹⁾, and EMOC⁽⁴⁾. The following work has been considered for comparative analysis because of the multi-objective optimization strategy adopted. From Table 1, we can say that all WSC models except WSC can allow execution of the complex

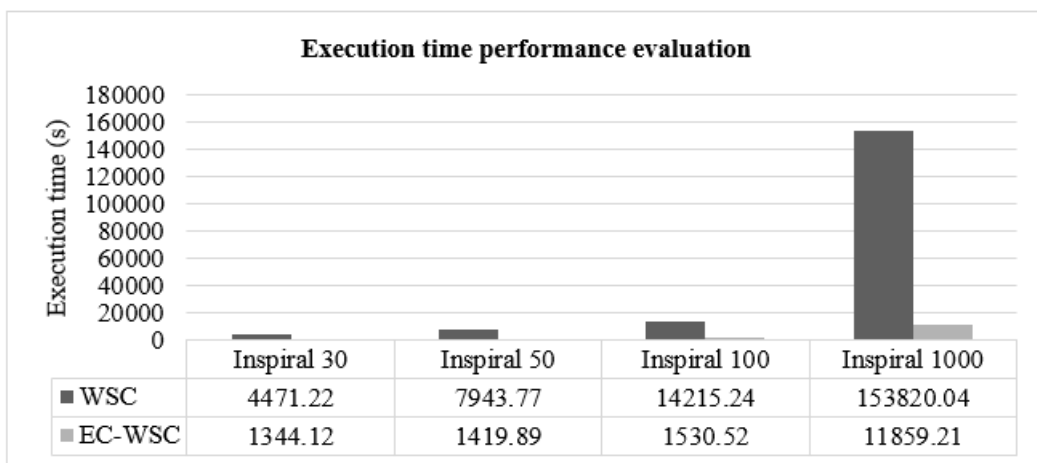


Fig 5. Inspiral Workload ExecutionTime using EC-WSC and standard WSC technique

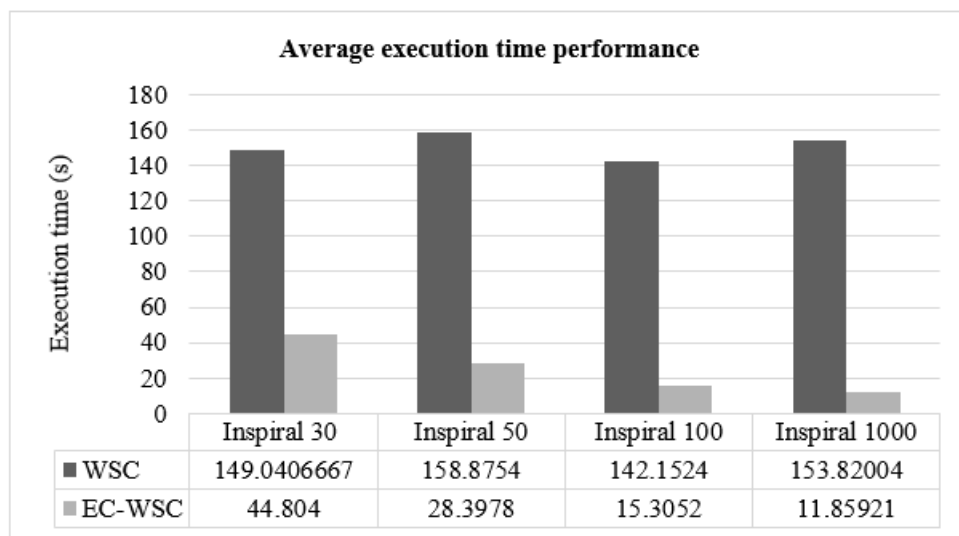


Fig 6. Inspiral Workload Average Execution Time Efficiency per Job using EC-WSC and standard WSC technique⁽³⁾

workload. Similarly, the existing model can support the execution of the complex workload of Small to medium-large. However, the proposed EC-WSC allows execution of complex workload of small-large. All existing work minimizes cost-makespan and no prior work has considered minimizing energy which plays a very important role in defining cost when resources are heterogeneous. However, the proposed EC-WSC is designed to minimize makespan and maximize resource utilization by optimizing energy constraints and meeting application deadlines. Most of the existing strategies have adopted genetic algorithms for solving multi-objective optimization problems; however, WSC⁽²⁾ showed dragonfly algorithms are much efficient in comparison with genetic algorithms for optimizing NP-Hard problems; thus, in our work, an improved version of the dragonfly is modeled for solving a multi-objective optimization problem. The result achieved proves the efficiency of the EC-WSC model. The EC-WSC model reduces energy consumption and at the same time improves execution efficiency for workload execution. Thus, proves that the new improved evolution computing algorithm of the dragonfly algorithm can search better (i.e., optimal) resources in executing the workload. As a result, achieves much better performance than the recently modeled WSC technique⁽¹⁻⁴⁾.

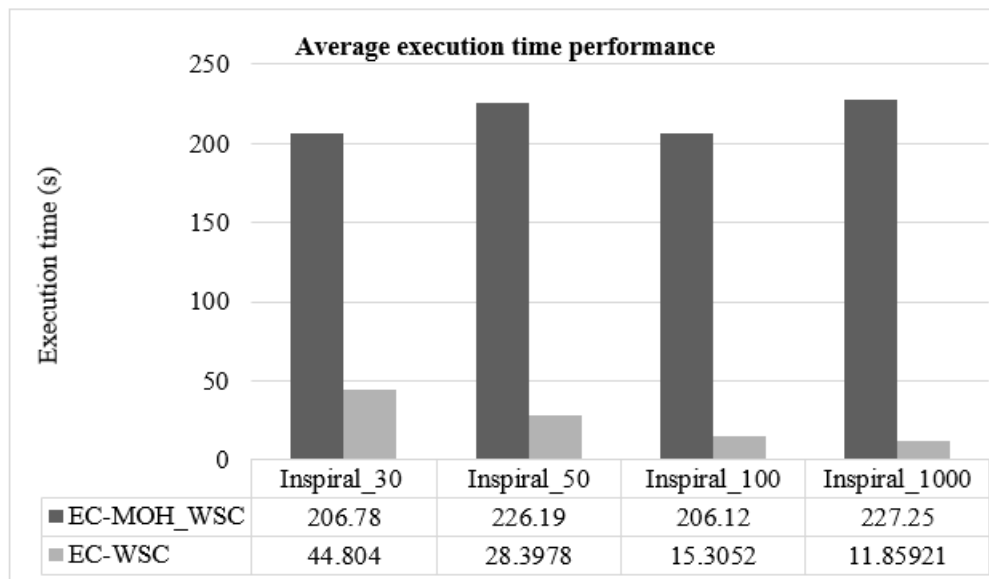


Fig 7. Inspiral Workload Average Execution Time Efficiency per Job using EC-WSC and existing WSC technique^{(2), (16)}

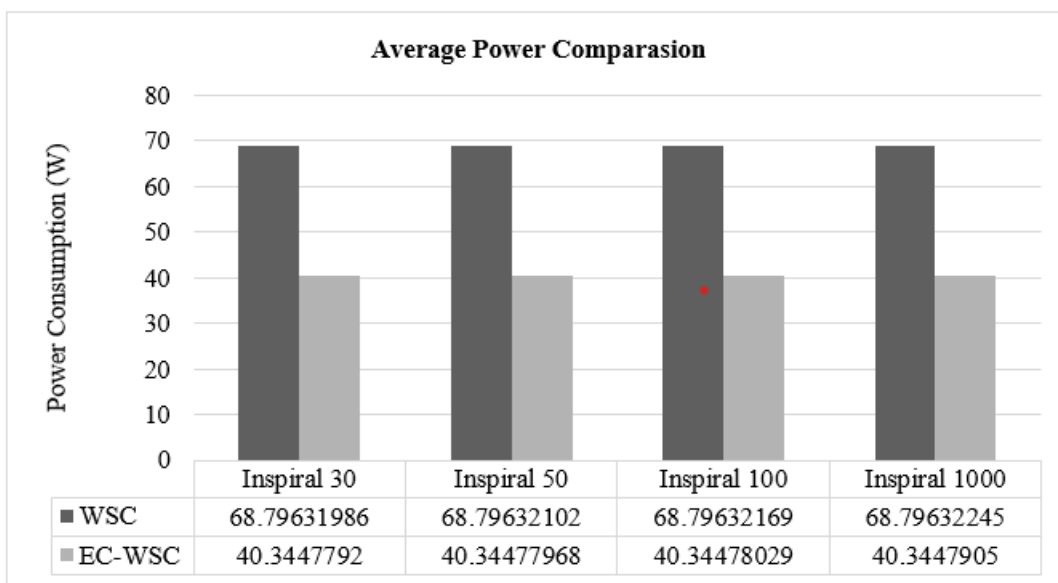


Fig 8. Inspiral Workload Average Energy Efficiency using EC-WSC and existing WSC technique

4 Conclusion

In this paper, various existing WSC techniques for executing workload in the cloud environment have been discussed. This paper also presents a solution for the trade-off problems (NP-Hard problems) using QoS metrics and a new improved searching mechanism through an evolutionary computing algorithm which is introduced to obtain the optimal solution in quick time. In this model of WCS, the multi-objective dragonfly evolutionary computing algorithm has been used. Experiment outcomes show that the EC-WCS is very efficient in reducing the energy consumption by 23% when compared with the exiting method EC-MOH-WSC and also meets the QoS requirement of workload application. Experiment outcomes also show EC-WSC model attains superior performance in execution time performance analysis and energy efficiency performance analysis

Table 1. Comparative analysis of proposed EC-WSC with various other existing WSC models

	EC-MOH ⁽²⁾ , 2019	WSC ⁽³⁾ , 2020	DBAWS ⁽¹⁾ , 2021	EMOC ⁽⁴⁾ , 2021	EC-WSC
Heterogeneous computing	Yes	No	No	No	Yes
Workload type	Complex	Simple	Complex	Complex	Complex
Workload size	Small to medium	small	Small to medium-large	Small to medium-large	Small to large
Multi-objective optimization	Yes	Yes	Yes	Yes	Yes
QoS Metrics	Makespan & cost	Makespan, cost, & load balancing	Deadline & budget	Makespan & cost	Energy, Processing time, resource utilization, & deadlines
Optimization strategy	Genetic Algorithm	Dragonfly algorithm	Monte Carlo & Genetic Algorithm	Chaotic system & Genetic Algorithm	Improved dragonfly algorithm

when compared with existing resource provisioning models of workload service composition such as EC-MOH⁽²⁾, WSC⁽³⁾, DBAWS⁽¹⁾, and EMOC⁽⁴⁾ in terms of heterogeneous computing, workload size, multi-objective optimization, QoS metric, and optimization strategy. Our model EC-WSC has proved to be more efficient in terms of energy efficiency by a reduction of 52.13% and also a reduction in execution time by 71% when compared with the WSC⁽³⁾ existing Web Service Composition model. As the size of the workload scheduling increases our model performs efficiently when compared with other existing models.

Future work would be to improvise the utilization of physical machines. Further, would consider the minimization of computation time and I/O overhead by effectively utilizing system resources (idle server) of the cloud.

References

- 1) Calzarossa MC, Vedova MLD, Massari L, Nebbione G, Tessera D. Multi-Objective Optimization of Deadline and Budget-Aware Workflow Scheduling in Uncertain Clouds. *IEEE Access*. 2021;9:89891–89905. Available from: <https://dx.doi.org/10.1109/access.2021.3091310>.
- 2) Zhou J, Wang T, Cong P, Lu P, Wei T, Chen M. Cost and makespan-aware workflow scheduling in hybrid clouds. *Journal of Systems Architecture*. 2019;100:101631–101631. Available from: <https://dx.doi.org/10.1016/j.sysarc.2019.08.004>.
- 3) Neelima P, Reddy ARM. An efficient load balancing system using adaptive dragonfly algorithm in cloud computing. *Cluster Computing*. 2020;23(4):2891–2899. Available from: <https://dx.doi.org/10.1007/s10586-020-03054-w>.
- 4) Paknejad P, Khorsand R, Ramezanpour M. Chaotic improved PICEA-g-based multi-objective optimization for workflow scheduling in cloud environment. *Future Generation Computer Systems*. 2021;117:12–28. Available from: <https://dx.doi.org/10.1016/j.future.2020.11.002>.
- 5) Ullah A, Nawi NM, Ouham S. Recent advancement in VM task allocation system for cloud computing: review from 2015 to 2021. *Artificial Intelligence Review*. 2021;p. 1–45. Available from: <https://dx.doi.org/10.1007/s10462-021-10071-7>.
- 6) Devarasetty P, Reddy S. Genetic algorithm for quality of service based resource allocation in cloud computing. *Evolutionary Intelligence*. 2021;14(2):381–387. Available from: <https://link.springer.com/article/10.1007/s12065-019-00233-6>.
- 7) Doppa JR, Kim RG, Isakov M, Kinsy MA, Kwon HJ, Krishna T. Adaptive Manycore Architectures for Big Data Computing. *Proceedings of the Eleventh IEEE/ACM International Symposium on Networks-on-Chip*. 2017;9:1–8. Available from: <https://doi.org/10.3390/pr9091514>.
- 8) Xie G, Zeng G, Li R, Li K. Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Heterogeneous Cloud Computing. *IEEE Transactions on Sustainable Computing*. 2017;2(2):62–75. Available from: <https://dx.doi.org/10.1109/tsusc.2017.2705183>.
- 9) Li Z, Ge J, Hu H, Song W, Hu H, Luo B. Cost and Energy Aware Scheduling Algorithm for Scientific Workflows with Deadline Constraint in Clouds. *IEEE Transactions on Services Computing*. 2018;11(4):713–726. Available from: <https://dx.doi.org/10.1109/tsc.2015.2466545>.
- 10) Lakhani B, Agrawal A. A Task Scheduling Approach for Cloud Environments Employing Evolutionary Algorithms. *Journal of Scientific Research*. 2021;13(2):423–438. Available from: <https://dx.doi.org/10.3329/jsr.v13i2.49944>.
- 11) Mubeen A, Ibrahim M, Bibi N, Baz M, Hamam H, Cheikhrouhou O. Alts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing. *Processes*. 2021;9(9):1514–1514. Available from: <https://dx.doi.org/10.3390/pr9091514>.
- 12) Zhu Z, Zhang G, Li M, Liu X. Evolutionary Multi-Objective Workflow Scheduling in Cloud. *IEEE Transactions on Parallel and Distributed Systems*. 2016;27(5):1344–1357. Available from: <https://dx.doi.org/10.1109/tpds.2015.2446459>.
- 13) Khorramnejad K, Ferdouse L, Guan L, Anpalagan A. Performance of integrated workload scheduling and pre-fetching in multimedia mobile cloud computing. *Journal of Cloud Computing*. 2018;7(1):1–14. Available from: <https://dx.doi.org/10.1186/s13677-018-0115-6>.
- 14) Chunlin L, Jianhang T, Youlong L. Hybrid Cloud Adaptive Scheduling Strategy for Heterogeneous Workloads. *Journal of Grid Computing*. 2019;17(3):419–446. Available from: <https://dx.doi.org/10.1007/s10723-019-09481-3>.
- 15) Zhu X, Yang LT, Chen H, Wang J, Yin S, Liu X. Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds. *IEEE Transactions on Cloud Computing*. 2014;2(2):168–180. Available from: <https://dx.doi.org/10.1109/tcc.2014.2310452>.
- 16) Bharathi S, Chervenak A, Deelman E, Mehta G, Su MH, Vahi K. Characterization of scientific workflows. In: 2008 Third Workshop on Workflows in Support of Large-Scale Science. IEEE. 2008. doi:10.1109/WORKS.2008.4723958.